République Algérienne Démocratique et Populaire
Ministère De l'Enseignement Supérieur Et De La Recherche Scientifique
Université Mouloud Mammeri De TIZI-OUZOU
Faculté de Génie Electrique et Informatique
Département Informatique

MEMOIRE DE MASTER
Filière: Informatique
Option: Conduite de projets informatiques

Thème:

Construction d'une ressource lexicale pour l'analyse d'opinion dans un système de e-Education.

Présenté par:
KACIMI Ania
OUSSAID Melissa

Devant le jury composé de :

Président du jury	Mme. T.BERKANE	UMMTO
Examinatrice	Mme. L.LAZIB	UMMTO
Encadreur	Mme. F.BOUARAB	UMMTO
Co-encadreur	Mme. S.LAZIB	UMMTO

Année universitaire : 2019 / 2020

Remerciement

Avant tout, nous remercions Dieu le tout puissant en qui nous avons trouvé la force, le courage et la volonté pour la réalisation de ce mémoire.

Nous adressons le grand remerciement à notre promotrice Mme BOUARAB pour l'honneur qu'elle nous a fait en acceptant de nous encadrer.

Ce travail ne serait pas aussi riche et n'aurait pas pu voir le jour sans l'aide et l'encadrement de Mme LAZIB, nous la remercions pour la qualité de son encadrement exceptionnel, pour sa patience, sa rigueur et sa disponibilité durant notre préparation de ce mémoire.

Nos remerciement les plus sincères aux membres de jury pour l'honneur qu'ils nous ont fait d'avoir accepté de juger notre travail, et d'avoir consacré leur temps pour sa lecture.

Nos remerciement s'adressent également à tous nos enseignants qui ont veillé sur notre formation.

Enfin nos chaleureux remerciements à nos familles et nos amis(es).

Dédicaces

Je dédie ce travail à tous ceux qui me sont chers,
A ma grande mère, aucun hommage ne pourrait être à la hauteur de l'amour dont elle ne cesse de me combler. Que Dieu lui procure bonne santé et longue vie.
A mon exemple éternel, ma source de joie et de bonheur, celle qui s'est toujours sacrifié pour me voir réussir, qui éclaire mon chemin et m'illumine de douceur et d'amour, que dieu te garde pour nous mama.
A mon très cher papa en signe d'amour, de reconnaissance et de gratitude pour tous les soutiens et les sacrifices dont il a fait preuve à mon égard.
A la mémoire de mon grand-père, que son âme repose en paix.
A la prunelle de mes yeux, ma sœur Sarah, la bougie de la maison.
A mon petit frère Mebarek que j'adore, ma vie ne serait pas aussi magique sans sa présence.
A ma tante Nadia, qui m'a accompagné par ses prières, sa douceur, puisse Dieu lui prêter longue vie, beaucoup de santé et de bonheur.
A mes tantes Fadila & Samia, qui m'avaient toujours soutenu et encouragé durant ces années d'études. Que dieu leur donne une longue et joyeuse vie.
A ma tante Nora et mes oncles spécialement : Kaci & Ferhat que j'aime beaucoup.
A mes cousins et cousines.
A la personne qui m'a accompagné tout au long, qui a partagé tous les moments de ma vie, ma perle rare Lydia.
A mes aimables amis(es), qui ont été toujours à mes côtés spécialement : Assia, Rosa & Kahina.
A toute la promotion CPI 2020.
Je termine avec la personne qui a partagé tout le travail, qui a supporté mon humeur au moment de stresse, ma binôme Ania.

Melissa

Dédicaces

Je dédie cet humble travail
A mes très chers parents qui ont fait de moi ce que je suis aujourd'hui grâce à leurs sacrifices. Qu'ils trouvent ici le témoignage de ma plus profonde reconnaissance.

A ma petite sœur chérie Imilia, la gaieté de la maison et mon repère dans cette vie.

A tous mes petits anges spécialement Sadia, Ilyas, Ilyane et Sofia qui illuminent nos vies avec leurs joies de vivre.

A mes chers oncles et tantes

A mes cousins et cousines

A ma binôme qui a rendu ces deux dernières années exceptionelles.

A toute mes copines avec qui j'ai partagé un bout de chemin spécialement Fifi et Avzim deux personnes en or et Yamina qui restera ma plus belle rencontre à la fac.

Au deux soeurs que la vie m'a donné Kahina et Sassy.

A toute la promo CPI 2020

Ania

Résumé

La fouille d'opinion est considérée actuellement comme l'un des domaines de recherche les plus actif . Il a pour but d'analyser les sentiments des gens, ce qui fait de lui un élément important dans le processus de prise de décision. Il est introduit dans quasiment tous les domaines, notamment dans le domaine de l'éducation.

L'objectif de ce travail, est l'amélioration d'une ressource lexicale appelée DICO dédiée à l'analyse d'opinion pour l'éducation, ceci en effectuant un recalcule de polarités suivant une méthode statistique qui permet la génération des modèles de classification qui vont servir d'appuis pour ce recalcule. La nouvelle ressource lexicale sera évaluée en comparant les indicateurs de performances obtenus lors de la classification du corpus EDUCA.

Mots clés : fouille d'opinion, subjectivité, classification, apprentissage automatique, apprentissage supervisé, polarité.

Abstract

Opinion mining is currently considered one of the most active areas of research. It aims to analyze people's feelings, which makes it an important part in the decision-making process. It is introduced in almost all fields, especially in the field of education.

The objective of this work is to improve a lexical resource called DICO dedicated to opinion analysis for education, by performing a recalculation of polarities according to a statistical method which allows the generation of classification models. which will be used as supports for this recalculation. The new lexical resource will be evaluated by comparing the performance indicators obtained during the classification of the EDUCA corpus.

Keywords: opinion research, subjectivity, classification,machine learning ,supervised learning

Liste des abbreviations

AM	apprentissage automatique
ARFF	Attribute-Relation File Format
CLI	Interface de commande en ligne
FN	Faux négatif
FNT	Faux neutre
FP	Faux positif
IBK	Le nom de K plus proche voisin dans Weka
KNN	k plus proches voisins(k nearest neighbors)
POS	Partie du discour(Part of speech)
PWN	Princeton WordNet
SMO	nom de machines à support de vecteurs dans Weka
SVM	Machines à support de vecteurs
SWN	SentiWordNet
TAL	traitement automatique des langues
VN	Vrai négatif
VNT	Vrai neutre

VP Vrai positif

Table des figures

1.1 Axes de la fouille d'opinion. [2]. 6
1.2 Processus de fouille d'opinion[8]. 7
1.3 Domaine d'application de fouille d'opinion[2]. 9

2.1 Techniques de fouille d'opinion [2]. 28
2.2 Fragment de données de sentiwordnet 3.0 41

3.1 Description de la démarche adaptée. 45
3.2 Processus de construction de la ressource lexical DICO[2] 46
3.3 La ressource lexicale DICO 47
3.4 Construction de EDUCA. 49
3.5 Le correcteur CORDIAL [44] 52
3.6 Partie du corpus EDUCA annoté. 53
3.7 Interface graphique de l'outil TreeTager. 54
3.8 Interface utilisateur WEKA 56
3.9 Interface graphique "Explorer". 57
3.10 Extrait du fichier ARFF. 59
3.11 Résultat d'entrainement du classifieur KNN. 61
3.12 Résultat d'évaluation de classifieur KNN 62
3.13 Construction du nouveau DICO. 67
3.14 Algorithme de recalcule de polarité de DICO. 69
3.15 Fragment de la ressource lexicale DICO après recalcule de polarité. . . . 70

4.1 Evaluation de DICO. 73
4.2 Algorithme d'extraction de l'opinion libre [2]. 74

Liste des tableaux

2.1 comparaison entre les méthodes de fouille d'opinion.[2] 26

3.1 Détails d'opération de récolte de donnée 51
3.2 Matrice de confusion du classifieur KNN. 63
3.3 Les indicateurs de performances de classifieurs KNN. 63
3.4 Matrice de confusion du classifieur SVM. 64
3.5 Les indicateurs de performances de classifieur SVM. 64
3.6 Matrice de confusion du classifieur C.4.5 64
3.7 Les indicateurs de performances de classifieur C.4.5 65
3.8 Matrice de confusion du classifieur Naïve Bayes. 65
3.9 Les indicateurs de performances de classifieur Naïve Bayes. 66
3.10 Récapitulation indicateurs de performances des 4 classifieurs. 66
3.11 Statistique de DICO. 70

4.1 Matrice de confusion de classification selon DICO initial. 75
4.2 Indicateurs de performances de la classification selon DICO initial. 75
4.3 Matrice de confusion de la classification selon le nouveau DICO. 75
4.4 Indicateurs de performances de la classification selon le nouveau DICO. . 75
4.5 Récapitulation des deux classifications 76

Table des matières

Introduction générale **1**

1 Fouille d'opinion **3**

I Introduction 3

II Notions de base 3

II.1 Fouille de texte 4

II.2 subjectivité 4

II.3 Faits et opinions 4

II.4 Opinion mining et analyse de sentiment 5

III Définition de fouille d'opinion 5

IV Les taches de fouille d'opinion 6

V Processus de fouille d'opinion 7

V.1 L'acquisition et le prétraitement des données 7

V.2 La pertinence par rapport au sujet 8

V.3 La détection d'opinion 8

VI Domaine d'application de fouille d'opinion 8

VI.1 Domaine politique 9

VI.2 Domaine de l'éducation 9

VI.3 Domaine commercial 9

VI.3.1 Point de vue des entreprises 9

VI.3.2 Point de vue des Clients 10

VII Difficultés de la fouille d'opinion 10

Conclusion 11

2 Méthodes de fouille d'opinion **12**
I Introduction 12
II Les méthodes de détection d'opinions 12
II.1 Les méthodes symboliques 12
II.1.1 Méthodes de construction de ressource lexicale 13
II.1.1.1 Les méthodes manuelles 13
II.1.1.2 La méthode basée sur les corpus 14
II.1.1.3 La méthode basée sur les dictionnaires 14
II.2 Les méthodes statistiques 14
II.2.1 Introduction 14
II.2.2 Techniques d'apprentissage automatiques 15
II.2.2.1 Méthodes d'apprentissage supervisées 15
II.2.2.1.1 Les algorithmes d'apprentissage 16
II.2.2.1.1.1 Classification bayesienne naïve . 16
II.2.2.1.1.2 Arbre de décision 17
II.2.2.1.1.3 Machine à support vectorielle . . 19
II.2.2.1.1.4 K-voisins 21
II.2.2.2 Méthodes d'apprentissage semi-supervisées 22
II.2.2.3 Méthodes d'apprentissage non-supervisées 22
II.2.3 Outil logiciel WEKA 22
II.2.3.1 Traitement de données 23
II.2.3.2 Les méthodes de traitement de données dans WEKA . 23
II.2.3.2.1 La classification : 24
II.2.3.2.2 Le clustering : 24
II.2.3.2.3 L'association : 24
II.2.3.3 L'apprentissage automatique proposé par WEKA 24
II.2.3.3.4 Apprentissage non-supervisé : 24
II.2.3.3.5 Apprentissage supervisé : 24
II.3 Les points forts et points faibles des méthodes de fouille d'opinion 25
II.4 les méthodes hybrides 27
II.5 Evaluation de la classification 28
III corpus 30

III.1 Définition d'un corpus 30
III.1.1 Le processus de constitution du corpus 30
III.1.2 Les types du corpus 31
III.1.2.1 Corpus spécialisé 31
III.1.2.2 Corpus de référence 31
III.1.2.3 Corpus ouvert 31
III.1.2.4 Corpus fermé 32
III.1.2.5 Corpus d'apprenants (Learner corpus) 32
III.1.2.6 Corpus enrichi/annoté 32
III.2 Annotation de corpus 32
III.2.1 Définition d'annotation 32
III.2.2 Types d'annotation 33
III.2.2.1 Annotation phonétique 33
III.2.2.2 Annotation prosodique 33
III.2.2.3 Annotation syntaxique 33
III.2.2.4 Annotation sémantique 34
III.2.2.5 Annotation pragmatique 34
III.2.2.6 Annotation de discours 34
III.2.2.7 Annotation stylistique 34
III.2.2.8 Annotation lexicale 34
III.3 Les approches d'annotation 35
III.3.1 Approche d'annotation manuelle 35
III.3.2 Approche d'annotation automatique 35
III.3.2.1 Méthodes symboliques 35
III.3.2.2 Méthodes par apprentissage 36
III.3.3 Approche d'annotation semi-automatique 36
III.4 Les normes d'annotation d'un corpus 37
III.5 Les avantages et les inconvénients de l'annotation 38
III.5.1 Les avantages de l'annotation 38
III.5.2 Les inconvénients d'annotation 38
IV Wolf et Sentiwordnet 38
IV.1 Sentiwordnet 38

IV.1.1 Introduction 38
IV.1.2 Présentation de SentiWordNet 39
IV.1.3 Les versions de SentiWordNet 39
IV.1.4 L'approche de construction de SentiWordNet3 40
IV.1.5 Structure de SentiWordNet 40
IV.2 Wolf 41
IV.2.1 Présentation de WOLF 41
IV.2.2 L'approche de construction de Wolf 42
IV.2.3 Structure de Wolf 42
Conclusion 43

3 Réalisation **44**
I Introduction 44
II Description de la construction initiale de DICO 45
II.1 Processus de construction de DICO 45
II.2 La structure de DICO 46
III Construction du corpus EDUCA 47
III.1 Rappel sur les démarches de construction du corpus 47
III.1.1 Démarche de construction retenue 48
IV Lemmatisation du corpus EDUCA 53
V Entrainement de WEKA avec le corpus EDUCA 55
V.1 Interface utilisateur 55
V.2 Présentation de l'interface graphique Explorer 57
V.3 Format des données d'entrées de WEKA 58
V.4 Démarche de construction des modèles de classifications : 59
V.4.1 Entrainement des classifieurs 60
V.4.2 Evaluation des modèles obtenus 61
V.5 Le modèle basé sur KNN 62
V.6 Le modèle basé sur SVM 63
V.7 Le modèle basé sur arbre de décision 64
V.8 Le modèle basé Naïve bayes 65
V.9 Récapitulation des indicateurs de performance obtenus 66
VI Recalcule des polarités d'opinion de DICO 66

VI.1 Statistique de DICO 70
Conclusion 71

4 Evaluation **72**
Introduction 72
I Description de la démarche 72
II Classification selon la ressource lexicale DICO 73
II.1 Classification selon DICO initial 74
II.2 Classification selon le nouveau DICO 75
III Discussion des résultats 76
Conclusion 76

Conclusion générale **77**

A le programme de conversion de fichier txt en fichier Arff **83**

B le programme de classification du corpus selon DICO **87**

C le programme d'evaluation **92**

Introduction générale

La révolution de l'information bousculée par le développement à grande échelle de l'internet ou intranet a fait exploser la quantité d'informations, ce qui a permit à l'internaute de ne plus être simple spectateur mais de devenir un acteur qui crée son espace sur Internet. Il peut, collaborer, partager et échanger des informations, des outils, des fichiers multimédias, donner ses opinions, commenter, réagir, etc. Et tout ceci sans connaissances spécifiques.

De nombreux sites présents sur Internet aujourd'hui offrent la possibilité à tous leurs visiteurs de laisser, au minimum, une trace textuelle et ainsi s'exprimer librement, comme est le cas des étudiants qui expriment leurs avis, leurs critiques dans les groupes de discussion en sortes : des blogs personnels, des commentaires, des forums, citations d'avantages ou des inconvénients rédigés dans un langage libre dit naturel.

Ces données textuelles, peuvent être analysées dans différents buts pour différents domaines où chaque contenu exprimé par les utilisateurs sous forme de données textuelles engendre un ensemble d'opinions et de sentiments.

Le domaine de la fouille d'opinion est un sous domaine du text-mining qui s'intéresse principalement à l'extraction et la classification d'opinion. Il permet d'identifier la position d'un émetteur d'opinion sur un sujet donné par le biais d'approches et outils. La fouille d'opinion repose sur trois principales méthodes, lexicales, statistiques et hybrides. Ces méthodes nécessitent des ressources textuelles pour leur fonctionnement.

La fouille d'opinion trouve des applications dans différents domaine comme celui du

commerce, politique, médical et même éducatif [1]. Néanmoins, les ressources nécessaires pour la classification des opinions sont rattachées à ces domaines vu que l'orientation des mots peut changer selon le contexte dans lequel ils sont employés.

Notre travail s'inscrit principalement dans le contexte de la fouille d'opinion dans le domaine de l'éducation et notre objectif est de contribuer dans la construction d'une ressource lexicale destinée précisément à l'analyse d'opinion dans le domaine de l'éducation.

Pour ce faire, nous avons organisé notre mémoire en quatre chapitres, le premier aborde les concepts de base de la fouille d'opinion. Nous donnons les définitions les notions fondamentales de la fouille d'opinion, et nous mettons en évidence ses domaines d'application.

Le deuxième chapitre présente les méthodes de classification d'opinion ainsi que la construction de corpus pour finir par présenter la plateforme WEKA destiné à la classification d'opinion basée sur des méthodes statistiques.

Le troisième chapitre est consacré à la présentation de notre proposition. Nous développons la démarche de construction initiale de la ressource lexicale DICO [2]. Nous abordons ensuite la construction d'un corpus que nous nommons EDUCA à partir de textes issus du domaine de l'éducation. Ce corpus sert à entrainer des classifieurs dans la plateforme WEKA pour générer quatre modèles de classification d'opinion. Ces modèles sont utilisés pour recalculer les polarités d'opinion de DICO.

Le dernier chapitre est dédié à la validation de notre proposition par la comparaison entre la classification du corpus EDUCA selon les polarités initiales de DICO et selon ses nouvelles polarités.

Chapitre 1

Fouille d'opinion

I Introduction

Le développement d'internet à grande échelle, à engendré la disponibilité des avis et des points de vue des personnes sous plusieurs formes. D'ailleurs, il existe un nombre important de documents textuels qui sont porteurs d'opinions et de sentiments, d'où la nécessité de l'apparition d'une discipline qui s'occupe du traitement de la subjectivité.

La fouille d'opinion est une sous-discipline récente de la linguistique informatique qui s'intéresse à l'opinion véhiculé par le document qu'au sujet sur lequel porte. Elle se concentre sur l'identification et la classification des opinions dans les données textuelles. Elle consiste à analyser une grande quantité de données (données textuelles) afin d'en déduire les différents sentiments qui y sont exprimés.

II Notions de base

La terminologie utilisée en fouille d'opinion est multiple : subjectivité, opinion, sentiment etc.

II.1 Fouille de texte

La fouille de textes, ou text mining en anglais, est l'extraction de connaissances à partir de textes en langage naturel. Elle est une spécialisation de la fouille de donnée au sens où elle est l'extension du même but et du même processus vers des données textuelles. L'objectif de la fouille de textes est le traitement de grandes quantités d'information qui sont disponibles sous une forme textuelle non structurée. Notons, que la fouille d'opinion est un sous domaine de la fouille de texte [2].

II.2 subjectivité

La subjectivité se définie comme la capacité de la personne à exprimer linguistiquement ses opinions, sentiments, émotions, évaluations, croyances, spéculations (états privés), de là, un état privé a été définit comme étant un état qui n'accepte pas d'observation objective et moins une vérification[3].
On peut distinguer deux niveaux de subjectivité dans le langage :

- Le premier niveau n'implique pas l'expression d'une évaluation. Il témoigne simplement du degré de présence de l'énonciateur dans son énoncé. Cette présence peut être implicite ou bien explicite en fonction de la présence ou l'absence de certains marqueurs [4].

- Le second niveau est celui des évaluations exprimées par l'énonciateur. Elles se caractérisent par la présence d'un prédicat exprimant l'évaluation. Ce prédicat peut avoir ou non une valeur axiologique (positif, négatif, neutre...) [4].

II.3 Faits et opinions

L'information textuelle est généralement repartitionnée en deux catégories principales qui sont : faits et opinions [5].

- **Faits :**Il s'agit des énoncés objectifs sur les événements et leurs propriétés [5].

- **Opinions :**Il s'agit d'expressions subjectives décrivant les sentiments d'une personne envers une entité ça peut être un jugement, un avis..etc. ces opinions peuvent

être décrites avec certains attributs. L'attribut d'opinion le plus étudié est sans doute la polarité (positive, négative et parfois neutre) qui définit si l'opinion est favorable ou défavorable [5].

II.4 Opinion mining et analyse de sentiment

L'étude des opinions et sentiments dans les textes est un nouveau axe de recherche qui est apparu au début des années 2000.plusieurs termes sont utilisés pour le décrire, mais ces derniers ne sont pas toujours normés. Les termes plus couramment utilisés sont ceux de "fouille d'opinion" et "d'analyse des sentiments" parfois réunis sous l'appellation d'analyse de la subjectivité", parfois utilisés de manière interchangeable. Le terme "fouille d'opinion" est apparu pour la première fois et a été plus utilisé par la communauté de recherche d'information. Le terme "d'analyse des sentiments" a quant à lui fait son apparition plus au cœur de la communauté de traitement automatique des langues. Pour la suite de notre travail nous allons retenir l'appellation de " fouille d'opinion "[4]. opinion mining et sentiment analysis opèrent dans le même champ car l'analyse de sentiment permet de spécifier ce qui est important dans l'analyse d'opinion à savoir les jugements de valeur positifs ou négatifs portés sur des entités[6].

III Définition de fouille d'opinion

La fouille d'opinion est une discipline récente qui s'intéresse particulièrement au traitement automatique des opinions, des sentiments et de la subjectivité dans les textes[7].

C'est un domaine de recherche qui vise à récolter les avis des utilisateurs afin de procéder à l'identification et la classification des opinions exprimés dans ces données textuelles [6]. Il a été récemment interprété de manière plus générale pour inclure de nombreux types d'analyse d'évaluation de texte [7].

L'analyse d'opinion peut être étudiée selon trois axes principaux à savoir : les tâches de l'analyse d'opinion, les approches qu'elle met en oeuvre et ses applications comme illustré dans la figure 1.1.

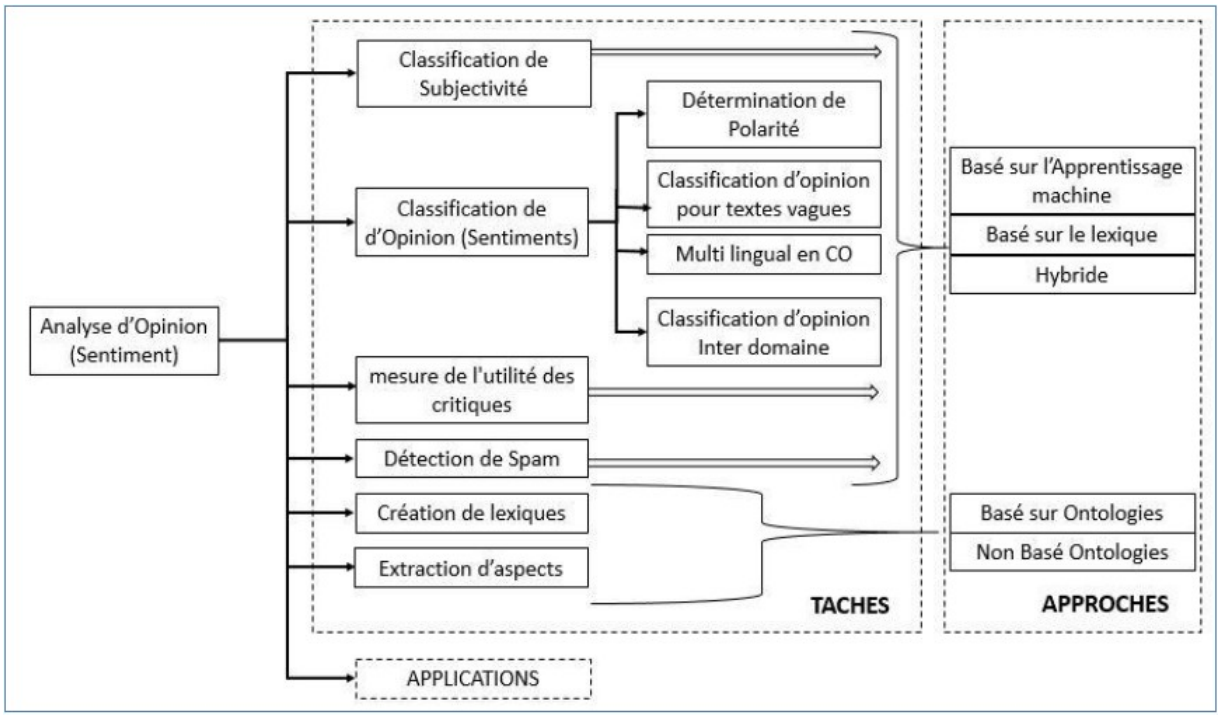

FIGURE 1.1 : Axes de la fouille d'opinion. [2].

IV Les taches de fouille d'opinion

Elle se compose de plusieurs tâches, qu'il est utile ou non de mettre en œuvre selon les applications visées [4] :

- Détection de la présence ou non de l'opinion,donc la subjectivité du document.
- Classification de l'axiologie de l'opinion c'est-à-dire décider si un texte subjectif donné exprime une opinion positive, négative ou même neutre sur son sujet.
- Classification de l'intensité de l'opinion. S'il est faiblement positif, légèrement positif ou fortement positif, ce qu'on appelle la force de l'opinion.
- Identification de l'objet de l'opinion, ce sur quoi porte l'opinion.
- Identification de la source de l'opinion,qui exprime l'opinion.

Toutes ces tâches peuvent se pratiquer à différents niveaux selon les applications envisagées.

V Processus de fouille d'opinion

La détection d'opinions est une tâche qui permet d'extraire les opinions d'un ensemble de documents pertinents (des corpus récupérés à partir de différentes sources comme par exemple internet) pour un sujet donné.
La fouille d'opinion passe par un ensemble d'étapes essentielles, illustrées dans la Figure 1.2 :

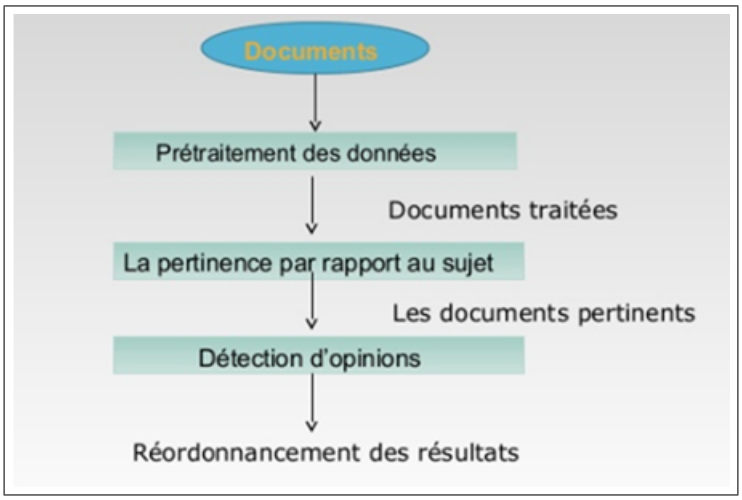

FIGURE 1.2 : Processus de fouille d'opinion[8].

V.1 L'acquisition et le prétraitement des données

Une étape cruciale dans l'automatisation des tâches, offrant la possibilité d'accès aux différentes données répartis sur différents sites afin de construire des corpus de données spécifiques.

Une fois que les données soient collectées, place à l'acquisition du corpus, où un prétraitement du langage naturel serait effectué via un ensemble de traitement [8] :

- Elimination de mots vides, caractères spéciaux. . . etc.
- Effectuer une analyse lexicale pour éliminer les répétitions.
- Structurer la phrase de façon hiérarchique.

V.2 La pertinence par rapport au sujet

Cette phase est considérée comme étant une recherche d'information qui se base sur la définition de quelques modèles, qui permettent l'accès facile à un ensemble de documents sur le web, il fait en sorte de retourner l'ensemble de documents pertinents dont le contenu répond aux besoins d'utilisateurs.

La pertinence des documents par rapport à un sujet donné est étudiée en utilisant différentes méthodes comme la méthode probabiliste. Les documents seront classés, et les plus pertinents sont extraits, et seront utilisés pour la prochaine étape [8].

V.3 La détection d'opinion

Dans cette phase, plusieurs méthodes de classification sont utilisées afin de réordonner les documents pertinents selon un score d'opinion [8].

VI Domaine d'application de fouille d'opinion

L'importance de la détection d'opinion est présente dans plusieurs domaines, nous allons citer quelques cas comme on peut le voir dans la figure1.3.

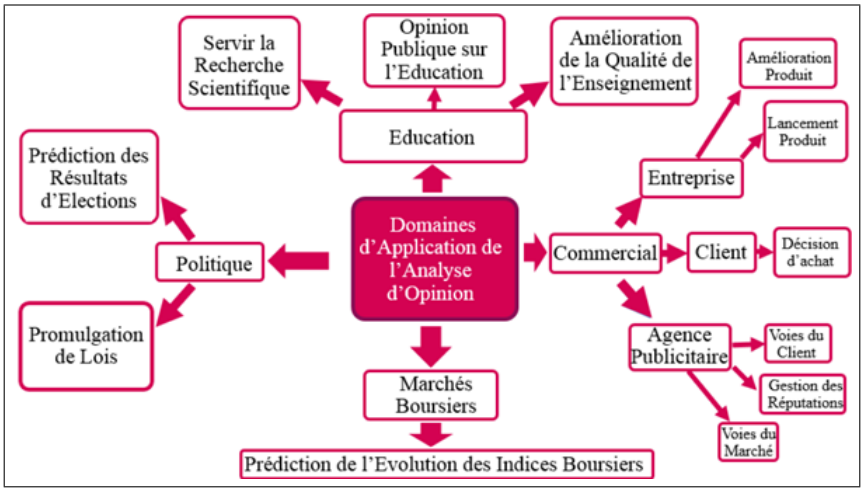

FIGURE 1.3 : Domaine d'application de fouille d'opinion[2].

VI.1 Domaine politique

L'avis des internautes est important pour les politiciens lors de la promulgation d'une nouvelle loi, ainsi que lors d'une élection présidentielle, il est nécessaire de connaitre l'avis des internautes sur les hommes politiques [1].

VI.2 Domaine de l'éducation

Plusieurs travaux ont été menés dans ce domaine. Par exemple, il est important de savoir l'avis des étudiants sur la méthode d'enseignement afin de l'améliorer[2].

VI.3 Domaine commercial

L'utilisation de la fouille d'opinion dans le domaine commercial est importante à différents points de vue citons :

VI.3.1 Point de vue des entreprises

À travers la fouille d'opinions, les entreprises peuvent connaître mieux le consommateur et savoir son avis sur le produit afin de l'améliorer de sorte à satisfaire le plus de consom-

mateurs.

Le domaine de la fouille d'opinion est un outil majeur pour toute entreprise désireuse de mieux comprendre ce qui plait et déplait à ses clients [1].

VI.3.2 Point de vue des Clients

Les clients s'inspirent des opinions des autres gens sur un produit donné en comparant les avis négatifs et positifs afin de prendre une décision d'acheter ou non. Et de son coté, ce client aussi donne son opinion [8].

VII Difficultés de la fouille d'opinion

La classification d'opinion est le plus souvent décrite par une polarité, qui est en générale soit positive, négative ou neutre. Cependant, cette classification rencontre quelques difficultés :

- La difficulté de décider de la polarité de certains mots à cause de leur ambiguïté dans certains contextes [3].
- Difficulté due au contexte : l'importance de l'analyse syntaxique du texte qui peut aider à trouver les expressions qui contiennent des opinions. Cette analyse peut s'avérer particulièrement difficile dans le cas de la coordination entre les parties d'une phrase [9].
- Difficulté due au langage naturel pour l'analyse automatique de sentiments selon les contextes intentionnels, pour lesquels l'expression d'opinion n'est pas un vrai sentiment [8].
- Difficulté due à la structuration de la phrase. Dans le cas d'une phrase se composant de deux parties liées par une conjonction mais, alors la polarité de la deuxième partie est opposée à la première [8].
- Difficulté due au vocabulaire utiliser pour l'expression de l'opinion. Ce dernier diffère d'une personne à une autre chacun sa façon d'exprimer ses sentiments [8].

- Difficulté due à l'utilisation d'une thématique, cette dernière peut être utilisé dans différent contexte pour exprimer différentes significations [8].
- Difficulté de déterminer un lexique adapté à l'analyse de l'ensemble des textes d'opinion [8].
- Difficulté à trouver une association entre l'opinion et la requête. l'opinion acheminée dans un document ne porte pas forcement sur la requête considéré [8].

Conclusion

Dans ce chapitre, nous avons présenté les principales notions et différents concepts propre à l'analyse de l'opinion entre subjectivité, opinion mining et sentiments analysis, nous avons abordé la distinction entre l'opinion et fait, nous avons expliqué le processus de fouille d'opinion et ses étapes, nous avons montré son domaine d'application ainsi que les difficultés qui empêche son bon acheminement.

Chapitre 2

Méthodes de fouille d'opinion

I Introduction

L'exploitation de donnée manuellement à montré des limites rapidement ce qui a poussé à l'utilisation des approches automatiques qui facilitent la fouille d'opinion. Plusieurs méthodes de ce domaine sont apparues pour rendre l'analyse d'opinion rentable en termes de temps et de coût. Ces méthodes seront détaillées dans ce qui suit.

II Les méthodes de détection d'opinions

Il existe trois différents types de méthodes pour la détection d'opinions qui donnent lieu à des techniques variées de classification et de traitement de l'opinion :

II.1 Les méthodes symboliques

Appelées aussi " approches linguistiques ou approches basées sur le lexique". Ces méthodes utilisent des ressources lexicales qui se présentent sous forme de dictionnaire d'opinion ou lexique. Les ressources lexicales sont des sortes de rassemblement de connaissances sur les mots, leurs sens et leurs usages etc. Pendant des siècles elles ont été sous format textuel, ce qui a changé de nos jours puisque une grande variété d'outils et de ressources accessibles sous des formats électroniques existent [10].

Afin de construire ces ressources lexicales, elle s'appuie généralement sur des systèmes d'extraction d'information. Ces systèmes sont fondés sur une analyse syntaxique du texte faite par un analyseur lexico-syntaxique qui contient un lexique de mots qui utilise des règles de grammaire. Un grand nombre de mots porteurs d'opinions est ainsi généré et rangé dans ces dictionnaires d'opinions [11].

Dans ces ressources lexicales, une polarité est associée a priori à chacun des mots [3]. Ces mots vont servir à classer les textes en trois catégories (positives, négatives et neutres) [10] selon le score d'opinion d'un document. Le score d'opinion est calculé en fonction du nombre total de mots dans le texte issus de ces dictionnaires (qui contiennent une opinion) suivant une méthode simple qui se résume à donner à chaque document un score d'opinion égal au score d'opinion des mots majoritaire [3].

II.1.1 Méthodes de construction de ressource lexicale

Pour construire une ressource lexicale d'opinion trois genres de techniques sont utilisées [10] :

- Les méthodes manuelles.
- La méthode basée sur les corpus.
- La méthode basée sur les dictionnaires.

II.1.1.1 Les méthodes manuelles

Cette méthode consiste à enrichir le lexique de mots d'opinions sans aucun outil particulier, seulement les experts font la sélection de mots et expressions porteurs d'opinions. Cet ensemble de mots est appelé graine ou germe (en anglais, seedwords). Ils construisent une première liste de mots et d'expressions qu'ils vont utiliser par la suite à trouver, répertorier et classer d'autres mots et expressions porteurs d'opinions. L'inconvénient de cette méthode est qu'elle demande un temps important [10] et elle est très coûteuse [12].

II.1.1.2 La méthode basée sur les corpus

Avec cette méthode, les mots qui contiennent une opinion sont extraits directement de corpus. Cette méthode consiste à utiliser les conjonctions de coordination présentes entre un mot déjà classé et un mot non classé. Par exemple, si la conjonction ET sépare un mot classé positif dans la ressource lexicale d'opinion et un mot non classé, alors le mot non classé sera considéré comme étant positif. À l'inverse, si la conjonction MAIS sépare un mot classé positif et un mot non classé, alors le mot non classé sera considéré comme étant négatif. Les conjonctions utilisées sont les suivantes : ET, OU, MAIS, OU BIEN, et NI-NI [10].

II.1.1.3 La méthode basée sur les dictionnaires

Cette méthode consiste à utiliser des dictionnaires qui peuvent être externes c'est-à-dire construits indépendamment de tout corpus, comme ils peuvent être généraux (SentiWordNet,SUBJ lexique, General Inquiry, Wilson lexicon) [10]. Dans ces dictionnaires, une polarité est associée a priori à chacun des mots. On donne ensuite au document un score d'opinion en fonction de la présence de mots issus de ces dictionnaires dans le texte. Ils sont souvent utiliser pour classifier du texte et déterminer l'orientation sémantique des nouveaux mots [3].

II.2 Les méthodes statistiques

II.2.1 Introduction

Les méthodes statistiques sont appelées aussi " approches basées sur l'apprentissage machine ou approches basées sur corpus ou encore classification supervisé". Ce sont des techniques d'apprentissage automatique qui se base sur des approches mathématiques et statistiques pour donner aux ordinateurs la capacité d' " apprendre " à partir de données [2].

Ces méthodes se focalisent sur la classification, qui a comme objectif la construction d'un modèle de classification à l'aide de corpus d'apprentissage dont on connaît déjà le label qui va permettre par la suite de prédire l'appartenance d'un nouvel exemple qui peut être commentaire non étiqueté à une classe [10].

Ces méthodes utilisent des "features" pour l'apprentissage tels que les bigrammes, les n-grammes, POS étiquettes morpho-syntaxiques etc. Ainsi que plusieurs types de classifieurs tels que : SVM, Naïve Bayes, Multiples Classifieur, Naïfs de Bayes, ainsi que la régression logistique [10].

Ces approches classent les documents ou les mots selon deux axes de classification, soit selon subjectivité ou objectivité du texte, soit selon les opinions exprimés positif ou négatif. Dans le cas de classification des opinions, plusieurs techniques d'apprentissage automatique sont utilisées :

II.2.2 Techniques d'apprentissage automatiques

L'apprentissage automatique (AM) est un domaine de l'informatique qui utilise des techniques statistiques pour donner aux systèmes informatiques la capacité d'" apprendre ", sans être explicitement programmés. Toutefois, les machines ont besoin de données à analyser et sur lesquelles s'entraîner pour " apprendre " et construire des modèles.

Il existe principalement deux types de techniques d'apprentissage automatique : l'apprentissage supervisé et l'apprentissage non supervisé.

II.2.2.1 Méthodes d'apprentissage supervisées

L'apprentissage supervisé ou supervised learning en anglais, est une forme d'apprentissage machine qui se base sur des données d'apprentissage " étiquetées "afin de créer des modèles d'intelligence artificielle [13].

Dans l'apprentissage supervisé, le processus d'apprentissage se base sur un ensemble d'exemples d'apprentissage, où chaque exemple est un couple constitué d'un objet d'entrée qui est généralement un vecteur et d'une valeur de sortie souhaitée également appelée signal de supervision [13]. Un algorithme d'apprentissage supervisé analyse les données d'apprentissage et produit un modèle, qui peut être utilisée pour mapper de nouveaux exemples. Les classes sont initialement spécifiées et les données d'apprentissage sont attribuées à des classes spécifiques [13].

Les méthodes d'apprentissage supervisé reposent sur deux types d'approche probabiliste et non probabiliste de classification :

- **Probabiliste :** Ces approches sont dérivées de modèles probabilistes. Elles permettent la réalisation d'une classification statistique dans les domaines tels que le langage naturel. Par conséquent, il a une application réelle dans l'analyse d'opinion. Parmi les méthodes d'analyse d'opinion les plus connues dans cette catégorie : Naïve Bayes, Réseaux Bayésiens et Entropie Maximale [13].

- **Non probabiliste :** Il existe aussi des méthodes d'apprentissage automatique supervisé non probabilistes. Les classifieurs de cette catégorie, les plus connus et les plus utilisés en analyse d'opinion sont : les réseaux de neurone, machine à vecteurs de support (SVM), le plus proche voisin, arbre de décision et les méthodes basées sur des règles [2].

II.2.2.1.1 Les algorithmes d'apprentissage

Différents types de classifieurs ont été mis au point afin d'atteindre un degré maximal de précision et d'efficacité, chacun d'eux dispose d'un ensemble d'avantage et d'un ensemble d'inconvénients, tout en partageant quelques caractéristiques communes. Les algorithmes de classifications sont souvent regroupés en familles, ce qui permet de distinguer plusieurs grandes familles. Etant donné que nous allons travailler avec le logiciel WEKA, nous allons présenter quelques algorithmes parmi ces familles :

II.2.2.1.1.1 Classification bayesienne naïve

Les classifieurs bayésiens naïfs sont des classifieurs probabilistes se basant sur le théorème de Bayes. Ils considèrent qu'il y a une forte indépendance entre les différentes caractéristiques d'une classe.

Un classifieur bayésien calcule la probabilité d'un commentaire de faire partie de chaque classe en connaissant ses autres attributs, puis l'étiquette en fonction de la classe ayant la plus haute probabilité [14]. Classifieurs bayésiens naïfs sont très efficaces, même avec de faibles quantités de données. Le théorème de Bayes permet de calculer la probabilité postérieure P (c | x) à partir de P (c), P (x) et P (x | c). En utilisant l'équation ci-dessous :

$$P(c|x) = \frac{P(c|x)P(c)}{P(x)} \tag{2.1}$$

où $P(c|X) = P(x1|c) * P(x2|c) * \ldots * P(xn|c) * P(c)$

- P (c | x) est la probabilité postérieure de classe (c, cible) donnée prédicteur (x, attributs).
- P (c) est la probabilité a priori de classe.
- P (x | c) est la vraisemblance qui est la probabilité du prédicteur de la classe donnée.
- P (x) est la probabilité antérieure du prédicteur.

Le classificateur Naive Bayes calcule la probabilité d'un événement dans les étapes suivantes :

- Étape 1 : Calculer la probabilité antérieure pour des étiquettes de classe données.
- Étape 2 : Trouver la probabilité de vraisemblance avec chaque attribut pour chaque classe.
- Étape 3 : Mettre ces valeurs dans la formule de Bayes et calculer la probabilité postérieure.
- Étape 4 : voir quelle classe a une probabilité plus élevée, étant donné que l'entrée appartient à la classe de probabilité la plus élevée.

II.2.2.1.1.2 Arbre de décision

L'idée générale d'un algorithme de règles de décision, consiste globalement à trouver un ensemble de règles qui sépare complètement les deux classes. Ensuite, on divise l'ensemble des règles en plusieurs petits ensembles qui contiennent, évidement, moins de règles que l'ensemble de départ. Nous répétons cette étape d'élagage jusqu'à avoir une seule règle dans chaque ensemble. Enfin, on évalue ces ensembles de règles et on sélectionne le meilleur ensemble comme une solution finale. Ce concept d'élagage est partagé

par de nombreux programmes d'apprentissage automatique comme C4.5, CART et l'algorithme ID3. Ce type d'élagage nous permet de créer des ensembles de règles de taille différente puis choisir un seul ensemble, comme solution au problème, par l'établissement d'une norme.
Les algorithmes utilisant l'approche de règles de décision se basent généralement sur le modèle d'arbre de décision [15].

1. **Définition**
   Les arbres de décision représentent une méthode très efficace d'apprentissage supervisé. Il s'agit de partitionner un ensemble de données en des groupes les plus homogènes possible du point de vue de la variable à prédire. On prend en entrée un ensemble de données classées, et on fournit en sortie un arbre qui ressemble beaucoup à un diagramme d'orientation où chaque nœud final c'est à dire feuille représente une décision qui peut être une classe : négatif, positif ou neutre, et chaque nœud non final c'est-à-dire interne représente un test. Chaque feuille représente la décision d'appartenance à une classe des données vérifiant tous les tests du chemin menant de la racine à cette feuille. Pour construire l'arbre de décision, il faut trouver quel attribut tester à chaque nœud, c'est un processus récursif. Pour déterminer quel attribut tester à chaque étape, on utilise un calcul statistique qui détermine dans quelle mesure cet attribut sépare bien les exemples oui/non [15].

2. **Algorithme générique**
   Idée centrale : Diviser récursivement et le plus efficacement possible les commentaires de l'ensemble d'apprentissage par des tests définis à l'aide des attributs jusqu'à ce que l'on obtienne des sous-ensembles d'exemples ne contenant presque que des exemples appartenant tous à une même classe. Dans toutes les méthodes, on trouve les trois opérateurs suivants [16] :
   Décider si un nœud est terminal : c'est-à-dire décider si un nœud doit être étiqueté comme une feuille. Par exemple : tous les exemples sont dans la même classe, il y a moins d'un certain nombre d'erreurs, ...
   Sélectionner un test à associer à un nœud : Par exemple : aléatoirement, utiliser des critères statistiques, ...
   Affecter une classe à une feuille : On attribue la classe majoritaire sauf dans le cas où l'on utilise des fonctions coût ou risque. Les méthodes vont différer par les

choix effectués pour ces différents opérateurs, c'est-à-dire sur le choix d'un test par exemple, utilisation du gain et de la fonction entropie et le critère d'arrêt quand arrêter la croissance de l'arbre, soit quand décider si un nœud est terminal.

3. **Algorithme C4.5**
   Cet algorithme a été proposé en 1993, toujours par Ross Quinlan, pour pallier les limites de l'algorithme ID3 comme :

   - Une adaptation de la fonction de gain qui n'a plus tendance à aller vers l'attribut avec le plus de valeurs possibles.
   - La possibilité de gérer des attributs avec des valeurs manquantes.
   - La possibilité de post-élaguer son arbre pour éviter " l'overfitting ".
   - La possibilité de manipuler des valeurs continues en les " discrétisant " lors de la mise en arbre.

   Le déroulement de l'algorithmeC4.5 se passe en grande partie comme l'ID3. Les différences se trouvent en plusieurs endroits :

   - Quand on rencontre une valeur nulle, on ne prend pas en compte l'enregistrement pour les calculs sur ce champ.
   - Quand on rencontre un champ continue, on procède à la discrétisation, il faut noter que cette opération est souvent répétée plusieurs fois.
   - Quand on rentre dans la création d'un sous-arbre on vérifie qu'il n'a pas besoin d'être élaguer en comparant le cardinal de chacune des classes de la valeur " cible "
   - Le calcul du ratio de gain remplace le calcul du gain.

   Ces améliorations sont très importantes dans le sens ou ID3 est inutilisable dans la pratique sans la gestion des valeurs nulles et même souvent sans le remplacement du gain par le ratio du gain [17].

II.2.2.1.1.3 Machine à support vectorielle

Les machines à support de vecteurs, elles sont sous le nom SMO dans weka sont à l'origine

de nouvelles méthodes de catégorisation, bien que les premières publications sur le sujet datent des années 60.
Avant d'aborder le principe de fonctionnement général des SVM, voici ses notions de base :

- Hyperplan : un séparateur d'objets des classes. De cette notion, nous pouvons dire qu'il est évident de trouver une mainte d'hyperplan mais la propriété délicate des SVM est d'avoir l'hyperplan dont la distance minimale aux exemples d'apprentissage est maximale, cet hyperplan est appelé l'hyperplan optimal, et la distance appelée marge [18].

- Vecteurs support : ce sont les points qui déterminent l'hyperplan tels qu'ils soient les plus proches de ce dernier [18].

Les SVM sont des algorithmes qui utilisent une transformation non linéaire des données d'apprentissage. Ils projettent les données d'apprentissage dans un espace de plus grande dimension que leur espace d'origine. Le principe de SVM consiste en une stratégie de minimisation structurelle du risque mais le problème revient à trouver une frontière de décision qui sépare l'espace en deux régions, Dans ce nouvel espace, ils cherchent l'hyperplan qui permet une séparation linéaire optimale des données d'apprentissage en utilisant les vecteurs de support et les marges définies par ces vecteurs [15].

Dans le cas de la catégorisation de texte, les entrées sont des commentaires et les sorties sont des catégories. En considérant un classificateur binaire, on voudra lui faire apprendre l'hyperplan qui sépare les commentaires appartenant à la catégorie et ceux qui n'en font pas partie [18].

Les SVM conviennent bien pour la classification de textes parce qu'une dimension élevée ne les affecte pas puisqu'ils se protègent contre le sur apprentissage. Autrement dit, il affirme que peu d'attributs sont totalement inutiles à la tache de classification et que les SVM permettent d'éviter une sélection agressive qui aurait comme résultat une perte d'information. On peut se permettre de conserver plus d'attributs. Également, une caractéristique des documents textuels est que lorsqu'ils sont représentés par des vecteurs, une

majorité des entrées sont nulles [18].

L'ensemble de données d'entrainement de ce classifieur sera présenté comme une matrice binaire, indiquant si un tel terme apparait dans une catégorie ou non [18].

II.2.2.1.1.4 K-voisins

1. **Définition**
   L'algorithme des k plus proches voisins est un algorithme d'apprentissage supervisé, qui figure parmi les plus simples algorithmes d'apprentissage artificiel il est nécessaire d'avoir des données labellisées. À partir d'un ensemble E de données labellisées, il sera possible de classer une nouvelle donnée qui n'appartient pas à E. À noter qu'il est aussi possible d'utiliser l'algorithme des k plus proches voisins à des fins de régression où on cherche à déterminer une valeur à la place d'une classe. L'algorithme des k plus proches voisins est une bonne introduction aux principes des algorithmes d'apprentissage automatique [19]. La méthode KNN est donc une méthode à base de voisinage, non-paramétrique ; Ceci signifiant que l'algorithme permet de faire une classification sans faire d'hypothèse sur la fonction y=f(x1,x2,...xp) qui relie la variable dépendante aux variables indépendantes [20].

2. **Principe de l'algorithme**
   L'algorithme de k plus proches voisins ne nécessite pas de phase d'apprentissage à proprement parler, il faut juste stocker le jeu de données d'apprentissage. Soit un ensemble E contenant n données labellisées : E = (yi , $\overrightarrow{xi}$) avec i compris entre 1 et n, où yi correspond à la classe (le label) de la donnée i et où le vecteur $\overrightarrow{xi}$ de dimension p ($\overrightarrow{xi}$ = (x1i , x2i , ..., xpi)) représente les variables prédicatrices de la donnée i. Soit une donnée u qui n'appartient pas à E et qui ne possède pas de label (u est uniquement caractérisé par un vecteur $\overrightarrow{xu}$ de dimension p). Soit d une fonction qui renvoie la distance entre la donnée u et une donnée quelconque appartenant à E. Soit un entier k inférieur ou égal à n. Voici le principe de l'algorithme de k plus proches voisins [20] :

   - On calcule les distances entre la donnée u et chaque donnée appartenant à E à l'aide de la fonction*d*

- On retient les k données du jeu de données E les plus proches de u
- On attribue à u la classe qui est la plus fréquente parmi les k données les plus proches.

II.2.2.2 Méthodes d'apprentissage semi-supervisées

L'apprentissage semi supervisé utilise un ensemble de données étiquetées et non-étiquetées. Il se situe ainsi entre l'apprentissage supervisé qui n'utilise que des données étiquetées et l'apprentissage non-supervisé qui n'utilise que des données non-étiquetées. Le fait de lié l'apprentissage supervisé au non supervisé permet d'avoir de meilleures performances [2].

II.2.2.3 Méthodes d'apprentissage non-supervisées

Contrairement à l'apprentissage supervisé, l'apprentissage non-supervisé n'a pas de sortie cible associée à l'entrée. Le clustering est une technique utilisée dans l'apprentissage non supervisé. Elle consiste à placer des données dans différents groupes dans lesquels les membres de chaque groupe sont similaires entre eux. Ces méthodes évoquent souvent la notion de similarité entre les documents comme critère de regroupement [6].

Les techniques d'apprentissage automatique sont une partie essentielle d'un nombre croissant d'applications en science, elles permettent en partie l'exploration de données et l'exploration du Web en utilisant des techniques d'apprentissage-machine de base. Il existe plusieurs logiciels comportant un nombre important d'algorithmes qui peuvent aider à l'apprentissage machine comme CRF++, Wapiti, LibSVM, SVMLight, WEKA. Pour la suite de notre travail nous avons choisis d'utiliser WEKA.

II.2.3 Outil logiciel WEKA

WEKA (Waikato Environment for Knowledge Analysis) est un logiciel libre et gratuit qui implémente plusieurs algorithmes d'apprentissage permettant de manipuler et d'analyser des fichiers de données. Cet outil a été développé en java par une équipe de chercheurs de l'université de Waikato en nouvelle Zélande en 1992 [21].

II.2.3.1 Traitement de données

Le logiciel WEKA se présente comme une solution efficace pour le traitement de données même celles de grandes envergures appelées " Big Data ". Pour être traitées, les données doivent être entrées sous les formats ARFF, CSV, Binaire, BDD, SQL ou URL. Le format le plus utilisé est le format ARFF (Attribute-Relation File Format). Les données sous ces formats sont compatibles si elles sont bien structurées. La structure des données se composent de la séquence :

- De noms des données(@relation).
- Des attributs (@attribut).
- De la variable de classe à prédiction (@data).

Les données peuvent être de types :

- numérique continue.
- numérique discrète.
- catégorie, avec ou sans relation d'ordre.
- binaire (vrai/faux).

Les données structurées : arbres, graphes. Les attributs sont des réels (real), des chaines de caractères (string) et des dates (date) [6].

II.2.3.2 Les méthodes de traitement de données dans WEKA

Le logiciel WEKA présente un ensemble de méthodes liées au datamining qui sont : la classification, le clustering et l'association où chaque méthode propose un ensemble d'algorithmes.

II.2.3.2.1 La classification :

Il s'agit d'une tâche d'analyse de données, considéré comme un processus de recherche d'un modèle qui décrit et distingue les classes de données et les concepts. Elle consiste à étudier les caractéristiques d'un nouvel objet, pour lui attribuer une classe prédéfinie. WEKA propose 134 algorithmes subdivisés en 8 méthodes : Bayes, function, trees, lazy, rules, meta, multi-instance et miscellaneous [22].

II.2.3.2.2 Le clustering :

Est une méthode d'analyse statistique utilisée pour organiser des données brutes. A l'intérieur de chaque grappe, les données sont regroupées selon une caractéristique commune. L'outil d'ordonnancement est un algorithme qui mesure la proximité entre chaque élément à partir de critères définis [23]. WEKA propose 13 algorithmes assurant cette tâche.

II.2.3.2.3 L'association :

Contenir un ensemble de règles d'association permettant de détecter les relations entre variables. WEKA propose 7 règles d'associations.

II.2.3.3 L'apprentissage automatique proposé par WEKA

Le logiciel WEKA contient deux sortes d'apprentissage : l'apprentissage supervisé et l'apprentissage non supervisé. L'apprentissage est fait par regroupement de données en sous ensemble cohérents ou clusters. Généralement en entrée nous avons un ensemble de données non étiquetées et en sortie plusieurs sous ensemble de données cohérents.

II.2.3.3.4 Apprentissage non-supervisé : l'objectif principal consiste à former des groupes "cluster" à partir d'un ensemble de documents et ensuite, assigner une catégorie à chaque groupe [15].

II.2.3.3.5 Apprentissage supervisé : implique l'apprentissage d'une correspondance entre un ensemble de variables d'entrée et une grandeur de sortie, il est appliqué pour prédire les sorties de données invisibles. C'est la méthodologie la plus importante dans la machine apprentissage, il met en œuvre plusieurs algorithmes, parmi eux on trouve BayesNaive, SMO, KNN, arbre de décision, réseau de neurones . . . etc [24].

II.3 Les points forts et points faibles des méthodes de fouille d'opinion

Chacune des méthodes étudiées dessus possède un ensemble de points forts et de points faibles résumé dans le tableau 2.1 :

Méthode	**Points forts**	**Points faibles**
Méthode statistique	-Possède la capacité d'analyser de nombreuses catégories. -Efficacité dans la classification de la subjectivité. -Bonne performance notamment lors de l'ambigüité. -Aide à atteindre de plus haute précision avec seulement un petit effort humain d'annotation. -Efficace et largement applicable.	-Dépendance à la formation de documents étiquetés. -Nécessite un effort humain considérable et des experts linguistiques. coût élevées. -Difficultés de classification en présence de bruit. -Le nombre de clusters dans la plupart des cas sont inconnus. -La justesse peut parfois être relativement faible. -Instabilité des résultats.
Méthode symbolique	-Ne nécessite pas de formation d'échantillons annotés. -Accès facile au lexique d'opinion et leur orientation. -Fournit de meilleurs résultats sur domaine moins segmenté. -La capacité de trouver les mots d'opinion avec l'orientation spécifique du contexte. -Donne de bons résultats lorsque l'analyse porte sur des domaines différents.	-Difficulté de trouver les mots d'opinion avec des orientations spécifiques à des domaines autre que ceux du dictionnaire. -Incompatibilité dans les textes avec une certaine dépendance sémantique. -Moins précise lors d'analyse de différents domaines. -Performance variables en raison de l'étendue du domaine du lexique. -La difficulté de fournir des textes volumineux avec la capacité de couvrir tous les mots du texte. -Ne peut être utilisé seule

TABLE 2.1 : comparaison entre les méthodes de fouille d'opinion.[2]

Après avoir constaté que les deux méthodes présentées ci-dessus ont l'une comme l'autre des limitations, une nouvelle méthode est apparue sous le nom de méthode hybride.

II.4 les méthodes hybrides

Les méthodes hybrides sont une combinaison entre les techniques des deux approches statistique et symbolique pour aboutir à des résultats très précis. Elles prennent en compte tout le traitement linguistique puis lancer le processus d'apprentissage, les opinions sont extraits des phrases du document qui sont traitées phrase par phrase, ensuite une valeur globale est attribuée au classe [6].

Ces méthodes semblent efficaces à condition d'effectuer les bonnes combinaisons [12]. La figure 2.1 résume les techniques de classification pour l'analyse d'opinion.

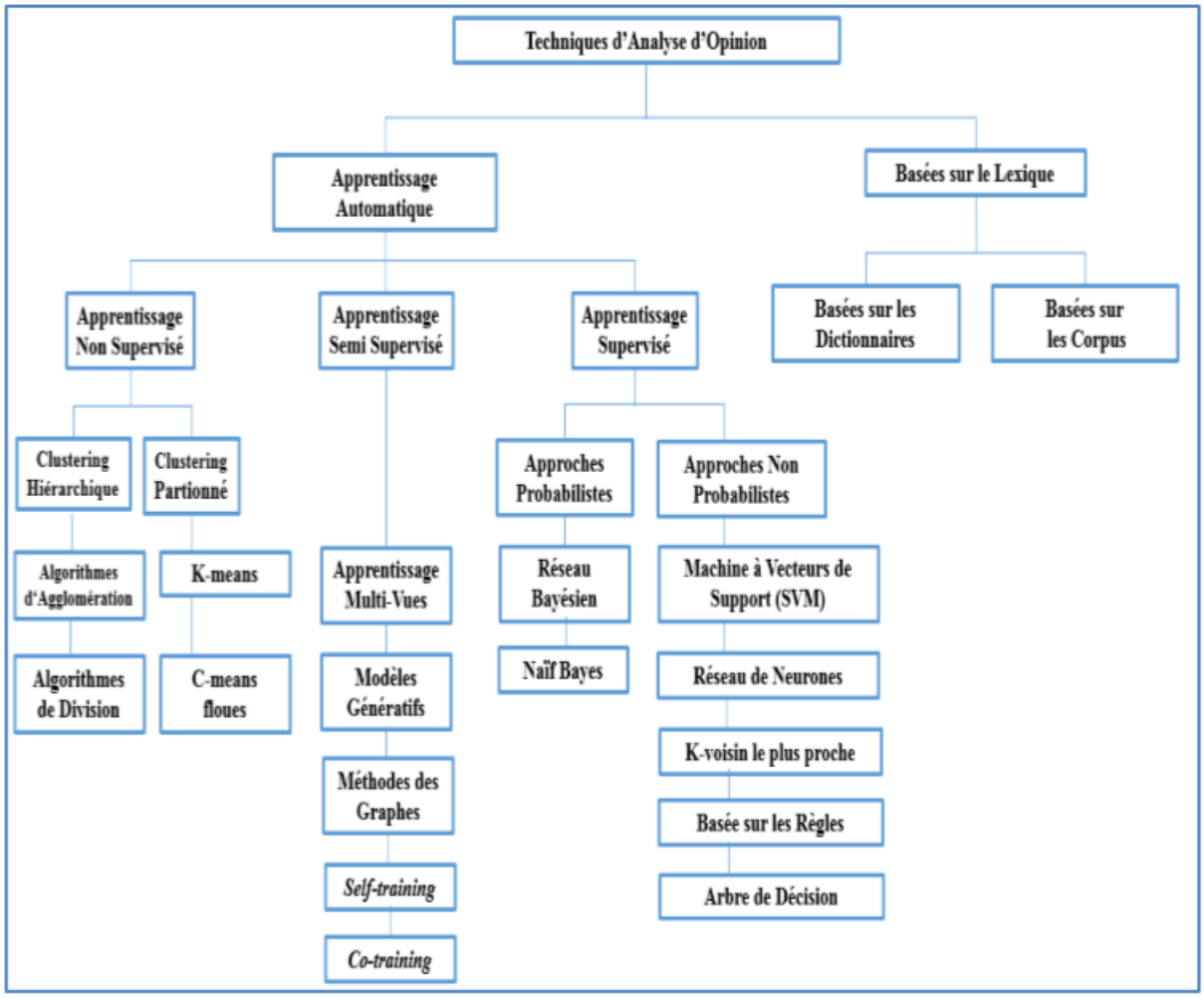

FIGURE 2.1 : Techniques de fouille d'opinion [2].

II.5 Evaluation de la classification

L'évaluation de la performance des méthodes de classifications s'effectue par le calcul des mesures de similarités tel que la justesse (accuracy), la précision, le rappel, le F-score(f-mesure) [25].

- La précision :est le nombre de données correctement classées dans une classe sur le nombre totale de données dans cette même classe [2], la formule est donnée comme

suit pour la classification binaire :

$$Precision = \frac{Vpi}{Vpi + Fpi} \tag{2.2}$$

Pour la multi classification, elle est considéré comme la moyenne de précision de toutes les classes :

$$PrecisionMoyenne = \frac{\sum_{i=1}^{N} \frac{Vpi}{Vpi+Fpi}}{N} \tag{2.3}$$

- Le rappel : le nombre de données correctement classées dans une classe sur le nombre de données attendues pour cette classe [2], la formule est donnée comme suit pour la classification binaire :

$$Rappel = \frac{Vpi}{Vpi + Fni} \tag{2.4}$$

Pour la multi classification, elle est considéré comme la moyenne de toutes les classes :

$$Rappelmoyen = \frac{\sum_{i=1}^{N} \frac{Vpi}{Vpi+Fni}}{N} \tag{2.5}$$

- La F-score : moyenne harmonique de la précision et du rappel, elle donne un score de performance global du classifieur [26],la formule est donnée comme suit pour la classification binaire :

$$F - score = \frac{2 * (P * R)}{(P + R)} \tag{2.6}$$

Pour la multi classification, elle est considéré comme la moyenne de toutes les classes :

$$F - scoremoyen = 2 * \frac{(P_N * R_N)}{(P_N + R_N)} \tag{2.7}$$

- Accuracy(justesse) : nombre de prédictions correctes sur le nombre totale de prédictions[26],la formule est donnée comme suit pour la classification binaire :

$$Accuracy = \frac{V_p + V_n}{V_p + V_n + F_p + F_n} \tag{2.8}$$

Pour la multi classification, elle est considéré comme la moyenne de toutes les classes :

$$Accuracy = \frac{\sum_{i=1}^{N} \frac{V_p+V_n}{V_p+V_n+F_p+F_n}}{N} \tag{2.9}$$

III corpus

III.1 Définition d'un corpus

Un corpus est un ensemble de documents (audios, vidéos, textes) contenant des données à exploiter. Un corpus textuel est une collection de textes supposés être représentatifs d'une langue, d'un dialecte ou d'un autre sous-ensemble d'une langue à utiliser pour l'analyse linguistique [27].

Il se conçoit selon des critères de conception de corpus spécifiques pour être représentatif au maximum de langage ou d'autres systèmes sémiotiques [28].

De nombreuses ressources telles que les sites web, forums, blogs ... etc., relatifs aux différents domaines, peuvent être utilisées pour créer des corpus textuels.

III.1.1 Le processus de constitution du corpus

Souvent lors de la conception de tout corpus, il faut se poser des questions comme :

- Le type de corpus.
- L'adéquation pour le projet visé.
- La possibilité de réutiliser ou d'interchanger ces corpus.
- La taille.
- La représentativité (c'est-à-dire, la variété de textes, d'auteurs, de sources, etc.).
- L'utilisation de textes complets ou d'échantillons.

Bien sûr, certains de ces critères sont difficiles à équilibrer entre eux et représentent des difficultés dans la construction du corpus. Donc la constitution du corpus peut se résumer en ces phases distinctes [29] :

- La sélection des sources.

- Les critères de sélection des textes et la décision de savoir s'il faut rendre le texte complet ou des fragments du même texte.
- Les décisions quant à l'infrastructure logicielle et matérielle (système de gestion de corpus textuels).
- La sélection des conventions pour la représentation des textes.
- Les critères, langage et système de balisage structurel.

III.1.2 Les types du corpus

Il existe plusieurs types de corpus qu'on citera ci-dessous :

III.1.2.1 Corpus spécialisé

Un corpus spécialisé est axé sur l'aspect particulier du vocabulaire d'un domaine de la connaissance [30].

III.1.2.2 Corpus de référence

Un corpus de référence reflète une langue et permet de faire des observations d'ordre général. Ce type de corpus contient des données orales et écrites, c'est un mélange de plusieurs textes de différentes natures [30].

Utilisés conjointement, un corpus de référence et un corpus spécialisé peuvent permettre d'identifier les différences entre un langage spécialisé et la langue générale.

III.1.2.3 Corpus ouvert

Est constamment étendu, implique un entretien constant et méticuleux c'est à dire auxquels on ajoute constamment de nouveaux textes. C'est le type généralement utilisé en lexicographie [30].

III.1.2.4 Corpus fermé

Une fois qu'il est compilé aucun texte n'est ajouté. Il va par conséquent rester tel quel [30].

III.1.2.5 Corpus d'apprenants (Learner corpus)

Ce type de corpus contient des textes écrits par les apprenants d'une langue étrangère. Il est intéressant pour effectuer des comparaisons avec des textes écrits par des natifs. Il fait ressortir les erreurs types des apprenants [30].

III.1.2.6 Corpus enrichi/annoté

Ce sont des corpus dont les documents ont bénéficié de traitements complémentaires tels que l'étiquetage morphosyntaxique, sémantique, méta-informations...etc [30].

III.2 Annotation de corpus

L'utilité d'annotation consiste à l'ajout des informations dites métadonnées au texte. Ces informations peuvent se rapporter à la structure de documents, paragraphes, phrases, etc.

III.2.1 Définition d'annotation

Globalement, l'annotation consiste dans l'apport d'informations de nature différente. On parle à ce sujet d'une " valeur ajoutée " aux données brutes. C'est la pratique qui consiste à appliquer au corpus des données qui ne sont pas explicitement présentes lors de la compilation de données.

En d'autres termes, l'annotation permet d'ajouter des structures linguistiques spécifiques aux données brutes du corpus, comme les jeux d'étiquettes et l'analyse syntaxique [3]. Par exemple, un type d'annotation courant est l'ajout de balises ou d'étiquettes, indiquant la classe de mots à laquelle les mots d'un texte appartiennent. Il s'agit de ce qu'on appelle le balisage de la partie du discours (ou balisage POS).

- Présent NN1 (nom commun singulier).
- Présent VVB (forme de base d'un verbe lexical).
- Présent JJ (adjectif général).

III.2.2 Types d'annotation

Il existe trois types d'annotations qui s'appliquent à trois domaines différents et à des applications distinctes [30] :

- l'annotation dans son sens premier comme ajout manuel de remarques, commentaires, notes sur le texte.
- l'annotation du document et/ou du corpus avec les métadonnées caractérisant et décrivant le document numérique.
- l'annotation d'ordre linguistique dans le cas de l'étiquetage morphosyntaxique ou de l'annotation sémantique.

L'annotation d'analyse linguistique d'un corpus possède plusieurs types, par exemple :

III.2.2.1 Annotation phonétique

Cette annotation consiste à ajouter des informations sur la façon dont un mot dans un corpus parlé a été prononcé [31].

III.2.2.2 Annotation prosodique

Cette annotation est utilisée dans un corpus parlé. Elle consiste à ajouter des informations sur les caractéristiques prosodiques telles que le stress, l'intonation et les pauses [31].

III.2.2.3 Annotation syntaxique

L'annotation syntaxique signifie marquer des termes ou des phrases avec leur type de mot ou leur fonction syntaxique. Il est généralement utilisé en combinaison avec le marquage morphologique, qui identifie les unités linguistiques telles que les préfixes, les suffixes et les morphèmes. Cette combinaison est appelée étiquetage POS [31].

III.2.2.4 Annotation sémantique

L'annotation sémantique fournit des connaissances sur la signification d'un terme ou d'une phrase, on ajoute des informations sur la catégorie sémantique des mots le mot cricket en tant que terme pour un sport et en tant que terme pour un insecte appartient à différentes catégories sémantiques, bien qu'il n'y ait pas de différence d'orthographe ou de prononciation [31].

III.2.2.5 Annotation pragmatique

Ce type d'annotation consiste à ajouter des informations sur les types d'actes de langage ou de dialogue qui se produisent dans un dialogue parlé. Ainsi, l'énoncé correct à différentes occasions peut être une reconnaissance, une demande de rétroaction, une acceptation ou un marqueur pragmatique amorçant une nouvelle phase de discussion [31].

III.2.2.6 Annotation de discours

Cette annotation consiste à ajouter des informations sur les liens anaphoriques dans un texte, par exemple connecter le pronom leur et son antécédent les chevaux dans : Je vais seller les chevaux et leur faire les tours. [Un exemple du corpus Brown] [31].

III.2.2.7 Annotation stylistique

Ce type d'annotation consiste à ajouter des informations sur la parole et la présentation de la pensée c'est à dire monter si c'est un discours direct ou indirect, ou c'est une pensée indirecte libre, etc [31].

III.2.2.8 Annotation lexicale

Ce type d'annotation consiste à ajouter l'identité du lemme de chaque forme de mot dans un texte, c'est-à-dire la forme de base du mot [31].

Les types d'annotation syntaxique, sémantique et lexicale sont souvent utilisés ensemble pour obtenir un meilleur modèle. Ils permettent un système plus performant grâce

aux effets mutuellement bénéfiques de différents types d'annotations. Dans notre cas nous avons utilisé une annotation sémantique.

III.3 Les approches d'annotation

Les approches d'annotation des corpus peuvent être classées en trois approches : automatique fondée sur le TAL (traitement automatique des langues), semi-automatique et en approche manuelle [32]. Pour annoter des documents, il est nécessaire de disposer d'un guide d'annotation sous forme de document qui indique quoi annoter et comment le faire et dans quel objectif.
Le résultat de l'annotation manuelle peut être utilisé pour évaluer la qualité des annotations automatiques.

III.3.1 Approche d'annotation manuelle

L'annotation manuelle est le processus d'annotation réalisé par un humain. Cette approche est appelée aussi annotation collaborative. Elle a pour but la construction d'une base de connaissances consensuelles.

Un même corpus peut être annoté par plusieurs êtres humains sans visualiser les annotations des autres humains. Par la suite les annotations peuvent alors être comparées afin d'identifier les points d'accords et de désaccords.

III.3.2 Approche d'annotation automatique

L'annotation automatique est le processus d'annotation réalisé par une machine [33]. Cette approche regroupe deux méthodes :

III.3.2.1 Méthodes symboliques

Méthodes permettent de réaliser une annotation en se basant sur les connaissances d'expert qu'ils ont formalisés en listes et expressions régulières [34].

- Listes : fichier contenant des données relevant d'une seule catégorie (adjectifs, noms)

- Dictionnaire : ensemble des mots (lemmes) d'une langue classés par ordre alphabétique (mot, définition, exemple, information morphologique, prononciation).
- Thesaurus : ensemble des mots (lemmes) d'une langue organisés par thématiques en distinguant les différents sens.
- Expressions régulières : séquences de caractères qui définissent un patron de recherche.
- Patron syntaxique/motif : motif particulier cherché dans un texte (un mot commençant par une majuscule suivie de minuscules non accentuées) : Paris, France.

III.3.2.2 Méthodes par apprentissage

Méthodes qui reposent sur des observations statistiques ; l'utilisateur fournit à la machine des exemples de sorties attendues en simulant un corpus annoté. Afin d'y arriver un ensemble de formalismes, outils et caractéristiques sont utilisés [34] :

- Formalismes : CRF (champs aléatoires conditionnels), SVM (séparateurs à vaste marge), arbres de décision.
- Outils : CRF++, Wapiti, LibSVM, SVMLight, WEKA.
- Caractéristiques : ensemble des informations associées à chaque token permettant de construire des modèles. Il en existe trois types :
  - Caractéristiques de surface : propriétés inférées du token (capitalisation, taille).
  - caractéristiques profondes : informations morphosyntaxiques, syntaxiques, sémantiques.
  - caractéristiques externes : position dans le document, fréquence globale, cluster.

III.3.3 Approche d'annotation semi-automatique

Cette méthode regroupe les deux approches précédentes ; manuelle et automatique. Elle est utilisée de manière incrémentale.

Cette approche procède d'abord à une annotation manuelle. Il en résulte une annotation de référence. Cette dernière sert à la construction des ensembles d'amorçage qui permettront l'apprentissage d'un système d'annotation automatique [35].

Apres avoir effectuer une annotation manuelle qui permet de construire un système d'annotation automatique, l'annotation semi-automatique du corpus s'effectue en deux étapes :

- un nouveau corpus est annoté automatiquement.
- Ce corpus annoté est soumis aux experts pour une annotation complémentaire.

III.4 Les normes d'annotation d'un corpus

L'utilité des corpus annotés, dépend essentiellement de la bonne planification et de la bonne exécution de l'annotation. Il est donc essentiel de recommander un ensemble de normes de bonnes pratiques à respecter par les annotateurs :

- Les annotations doivent être séparables : Il devrait toujours être facile de séparer les annotations du corpus brut, afin que le corpus brut puisse être récupéré exactement sous la forme qu'il avait avant l'ajout des annotations.
- Une documentation détaillée et explicite doit être fournie : Il est important de fournir une documentation explicite et détaillée sur les annotations dans un corpus annoté. Afin que les utilisateurs sachent précisément ce qu'ils obtiennent, la documentation à fournir sur les annotations doit inclure les éléments suivants :
    - Comment, où, quand et par qui les annotations ont-elles été appliquées ?
    - Quel schéma d'annotation a été appliqué ?
    - Quel schéma de codage a été utilisé pour les annotations ?
    - Quelle est la qualité de l'annotation ?

III.5 Les avantages et les inconvénients de l'annotation

L'annotation a connu un grand débat sur son utilité et les dangers qu'elle peut bien causer. Certaines personnes préfèrent ne pas s'engager dans l'annotation de corpus, pendant que d'autres le valorisent . Delà, un ensemble d'avantages et d'inconvénients ont été cité.

III.5.1 Les avantages de l'annotation

- L'annotation est un moyen de rendre un corpus beaucoup plus utile [37].
- L'annotation permet de donner une valeur ajoutée au corpus ce qui le rend plus avantageux [37].
- L'annotation est un enrichissement du corpus brut d'origine [37].

III.5.2 Les inconvénients d'annotation

- Un corpus annoté est un corpus falsifié qui peut contenir des informations suspectes [6].
- Risque d'erreurs d'annotateurs [6].
- Perte de données pendant le processus d'annotation [37].
- Les techniques utilisant un corpus annoté se limitent généralement à utiliser exclusivement les balises. En utilisant des textes balisés, le texte ne sera observé qu'à travers ses balises et l'ajout de balises entraîne une perte d'intégrité du texte [37].

IV Wolf et Sentiwordnet

IV.1 Sentiwordnet

IV.1.1 Introduction

Les chercheurs ont tenté de développer des systèmes pour étiqueter automatiquement les mots qui indiquent les opinions comme étant soit positives soit négatives. La question préalable et connexe est de savoir si un mot est en fait un marqueur d'opinion ou non, qu'il soit

subjectif ou objectif, a reçu moins d'attention. Les premières tentatives ont consisté à étiqueter les mots sans faire de distinction entre différents sens dans lesquels un mot peut être utilisé, de sorte que le mot plutôt que son sens est classé. Cela a des limites car le même mot a très souvent plusieurs sens et tout système qui ne parvient pas à capturer ces variations de sens est sévèrement limité en fonctionnalités et fiabilité. Ils ont tenté de remédier à cette limitation en développant une ressource qui est le SentiWordNet (wordnet+sentiment information)

IV.1.2 Présentation de SentiWordNet

SentiWordNet est une ressource lexicale, résultante de l'annotation automatique de tous les synsets de WORDNET selon les notions de "positivité", de "négativité" et de "neutralité". Chaque synset est associé à trois scores numériques Pos (s), Neg (s), et Obj (s) qui indiquent comment sont les termes contenus dans le synset positif, négatif ou objectif(neutre). Chacun des trois scores varie dans l'intervalle [0,1], et le score d'objectivité est calculé via la formule suivante ObjScore=1-(PosScore+ NegScore). SWN est utilisé dans les taches d'analyse de sentiments, cette ressource contient une base de données de termes anglais facilement utilisable [38].

IV.1.3 Les versions de SentiWordNet

SENTIWORDNET a connu quatre versions :

- SENTIWORDNET 1.0, présenté dans (Esuli et Sebastiani,2006) est rendu public pour la recherche ; se compose d'une annotation de l'ancien WORDNET 2.0 qui est réalisé par une annotation automatique semi-supervisé, il utilise les glosses de WORDNET synsets en tant que représentations sémantiques [39].
- SENTIWORDNET 1.1, discuté dans un rapport mais n'a pas atteint la publication (Esuli et Sebastiani, 2007b) [39].
- SENTIWORDNET 2.0, uniquement discuté dans la deuxième thèse de l'auteur (Esuli, 2008) [39].
- SENTIWORDNET 3.0, la version que nous utilisons, conçue pour soutenir la classification sentiment et applications d'extraction d'opinion (Pang et Lee, 2008). Il est

composé d'une a une annotation du WORDNET 3.0. qui est réalisé par une annotation automatique semi-supervisé suivi d'un processus de marche aléatoire, il utilise les glosses désambiguïsées manuellement du Princeton WordNet Gloss Corpus [39].

Les versions 1.1 et 2.0 n'ont pas eu de publications officielles largement connues.

IV.1.4 L'approche de construction de SentiWordNet3

SENTIWORDNET 3 est généré selon un processus d'annotation automatique. Ce processus se compose de deux étapes, une étape d'apprentissage à supervision faible, semi-supervisée, et une étape de marche aléatoire.

L'étape d'apprentissage semi-supervisée implique l'utilisation de 8 classifieurs pour décider si le synset est positif, négatif ou objectif. Ce processus fournit un facteur de généralisation plus élevé et un risque de sur-ajustement faible. De plus, le synset est classée selon différentes classifications et ces résultats sont ensuite combinés pour atteindre une précision élevée pour qu'il soit plus positif, négatif ou objectif. En faisant la moyenne des classifications obtenues à partir de chaque classificateur, une valeur comprise entre 0,0 et 1,0 est obtenue pour chaque catégorie de synset. Si tous les classificateurs décident de la même classification, ce synset aura une valeur maximale de 1,0 pour ce sentiment.

L'étape de marche aléatoire consiste à visualiser SentiWordNet 3.0 sous forme de graphique, et un processus itératif de marche aléatoire est généré en termes positifs, négatifs et objectifs. Le processus démarre à partir de ces termes déterminés à l'étape précédente avec une possibilité de changer l'orientation des sentiments à chaque itération [38].

IV.1.5 Structure de SentiWordNet

La structure de sentiwordnet 3.0 contient les rubriques suivantes :

- POS : indique la fonction grammaticale du synset (nom, verbe, adjectif, adverbe).
- ID : représente l'identifiant du synset.

Et la paire (POS, ID) identifie le synset de wordnet.

- PosScore : indique le score positif du synset.
- NegScore : indique le score négatif du synset.
- SynsetTerms : les mots inclus dans un synset.
- Gloss : c'est la définition du synset.

La figure 2.2 ci-dessous montre un fragment de données de SentiWordNet que nous avons téléchargé, en illustrant la structure précédemment décrite [40].

# POS	ID	PosScore	NegScore	SynsetTerms	Gloss
a	00001740	0.125	0	able#1	(usually followed by `to') having the necessary means or skill or know-how
a	00002098	0	0.75	unable#1	(usually followed by `to') not having the necessary means or skill
a	00002312	0	0	dorsal#2 abaxial#1	facing away from the axis of an organ or organism; "the aba
a	00002527	0	0	ventral#2 adaxial#1	nearest to or facing toward the axis of an organ or organis
a	00002730	0	0	acroscopic#1	facing or on the side toward the apex
a	00002843	0	0	basiscopic#1	facing or on the side toward the base

FIGURE 2.2 : Fragment de données de sentiwordnet 3.0

IV.2 Wolf

IV.2.1 Présentation de WOLF

Le WOLF (Wordnet Libre du Français) est une ontologie en licence libre développée pour le français par l'Inria (Institut national de recherche en informatique et en automatique) en 2008 [41]. Elle a été construite à partir du Princeton WordNet (PWN)[1] et de diverses ressources multilingues. Son but est de répertorier, classifier et mettre en relation de diverses manières le contenu sémantique et lexical de la langue française.

La ressource WOLF est publiée sous la forme d'un fichier XML, avec XML les données peuvent se décrire elles-mêmes si des noms d'étiquettes appropriés sont choisis correctement. Ceci permet à un humain de s'y retrouver plus facilement que la liste de caractères ou codes hexadécimaux utilisés dans la distribution de WordNet.

[1]C'est une base de données lexicales de la langue anglaise développée par des linguistes du laboratoire des sciences cognitives de l'université de Princeton.

L'otologie se compose donc d'un nœud principal " Wordnet " ayant pour fils des nœuds Synset. Chacun de ces nœuds représente une unité de sens pour le français. Ce principe de synsets vient de l'ontologie anglaise Wordnet de Princeton.

Le WOLF est une ressource ontologique intéressante puisqu'elle propose un lien direct avec l'ontologie Wordnet de Princeton via les identifiants des synsets. Cela permet d'obtenir facilement un couplage entre cette ressource et le Wordnet anglais qui a servi à la construire[41].

IV.2.2 L'approche de construction de Wolf

WOLF a été crée par une méthode d'extension [42], dans laquelle un ensemble de synsets (ensembles de synonymes) du PWN ont été traduits en français afin d'avoir cette ressource lexicale de termes français [41].

IV.2.3 Structure de Wolf

Wolf est structuré de la manière suivante [41] :

- Un numéro d'identification :Un identifiant unique (un nombre hexadécimal), présent dans sa sous balise ID. Cet identifiant est en fait le même que celui du synset original dans le Wordnet de Princeton.
- Une fonction grammaticale :Cette fonction se trouve dans la balise POS. Tous les mots composant un synset partagent la même fonction grammaticale.
- Une liste de mots (ou encore lexèmes) : Cette liste est présente dans la balise Synonym. Elle représente la liste des mots composants le synset.
- Une liste de liens avec d'autres synsets : Cette liste est présente dans la balise ILR.Les liens ont été décrits en utilisant les identifiants des synsets.
- une note BCS : Indiquant l'importance du synset. 1 : très important, 2 : important, 3 : relativement important, vide : peu important.
- Balise Def : Contient une définition, des exemples ou encore des renseignements par rapport à la traduction du synset depuis le Wordnet de Princeton.

Conclusion

Dans ce chapitre, nous avons présenté les différentes méthodes de fouille d'opinion en mettant l'accent sur les caractéristiques de chacune d'elles. Nous avons présenté l'apprentissage automatique et ses méthodes de classification en passant par quelques classifieurs de la méthode supervisé.

Nous avons introduit WEKA qui est un logiciel très puissant dans le domaine du traitement automatique de données.

Nous avons abordé les différents types de corpus ainsi que la démarche de sa construction, les différents types d'annotation existants et les différentes techniques de la réalisation de l'annotation. Nous avons cité les points forts et points faibles de l'annotation de corpus.

A la fin de ce chapitre, nous avons arboré les ressources lexicales SentiWordNet et Wolf en précisant leurs structures les processus de leur création.

Chapitre 3

Réalisation

I Introduction

Depuis le développement d'Internet, les personnes rendent leurs avis disponibles. Aujourd'hui il existe plein de documents textuels qui expriment les idées, les opinions et les sentiments des gens sur le web. pour analyser les opinions exprimées dans divers domaines il est nécessaire d'utiliser des ressources lexicales propre à ces derniers.Notre travail s'insère dans l'analyse d'opinion, nous proposons la création d'une ressource lexicale pour l'analyse d'opinion dans le domaine de l'éducation pour avoir une meilleure classification des opinions issues de ce domaine. Pour ce faire, nous suivons les étapes ci-dessous illustrées dans la figure 3.1 :

- Construire une ressource lexicale appelée " DICO ".
- Construire un corpus annoté nommé EDUCA.
- Lemmatisation de corpus EDUCA.
- Entrainement des classifieurs avec l'outil weka.
- Recalcule des nouvelles polarités de la ressource lexicale DICO.
- Validation de notre ressource lexicale.

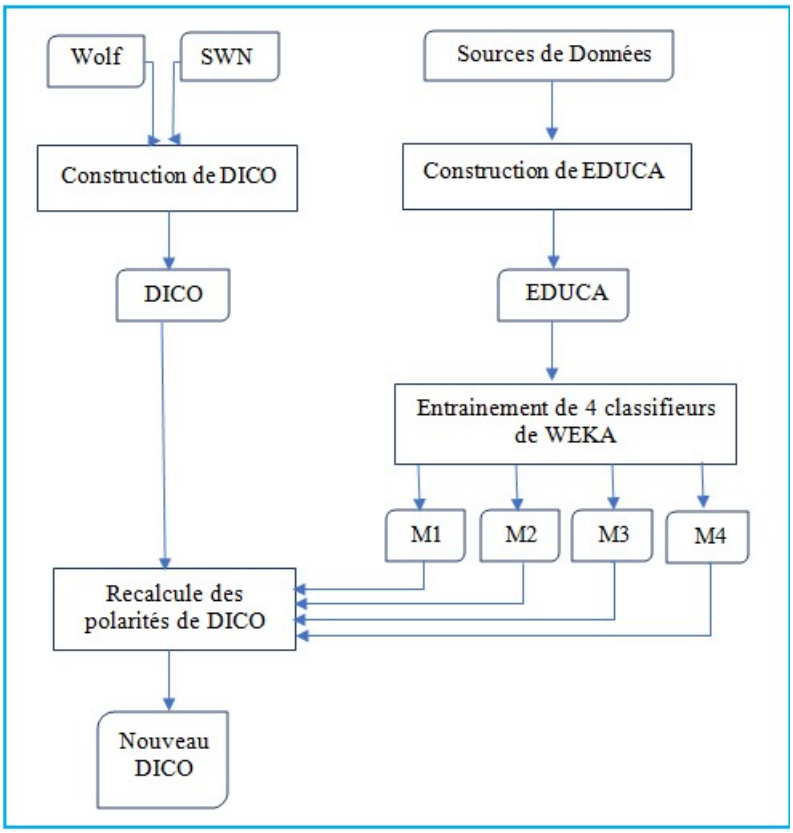

FIGURE 3.1 : Description de la démarche adaptée.

II Description de la construction initiale de DICO

Dans le cadre de ce travail, nous avons repris la démarche de construction de la ressource lexicale de la langue française dédiée à l'analyse d'opinion pour l'éducation appelée " Dico " [2].

II.1 Processus de construction de DICO

La construction de cette ressource est basée sur le regroupement des synsets de WOLF 0.6 avec les polarités correspondantes données par SentiWordNet 3.0. [2]. Ce regroupement est réalisé en faisant la correspondance entre le (ID, POS) de SentiwordNet3.0. et le (ID, POS) de WOLF 0.6.

En d'autres termes la ressource construite peut se mettre à jour par des experts en rajoutant des mots inexistants auparavant dans cette ressource. La mise à jour s'effectue lors du traitement de nouveaux textes contenant de nouveaux mots qui ne sont pas inclus dans la ressource en leur attribuant des polarités et les insérer dans la ressource lexicale.

La figure 3.2 illustre le processus de construction de cette ressource lexicale

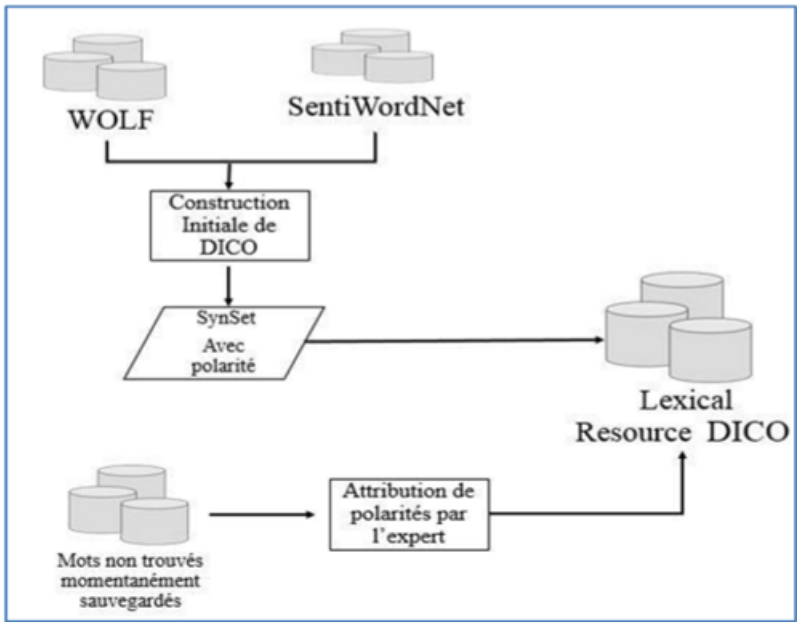

FIGURE 3.2 : Processus de construction de la ressource lexical DICO[2]

II.2 La structure de DICO

La ressource lexicale DICO est composée de Synset décrits comme suit :

- ID : Indique l'identifiant du synset.
- POS : Indique la catégorie de la partie de discours (Nom, verbe, adjectif, adverbe) :
  - n : Nom.
  - v : Verbe.

  - a : Adjectif.
  - b : Adverbe.

- Termes : Indique le terme du synset.
- Def : Indique la définition du synset.
- PosScore : Indique la valeur de la polarité positive.
- NegScore : Indique la valeur de la polarité négative.
- ObjScore : Indique la valeur de la polarité objective.

La figure 3.3 illustre un échantillon de la ressource lexicale DICO.

ID	POS	Termes	Posscore	Negscore	Objscore	Def
00001740	a	capable	0	0.125	0.875	(généralement suivi de `` à '') avoir les moyens, ...
00002098	a	incapable	0.75	0	0.25	(généralement suivi de `` à '') ne pas avoir les m...
00002312	a	abaxial	0	0	1	face à l'axe d'un organe ou d'un organisme
00002527	a	ventral	0	0	1	le plus proche ou faisant face à l'axe d'un organe...
00003553	a	naissant	0	0	1	venir à l'existence

FIGURE 3.3 : La ressource lexicale DICO

III Construction du corpus EDUCA

La construction de notre corpus EDUCA consiste à collecter un ensemble de 20001commentaires subjectif des sources les plus importantes et les plus influentes pour le domaine d'éducation. Ces commentaires vont être soumis à une classification selon l'opinion.

III.1 Rappel sur les démarches de construction du corpus

Comme nous l'avons vu précédemment dans le chapitre 2, la construction d'un corpus passe par plusieurs étapes :

- La sélection des sources.
- Les critères de sélection des textes.
- Les décisions quant à l'infrastructure logicielle et matérielle (système de gestion de corpus textuels).
- La sélection des conventions pour la représentation des textes.
- Les critères, langage et système de balisage structurel [29].

III.1.1 Démarche de construction retenue

Notre démarche de construction du corpus EDUCA est décrite dans la figure 3.4 :

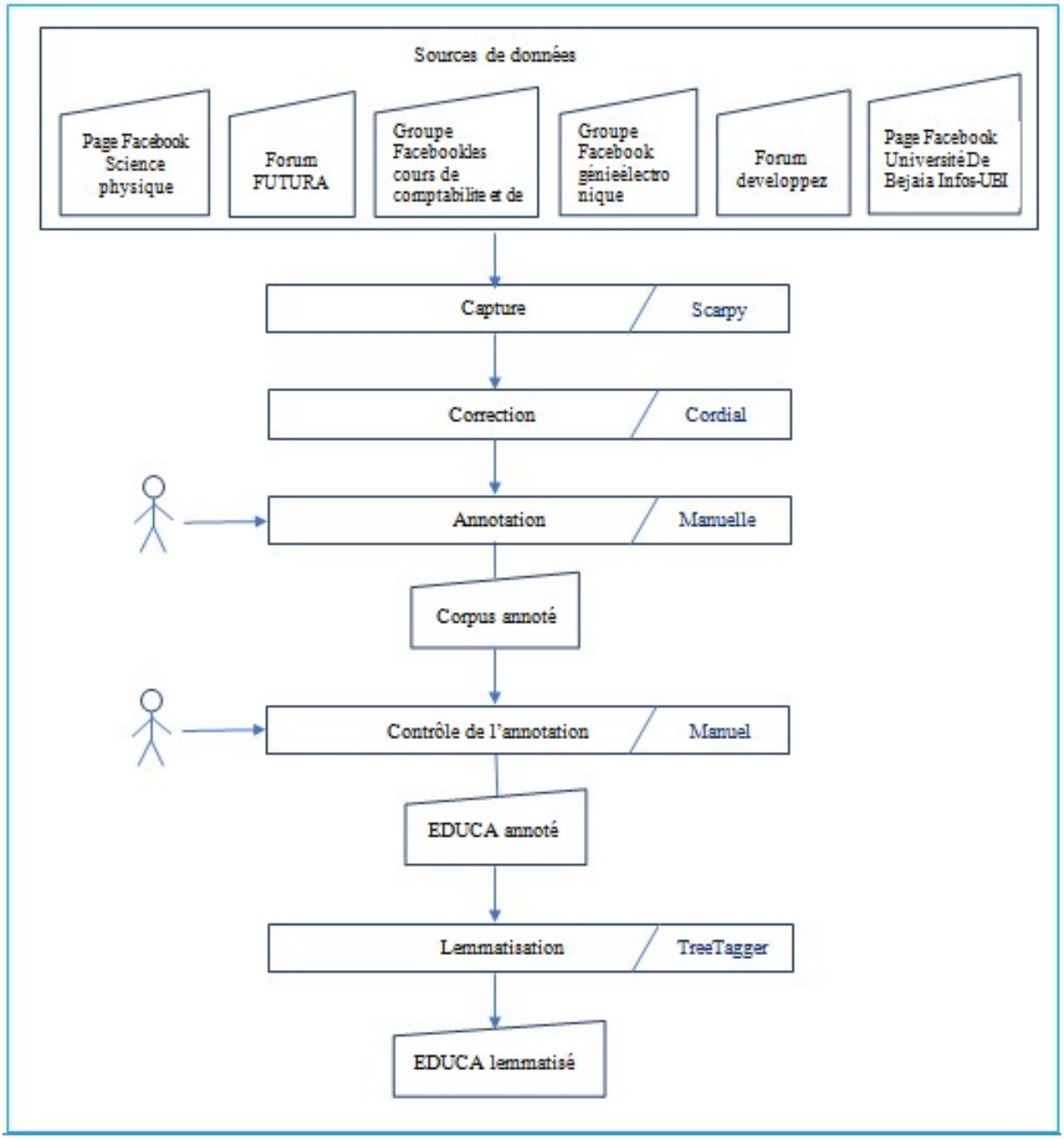

FIGURE 3.4 : Construction de EDUCA.

Les étapes illustrée dans la figure 3.4 seront approfondies dans ce qui suit :

1. **Le choix des sources** :

La première étape de notre démarche consiste à choisir les sources de collecte de commentaires ceci en listant toutes les sources qui sont les plus reconnues dans le domaine de l'éducation, les plus visibles et les plus consultées.

Notre choix est porté sur différents blogs, forums ainsi que les pages et groupes Facebook issus de domaine de l'éducation.

2. **Le choix de l'outil de la collecte :**

Lors de la construction de notre corpus, nous avons opté pour l'utilisation d'un scraper du nom Scrapy afin de rendre la collecte de commentaires facile et moins couteuse en termes de temps.

Scrapy est un Framework collaboratif et open source qui permet l'extraction automatique de données dont on a besoin à partir des sites web. Il est développé en Python. Son fonctionnement peut se résume à l'utilisation de la console shell qui permet l'extraction de données à partir de sources HTML, à l'aide d'un sélecteur CSS, comme est le cas dans notre travail [43].

Nous avons opté pour cet outil principalement pour ses avantages, il est gratuit, multiplateforme, rapide et robuste.

Le tableau 3.1 ci-dessous illustre les sources utilisées, le nombre de commentaire collecté de chacune de ces sources ainsi que les dates d'effet de cette opération.

Sources de données	**Nombre de commentaires collectés**	**Date de la récolte de commentaires**
Page Facebook science physique. Url :https ://www.facebook.com/Science-Physique-311189628517	2045	16 juillet2020
Forum FUTURA. Url :https ://forum.futura-sciences.com/technologies/883056-aide-une -presse-a-lhorizontale.html	1605	20juillet2020
Groupe Facebook LES COURS DE COMPTABILITÉ ET DE GESTION OHADA. Url : https ://www.facebook.com/groups/2050 098611943643	2956	26 juillet2020
Groupe Facebook genie electronique. Url :https ://www.facebook.com/groups/ 135057557248276	1525	28juillet 2020
Forum developpez. Url : https ://www.developpez.net/forums /	8492	02 aout2020
Page Facebook Université De Béjaia Infos-UBI Url : https ://www.facebook.com/universitedebejaianews/	3377	12aout2020

TABLE 3.1 : Détails d'opération de récolte de donnée

3. **La correction du corpus**

Une fois, l'ensemble de commentaires est collecté nous avons procéder à la correction de ce dernier, en éliminant le maximum de fautes d'orthographe à l'aide de correcteur CORDIAL, qui est un correcteur français le plus performant en orthographe et en grammaire, il souligne les erreurs soit en rouge pour l'orthographe soit en bleu pour la grammaire[44].

La figure 3.5 illustre un exemple d'utilisation du correcteur cordial.

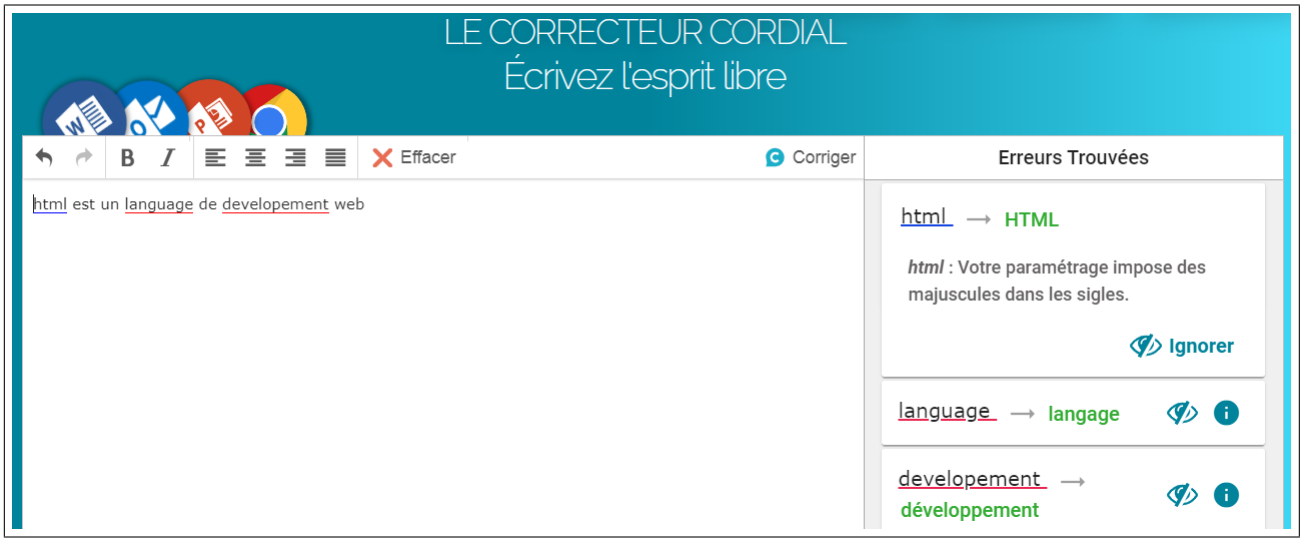

FIGURE 3.5 : Le correcteur CORDIAL [44]

4. **Annotation du corpus**

Une fois que la correction du corpus est achevée, il a fallu mener une compagne d'annotation pour pouvoir réaliser une classification supervisée et son évaluation.

Comme nous avons vu précédemment dans le chapitre2 , il existe 3 approches d'annotation, l'annotation automatique qui se fait à l'aide d'un outil d'annotation, l'annotation manuelle qui est réalisé par des êtres humains et l'annotation semi-automatique qui regroupe les deux annotations automatique et manuelle.

Puisque nous n'avons trouvé aucun logiciel correspondant à nos besoins, nous avons opté pour l'annotation manuelle d'opinion. Chaque commentaire, nous lui avons affecté une étiquète de positivité, négativité ou neutre selon la nuance de l'opinion exprimé dans le commentaire. Par exemple :

- J'adore coder avec PHP. $\doteq>$ positif
- C'est incroyable comment les étudiants peuvent être parfois insouciants bon après pour lúniversité de bejaia ses étudiants sont toujours insouciants. $\doteq>$ negatif

La figure 3.6 illustre une partie du corpus EDUCA annoté construit.

```
'Félicitation pour cette FAQ. C\'est une très bonne chose pour la rubrique d\'avoir
'Bravo et merci pour cet énorme boulot  positif
'Merci pour la FAQ    neutre
'Bravo, excellent  positif
'je n\'ai pas retrouvé ce passage dans la nouvelle faq  negatif
'comme j\'avais posé une question sur l\'initialisation des valeurs , j\'espérais y
'Bonjour à tous, et merci pour toutes les FAQ  positif
'Bonjour,Je voudrais participer dans la rubrique FAQ mais je bloque sur l\'insertion
'Est-ce possible d\'insérer des images depuis mon ordinateur ?    neutre
'Si oui, comment procéder ?Merci d\'avance.    neutre
'Rédige ta QR en indiquant les emplacements des images et ensuite on verra pour que
'Les images se trouvent dans mon PC ; donc si j\'ai bien compris, pour indiquer leur
'Tu peux mettre simplement [IMAGE1] et mettre entre parenthèses la description de l\
'Ok je comprends maintenant.Merci.  positif
'Dans la FAQ tableur "Comment scinder votre feuille ?", le menu de "scinder la fenêt
'Par ailleurs, FAQ très instructive.  positif
' C\'est de bonne guerre, non ?  positif
'Tous les jours.L\'obligation de passer sur PC.    neutre
'je comprends pas quel est l'intérêt d\'acheter un mac (qui est un pc 2 fois plus ch
'ceux qui le font pourriez vous dire pourquoi ?    neutre
'Selon un test que j\'ai vu il apparaît que la version de Windows 10 pour ARM n\'est
'Cela parait tout à fait normal. Sous RPi 4 c\'est un peu la même chose, certains so
```

FIGURE 3.6 : Partie du corpus EDUCA annoté.

IV Lemmatisation du corpus EDUCA

Avant de classifier ou regrouper du texte, nous devons obligatoirement passer par une étape primordiale qui consiste aux lemmatisations des données textuelles. Ce concept représente la première étape pour former des données bien structurées sur lesquelles nous serons capables d'appliquer les méthodes d'apprentissage automatique.

La lemmatisation est par définition une action consistant à l'analyse lexicale d'un texte avec pour but de regrouper les mots d'une même famille. On parle ici de donner la forme canonique d'un mot ou d'un ensemble de mots : Chacun de ces mots d'un contenu donné se trouve réduit en une entité appelée en lexicologie lemme ou encore "forme canonique d'un mot". Les lemmes d'une langue utilisent plusieurs formes en fonction[46] :

- Du genre (masculin ou féminin).
- De leur nombre (un ou plusieurs).
- De leur personne (moi, toi, eux…).
- De leur mode (indicatif, impératif…).

Pour ce faire, nous avons opté pour TreeTagger, qui est un outil permettant d'annoter un texte avec des informations sur les parties du discours il spécifie noms, verbes, infinitifs et particules et des informations de lemmatisation. Il a été développé par Helmut Schmid dans le cadre d'un projet dans le ICLUS (Institute for ComputationalLinguistics of the University of Stuttgart)[47].

La figure 3.7 ci-dessous représente l'interface graphique de l'outil TreeTagger, à travers laquelle nous avons effectué la lemmatisation ceci en chargeant dans l'onglet "input file" un fichier txt contenant notre corpus et en spécifiant le fichier txt de sortie dans l'onglet "output file" qui va contenir l'ensemble des lemmes retourné par TreeTagger.

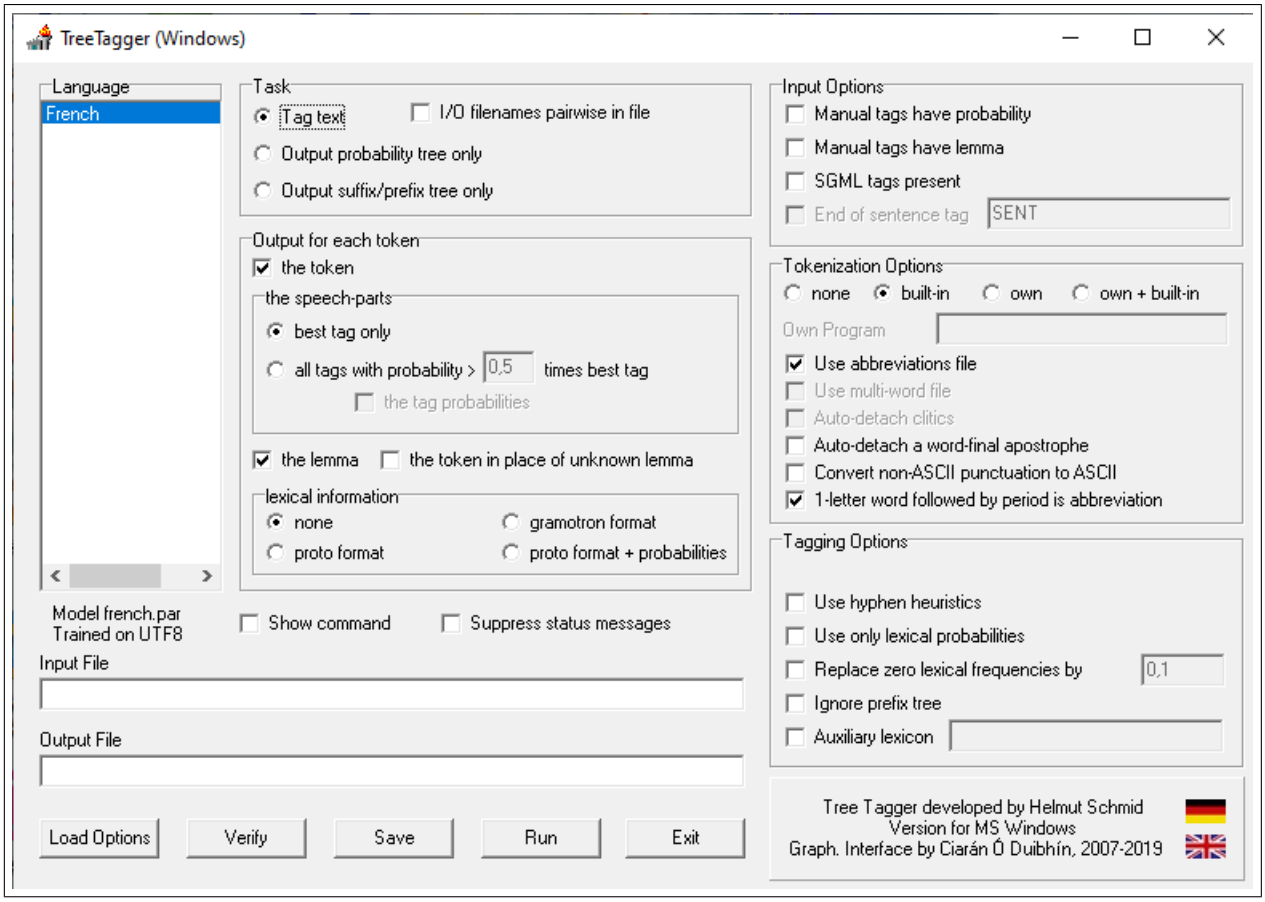

FIGURE 3.7 : Interface graphique de l'outil TreeTager.

V Entrainement de WEKA avec le corpus EDUCA

Ce travail consiste à entrainer des classifieurs avec notre corpus afin d'attribuer de nouvelles polarités aux mots de DICO, pour ce faire, nous allons utiliser le logiciel WEKA qui est une plateforme de référence lorsqu'on veut mettre en place des solutions en apprentissage automatique rapidement et simplement. Dans notre cas, puisque les classes sont connues à l'avance on va utiliser l'apprentissage supervisé. Elle permet d'utiliser rapidement différents algorithmes de classification avec des paramètres par défaut .

Dans le cadre de ce projet, quatre types de classifieurs sont utilisés :

- Les K-Voisins les plus proches.
- Les machines à vecteurs de support (SVM).
- Les arbres de décision.
- Les classifieurs bayésiens naïfs.

Etant donner que weka est un élément majeur dans notre travail, il est important de bien comprendre et savoir ses bases de fonctionnement.
IL peut être utilisé à trois niveaux :

- Via l'interface graphique, pour charger un fichier de données, lui appliquer un algorithme, vérifier son efficacité.
- Invoquer un algorithme sur la ligne de commande.
- Utiliser les classes définies dans ses propres programmes pour créer d'autres méthodes, implémenter d'autres algorithmes, comparer ou combiner plusieurs méthodes [45]

V.1 Interface utilisateur

L'interface de démarrage WEKA est constitué d'un ensemble de fonctionnalités comme l'illustre la figure 3.8 [48]

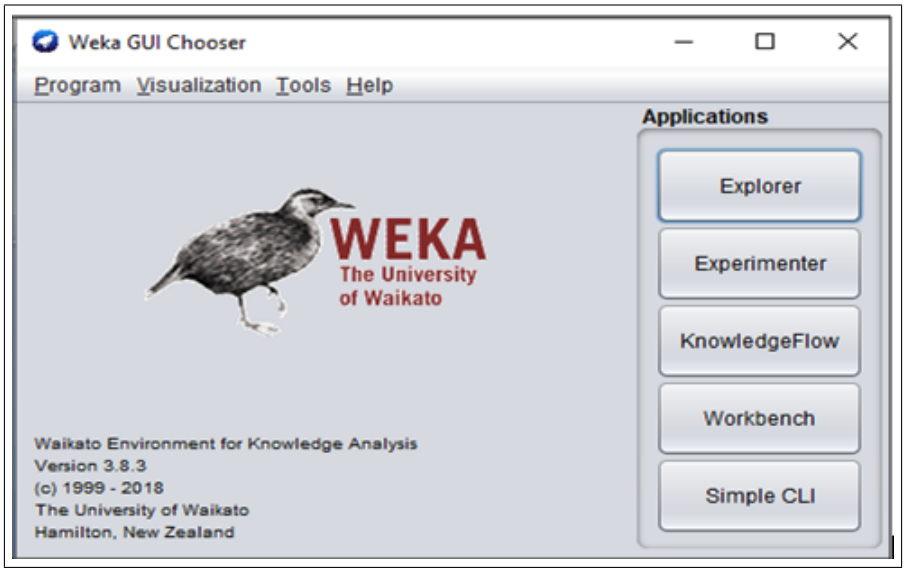

FIGURE 3.8 : Interface utilisateur WEKA

- **Explorer :** Interface permettant de paramétrer et réaliser une analyse sur un jeu de données.
- **Experimenter :** Environnement pour la réalisation d'expériences de teste et de comparaison de modèles statistiques.
- **knowledgeFlow :** Interface " drag-and-drop " permettant de créer un processus de workflow complet d'analyse d'un ou plusieurs jeux de données.
- **Workbench :** C'est une interface regroupant en un seul endroit le simpleCli, Explorer, KnwledgeFlow et Experimenter
- **Simple Cli :** Interface simple (shell) qui permet l'exécution directe des commandes WEKA en ligne de commandes.

La meilleure façon d'utiliser WEKA est par le biais d'une interface utilisateur graphique appelée Explorer.

V.2 Présentation de l'interface graphique Explorer

La figure 3.9 représente l'interface graphique Explorer. Cette interface regroupe un ensemble d'options permettant d'effectuer le prétraitement des données ainsi que l'accès aux différentes méthodes d'apprentissage automatique.

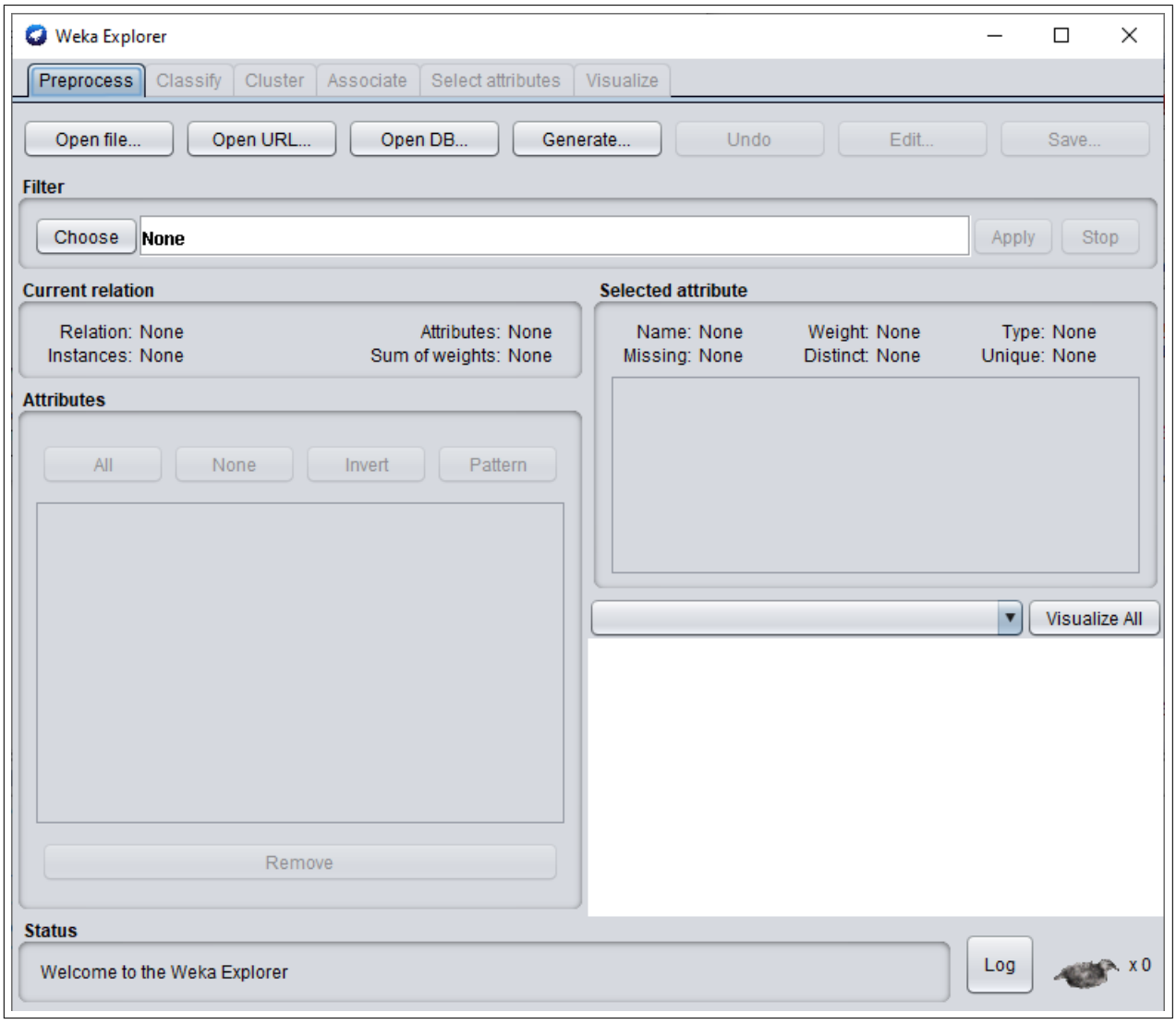

FIGURE 3.9 : Interface graphique "Explorer".

Les différentes options offertes par l'interface graphique Explorer se résument en[24] :

- Preprocess : La saisie des données, l'examen et la sélection des attributs, les transformations d'attributs.
- Classify : Mise en œuvre des différents algorithmes de classification.
- Cluster : Accès aux techniques et méthodes de segmentation (clustering).
- Associate : Accès aux apprentissages par règles d'association qui essaient d'identifier toutes les relations importantes entre les variables.
- Select attributes : L'étude et la recherche de corrélations entre attributs.
- Visualize : Représentations graphiques des données.

V.3 Format des données d'entrées de WEKA

Les données à classifier avec WEKA doivent être au format ARFF quelque soit le classifieur utilisé. Ceci permet de facilement comparer les résultats de différents classifieurs avec exactement les mêmes fichiers d'entrée. C'est une norme de format de fichier texte créée spécialement pour la lecture de données dans WEKA.

Afin d'avoir le fichier ARFF, nous avons effectuer une conversion en implémentant un programme java qui a eu en entrée notre corpus EDUCA sous format txt, et a donné en sortie le corpus sous format ARFF prêt à être exploité par WEKA. La figure 3.10 montre un fragment du résultat de l'implémentation.

```
@relation corpus

@attribute text string
@attribute @@class@@ {negatif,neutre,positif}

@data
'Il y a quelques année avec Delphi j\'ai repris des applis d\'il y a plus de 15 ans (en Delphi 4.0
' Récemment la même appli a été migré en 10.3 en 64 bits.\r\n',neutre
' et ça fonctionne toujours sans avoir eu à tout casser. \r\n',neutre
'Le gars qui avait développé à l\'époque avait fait un code très propre faut dire.\r\n',positif
' Je ne suis pas sur que l\'on aurait pu faire la même choses avec des langages plus en vue.\r\n',
'Donc pour moi, et ce n\'est que mon point de vue, ça reste une techno viable même dans le context
' Les ajouts et évolution de ces dernières années font que le langage n\'a pas à rougir coté fonct
'C\'est vrai, ça craint. Et dire que j\'ai laissé le C++ pour Delphi juste par fainéantises (marre
'En même temps, avec un vendredi férié, il fallait bien décaler trolldi, sauf à ce qu\'ils parvien
' Et vous, vous travaillez dans quelle société ?  Non vous n\'êtes pas comme la majorité des parti
'Cela peut faire aussi débat  maintenance longue contre évolution régulière (5-6 ans)\r\n',neutre
'Mais là aussi c\'est mensonger  : la bibliothèque VCL est pérenne (mais Windows seulement) pas la
'La France n\'est effectivement pas représentative sur Delphi.\r\n',negatif
' Mais il est possible de consulter ce lien\r\n',neutre
'Non je suis pas à la retraite et il me reste quelques belles années à faire\r\n',neutre
'Et je suis pas à mon compte non plus, la boite pour laquelle je bosse n\'a aucun intérêt ici, mai
'Dans la "maintenance longue", y\'a toujours des évolutions, du fait que les process de fabricatio
'FMX a déjà 8 ans si je ne m\'abuse... C\'est sur que c\'est pas encore 25 ans comme la VCL mais b
'Microsoft maintient encore la compatibilité Win32 eu égard aux milliers de logiciels qui sont enc
```

FIGURE 3.10 : Extrait du fichier ARFF.

- Données initiales :

  Dans notre cas, le fichier ARFF a comme attribut :

  - text : Un attribut du type 'string' qui va contenir chaque commentaire apparaissant dans le corpus EDUCA .
  - class : Représente les différentes annotations attribuées à chaque commentaire, dans notre cas, positif, négatif ou neutre.

V.4 Démarche de construction des modèles de classifications :

Pour générer les modèles de classifications avec les quatre classifieurs décris ci-dessus, nous avons utilisé le corpus EDUCA de 20001 commentaires. Nous avons utilisé 15000 commentaires pour l'entrainement de chacun des classifieurs afin d'obtenir des modèles de classifications. Les modèles ainsi obtenus seront évalués pour s'assurer de leur capacité de généralisation pour la classification d'opinion en utilisant 5001 commentaires annotés restant du corpus EDUCA.

Les commentaires sont classés en fonction d'une catégorie donnée, c'est à dire en

fonction de l'opinion qu'il véhicule, comme défini auparavant. La classification porte donc sur les différentes classes, il s'agit de positif, négatif et neutre.

V.4.1 Entrainement des classifieurs

Cette étape consiste à construire des modèles de classifications en apprenant du corpus EDUCA les exemples d'entrainement avec leurs classes respectives.
Pour ce faire, nous avons d'abord procédé au prétraitement des données d'entrée en utilisant les configurations de prétraitement proposés par WEKA.

- **Le prétraitement des données :**

Les prétraitements dans Weka sont effectués grâce aux filtres. Les filtres permettent de modifier l'ensemble de données, supprimer ou ajouter des attributs, selon le besoin.
Dans notre cas, nous avons utilisé les filtres suivants :

- classAssigner : Un filtre pour désigner que le dernier attribut sera considéré comme étant une classe.

- FiltredClassifier : est un filtre classifier qui combine entre le classifieur et le filtre , à ce niveau s'effectue le choix de classifieur ainsi que le filtre à utiliser.

- StringToWordVector : est un filtre qui permet la conversion des attributs de forme chaine de caractères en un ensembles d'attributs numériques représentants la fréquence de chaque mot dans la chaine. Chaque mot devient par défaut un attribut dont la valeur est 1 ou 0, ce qui reflète la présence de ce mot dans la chaîne. Ce filtre est utilisé suivant un réglage personnalisé,nous avons choisi les options suivantes :

    - Word tokenizer : qui est un tokenizer simple permet de segmenter les chaines à l'aide des délimiteurs.

- **Option de test :**

Pour entrainer nos classifieurs, nous avons opté pour une validation croisé 10 plis [49]. Dans cette méthode d'entrainement, un échantillon du corpus sert à entrainer le modèle, qui sera testé sur le reste du corpus. Ce procédé est répété dix fois, en changeant l'échantillon d'entrainement à chaque itération.

- **Génération du modèle KNN**

Les résultats obtenus lors l'entrainement de l'algorithme IBK sont illustrés dans la figure 3.11 :

```
Correctly Classified Instances        7875               52.5    %
Incorrectly Classified Instances      7125               47.5    %
Kappa statistic                          0.1491
Mean absolute error                      0.3298
Root mean squared error                  0.5191
Relative absolute error                 82.1539 %
Root relative squared error            115.8767 %
Total Number of Instances            15000

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0,264    0,151    0,323      0,264    0,290      0,122    0,580     0,272     negatif
               0,747    0,612    0,589      0,747    0,659      0,146    0,591     0,600     neutre
               0,264    0,100    0,464      0,264    0,336      0,204    0,609     0,355     positif
Weighted Avg.  0,525    0,387    0,501      0,525    0,501      0,155    0,593     0,470

=== Confusion Matrix ===

    a    b    c   <-- classified as
  847 2017  348 |    a = negatif
 1266 6054  779 |    b = neutre
  512 2203  974 |    c = positif
```

FIGURE 3.11 : Résultat d'entrainement du classifieur KNN.

V.4.2 Evaluation des modèles obtenus

Cette étape consiste à évaluer les performances des modèles déjà générés en utilisant un corpus de test. Pour ce faire, nous avons utilisé un corpus de 5001 commentaires afin de prédire les classes de ces dérniers. Lors de chaque évaluation, WEKA calcule quatre scores pour mesurer l'efficacité de la classification comme déjà vu dans le chapitre 2 : la précision, le rappel, la F-mesure et l'accuracy. :

Les résultats obtenus lors de l'évaluation du classifieur KNN sont illustrés dans la figure3.12 :

```
Correctly Classified Instances        2918               58.3483 %
Incorrectly Classified Instances      2083               41.6517 %
Kappa statistic                          0.1125
Mean absolute error                      0.3047
Root mean squared error                  0.488
Total Number of Instances             5001

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,248    0,161    0,267      0,248    0,257      0,090    0,588     0,242     negatif
                 0,742    0,635    0,714      0,742    0,728      0,109    0,568     0,723     neutre
                 0,241    0,096    0,268      0,241    0,254      0,151    0,613     0,197     positif
Weighted Avg.    0,583    0,476    0,572      0,583    0,577      0,111    0,578     0,564

=== Confusion Matrix ===

   a    b    c   <-- classified as
 237  630   87 |    a = negatif
 547 2527  333 |    b = neutre
 104  382  154 |    c = positif
```

FIGURE 3.12 : Résultat d'évaluation de classifieur KNN

Pour évaluer les performances des modèles générés, nous nous intéressons à la matrice de confusion, et les critères accuracy, précision, rappel (recall) et F-Score.
Les résultats qui se trouvent au fil de ces pages ont été obtenus en utilisant le même corpus de test, pour chaque classifieur nous avons donné la matrice de confusion obtenu ainsi que les indicateurs de performances de ce dernier.

V.5 Le modèle basé sur KNN

Le tableau 3.2 représente la matrice de confusion obtenu lorsque nous avons testé le modèle généré par le classifieur KNN.

	Classe prédite negatif	Classe prédite neutre	Classe prédite positif
Vraie negatif	(VN) 237	(FNT) 630	(FP) 87
Vraie neutre	(FN) 547	(VNT) 2527	(FP) 333
Vraie positif	(FN) 104	(FNT) 382	(VP) 154

TABLE 3.2 : Matrice de confusion du classifieur KNN.

D'après le résultat de la matrice de confusion (Figure 3.2) on distingue que les erreurs de classification ont concerné les 3 classes (positif, négatif et neutre) :

-Le nombre des instances correctement classées = 2918(237 : classe " negatif ", 2527 : classe " neutre ", 154 : classe " positif").

-Le nombre des instances mal classées = 2083 (651 : classe " negatif ", 1012 : classe " neutre ", 420 : classe " positif ").

Les indicateurs de performance relatifs à cette matrice sont illustrés dans le tableau 3.3

Indicateurs de performance	**Accuracy**	**Précision**	**Rappel**	**F-measure**
	0.58	0.57	0.58	0.57

TABLE 3.3 : Les indicateurs de performances de classifieurs KNN.

V.6 Le modèle basé sur SVM

Le tableau 3.4 représente la matrice de confusion obtenu lorsque nous avons testé le modèle généré par le classifieur SVM.

	Classe prédite negatif	Classe prédite neutre	Classe prédite positif
Vraie negatif	(VN) 301	(FNT) 600	(FP) 53
Vraie neutre	(FN) 389	(VNT) 2690	(FP) 328
Vraie positif	(FN) 68	(FNT) 301	(VP) 271

TABLE 3.4 : Matrice de confusion du classifieur SVM.

D'après le résultat de la matrice de confusion (Tableau3.4) on distingue que :

-Le nombre des instances correctement classées = 3262 (301 : classe " negatif ", 2692 : classe " neutre ", 271 : classe " positif ")

-Le nombre des instances mal classées = 1739(457 : classe " negatif ", 309 : classe " neutre ", 381 : classe " positif ")

Les indicateurs de performance relatifs à cette matrice sont illustrés dans le tableau 3.5

Indicateurs de performance	**Accuracy**	**Précision**	**Rappel**	**F-measure**
	0.65	0.63	0.65	0.64

TABLE 3.5 : Les indicateurs de performances de classifieur SVM.

V.7 Le modèle basé sur arbre de décision

Le tableau 3.6 représente la matrice de confusion obtenu lorsque nous avons testé le modèle généré par le classifieur C.4.5.

	Classe prédite negatif	Classe prédite neutre	Classe prédite positif
Vraie negatif	(VN) 312	(FNT) 547	(FP) 95
Vraie neutre	(FN) 530	(VNT) 2440	(FP) 437
Vraie positif	(FN) 77	(FNT) 292	(VP) 271

TABLE 3.6 : Matrice de confusion du classifieur C.4.5

D'après le résultat de la matrice de confusion (Tableau3.6) on distingue que : -Le nombre des instances correctement classées = 4166 (312 : classe " negatif ", 2440 : classe " neutre ", 271 : classe " positif ")

-Le nombre des instances mal classées = 1978 (601 : classe " negatif ",839 : classe " neutre ", 532 : classe " positif ")

Les indicateurs de performance relatifs à cette matrice sont illustrés dans le tableau 3.7

Indicateurs de performance	**Accuracy**	**Précision**	**Rappel**	**F-measure**
	0.60	0.61	0.60	0.60

TABLE 3.7 : Les indicateurs de performances de classifieur C.4.5

V.8 Le modèle basé Naïve bayes

Le tableau 3.8 représente la matrice de confusion obtenu lorsque nous avons testé le modèle généré par le classifieur Naïve Bayes.

	Classe prédite negatif	Classe prédite neutre	Classe prédite positif
Vraie negatif	(VN) 395	(FNT) 375	(FP) 184
Vraie neutre	(FN) 765	(VNT) 1949	(FP) 693
Vraie positif	(FN) 121	(FNT) 185	(VP) 334

TABLE 3.8 : Matrice de confusion du classifieur Naïve Bayes.

D'après le résultat de la matrice de confusion (Tableau3.8) on distingue que :

-Le nombre des instances correctement classées = 2678 (395 : classe " negatif ", 1949 : classe " neutre ", 334 : classe " positif ")

-Le nombre des instances mal classées = 2323 (886 : classe " negatif ",560 : classe " neutre ", 877 : classe " positif ")

Les indicateurs de performance relatifs à cette matrice sont illustrés dans le tableau 3.9 :

Indicateurs de performance	**Accuracy**	**Précision**	**Rappel**	**F-measure**
	0.53	0.62	0.53	0.56

TABLE 3.9 : Les indicateurs de performances de classifieur Naïve Bayes.

V.9 Récapitulation des indicateurs de performance obtenus

Les indicateurs obtenus / Les classifieurs	Accuracy	Précision	Rappel	F-measure
SVM	0.65	0.63	0.65	0.64
Naïve Bayes	0.53	0.62	0.53	0.56
Arbres décisions	0.60	0.61	0.60	0.60
KNN	0.58	0.57	0.58	0.57

TABLE 3.10 : Récapitulation indicateurs de performances des 4 classifieurs.

Les résultats obtenus avec nos quatre classifieurs sont compris entre 0 et 1 et ce qui indique que ces derniers sont acceptables.

VI Recalcule des polarités d'opinion de DICO

L'objectif de notre présente étude est de proposer une amélioration dans la classification d'opinion donnée par la ressource lexicale DICO. Pour rappel, les polarités de DICO sont celle de SentiWordNet qui à priori ne correspondent pas à un domaine particulier. Nous proposons alors de recalculer ces polarités en nous basons sur des classifieurs entrainés avec un corpus issus du domaine de l'éducation comme le montre la figure 3.13.

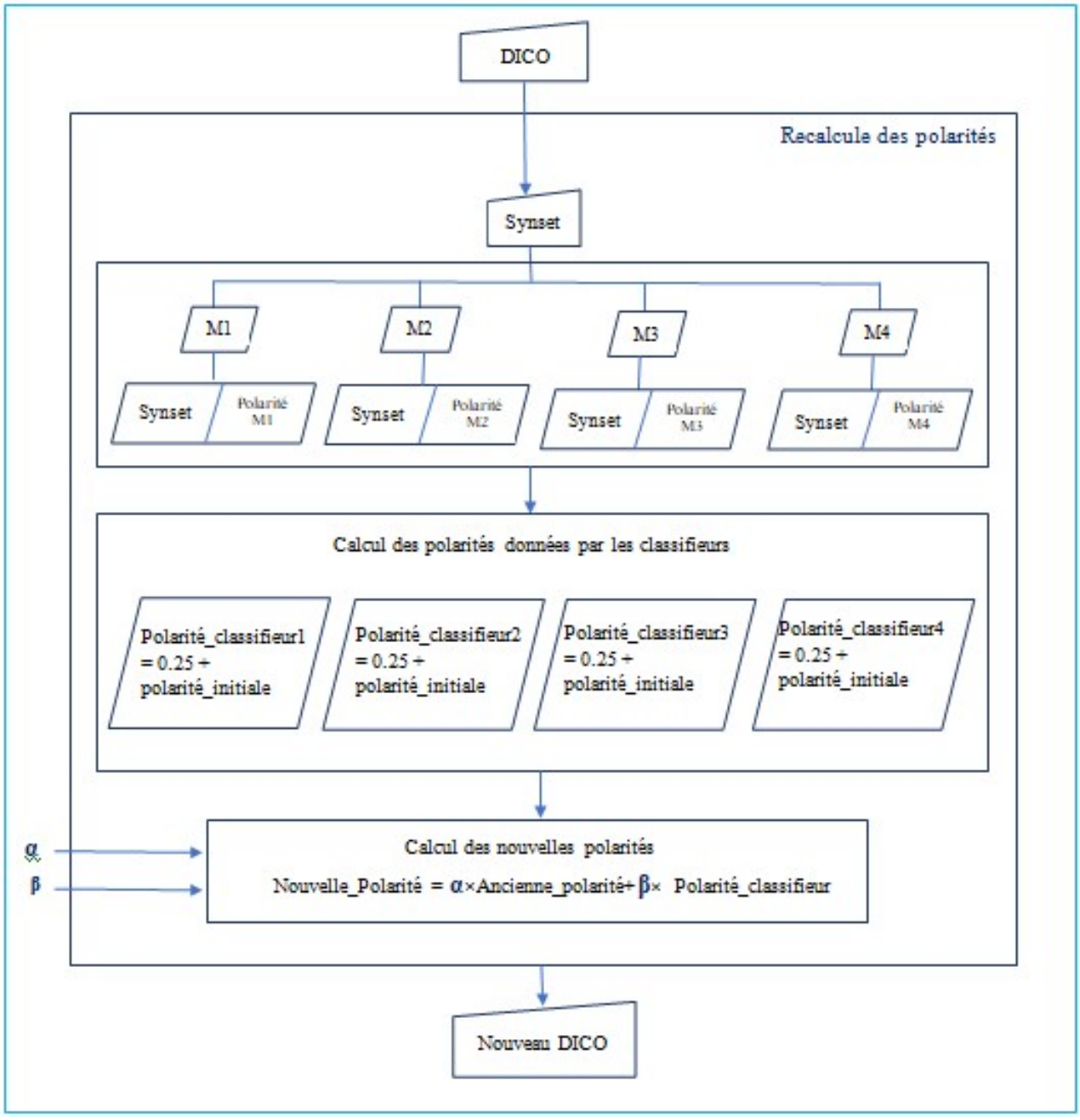

FIGURE 3.13 : Construction du nouveau DICO.

Le recalcule des polarités de DICO revient à attribuer pour chaque synset de cette dernière de nouvelles polarités calculées à partir des classifications de ces Synset avec les modèles construits dans la section précédente. Les polarités calculées sont données par la formule :

$$PolariteeCalculee = 0.25 + polariteeInitiale \tag{3.1}$$

soit :

$$PolariteePositiveCalculee = PolariteePositiveInitiale + 0.25 \tag{3.2}$$

$$PolariteeNegativeCalculee = PolariteeNegativeInitiale + 0.25 \tag{3.3}$$

$$PolariteeObjectiveCalculee = PolariteeObjectiveInitiale + 0.25 \tag{3.4}$$

Les nouvelles polarités de DICO sont données selon la formule :

$$NouvellePolariteePositive = \alpha * ancienne PolariteePositive + \beta * polariteePositiveCalculee. \tag{3.5}$$

$$NouvellePolariteeNegative = \alpha * ancienne PolariteeNegative + \beta * polariteeNegativeCalculee. \tag{3.6}$$

$$NouvellePolariteeNeutre = \alpha * ancienne PolariteeNeutre + \beta * polariteeNeutreCalculee. \tag{3.7}$$

Le détail de ce recalcule est donné par l'algorithme de la figure 3.14. Où α et β sont à 0.5.

```
Algorithm Recalcule_polarité

Input: Listes polarité classifieurs L1, L2, L3, L4 ;
Listes anciennes polarités dico : Liste-ancienne-polarité-pos, Liste-ancienne-polarité-neg,
Liste-ancienne-polarité-net ;
Output : nouvelles polarités de dico : Liste-nouvelles-p-pos, Liste-nouvelles-p-neg
Liste-nouvelles-p-net ;

VAR Positif, negatif, objectif : integer ;
   L1, L2, L3, L4, Liste-polarité-classifieur-positif_i, Liste-polarité-classifieur-negatif_i,
Liste-polarité- classifieur-neutre_i : list ;
Begin
Positif:=0; Negatif:=0; Objectif:=0;
Load (L1, L2, L3, L4);
Foreach Li Do
    Begin
      If Li = Positif then
         Positif := Positif +0.25 ;
         Liste-polarité-classifieur-positif_i := Positif ;
     If Li = Negatif then
        Negatif := Negatif + 0.25;
        Liste-polarité-classifieur-negatif_i := Negatif ;
     If Li = Neutre then
        Objectif := Objectif + 0.25;
        Liste-polarité-classifieur-neutre_i := Neutre;
   End
For L1 Do
   Begin
      Liste-polarité_positif := Somme (Liste-polarité-classifieur-positif_i);
      Liste-polarité_negatif:= Somme (Liste-polarité-classifieur-negatif_i);
      Liste-polarité_neutre := Somme (Liste-polarité-classifieur-neutre_i);
   End
Load(Liste-ancienne-polarité-pos, Liste-ancienne-polarité-neg, Liste-ancienne-polarité-net) ;
For L1Do
  Begin
     Liste-nouvelles-p_pos:=0.5*Liste-polarité_positif+0.5*liste_ancienne_ polarité_pos ;
     Liste-nouvelles-p_neg:= 0.5*Liste-polarité_negatif+0.5*liste_ancienne_polarité_neg ;
     Liste-nouvelles-p_net:= 0.5*Liste-polarité_neutre+0.5*liste_ancienne_ polarité_net ;
  End
End
```

FIGURE 3.14 : Algorithme de recalcule de polarité de DICO.

La figure 3.15 ci-dessous illustre un échantillon de la ressource lexicale DICO après recalcule de polarités.

ID	POS	Termes	Posscore	Negscore	Objscore	Nouveau_Posscor	Nouveau_Negsco	Nouveau_Objscore
00001740	a	capable	0	0.125	0.875	0.125	0.0625	0.8125
00002098	a	incapable	0.75	0	0.25	0.5	0	0.5
00002312	a	abaxial	0	0	1	0.125	0	0.875
00002527	a	ventral	0	0	1	0.125	0	0.875
00003553	a	naissant	0	0	1	0.125	0	0.875

FIGURE 3.15 : Fragment de la ressource lexicale DICO après recalcule de polarité.

VI.1 Statistique de DICO

Le tableau 3.11 ci-dessous illustre les informations concernant la taille et le nombre de synsets ayant une polarité (positive, negative, neutre) pour chacun de nouveau et ancien DICO.

Données	ancien DICO	nouveau DICO
Taille de DICO	56475	56475
Nombres des synset avec polarité	12755	53113
Nombres des synset sans polarité (neutre)	52091	55544
Nombre de synset positif dans DICO	1694	31
Nombre de synset négatif dans DICO	1057	356

TABLE 3.11 : Statistique de DICO.

Conclusion

Dans ce chapitre nous avons expliqué les étapes de réalisation de notre projet. A travers les différentes sections nous avons présenté la description de la construction initiale de la ressource lexicale DICO : processus de construction, la structure. Nous avons abordé la démarche suivie pour la construction du corpus EDUCA.Nous avons expliqué les étapes suivies pour l'entrainement des différents classifieurs avec WEKA ainsi que les différents résultats obtenus. Nous avons, enfin développé la démarche de recalcule des polarités de DICO en se basant sur les modèles de classification précédemment obtenus.

Dans chapitre suivant nous allons présenter la validation de notre proposition en discutant les résultats obtenus.

Chapitre 4

Evaluation

Introduction

Dans ce chapitre, nous passons à la phase d'évaluation, dans laquelle nous serons amenés à évaluer notre ressource lexicale DICO, et ceci en se basant sur une certaine démarche décrite dans ce qui suit.

I Description de la démarche

Une fois l'étape de recalcule de polarités est achevée, nous avons procédé à l'évaluation de notre ressource avec une démarche décrite dans la figure 4.1 qui consiste en ces étapes :

- Reprendre le corpus EDUCA.
- Classification de corpus EDUCA selon les anciennes polarités de DICO.
- Classification de corpus EDUCA selon les nouvelles polarités de DICO.
- Comparaison des résultats de classification.

Les résultats des deux classifications ont été sauvegardés, pour être comparés ultérieurement. Dans ce qui suit nous allons approfondir ces étapes et voir de plus près les résultats obtenus.

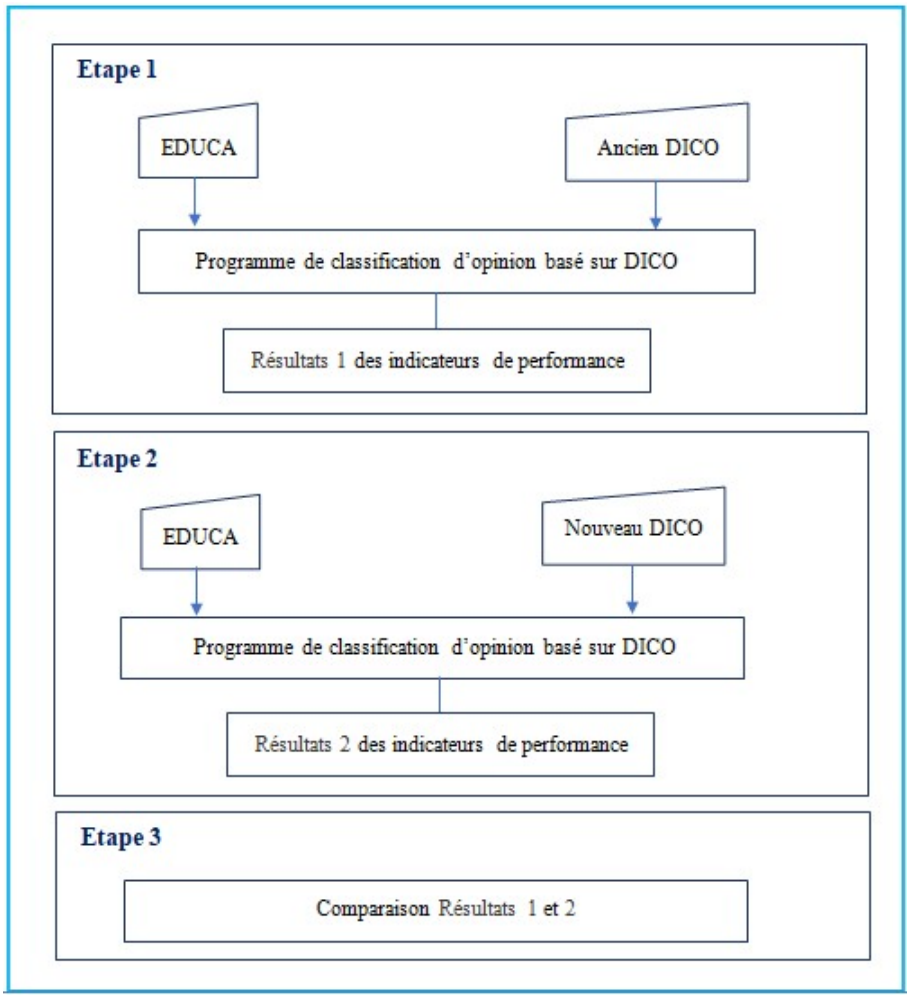

FIGURE 4.1 : Evaluation de DICO.

II Classification selon la ressource lexicale DICO

Afin de procéder à une classification d'opinion de textes selon la ressource lexicale DICO, nous nous basons sur une méthode de classification symbolique développée dans [2].Cette méthode reçoit en entrée un texte court et donne en sortie une classification de ce texte selon l'algorithme suivant 4.2 :

```
Algorithm 2 Extratct_Free_Opinion
1:  Input FA: Text; /*Free answer */
2:  Output   VOL: Integer; FA_Tab: List of Word with POS;
3:  VAR FA_Tab: List_of_tagued_Words;
4:          S_Positif, S_Negatif, S_Neutral :integer;
5:          IDS: ID_Synset;
6:          L_S : liste_de_synset ;
7:  Begin
8:     VOL :=0 ; S_Positif :=0 ; S_Negatif :=0 ; S_Neutral :=0 ;
9:     Pretreatment (FA, FA_Tab);
10: Foreach Word in FA_Tab Do
11: Begin
12:   If Found(Word in DICO) Then
13:    Begin
13:      If Unique (Word in DICO) Then
14:        Begin
15:          S_Positif:= S_Positif + synset(Word, PosScore);
16:          S_Negatif:= S_Negatif+ synset(Word, NegScore);
17:          S_Neutral:= S_Neutral+ synset(Word, ObjScore);
18:        End
19:      Else
20:        Begin
21:          L_S := Liste_synset (Word, Dico) ; /*liste des synset correspondants au
                                                  mot dans DICO*/
22:          Desembiguisation (Word, FA_Tab, L_S, IDS);
23:          S_Positif:= S_Positif + synset(ID_S, PosScore);
24:          S_Negatif:= S_Negatif+ synset(ID_S, NegScore);
25:          S_Neutral:= S_Neutral+ synset(ID_S, ObjScore);
26:        End
27:    Else
28:      Begin
29:        Store(Word,Temporary);
30:        Attribute_Polarity(SynSet (Word),Polarity(Pos, Neg, Neut));
31:        Store(Synset(Word, Polarity), DICO);
32:      End
33:    End
34:    If (S_Positif>S_Negatif) and (S_Positif>=S_Neutral) then VOL=1
35:     Else If (S_Negatif>S_Positif) and (S_Negatif>=S_Neutral) then VOL=-1
36:      Else If (S_Neutral >S_Positif) and (S_Neutral >S_Negatif) then VOL= 0
37:       Else If  (S_Positif=S_Negatif) then VOL= 0
38: End
```

FIGURE 4.2 : Algorithme d'extraction de l'opinion libre [2].

II.1 Classification selon DICO initial

Nous avons procédé à la classification du corpus EDUCA selon les polarités initiales de DICO et nous avons obtenu la matrice de confusion illustrée dans le tableau 4.1.

	Classe prédite positif	Classe prédite negatif	Classe prédite neutre
Vraie positif	8	31	4280
Vraie negatif	8	1	955
Vraie neutre	56	16	14647

TABLE 4.1 : Matrice de confusion de classification selon DICO initial.

Nous avons calculé les indicateurs de performance relatifs à cette matrice et nous les avons illustré dans le tableau 4.2.

Indicateurs de performance	**Accuracy**	**Précision**	**Rappel**	**F-measure**
	0.73	0.29	0.33	0.31

TABLE 4.2 : Indicateurs de performances de la classification selon DICO initial.

II.2 Classification selon le nouveau DICO

Nous avons ensuite reclassifier le corpus EDUCA selon les nouvelles polarités de DICO et nous avons obtenu la matrice de confusion indiquée dans le tableau 4.3.

	Classe prédite positif	Classe prédite negatif	Classe prédite neutre
Vraie positif	64	1	4254
Vraie negatif	5	6	953
Vraie neutre	34	26	14659

TABLE 4.3 : Matrice de confusion de la classification selon le nouveau DICO.

A cette matrice de confusion correspondent les indicateurs de performance montrés dans le tableau 4.4

Indicateurs de performance	**Accuracy**	**Précision**	**Rappel**	**F-measure**
	0.74	0.50	0.34	0.40

TABLE 4.4 : Indicateurs de performances de la classification selon le nouveau DICO.

III Discussion des résultats

	Accuracy	Précision	Rappel	F-measure
DICO initial	0.73	0.29	0.33	0.31
Nouveau DICO	0.74	0.50	0.34	0.40

TABLE 4.5 : Récapitulation des deux classifications

A partir des résultats obtenus on remarque une légère amélioration des performances du nouveau DICO par rapport à l'ancien DICO.

Conclusion

Dans ce chapitre, nous avons effectué deux classifications du corpus EDUCA. La première est effectuée selon les polarités initiales de DICO obtenues de SentiWordNet. La deuxième est réalisée selon les nouvelles polarités de DICO calculées à partir des modèles de classifieurs entrainés sur des données issus du domaine de l'éducation. Nous avons ensuite comparé les résultats obtenus pour chacun des cas. Nous avons constaté une légère amélioration dans les indicateurs de performance avec les nouvelles polarités de DICO.

Conclusion générale

Le travail présenté dans ce mémoire s'inscrit dans le cadre de la fouille d'opinion. Il consiste à contribuer dans la construction d'une ressource lexicale dédié au domaine de l'éducation.

La fouille d'opinion est un nouvel axe de recherche qui offre une multitude de façons de traitement de langues. Elle utilise des méthodes qui assurent la classification automatique des opinions qui nécessite l'utilisation des ressources pour avoir des résultats concret.

Ces ressources sont tributaires du domaine d'application puisque les mots n'ont pas la même orientation dans tous les domaines, elle change selon le contexte, tel que, la polarité d'un mot employé dans un tel domaine est différente de la polarité du même mot employé dans un autre domaine.

Notre objectif consiste à améliorer une ressource lexicale existante DICO construite à partir de WOLF et SentiWordNet dédiée au domaine de l'éducation.

Pour ce faire nous avons construit un corpus de commentaires nommé EDUCA collectés de différentes sources du domaine de l'éducation. Ce corpus a été annoté manuellement selon trois classes d'opinion, positive, négative et neutre.

Nous avons procédé à l'entrainement des classifeurs à base du corpus EDUCA afin d'avoir des modèles de classification de l'opinion.

L'objectif visé est d'utiliser ces modèles pour le recalcule de polarités de DICO et

d'avoir une nouvelle ressource lexicale.

Nous avons testé notre proposition en classifiant notre corpus EDUCA selon la ressource initiale DICO et selon la nouvelle ressource lexicale.

Les résultats obtenus ont montré une légère amélioration des indicateurs de performance dans la classification selon la nouvelle ressource par rapport à la classification selon DICO initiale.

Néanmoins les résultats obtenus ouvrent des perspectives de perfectionnement des résultats à court terme comme :

- Travailler avec un corpus plus consistant avec au minimum 50000 commentaires.
- Utiliser une annotation semi-automatique pour pouvoir ainsi améliorer la qualité de l'annotation.
- Utiliser plus de classifieurs pour raffiner les valeurs de polarités.

A long terme :

- Généraliser la démarche pour construire une ressource lexicale multi langue (arabe et thamazight) pour mieux l'adapter aux contextes Algérien.
- Généralisation à d'autres domaines où l'analyse d'opinion peut trouver des applications.

Bibliographie

[1] Saliha GAGUI Randa BENKHELIFA. Fouille de données d'opinion des usagers de sites E-commerce. Master, université d'Ouargla, Ouargla, 2013. Url :https ://dspace.univ-ouargla.dz/jspui/bitstream/123456789/1620/1/Master-

[2] samia LAZIB ep CHOUAKI. Un système pour la E-Orientation Scolaire Intégrant l'Analyse d'Opinion. Thèse, université de Mouloud Mammeri Tizi-Ouzou, Tizi-Ouzou, 2020.

[3] Dominique Boullier et Audrey Lohard. OPINION MINING ET SENTIMENT ANALYSIS :Méthodes et outils. OpenEdition Press, Marseille, 2012. URL : http ://-books.openedition.org/oep/214

[4] Morgan MARCHAND. Domaines et fouille d'opinion. Thèse, université de Paris sud, Paris, 2015. Url :https ://tel.archives-ouvertes.fr/tel-01157951/document

[5] Bing Liu. Sentiment analysis and subjectivity. University of Illinois at Chicago,2010 URL : http ://citeseerx.ist.psu.edu/viewdoc/summary ?doi=10.1.1.216.5533

[6] Fatma Zohra CHABBOU , Souhaila BAKHOUCHE. Fouille d'opinions méthodes et outils. Master, université de Tébessa,Tébessa, 2016. Url :http ://univ-tebessa.dz/fichiers/masters/sesnv160079.pdf

[7] Nadia LAMAMRI, Taher GUERRAM . Analyse des opinions exprimées sous forme de textes arabes.Master Université Oum El Bouaghi,Oum El Bouaghi,2013.

[8] Faiza BELBACHIR. Expérimentation de fonctions pour la détection d'opinions dans les blogs. Mster, université de Paul Sabatier, Toulouse, 2010. Url : http ://bib.univ-oeb.dz :8080/jspui/bitstream/123456789/6819/1/m

[9] Sigrid MAUREL, Paolo CURTONI et Luca DINI. L'analyse des sentiments dans les forums. URL :http ://www2.lirmm.fr/ mroche/FODOP08/ArticlesFODOP08/Article2.pdf

[10] Damien POIRIER et al. Approches Statistique et Linguistique Pour la Classi ?cation de Textes d'Opinion Portant sur les Films. 2010 Url : https ://hal.archives-ouvertes.fr/hal-00466412/document

[11] https ://hal-lirmm.ccsd.cnrs.fr/lirmm-00764371/document

[12] Caroline COLLET, Alexandre PAUCHET, Laurent VERCOUTER, Khaled KHELIF. ?LITIS-Avenue de l'Université - BP 8 76801 Saint-Étienne-du-Rouvray Cedex. Url : https ://editions-rnti.fr/renderpdf.php ?p=1001964

[13] https ://www.24pm.com/117-definitions/512-apprentissage-supervise

[14] https ://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/

[15] Oussama ZIRI. Classification de courriels au moyen de diverses méthodes d'apprentissage et conception d'un outil de préparation des données textuelles basé sur la programmation modulaire,université du Québec à Montréal . Url : https ://-core.ac.uk/download/pdf/19501296.pdf

[16] https ://inf1421.teluq.ca/teluqDownload.php ?file=2016/09/INF1421-Module7-ArbreDeDecisonJan2019.pdf

[17] Hanane EZZIKOURI, Mohamed FAKIR. Algorithmes de classification :ID3 et C4.5. Url : https ://www.academia.edu/33701469/Algorithmes de classification ID3 and C4 5

[18] Ahmed ZEGGADA Rabah MOULAI catégorisation des textes arabes blida 2019. Url : http ://di.univ-blida.dz :8080/xmlui/bitstream/handle/123456789/3524/Zeggada

[19] https ://hal.inria.fr/inria-00494814/document

[20] https ://cache.media.eduscol.education.fr/file/NSI/76/6/RALyceeGNSIalgoknn1170766.pdf

[21] https ://cours.etsmtl.ca/gti770/private/labos/Lab1/LOG770-Labo1-Enonce.pdf

[22] https ://wiki.pentaho.com/display/DATAMINING/Data+Mining+Algorithms+and+ Tools+in+Weka

[23] https ://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203345-clustering-definition/

[24] https ://www.caiac.ca/sites/default/files/publications/ProjetMaitrise1430188.pdf

[25] Gillot Sébastien.Fouille d'opinion. Mémoire de master .2010 .Url : http ://www.univ-tebessa.dz/fichiers/masters/sesnv160079.pdf

[26] www.developers.google.com/machine-learning/crash-course/classification/precision-and-recall ?h1=fr

[27] Barbera, Manuel. Linguistica dei corpora e linguistica dei corpo-raitaliana. Un'introduzione. Milano, 2013 : Qu. A.S.A.R. s.r.l. Url : http ://www.bmanuel.org/man/BarberaIntroduzioneCL2013=Ver1-54.pdf

[28] https ://www.youtube.com/watch ?v=XAMxYZu36C4

[29] Teresa CABRÉ. Constituer un corpus de textes de specialité M, Institut Universitari de LingüísticaAplicadaUniversitatPompeuFabra, Barcelone.

[30] https ://theses.univ-lyon2.fr/documents/getpart.php ?id=lyon2.2005.ahroniancpart=90677

[31] Geoffrey Leech,Developing Linguistic Corpora :a Guide to Good Practice Adding Linguistic Annotation , Lancaster University, 2004 Url :http ://users.ox.ac.uk/ martinw/dlc/chapter2.htm

[32] https ://journals.openedition.org/corela/4857

[33] Poudat, Frédéric Landragin. Explorer un corpus textuel, 2017. Url : https ://-books.google.dz/books ?id=sawzDgAAQBAJprintsec=frontcoverhl=frsource= gbsge-summaryrcad=0v=onepageqf=false

[34] Cyril Grouin. Annotations manuelles et automatiques de corpus. url :cyril.grouinlimsi.fr - https ://perso.limsi.fr/grouin/tbilissi/

[35] SHS 27 shsconf/201627 Congrès Mondial de Linguistique Française - CMLF 2016. Url : https ://halshs.archives-ouvertes.fr/halshs-01350795/document

[36] Bruno Guillaume, Guy Perrier. Reflexion sur l'annotation de corpus ecrits du français en syntaxe et en semantique,Orléans,2017. Url : https ://hal.inria.fr/hal-01651753/document

[37] https ://www.researchgate.net/publication/309772011AnnotatedCorpusConstruction

[38] SentiMI : Introducing point-wise mutual information withSentiWordNet to improve sentiment polarity detection.

[39] SENTIWORDNET3.0 : An Enchanced Lexical Resource for sentiment Analysis and Opinion Mining . Url : https ://www.researchgate.net/publication/220746537SentiWordNet30AnEnhancedLexical ResourceforSentimentAnalysisandOpinionMining

[40] https ://github.com/aesuli/SentiWordNet/blob/master/data/SentiWordNet3.0.0.txt

[41] http ://blog.onyme.com/etude-de-lontologie-wordnet-libre-du-francais-wolf/

[42] https ://hal.inria.fr/inria-00614707/file/TALN08.pdf

[43] https ://fr.slideshare.net/franciscoyes/scrapy-42681497

[44] https ://www.cordial.fr/comment-fonctionne-correcteur-cordial

[45] https ://www.fil.univ-lille1.fr/ decomite/ue/APE/tp/tp1/weka2009.pdf

[46] https ://facemweb.com/referencement-naturel-seo/lemmatisation

[47] https ://www.cis.uni-muenchen.de/ schmid/tools/TreeTagger/

[48] http ://eric.univ-lyon2.fr/ ricco/tanagra/sise/LogicielsOct2016/3Weka.pdf

[49] Rémi Eyraud, Classification, Apprentissage,Décision. url :http ://pageperso.lif.univ-mrs.fr/ remi.eyraud/CAD/theorie.pdf

Annexe A

le programme de conversion de fichier txt en fichier Arff

```
package weka_api;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.util.ArrayList;

public class essai {
 static ArrayList inList=new ArrayList();
 static String colNames[];
 static String colTypes[];
 static String indata[][];
 static ArrayList clsList=new ArrayList();
 static ArrayList disCls=new ArrayList();
 static  String res="";
```

```
public String genTrain() throws IOException{
File fe=new File("corpus.txt");
FileInputStream fis=new FileInputStream(fe);
byte bt[]=new byte[fis.available()];
fis.read(bt);
fis.close();
String st=new String(bt);
String s1[]=st.trim().split("\n");
String col[]=s1[0].trim().split("\t");
colNames=col;
colTypes=s1[1].trim().split("\t");
for(int i=2;i<s1.length;i++){
 inList.add(s1[i]);}
ArrayList at=new ArrayList();
for(int i=0;i<inList.size();i++){
 String g1=inList.get(i).toString();
 if(!g1.contains("/")){
  at.add(g1);
  res=res+g1+"\n"; }}
  indata=new String[at.size()][colNames.length-1];
  for(int i=0;i<at.size();i++ {
    String s2[]=at.get(i).toString().trim().split("\t");
    for(int j=0;j<s2.length-1;j++){
     indata[i][j]=s2[j].trim();}
     if(!disCls.contains(s2[s2.length-1].trim()))
     disCls.add(s2[s2.length-1].trim());
     clsList.add(s2[s2.length-1]);}
          String ar="@relation terme\n";
          try{
     for(int i=0;i<colNames.length-1;i++){
          if(colTypes[i].equals("con"))
ar=ar+"@attribute "+colNames[i].trim().replace(" ","_")+" String\n";
```

```
else {ArrayList at1=new ArrayList();
  for(int j=0;j<indata.length;j++){
  if(!at1.contains(indata[j][i].trim()))
   at1.add(indata[j][i].trim());}
   String sg1="{";
   for(int j=0;j<at1.size();j++){
     sg1=sg1+at1.get(j).toString().trim()+","; }
     sg1=sg1.substring(0,sg1.lastIndexOf(","));
     sg1=sg1+"}";
     ar=ar+"@attribute "+colNames[i].trim().replace
(" ", "_")+" "+sg1+"\n";
   }
}
ArrayList dis=new ArrayList();
String c1="";
for(int i=0;i<clsList.size();i++){
  String g=clsList.get(i).toString().trim();
  if(!dis.contains(g)){
   dis.add(g);
   c1=c1+g+",";}}
   c1=c1.substring(0, c1.lastIndexOf(","));
   ar=ar+"@attribute class {"+c1+"}\n";
   ar=ar+"@data\n";
   for(int i=0;i<indata.length;i++) {
    String g1="";
    for(int j=0;j<indata[0].length;j++){
       g1=g1+indata[i][j]+",";}
       g1=g1+clsList.get(i);
       ar=ar+g1+"\n";}}
     catch(Exception e)
     {
       e.printStackTrace();
```

```
                }
                return ar;
            }
        public static void main(String[] args) throws IOException {
    essai T2A=new essai();
         String ar1=T2A.genTrain();
        File fe1=new File("C:\\Users\\Lenovo\\Documents\\arf\\essai.arff"
    FileOutputStream fos1=new FileOutputStream(fe1);
         fos1.write(ar1.getBytes());
         fos1.close();
        }
}
```

Annexe B

le programme de classification du corpus selon DICO

```
package memoire;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.IOException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.Collections;
import java.util.HashMap;
import java.util.Iterator;
import java.util.LinkedHashSet;
import java.util.List;
import java.util.Map;
```

```
import java.util.Map.Entry;

public class classificationdicolem {
public static void main(String[] args) throws InstantiationException,
IllegalAccessException, ClassNotFoundException,SQLException,IOException{
PreparedStatement inst = null;
Connection connexion = null;
PreparedStatement stmt = null;
Class.forName("com.mysql.cj.jdbc.Driver").newInstance();
connexion = DriverManager.getConnection("jdbc:mysql:
useTimezone=true&serverTimezone=UTC","root","");

List<String> mot_dico = new ArrayList<String>();
BufferedReader bf = new BufferedReader
(new FileReader("C:\\Users\\pc1\\Desktop\\Termes_ania.txt"));
for(String mot;(mot= bf.readLine()) != null;)
mot_dico.add(mot.trim());
br.close();
File file = new File
("C:\\Users\\pc1\\Desktop\\MEMOIRE 2020 FIN\\EDUCA-LEMMATISE.txt");
BufferedReader bufferedReader = null;
ArrayList <String> commentaire = null;
Map <Integer,ArrayList> EDUCA = new HashMap <Integer ,ArrayList>();
try {
                FileReader filereader = new FileReader(file);
                bufferedReader = new BufferedReader(filereader);
                String line;
                int cp =0;
                while ((line = bufferedReader.readLine()) != null ) {
                commentaire = new ArrayList <String>();
                cp++;
                String[] newString = line.split("['()''|/*+@%$#{_}  ,;:'\
```

```
                for (String ss: newString) {
                commentaire.add(ss);
                        }
                EDUCA.put(cp,  commentaire);
                }
        }
                }
                // le calcules
                double pos=0;
                double positif =0;
                double neg=0;
                double negatif=0;
                double net=0;
                double neutre=0;
                double positif1 =0;
                double negatif1=0;
                double neutre1=0;

List<Double> posi  = new ArrayList<>();
List<Double> negati  = new ArrayList<>();
List<Double> neut  = new ArrayList<>();
Iterator it = text.iterator() ;
while(it.hasNext()){
Object o = it.next() ;
if(  mot_dico.contains( o )) {
int frequence = Collections.frequency(mot_dico, o);
int nbr = frequence;
inst = connexion.prepareStatement("select Nouveau_Posscore,Nouveau_Negsco
Nouveau_Objscore from nouveaudicolem  where Termes='"+o+"'");
ResultSet rs = inst.executeQuery();
if (nbr == 1)  {
 while (rs.next()){
```

```
 positif =  rs.getDouble("Nouveau_Posscore");
 negatif =  rs.getDouble("Nouveau_Negscore");
 neutre=  rs.getDouble("Nouveau_Objscore");
                }
}else {
while (rs.next()){
 posi.add(rs.getDouble("Nouveau_Posscore")/nbr);
 negati.add(rs.getDouble("Nouveau_Negscore")/nbr);
 neut.add(rs.getDouble("Nouveau_Objscore")/nbr);
        }
                    }

                }
                }
        for(int i=0; i <posi.size(); i++) {
                positif1 = positif1+ posi.get(i);
        }
        for(int i=0; i <negati.size(); i++) {
                negatif1 = negatif1+ negati.get(i);
        }
        for(int i=0; i <neut.size(); i++) {
                neutre1 = neutre1+ neut.get(i);
        }

         pos = positif+ positif1;
             neg = negatif + negatif1;
             net = neutre + neutre1;


            int vol =0;
            if(pos>neg && pos>= net) {
                vol =1;}
```

```
                else if( neg>pos && neg >= net ){
                        vol = -1;
            }else if( net>pos && net >= neg ){
                vol = 0;
            } else if( pos>neg ){
                vol = 0;
            }
inst = connexion.prepareStatement("INSERT INTO
classificationnouvaudicolem (ID,polarite) VALUES (ID, '"+vol+"')");
                inst.executeUpdate();
          }
        } catch (IOException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
        }
}
```

Annexe C

le programme d'evaluation

```
package memoire;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;
public class predictiondicolem
{
 public static void main(String[] args) throws InstantiationException,
 IllegalAccessException, ClassNotFoundException, SQLException {
 PreparedStatement inst = null;
 Connection connexion = null;
 PreparedStatement stmt = null;
 Class.forName("com.mysql.cj.jdbc.Driver").newInstance();
 connexion = DriverManager.getConnection
 ("jdbc:mysql://localhost/dico?useTimezone=true&serverTimezone=UTC",
 "root","");
 List<Integer> annotation = new ArrayList<Integer>( );
 List<Integer> predictiondico = new ArrayList<Integer>( );
```

```
 int vrai_positif=0;          int Faux_positif=0;
 int vrai_negatif=0;          int Faux_positif_neg=0;
 int vrai_neutre=0;           int Faux_negatif=0;
 int Faux_positif_net=0;      int Faux_neutre=0;
 int Faux_negatif_net=0;      int Faux_neutre_neg=0;
 int Faux_negatif_pos=0;      int Faux_neutre_pos=0;
inst = connexion.prepareStatement("select polarite from annotation");
 ResultSet rs = inst.executeQuery();
 while (rs.next()){
 annotation.add(rs.getInt("polarite")); }
 PreparedStatement inst2 = connexion.prepareStatement("select polarite
 from classificationdicolem");
 ResultSet rs2 = inst2.executeQuery();
 while (rs2.next()){
 predictiondico.add(rs2.getInt("polarite"));}
 for(int i =0; i<annotation.size(); i++){
 if ((annotation.get(i)==1)&&(predictiondico.get(i)==1)){
  vrai_positif = vrai_positif +1; }
 if ((annotation.get(i)==-1)&&(predictiondico.get(i)==-1)) {
  vrai_negatif = vrai_negatif +1; }
 if ((annotation.get(i)== 0) &&(predictiondico.get(i)==0)) {
  vrai_neutre = vrai_neutre +1; }
 if ((annotation.get(i)==1) &&(predictiondico.get(i)==-1)) {
  Faux_negatif_pos = Faux_negatif_pos +1; }
 if ((annotation.get(i)==1) && (predictiondico.get(i)==0)) {
  Faux_neutre_pos = Faux_neutre_pos +1; }
 if ((annotation.get(i)==-1) &&(predictiondico.get(i)==1)) {
  Faux_positif_neg = Faux_positif_neg +1; }
 if ((annotation.get(i)==-1) &&(predictiondico.get(i)==0)) {
  Faux_neutre_neg = Faux_neutre_neg +1; }
 if ((annotation.get(i)==0) &&(predictiondico.get(i)==1)) {
```

```
  Faux_positif_net = Faux_positif_net +1; }
 if ((annotation.get(i)==0) && (predictiondico.get(i)==-1)) {
  Faux_negatif_net = Faux_negatif_net +1; }
       }
System.out.println(" Resultats classification  dico lem");
System.out.println("Vrai positif :"+vrai_positif+ " , faux positif netr :
" + Faux_positif_net + " ,faux positif neg : " + Faux_positif_neg);
System.out.println("Vrai negatif :"+vrai_negatif+ " , faux negatif netr :
"+ Faux_negatif_net + " ,faux negatif pos : " + Faux_negatif_pos);
System.out.println("Vrai neutre :"+vrai_neutre+ " , faux neutre neg
:
"+ Faux_neutre_neg + " ,faux neutre pos : " + Faux_neutre_pos );
   }
}
```