جامعة سيدي محمد بن عبد الله بفاس
UNIVERSITÉ SIDI MOHAMED BEN ABDELLAH DE FES
كلية العلوم ظهر المهراز فاس
FACULTE DES SCIENCES DHAR EL MAHRAZ FES

N° d'Ordre : **48 /2021**

Formation Doctorale : STIC
Discipline : Informatique
Spécialité : Informatique
Laboratoire d'Informatique, Signaux, Automatique et Cognitivisme (LISAC).

THESE DE DOCTORAT

Présentée par

Mariem Bounabi

Etude des systèmes d'analyse et de catégorisation des données non structurées basée sur l'apprentissage artificiel : Applications Text Mining.

Soutenue le 27 / 11 / 2021 devant le jury composé de :

Pr . *Hamid TAIRI*	***Faculté des Sciences Dhar Mahraz (USMBA– Fès)***	Président
Pr. *Said OUATIK EL ALAOUI*	***Ecole Nationale des Sciences Appliquées de Kenitra***	Rapporteur
Pr. *Mohammed OUMSSIS*	***Ecole Supérieure de Technologie de Salé***	Rapporteur
Pr. *Chakib SOSSÉ ALAOUI*	***Centre Régional des Métiers, Education et des Formations de Fès***	Rapporteur
Pr. *Abdelghani LAKEHAL*	***Faculté Poly-Disciplinaire de Larache***	Examinateur
Pr. *Hassan SATORI*	***Faculté des Sciences Dhar Mahraz (USMBA– Fès)***	Examinateur
Pr. *Abdellatif EL ABDERRAHMANI*	***Faculté des Sciences Dhar Mahraz (USMBA– Fès)***	Examinateur
Pr. *Khalid Satori*	***Faculté des Sciences Dhar Mahraz (USMBA– Fès)***	Directeur de thèse
Pr. *Karim EL Moutaouakil*	***Faculté Polydisciplinaire de Taza (USMB – Fès)***	Co-directeur de thèse

2021/2022

Remerciement

La réalisation de cette thèse a été possible grâce à DIEU et à l'aide de nombreuses personnes qui ont offert leur temps, leur énergie, leur soutient et leurs connaissances. Merci à toutes ces personnes, sans vous ce travail n'aurait pu aboutir.

Je tiens tout d'abord à remercier mon directeur de thèse, Monsieur *Khalid SATORI*, professeur à la Faculté des Sciences Dhar El Mehraz Fès, et mon Co-directeur de thèse Monsieur *Karim EL MOUTAOUAKIL*, Professeur à la Faculté Poly-Disciplinaire de Taza, pour leur patience, leurs précieux conseils et leur encouragement continuel durant toute la période d'encadrement.

Je tiens à exprimer mes sincères remerciements à Monsieur *Hamid TAIRI*, Professeur à la Faculté des Sciences Dhar El Mehraz Fès, pour avoir accepté de présider le jury, ainsi que les rapporteurs, Monsieur Said *OUATIK EL ALAOUI*, Professeur à l'Ecole Nationale des Sciences Appliquées de Kenitra, Monsieur *Mohammed OUMSSIS*, Professeur à l'Ecole Supérieure de Technologie de Salé, et Monsieur *Mohammed Chakib SOSSÉ ALAOUI* Professeur au Centre Régional des Métiers, Education et des Formations de Fès pour avoir accepté de rapporter ce travail, de la qualité de leurs remarques et de leurs critiques constructives. J'exprime mes profonds remerciements à Monsieur *Abdelghani LAKEHAL*, Professeur à la Faculté Poly-Disciplinaire de Larache, Monsieur *Hassan SATORI* et Monsieur *Abdellatif EL ABDERRAHMANI* Professeurs à la Faculté des Sciences Dhar El Mehraz Fès pour avoir accepté de participer au jury en tant qu'examinateur.

Mes remerciements s'adressent finalement à ma famille pour leur patience et leur encouragement infaillible durant toutes les années de mes études, sans oublier mes fidèles amis.

Dédicace

Je dédie ce mémoire à :

Mes chers parents pour leur patience, leur amour, leur soutien et leurs encouragements.

La mémoire de ma sœur

Mon frère et ma petite sœur

Ma famille, mes grand parents, tante, oncles cousins et cousines.

Mes amis

Mes professeurs

Et tous qui ont contribué de près ou de loin à la réussite de cette Thèse.

Résumé

Cette thèse s'inscrit dans le cadre d'apprentissage artificiel avec des applications concrètes du domine Texte Mining. Nous avons ainsi proposé des nouvelles approches en relation avec les applications d'analyse des sentiments telles que les sentiments des clients de l'entreprise de commerce en ligne, de la classification automatique et de la catégorisation multiple des textes. Les approches que nous avons proposées se focalisent, particulièrement, sur l'amélioration des processus d'extraction des caractéristiques, de même que sur la vectorisation des textes et la gestion de grande dimensionnalité des descripteurs.

Notre première contribution détermine les mécanismes nécessaires pour réussir la classification automatique, avec comparaison entre différents systèmes de classification. L'étude comparative réalisée, a prouvé qu'un bon choix des méthodes de racinisation, de vectorisation, et d'hybridation des algorithmes d'apprentissage, influence fortement la performance et la fiabilité de ces systèmes.

La deuxième contribution porte sur la proposition d'une nouvelle approche de vectorisation probabiliste pour rendre la catégorisation neuronale des textes faisable et performante.

Dans la troisième contribution, nous avons proposé une amélioration remarquable des descripteurs flous, avec une automatisation des systèmes d'inférences flous, ce qui améliore la sélection des règles d'inférence. Cette proposition renforce la vectorisation floue des textes et améliore le rendement des classifieurs automatiques.

Une quatrième et dernière contribution, sera présentée dans ce manuscrit, utilise l'amélioration de la logique floue par la logique Neutrosophique afin de présenter un nouveau modèle avancé pour la rénovation de la vectorisation floue des données textuelles. Nous proposons ainsi, la mise en œuvre d'un nouveau descripteur vectoriel qui se base sur la philosophie Neutrosophique pour représenter les termes pertinents des bases des données textuelles de références, ce qui permet d'avoir des systèmes de catégorisation des textes robustes avec des performances captivantes.

L'ensemble des outils, des comparaisons et des résultats avec les nouvelles contributions suscitées, seront affichés et discutés au sein de ce mémoire.

Mots clés :

Text Mining, Apprentissage Artificiel, Extraction des caractéristiques, Vectorisation des Textes, Logique Floue, Logique Neutrosophique, Descripteurs Neutrosophique, Classification multiple, Analyse des sentiments, Hybridation des classifieurs.

Abstract

This thesis is part of the artificial learning framework for Text Mining applications. In general, we have proposed new approaches that enhance the performance of sentiment analysis applications such as amazon customers' commercial sentiments, automatic classification, and multiple categorizations of texts. Indeed, the suggested approaches focus on improving feature extraction processes, including text vectorization, and handling high dimensionality of descriptors.

The first contribution outlines the mechanisms required for a successful text organization, where we compared different text classification systems. Hence, the comparative study showed that a good choice of the stemming methods, vectorization, and the learning algorithms combination strongly influence the performance and the reliability of these systems.

The second contribution introduces a new text vectorization approach based on probabilistic reasoning. The proposal allows making an efficient neural texts categorization system.

The third intervention propose a significant improvement on fuzzy descriptors, where we suggest the automation of fuzzy inference systems improving the selection of inference rules. This proposal boosts the fuzzy texts vectorization and the performance of automatic classifiers.

The last contribution uses the neutrosophic logic to present a new advanced neutrosophic model. The proposed approach replaces the fuzzy term representation and practices the neutrosophic inference system to calculate relevant term weight. Therefore, using the new neutrosophic descriptor for text classification applications shows satisfying results.

In this thesis, the tools and the results of the discussed contributions will be displayed and discussed.

Key Words

Text Mining, Artificial Learning, Features Extraction, Text Vectorization, Fuzzy Logic, Neutrosophic Logic, Neutrosophic Descriptor, Multiple Classification, Sentiment Analysis, Hybrid Classifiers.

Sommaire

Remerciement……...……...2
Dédicace…….....3
Résumé……..4
Abstract…….5
Sommaire……..6
Liste des abréviations……..10
Liste des figures……...11
Liste des tableaux……13
Introduction Générale……15
Chapitre 1 : Concepts et architectures générales du Text Mining et de l'apprentissage artificiel…… 20
I. Introduction……20
II. Généralités sur le Text Mining…… …….21
1. Apparition du Text Mining ……..22
2. Processus des applications du Text Mining……..23
2.1 Algorithmes de nettoyage et prétraitement des données textuelles 25
2.2 Modèles de la représentation numérique des données 27
2.2.1 Modèles vectoriels fréquentiels 29
2.2.2 Modèles neuronaux vectoriels (Word Embedding) 30
2.3 Réduction de la dimensionnalité 32
2.4 Applications du Text Mining 33
2.4.1 Classification des données non structurées 34
2.4.2 Analyse des sentiments 36
III. Généralité sur l'apprentissage statistique……..……36
1. Algorithmes d'apprentissage supervisé pour la catégorisation des Textes……...38
1.1 Méthodes de classification Probabilistes 39
1.1.1 Réseaux Bayésiens 40
1.1.2 Naïve Bayes 42
1.1.3 Bernoulli naïve Bayes 42
1.1.4 Naïve Bayes Complémentaire 42
1.2 Méthodes de classification discriminantes 43
1.2.1 Régression Logistique Simple 44
1.2.2 Machine à vecteurs supports SVM 44
1.2.3 Réseaux de Neurones Artificiels (RNA) et le Perceptron Multi Couches (PMC) 45
1.2.4 Forêts d'arbres décisionnels 48
2. Classification hybride…….…....49
3. Évaluation des classifieurs…….……50
3.1 Matrice de confusion 50
3.2 Formules des mesures d'évaluation des classifieurs 51
3.3 Fonction d'efficacité du récepteur (Courbe ROC) 51

IV. Conclusion…….52
Chapitre 2 : Comparaison et approche probabiliste pour les systèmes de catégorisation multi-classes…….....55
I. Introduction……....55
II. Systèmes de la recherche et d'analyse de l'information textuelle basés sur l'apprentissage artificiel analyse et comparaison……56
1. Utilité des systèmes de classification…….....56
2. Comparaison des systèmes de catégorisation multi-classes des textes…….....58
2.1 Base des données 58
2.2 Prétraitement des données..... 59
2.3 Outils de la classification..... 59
2.4 Résultats de l'étude comparative et analyse 59
III. Impact de la vectorisation sur la précision de la catégorisation des textes…….62
1. Plongement Lexical (PL) analyse et comparaison……....63
1.1 Architecture du Plongement Lexical adoptée..... 63
1.2 Analyse des paramètres neuronaux du Doc2vec 64
2. Comparaison des systèmes de classification multi-classes basée sur PL……....65
2.1 Système de classification adopté 65
2.2 Résultats de l'étude comparative 66
2.3 Classification des données à étiquettes multiples Résultats et discussion..... 67
2.4 Classification des données de sentiment commercial d'Amazon..... 71
IV. Approche probabiliste pour la vectorisation et la catégorisation neuronale…..73
1. Préambule……...73
2. Méthodologie et algorithme de pondération probabiliste…….……...75
2.1 Approche probabiliste pour la catégorisation neuronale 75
1.2 Architecture du système de catégorisation basée sur la nouvelle vectorisation probabiliste..... 75
2.2.1 Prétraitement..... 76
2.2.2 Sélection des Attributs (SA) 76
2.2.3 Algorithme de pondération probabiliste proposée..... 77
2.2.4 Outils de catégorisation neuronal..... 79
2. Expérimentation et résultats des systèmes adoptés…….…….……81
V. Conclusion…….……...…86
Chapitre 3 : Contributions au moteur d'inférence et à la représentation vectorielle flous pour le renforcement des systèmes de classification…….……...88
I. Introduction…….……...88
II. Logique floue et systèmes d'inférences flous…….……..89
1. Logique Floue (FL)…….……...89
2. Systèmes d'Inférences Flous (SIFs)……..……....90
III. Automatisation des Règles d'Inférences (RI) floues…….…..……...90
1. Règles d'inférences classiques……..…….....90
2. Nouvelle approche pour l'automatisation des règles d'inférence floues…..….....91
2.1 Modèles d'associations..... 91

2.1.1 Algorithme apriori 92
2.1.2 Filtre d'association..... 92
2.2 Nouvelle approche de sélection automatique des Règles d'Inférences 92
IV. Applications de la logique floue et l'automatisation des règles d'inférence à la classification……..…94
1. Application de la sélection automatique des règles d'inférence à la classification floue des données iris……...94
2. Application de la sélection automatique des règles d'inférence à la vectorisation floue et son impact sur la classification automatique……..98
2.1 Classification des textes basé sur la pondération floue et les modèles ML 98
2.2 Pondération TF-IDF floue (FTF-IDF) 98
2.3 Impact de la TF-IDF floue sur la performance des classifieurs automatique 102
2.4 Application de l'automatisation des règles d'inférence au processus FTF-IDF..... 105
2.4.1 Option automatique des règles d'inférence pour la FTF-IDF 107
2.4.2 Impact de la nouvelle FTF-IDF sur la performance des classifieurs ML 107
V. Conclusion……..……111
Chapitre 4 : Nouveau modèle neutrosophique avancé pour la rénovation de la vectorisation floue des données textuelles……..……..113
I. Introduction……..…….....113
II. Généralité sur la logique Neutrosophique ……114
1. Logique Boolean, floue et neutrosophique…….…..……...114
2. Logique Neutrosophique (LN)……...……..115
2.1 Sous-ensembles neutrosophique 115
2.2 Opérateurs Neutrosophiques..... 117
III. Systèmes d'Inférence Neutrosophique (SIN)……....118
IV. Application du raisonnement neutrosophique pour la rénovation de la vectorisation floue des données textuelles……....119
1. Principales observations et motivations……..…119
2. Pondération Floue FTF-IDF et ses lacunes……120
3. Nouvelle pondération neutrosophique NTF-IDF……121
3.1 Architecture du SIN pour la déduction de la NTF-IDF 121
3.2 Corrélation et conception des fonctions d'appartenances neutrosophiques pour les variables NTF-IDF..... 123
3.3 Calcul du poids final NTF-IDF..... 126
4. Expérimentation et résultats……....126
4.1 Matériels 126
4.2 Bases des données..... 127
4.3 Paramètres de la classification..... 127
a) Prétraitement des données 127
b) Paramètres des classifieurs..... 128
c) Mesures de performance..... 128
4.4 Résultats et discussions..... 128
V. Conclusion ……..……..…131
Conclusion générale et Perspectives……...……133

Résumé des contributions 133
Perspectives..... 134
Bibliographies……..……...136
Liste des publications……..……...145

Liste des abréviations

AA : Apprentissage Artificiel
AUC : Air Under Courbe (ROC)
CFS : Sélection de caractéristiques basée sur la corrélation
CNB : Complément Naïve Bayes
FA : Forêts Aléatoire(s)
FNN : Feed-Forward Neural network
KDD : Knowledge Discovery from Data bases
ML : Machine Learning
LF : Logique Floue
LN : Logique Neutrosophique
ML : Machine Learning
NB : Naïve Bayes
PL : Plongement Lexical
PMC : Perceptron Multi Couches
RB : Réseau Bayésien(s)
RF : Radom Forest
RI : Règles d'Inférence(s)
RLS : Régression Logistique Simple
RNA : Réseaux de Neurones Artificiels
ROC : Receiver Operating Characteristic Curve
RV : Représentation Vectorielle
SA : Sélection des Attributs
SIF : Système d'Inférence Flou
SIN : Système d'Inférence Neutrosophique
SVM : Machine à Vecteurs Supports
TALN : Traitement Automatique du Langage Naturel
IG : Gain d'Information

Liste des figures

Figure 1: Processus KDD [34] 22
Figure 2: Processus du Text Mining 24
Figure 3: Architectures du modèle word2vec (a) CBOW, (b) Skip-Gram. 31
Figure 4: Diagramme de Venn de l'intersection du Text Mining et des six champs associés[7] 34
Figure 5: Architecture des systèmes classique de la classification des textes. 35
Figure 6: Cas d'utilisation de l'intelligence artificielle et de l'apprentissage automatique pour les entreprises du monde entier. 38
Figure 7: Types des méthodes de classification des textes 39
Figure 8: Modèle de graphe probabiliste acyclique. 40
Figure 9: Exemple de séparateur à marges pour une séparation binaire [107] 44
Figure 10: Modèle et fonctionnement d'un réseau de neurones [111] 47
Figure 11: Exemple d'un perceptron à une seule couche cachée [114]. 47
Figure 12: Exemple de la courbe ROC [124]. 52
Figure 13: Architecture des systèmes classique de la recherche de l'information. 57
Figure 14: Architecture du système de la recherche d'information adoptée. 58
Figure 15: Architecture des systèmes classique de la recherche de l'information adoptée 58
Figure 16 : Courbe ROC du classifieur Vote. 62
Figure 17: Architecture Doc2vec adoptée (PV-DM). 64
Figure 18: Architecture du système de catégorisation adoptée. 66
Figure 19: Précision du vote basée sur la variation des paramètres PV-DM (avec une taille du vecteurfixe = 100, et une modification du nombre d'époques). 69
Figure 20: Courbe d'évolution de la performance du vote, appliquant plusieurs paramètres neuronaux 72
Figure 21: Les processus des applications du Text Mining. 73
Figure 22: Système adopté pour la catégorisation multiple des textes en employant la nouvelle vectorisation probabiliste. 76
Figure 23: Architecture du système neuronal adopté pour la classification multiple des textes. 81
Figure 24: Les 5 classes de la base de test BBCSport. 82
Figure 25: Changement de la performance des systèmes de catégorisation en fonction du nombre de nœuds dans la couche cachée du MLP utilisant les deux représentations numérique (TF-IDF, et la nouvelle pondération probabiliste). 83
Figure 26: Processus des systèmes d'inférence flous. 90
Figure 27: Processus de la génération automatique des règles d'inférence. 92
Figure 28: Organigramme du système d'inférence floue pour déterminer la pondérée floue FTF-IDF. 99
Figure 29: Les degrés d'appartenances des entrées FTF-IDF : (a) TF, (b) IDF, (c) N et (d) la sortie Wd,t. 102
Figure 30 : Organigramme du système d'inférence floue pour déterminer la pondérée floue FTF-IDF optimale. 106
Figure 31: Résultats de la classification utilisant les classificateurs Bayésiens, une collection des méthodes de pondération et les données BBCSPORT. 110
Figure 32: Résultats de la classification utilisant les classificateurs Bayésiens, une collection des méthodes de pondération et les données BBCNEWS. 110
Figure 33: Processus du système d'inférence Neutrosophique (SIN). 118

Figure 34: Exemple de chevauchement entre des ensembles flous pour une entrée donnée. 121
Figure 35: Système d'inférence neutrosophique pour déduire les poids NTF-IDF. 122
Figure 36: Corrélation et critères de conception des fonctions d'appartenances pour (a) la fréquence d'attribut flou TF, (b) la composante de vérité neutrosophique et (c) la composante d'indétermination neutrosophique. 123
Figure 37: Fonctions d'appartenance ambiguës pour les valeurs de TF, IDF, N et le poids indéterminé. 124
Figure 38: Fonctions d'appartenance de vérité pour les valeurs de TF, IDF, N et le poids Wt. ..125
Figure 39: Courbes ROC pour la classification SVM en utilisant : NFT-IDF (a) et FTF-IDF (b) comme méthode de représentation des caractéristiques. 130
Figure 40: Amélioration, selon les classifieurs et la méthode de pondération comparée (FTF-IDF et NTF-IDF) pour différents ensembles de données. 131

Liste des tableaux

Tableau 1: Exemple de représentation vectorielle des mots. 28
Tableau 2: Modèle binaire de la Matrice de confusion. 50
Tableau 3: Comparaison des différents systèmes de classification en employant différents stemmers et classifieurs. 60
Tableau 4: Matrice de confusion du système de classification basé sur le stemmer Lovin et le classifieur Vote. 60
Tableau 5: Comparaison des différents systèmes de classification en employant BBCSport data, différents Stemmers et classifieurs. 61
Tableau 6: Matrice de confusion du système de classification basé sur le stemmer Snow Ball et le classifieur Vote. 61
Tableau 7: Résultats de classification utilisant la représentation PV-DM (avec numéro d'époque = 1 et taille du vecteur= 100), un ensemble de méthodes de classifieurs. 67
Tableau 8: Résultats de classification utilisant la représentation PV-DM (avec numéro d'époque = 5 et taille du vecteur= 100) et un ensemble des classifieurs. 67
Tableau 9: Résultats de classification utilisant la représentation PV-DM (avec numéro d'époque = 1 et taille du vecteur= 300), un ensemble de classifieurs. 68
Tableau 10: Résultats de classification utilisant la représentation PV-DM (avec numéro d'époque = 5 et taille du vecteur= 300), un ensemble de classifieurs. 68
Tableau 11 : Matrice de confusion du système basé sur la représentation PV-DM avec un bon choix du nombre des Epoch, en employant la base des données BBCSport. 69
Tableau 12: Matrice de confusion du système basé sur la représentation PV-DM avec un mauvais choix du nombre Epoch pour la base des données BBCSport 70
Tableau 13 : Résumé des résultats de la classification des données multi-étiquetées à l'aide du classifieur vote et en variant les paramètres de la PV-DM. 70
Tableau 14: Résultats de classification des données Amazon à l'aide du classifieur Vote, changeant les paramètres de la PV-DM 71
Tableau 15: Matrice de confusion avec un bon choix des paramètres du Doc2vec. 71
Tableau 16: Matrice de confusion avec un mauvais choix des paramètres du Doc2vec. 71
Tableau 17: Résultats pour différents algorithmes d'apprentissage de la structure du réseau à l'aide de la nouvelle méthode probabiliste. 82
Tableau 18: Résultats pour un nombre différent de nœuds masqués. 82
Tableau 19: Résultat de classification en employant les classifieurs bayésiens et le Perceptron multi couches 84
Tableau 20: Matrice de confusion de la classification bayésien, utilisant la représentation TF-IDF. 85
Tableau 21: Matrice de confusion de la classification bayésien, utilisant la nouvelle représentation probabiliste. 85
Tableau 22: Matrice de confusion de la classification neuronale, utilisant le classifieur PMC et la représentation TF-IDF. 85
Tableau 23: Matrice de confusion de la classification neuronale, utilisant le classifieur PMC et la nouvelle représentation probabiliste. 85
Tableau 24: Comparaison de l'approche de classification neuronale, employant la base des données BBCSport, avec autres systèmes de la littérature. 85
Tableau 25: Représentation nominale de l'ensemble de données d'iris. 93
Tableau 26: Description des règles sélectionnées par type et mesure de confiance. 96
Tableau 27: Transformation des règles implicites aux règles explicites. 97

Tableau 28: Matrice de confusion pour la classification floue des données iris, basée sur la sélectionné automatique des règles d'inférence floues. 97
Tableau 29: Performances du système de classification floue adopté, basées sur les modèles d'association.....97
Tableau 30: Les composantes du système d'inférence FTF-IDF.....101
Tableau 31 : Résultats du système de classification, des BBCSport data, basés sur la pondération FTF-IDF et CFS comme méthode de sélection d'attributs**.** 103
Table 32: Résultats du système de classification, des BBCSport data, basés sur la pondération FTF-IDF et Relief comme méthode de sélection d'attributs. 104
Tableau 33: Résultats du système de classification, des BBCSport data, basés sur la pondération FTF-IDF et IG comme méthode de sélection d'attributs.....104
Tableau 34: Résultats du système de classification, des BBCNews data, basés sur la pondération FTF-IDF et IG comme méthode de sélection d'attributs.....105
Tableau 35: Résultats du système de classification, des BBCNews data, basés sur la pondération FTF-IDF et Relief comme méthode de sélection d'attributs. 105
Tableau 36 : Les résultats de la classification bayésienne pour BBCSport Data, en utilisant la représentation FTF-IDF simple et améliorée. 108
Tableau 37: Les résultats de la classification bayésienne pour BBCNews Data, en utilisant la représentation FTF-IDF simple et améliorée. 109
Tableau 38: Généralités sur la logique Boolean, floue et neutrosophique.....115
Tableau 39: Définitions des sous-ensembles pour l'ensemble des logiques : floue, intuitionniste et neutrosophique. 116
Tableau 40: Les opérateurs des logiques neutrosophique et floue.....117
Tableau 41: Résultats de la classification en employant la FTF-IDF dans la phase de pondération.....129
Tableau 42: Résultats de la classification, en employant la NTF-IDF dans la phase de pondération.....129

Introduction Générale

Dans ce monde follement compétitif, les sociétés qui possèdent plus d'information et qui l'exploitent, au maximum, réussissent le mieux à travers des prédictions vraisemblables au future du marché. Actuellement, le Data Mining offre une boîte à outils, bien comprises et maîtrisés, aux data analystes pour explorer les entrepôts des données et d'en extraire des décisions puissantes capables d'arracher les sociétés en difficulté pour les placer au sommet. Et portant, le métier du data analyste devient de plus en plus difficile devant les entrepôts hétérogènes de très grandes tailles (Big Data) contenant, entre autres, des images, audio, vidéos, et des textes exprimés en langage naturel. Une telle hétérogénéité a donné naissance à plusieurs sous disciplines de Data Mining, ainsi en parle de l'Image Mining, Vidéos Mining, Text Mining …etc.

Les travaux de recherche rapportés dans cette thèse entrent dans le cadre du Text Mining, notamment l'extraction de l'information [1] [2] [3], la conception des systèmes de l'analyse et de la catégorisation des textes écrits en langage naturel [4] [5], où l'Apprentissage Artificiel (AA) est la base de la conception de ces systèmes.

La mise en place des processus d'extraction des connaissances [6] contenues dans les textes (tweets, tags, post, commentaires, news, etc…) est la première phase de tout système automatique sous-jacent au domaine du Text Mining [7]. Part autre, la recherche de l'information, l'analyse des sentiments, la classification des textes et les systèmes de recommandation, sont initiés par l'extraction de l'information [8] [9] [10]. À cet égard, le prétraitement qui consiste au nettoyage des entrées qui sont redondons ou inutiles, la représentation numérique des entrés, ainsi que la sélection des descripteurs de taille optimale [9] lisibles par les algorithmes du Machine Learning (ML), ont toujours suscité l'intérêt des chercheurs dans le domaine du traitement des données textes [11] [12][13].

Généralement, le prétraitement joue un rôle primordial dans les systèmes du traitement des textes, pour cela plusieurs approches ont été suggérés [14]. Parmi les méthodes qui influencent le processus du prétraitement des textes nous trouvons la racinisation [15] [16] et la tokenisation des entrées [17], ces techniques évitent la redondance des termes constituant les indexe des textes d'entrer, tout en unifiant la forme des termes, et en réduisant l'occupation de la mémoire [15], permettant ainsi d'obtenir une étape préliminaire pour une bonne représentation des documents.

Cependant, différentes formes de représentation numériques ont été proposés dans la littérature, à savoir : la représentation binaire, représentation probabiliste et la représentation vectorielle [18] [19] [20]. La représentation binaire été proactive, connue par sa simplicité à la mise en œuvre. Malgré cela, elle a été exclue à la suite de sa déficience et son imprécision à représenter l'information. Le modèle probabiliste est apparu pour unifier les représentations des documents et les concepts, ce qui permet une correspondance approximative entre les documents et la demande. Mais ce modèle utilise un calcul de probabilité conditionnelle très complexe [19]. La vectorisation des textes (ou représentation vectorielle) est apparue, par la suite, pour corriger les lacunes des deux représentations précédentes, où un ensemble de modèles a été proposé dans la littérature [21] [22] [23]. En effet, la représentation fréquentiel

Term Frequency and Inverse Term Frequency (TF-IDF) est un des modèles vectoriels qui a connus un grand succès dans le domaine du traitement des textes [12]. Généralement, le modèle probabiliste est plus efficace que le modèle booléen mais moins efficace que le modèle vectoriel, c'est pourquoi nous choisissons le mode vectoriel dans nos systèmes de classification et d'analyse des documents. En revanche, la vectorisation des textes produit des descripteurs numériques de grande taille, où la sélection des caractéristiques pertinentes est devenue une nécessité [24]. Par conséquent, les méthodes de sélection des Attributs (SA) [24] ont connu une grande utilisation dans les systèmes relatifs au traitement des documents et pourtant la sélection peut provoquer une perte significative de l'information [25]. Relativement au problème de classification automatique des textes, les méthodes da la SA influence négativement la performance et l'interopérabilité des systèmes, comme S.Vora a indiqué dans son manuscrite [26]. Ainsi, en raison de la diversité des algorithmes d'apprentissage automatique, l'assurance de la fiabilité et la sécurité des systèmes de catégorisation est devenus une exigence primordiale lors du choix des outils de la classification.

Cette thèse présente nos contributions variées au problème de la classification binaire et multiple des textes, où le but principal est de concevoir des descripteurs robustes, visant à extraire des caractéristiques discriminantes de chaque document. Ultérieurement, les descripteurs suggérés seront intégrés dans des systèmes fiables et sécurisé, qui ont comme but la classification des textes et l'analyse des sentiments.

Pour atteindre nos objectives, notre première contribution porte sur une étude comparative entre différents systèmes de classification. Pendant cette étude, nous varions les méthodes utilisées dans les différents mécanismes constituant la tâche de catégorisation et la comparaison proposée doit montrer l'impact du bon choix de la méthode de racinisation et la vectorisation des textes, sur la performance des systèmes de catégorisation. En outre, nous comparons avec une variété de classifieurs artificiels individuels et les techniques de la classification hybride [27], pour avoir les meilleures performances.

Notre deuxième contribution porte sur l'utilisation du plongement de mots vectoriel (Word Embedding) pour la classification automatique. L'intérêt de cette contribution est de sélectionner les meilleures valeurs des paramètres neuronaux de la représentation doc2vec [28]. La comparaison présentée, montre qu'un bon choix des valeurs influence la pertinence du descripteur ainsi que la précision de la classification, par conséquent les résultats des contributions nous ont poussés vers l'amélioration du processus de la représentation et de la sélection des caractéristiques pour renforcer le processus de classification. Parmi les classifieurs qui exige certains critères sur les descripteurs est le perceptron multi couche (PMC). Pendant notre analyse nous avons remarqué l'incapacité des approches populaires, à produire des descripteurs compatibles à la structure des entrées du PMC, et pour surmonter ce problème nous proposons une nouvelle représentation vectorielle probabiliste des documents, où l'approche proposée résout les deux dimensions du problème qui sont :

- La haute dimensionnalité des descripteurs, où la pondération classique TF-IDF produit des descripteurs de grandes tailles ;
- La perte énorme de l'information provoqué par les méthodes de sélection des attributs.

En comparant notre nouvelle approche avec l'approche TF-IDF, l'étude expérimentale nous a révélé l'efficacité de notre proposition et ceci en faisant une comparaison avec les performances existantes.

Nous avons par la suite essayé de combiner la logique floue à la pondération TF-IDF, cette technique nous a révélé la découverte TF-IDF floue (FTF-IDF). Généralement, la FTF-IDF est une phase des systèmes d'extraction des connaissance flous [29], qui se base sur les systèmes inférence flous (SIF), pour composer la matrice descripteur de la base des données analysé. Notre contribution, dans ce stade, porte sur la modification du SIF par l'automatisation du processus de généralisation des Règles d'Inférence (RI), en bénéficiant des modèles d'association [30]. Cette automatisation permet d'avoir des systèmes experts performant avec une complexité optimale. La nouvelle approche est exploitée pour améliorer la qualité des descripteurs flous qui seront intégrer dans des systèmes de classification automatiques. Généralement, comme mentionner dans la littérature, la pondération TF-IDF et ses versions avancées, comme la FTF-IDF, néglige le degré d'ambiguïté des termes, c.à.d., être pertinent et représentatif pour un document donné.

Cependant, durant l'implémentation de la pondération FTF-IDF, nous avons remarqué l'échec des ensembles floues à distinguer, dans un certain niveau, entre l'appartenance des fréquences des mots à un sous ensemble précis. Cela provoque une ambiguïté sur la pertinence du terme et son degré d'informativité à représenter un document. Avec l'apparition de la Logique Neutrosophique (LN) qui étudie un ensemble de phénomènes (l'ambigüité, l'incertitude et la neutralité), dont la LF est incapable de le faire. La LN est devenue l'intérêt des développeurs des systèmes d'experts, et notre solution adoptés afin de surmonter les lacunes de la pondération floue (FTF-IDF). De ce fait, notre nouvelle proposition de pondération TF-IDF neutrosophique (NTF-IDF) relie la pondération fréquentielle avec le degré d'ambiguïté du terme, projetant les entrées FTF-IDF dans l'espace neutrosophique 3D. Cela autorise l'intégration du degré d'ambiguïté et marque une précision considérable sur les poids des termes. Pour prouver la performance du nouveau descripteur neutrosophique (DN), où ce dernier a été proposé comme entré à une collection des classifieurs, sensibles aux entrées fournies, afin d'analyser et de catégoriser des grands corpus de référence. Une telle approche a prouvé une excellente performance et robustesse pour les systèmes de catégorisation des textes.

Or, dans un contexte général, les moyens de l'apprentissage artificiel utilisés, ainsi que l'ensemble des contributions réalisé seront détaillés et discutés selon l'aperçu proposé dessous.

Aperçu de la thèse

Cette thèse contient un arrière-plan et un état de l'art sur le Text Mining et l'apprentissage artificiel, ainsi qu'un ensemble d'approches de l'extraction des connaissances à partir des données non structurées, pour l'objectif de la catégorisation automatique, en outre notre thèse constituée de quatre chapitres représentés comme suit :

Chapitre 1 : Nous présentons les concepts et l'architecture générale des systèmes du Text Mining (la fouille de textes) et les principaux processus qui les constituent. Ce chapitre donne un bref historique et survole les méthodes et les techniques les plus utilisées dans les différentes étapes constituantes d'un système ou d'une application du Text Mining (prétraitements,

représentation des termes, extraction des caractéristiques, et les tâches du Text Mining). Nous avons essayé de survoler quelques modèles d'apprentissage artificiel utilisés durant le développement des applications du Text Mining. En effet, nous présentons un ensemble des outils et des algorithmes avancés, efficacement utilisés pour la sélection des caractéristiques et l'apprentissage supervisé, afin de réaliser des systèmes d'analyse et de catégorisation des données textuelles non structurés. Également, les mesures des performances des systèmes étudiés seront discutées dans ce chapitre.

Chapitre 2 : Nous avons monté l'impact de chaque processus de l'architecture des systèmes de catégorisation des textes, sur ses qualité et performance. Focalisant sur l'impact important de la Représentation Vectorielle (RV), relevé par l'étude comparative citée dans la première partie de ce chapitre. Ainsi, nous présentons une de nos contributions apportées au domaine du traitement automatique du langage naturel, où une nouvelle version de RV probabiliste a été réalisé. La méthode introduite permet de créer des descripteurs pertinents, et de présenter un système de catégorisation neuronale robuste.

Chapitre 3 : Nous avons consacré ce chapitre à la Logique Floue (LF), où un aperçu général de cette logique ainsi que les Systèmes d'Inférences Flous (SIF) seront présentés. Ensuite, le chapitre élabore le problème de sélection des règles d'inférences, comme processus important des SIFs et la solution proposée pour l'automatisation de ce processus par l'utilisation des modèles d'association. Enfin, il survole l'application de la LF dans le Text Mining, et la génération des descripteurs vectoriels flous à base des modifications apportés sur le SIF.

Chapitre 4 : Nous avons présenté notre contribution pour la production d'un nouveau descripteur pertinent, qui s'appuie sur la logique neutrosophique pour l'achèvement du modèle vectoriel Flou. Notre nouvelle proposition est l'unique version Neutrosophic de la RV, qui permet de corriger les lacunes de la représentation flous des données textuelles. Le chapitre présente, aussi, des résultats compétitifs des systèmes d'analyse et de catégorisation multi classes, en employant des corpus de références d'analyse des sentiments et autres.

Nous clôturons notre rapport avec une conclusion générale, où nous résumons les travaux réalisés et cités dans les quatre chapitres précédents, ainsi que les directions de la recherche pour les futurs travaux.

Chapitre 1 : Concepts et architectures générales du Text Mining et de l'Apprentissage Artificiel.

Chapitre 1 : Concepts et architectures générales du Text Mining et de l'apprentissage artificiel

I. Introduction

La capacité à comprendre et à extraire des connaissances d'un texte est une exigence clé, dans l'objectif des chercheurs en intelligence artificielle, pour créer des machines qui peuvent simuler les pensées des cerveaux humains les plus complexe dans l'univers, qui est le cerveau humain. Pour cette raison plusieurs tentatives ont été expérimenté, et bientôt, le développement s'est déplacé vers l'extraction de données textuelles en utilisant des techniques avancées de ce qu'on appelle le Traitement Automatique du Langage Naturel (TALN) [31].

La Fouille de Textes (ou le Texte Mining) est un des champs de recherches du TALN qui est introduit pour l'extraction des connaissances à partir des corpus contenant des textes exprimés en langage naturel, et qui sont produits par des humains pour des humains [7]. Le Texte Mining a été lie aux autres concepts, comme le Web Mining, apparut à la fin de l'an 1990 [32], où lorsque cette discipline s'intéresse aux traitement et analyse des données présentes sur les sites web, spécialement les données exprimées en langage naturel, en se basant sur les applications et les techniques du Data Mining, nous parlons alors du Texte Mining [33].

L'enjeu principal de ce domaine est l'analyse, la manipulation rapide et efficace de grandes quantités de textes valables sur le web, ce qui a nécessité l'inclusion :

- des algorithmes simplifiés des théories linguistiques dans des systèmes informatiques d'apprentissage et de statistiques ;
- des techniques de compréhension du langage naturel [33].

Par conséquent les disciplines impliquées sont donc la linguistique calculatoire [11], l'ingénierie des langues, l'apprentissage artificiel, les statistiques et l'informatique [34]. Nous trouvons notamment comme domaines d'application du Texte Mining : la recherche d'informations (RI), la classification et la catégorisation des documents, l'analyse de sentiment, la recommandation automatique des documents, etc. Le dernier élément du développement des systèmes modernes du Text Mining a été apporté par la technologie d'Apprentissage Artificiel (AA), qui a commencé à être appliqué aux études de catégorisation de texte au début des années 1990 [7]. Les processus du Text Mining qui se basent principalement sur l'application des algorithmes artificiels, sont : le processus de sélection des caractéristiques et la phase de catégorisation des textes.

En effet, après l'extraction des caractéristiques les plus représentatives, pour chaque document du corpus, une phase de classification est requise pour décider de la classe convenable parmi l'ensemble des classes prédéfinie. Par conséquence, la tâche de catégorisation ne peut pas être réaliser et les algorithmes d'AA ne puissent pas être appliqués, avec succès à des documents texte, si et seulement si l'ensemble de caractéristiques d'entrée est fourni, et tous les documents de la base sont organisé selon des classes prédéfinis.

Ensuite, l'ensemble d'apprentissage classifié peut être soumis à l'algorithme artificiel. Une fois le modèle entraîné, les documents non classés peuvent être soumis au modèle pour

classification, en utilisant les modèles des catégories apprises lors de l'opération d'apprentissage [35].

Les algorithmes de l'AA les plus recommandés pour la catégorisation de texte sont les réseaux de neurones, les arbres de décision (ou règles de décision), la régression logistique (RLS), et la Machine à Vecteurs Supports (SVM). L'ensemble de ces algorithmes est employé durant l'implémentation des systèmes d'analyse et de catégorisation des textes que nous avons proposés dans cette thèse, pour cela, dans ce chapitre, nous discuterons les différents types des algorithmes de l'AA, les algorithmes de la classification supervisée, la classification hybride, ainsi que les moyens et les méthodes d'évaluation de ces classifieurs, après avoir discuté profondément la discipline Text Mining, ces principaux processus. L'objectif général est de choisir les meilleurs composantes et méthodes pour la conception d'un système d'étude et d'analyse des textes performant et compétitif.

II. Généralités sur le Text Mining

Le Text Mining est définie officiellement par le processus d'extraction des connaissances inconnues, valides et potentiellement exploitables dans les documents textuels, à travers la mise en œuvre des techniques statistiques ou de Machine Learning (ML).

Le terme Text Mining est lié, spécifiquement, avec autres applications comme le Résumé Automatique [39] et l'Extraction d'Information [3]. Les données textuelles traitées dans le cadre du Texte Mining peuvent être de type document, qui est considéré aussi bien qu'individu statistique, ou une collection de documents (connu par le mot corpus ou Base d'apprentissage). Généralement nous pouvons distinguer entre trois sortes de documents [37] qui sont :

- *Document structuré* : les données usuellement exploitées en Text Mining (TM), dont les algorithmes TM sont ajustés pour les traiter, est organisé sous forme de tableaux, où la ligne représente l'individu statistique, et la colonne contient l'attribut.
- *Document semi-structuré :* sont des données qui n'ont pas été organisées en référentiel spécialisé, comme c'est le cas dans une base de données, mais qui comportent néanmoins des informations associées, des métadonnées par exemple, qui les rendent plus faciles à traiter que des données non-structurées.
- *Document non structuré* : les données en Text Mining se présentent sous forme de textes bruts. Les algorithmes de data Mining ne savent pas les appréhender nativement, d'où la préparation de ce genre de données est fondamentale, car les techniques statistiques usuelles ne peuvent pas traiter des données non structurées.

En effet, les méthodes du texte Mining révèlent la globalité de leur puissance pour faire face au Big Data, quel que soit le type des données employé, à la multiplication des contenus en ligne et à leur souplesse d'accès. Dans ce qui suit, en s'intéresse au type non structuré des données, vue leurs complexités et sa large utilisation dans les plateformes actuelles.

1. Apparition du Text Mining

Le développement du Text Mining a été initié par la nécessité de cataloguer le texte des documents comme le cas de catégorisation des livres dans une bibliothèque. Mais bientôt, le développement s'est déplacé vers l'extraction de données textuelles en utilisant les techniques du TLAN. Officiellement, le Text Mining est apparue dans les années 90 comme partie du Data Mining (la fouille des données). En 1991, Piatetsky-Shapiro introduit comme titre de son ouvrage le terme de Knowledge Discovery from Data bases (KDD) [38], qui signifie l'Extraction de Connaissances à partir de bases de Données (ECBD). Vers 1995 l'usage des termes KDD et Data Mining se précise, où le KDD désigne le processus intégral pour découvrir les structures des connaissances qui permet de passer des données brutes à des connaissances.

Lorsqu'une des branches du KDD concerne plus particulièrement le texte, il s'agit alors de l'Extraction des Connaissances à partir des Textes (ECT) ou le Text Mining. Alors, tant que le Text Mining est un sous-champ de Data Mining, alors il suit automatiquement le processus KDD illustré dans la figure 1.

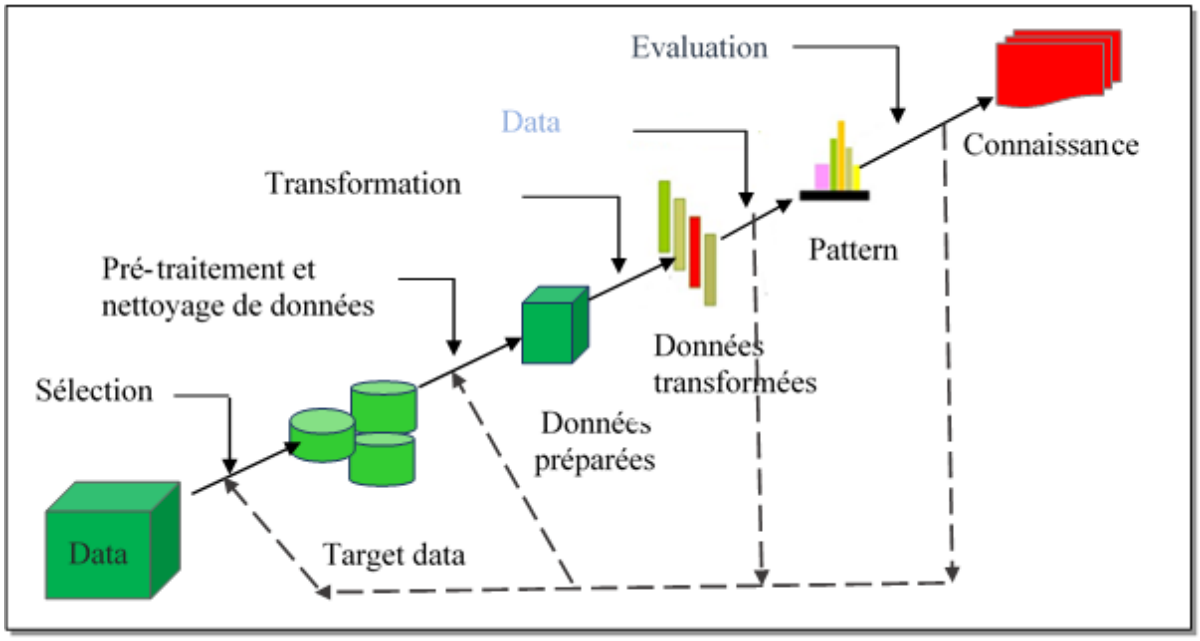

Figure 1: Processus KDD [34]

En effet le Text Mining est introduit, par Feldman and Degan en 1995 sous le terme Knowledge Discovery in Textual Databases (KDT) [41], ou Text Data Mining par Marti A. Hearst en 1999 [42], et traduit en français par [43] en Extraction des Connaissances à partir de Textes (ECT).

Depuis les années 2000, et l'arrivée du web 2.0 en 2005, la masse des données présente dans le web a connu un développement important. De nouvelles applications sont également apparues, comme celles qui reposent sur les opinions des internautes qui s'y expriment spontanément dans des différentes sites et plateformes. Le traitement automatique de ce type des données est appelé l'analyse des sentiments, ou comme il était nommé dans l'article de

Dave en 2003 par l'Opinion Mining (OP). Le sous-domaine du Texte Mining, l'OP, essaye de définir les opinions, sentiments et attitudes présentent dans un texte, afin d'améliorer la qualité des applications du e-service, qui utilisent le type non structuré des données textuelles.

Récemment la quantité des données non structurées disponibles est devenue incontrôlable et le passage vers le web 3.0 ou web sémantique est devenu nécessaire depuis 2010, pour faire face à l'explosion des données et améliorer le rendement des différents systèmes comme les systèmes e-commerce. Notamment en 2012, Gary Miner et son équipe [7] ont introduit l'emploi du Texte Mining et l'analyse statistique appliqués aux données textuelles (non structurées) et ils ont proposé aussi certaines procédures, pour créer des systèmes d'analyse des textes robustes.

Jusqu'à maintenant le Texte Mining a proposé des solutions qui revêtent plusieurs usages et services à base d'un ensemble de processus, comme :

- Filtrage du spam provenant des comptes [44].
- Produire une liste de documents (par exemple, les rapports d'erreurs) qui sont les plus similaires à un document d'intérêt. Obtenir un résumé rapide, mais représentatif, des thèmes d'une collection de documents [45].
- Évaluer le sentiment des clients sur les nouveaux produits [46] [47] (Twitter, groupes de discussion, plaintes, etc.).
- Automatiser la lecture de gros volumes de texte et déterminer les auteurs [48].
- Analyse des tendances [49].
- Détection de fraude dans les états financiers en se basant sur la fouille du texte [50].

Le plus souvent, le Text Mining respecte deux étapes principales, selon le domaine d'application, qui sont l'analyse des corpus de textes et l'interprétation de l'analyse [51]. L'étape suivante discute certaines notions du Text Mining ainsi que leurs processus principaux.

2. Processus des applications du Text Mining

Le processus global de recherche et d'interprétation des modèles à partir des données textuelles non-structurée, implique l'application répétée des étapes illustrées dans la figure 2.

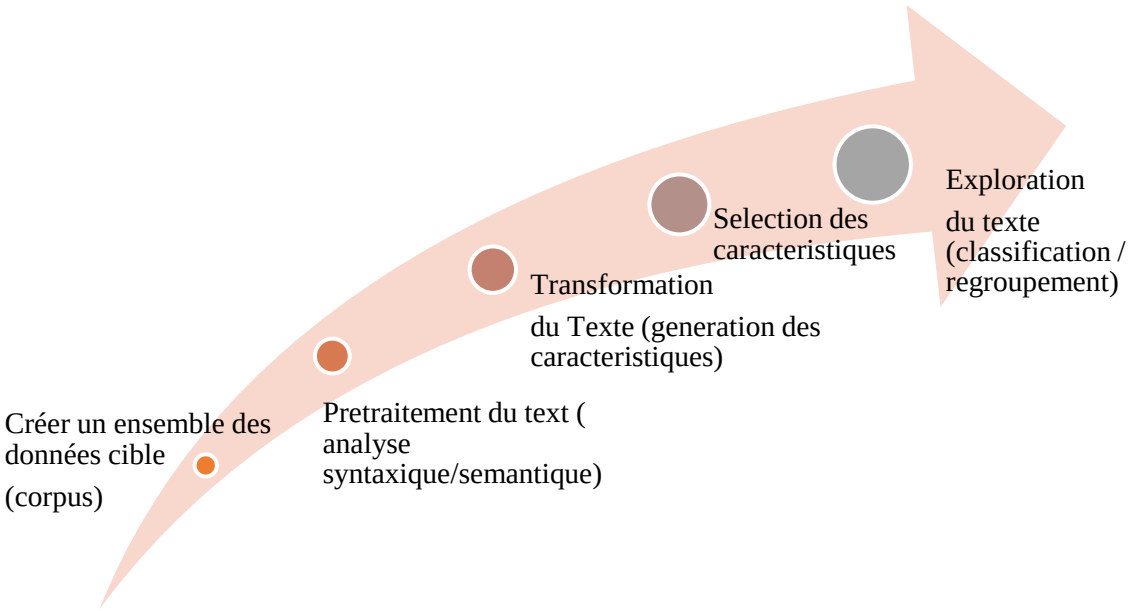

Figure 2: Processus du Text Mining.

La description détaillée de chaque étape figurée dans la figure au-dessus est la suivante :

- Créer un ensemble de données cible : ce qui signifier la sélection d'un ensemble de données, ou se concentrer sur un sous-ensemble de variables, ou d'échantillons de données, sur lesquels la découverte doit être effectuée.
- Nettoyage et prétraitement des données.
- Suppression du bruit ou des aberrations.
- Collecte des informations nécessaires pour modéliser ou prendre en compte le bruit.
- Stratégies de traitement des champs de données manquants.
- Réduction et projection des données.
- Trouver des caractéristiques utiles pour représenter les données en fonction de l'objectif de la tâche.
- Utilisation de méthodes de réduction de la dimensionnalité ou de transformation pour réduire le nombre effectif de variables considérées ou pour trouver des représentations invariantes pour les données.
- Choisir la tâche d'exploration des données.
- Décider si l'objectif du processus KDD est la classification, la régression, le regroupement, etc.
- Choisir les algorithmes d'exploration des données.
- Choisir les méthodes à utiliser pour rechercher des modèles dans les données.
- Décider quels modèles et paramètres peuvent être appropriés.
- Faire correspondre une méthode particulière d'exploration des données avec les critères généraux du processus de la BDOC.
- Exploration des données.

- Recherche de modèles d'intérêt dans une forme de représentation particulière ou un ensemble de représentations telles que les règles de classification ou les arbres, la régression, le regroupement, etc.
- Interprétation des modèles extraits.
- Consolidation des connaissances découvertes

Or, nous pouvons résumer ou regrouper ces processus en 4 étapes primordiales, après avoir énoncé le domaine de l'application, et les listés comme suivant :

- Nettoyage et prétraitement des données textuelles.
- Trouver des caractéristiques utiles pour la représentation numérique des données.
- Réduction de la dimensionnalité
- La tâche d'exploration des données

2.1 Algorithmes de nettoyage et prétraitement des données textuelles

Le prétraitement des données textes est élaboré afin de préparer l'information pertinente, et éliminer les éléments bruités, qui n'ont aucune importance, comme les mots vident, en utilisant une liste prédéfinie des termes vides. Par la suite il était essentiel de sauvegarder moins de termes significatifs non redondants pour minimiser le stockage dans la mémoire. Par conséquent, un ensemble d'algorithmes (dite de nettoyages) sont apparues pour résoudre le problème, comme les méthodes de lemmatisation [16] ou de la racinisation (Stemming) [52] [15].

Chacun de ces processus de nettoyage présente ses avantages et ses inconvénients, où le point fort de la racinisation est sa simplicité de mise en place, cependant la lemmatisation demande un dictionnaire complexe liant les différentes formes d'un mot, et elle est plus coûteuse en temps [53]. La racinisation est moins complexe, où en utilisant une série prédéfinie de préfixes et suffixes qui sont tronqués sur les mots du corpus. Plusieurs algorithmes de racinisation ont été présent dans la littérature [54] comme :

- ***Lovins Stemmer*** son algorithme affiché au-dessous a été proposé par Lovins [52],[15] Pour l'extraction des termes, Lovins utilise un tableau de 294 terminaisons, 29 conditions et 35 règles de transformation disposées selon le principe de la correspondance la plus longue, qui permet d'éliminer le suffixe le plus long d'un mot. Pour convertir ces tiges en mots valides, le mot est recodé à l'aide de différentes tables qui procèdent à divers ajustements. En tant qu'algorithme à passage unique, un maximum d'un suffixe est supprimé d'un mot. Parmi les principaux inconvénients de l'approche de Lovins, nous pouvons citer le fait qu'elle est longue et coûteux en données.

Algorithme : LOVINS

-Entrées : fichier contenant l'ensemble des mots
-Sorties : fichier contenant l'ensemble des mots raciné

Début
Tant que (non fin fichier) faire
/ Procédure de racinisation */*
'Déterminer l'emplacement du mot dans la liste des terminaisons'
'Chercher une correspondance entre le suffixe du mot et l'un des terminaisons de la liste'
si (terminaison trouvée) alors
Appliquer la règle
Fin si
/ Procédure de recodage */*
'Enlever doublement si existe
si (existe transformation) alors
Recoder stem selon règle
Fin si
Retourner (stem)
Fin tanque
Fin LOVINS

- ***Lovins Stemmer itératif :*** est une version itérative de l'algorithme de Lovins stemmer. En fait, si le mot comporte plus de deux caractères, cet algorithme exécute la racinisation du terme jusqu'à ce qu'il ne change plus [52] .
- ***Snow Ball :*** (ou Porter stemmer) est l'algorithme le plus courant pour la racinisation anglaise. L'algorithme original ne comporte que cinq étapes comme il est indiqué dans l'algorithme au-dessous. À chaque étape, les règles et les conditions sont appliquées. Une fois que la règle est correctement acceptée et selon la condition, les suffixes sont supprimés. l'étape suivante est réalisée [55] et qui se base sur l'enchainement Porter suivant :

Etape 1 : <suffix> → <new suffix>
Etape 2 : < Condition> <suffix> → <new suffix>

Algorithme : Snow-Ball

-Entrées : fichier contenant l'ensemble des mots du corpus
-Sorties : fichier contenant l'ensemble des mots raciné
Début
Tant que (non fin fichier) faire
/ Étape 1 */*
‘ Enlever et recoder forme plurielle
‘ Enlever « ed » et « ing » du verbe
/ étape 2 */*
si (existe voyelle dans **la racine**) alors
Transformer y en i
Fin si
/ étape 3 */*
‘Indexer la lettre avant dernière
‘ Enlever le doublement s'il existe
/ étape 4 */*
‘Indexer la lettre finale de la racine
‘Enlever la terminaison indiquée si possible
/ étape 5 */*
si (racine a la forme <c>vcvc<v>) alors
Enlever la terminaison
Fin si
si (plus qu'une séquence de consonne dans la racine) alors
Enlever terminaison
Fin si
Retourner (racine)
Fin tant que
Fin Snow-Ball

Ainsi, dans le cadre introduction au Text-Mining, nous nous contenterons d'aborder la technique la plus facile d'accès, celle de racinisation, ou *Stemming*. Ces méthodes sont, également, utilisées avec un certain nombre de classifieurs pour améliorer leurs performances et produire les meilleurs résultats de classification [15].

2.2 Modèles de la représentation numérique des données

La représentation numérique des données textes [56] est une étape importante pour la création des caractéristiques d'entrés aux modèles employés pour la réalisation d'une tâche du Text Mining. Cependant, plusieurs méthodes de représentation de l'information dans le domaine du

Text Mining ont été proposées dans la littérature, dont nous pouvons les classer en trois grandes catégories :

a. ***Un modèle booléen*** est le premier modèle de la représentation mathématique du contenu d'un document qui se base sur la théorie des ensembles et l'algèbre de Bool; la représentation binaire d'un terme est facile à mettre en œuvre mais elle n'est pas très informative [18].
b. ***Un modèle probabiliste*** repose sur la théorie des probabilités, où l'idée de base est de sélectionner des documents qui ont à la fois une forte probabilité d'être pertinents et une faible probabilité de ne pas être pertinents par rapport à la demande de l'utilisateur. Ce modèle unifie les représentations des documents et des concepts et permet une correspondance approximative entre les documents et la demande (requête), mais il utilise un calcul de probabilité conditionnelle complexe [19].
c. ***Un modèle vectoriel*** (où un modèle d'espace vectoriel MSV [57]) est un modèle algébrique introduit par Salaton [20] pour la modélisation des systèmes de recherche d'information (IR) spécialement pour représenter des documents et des requêtes comme des vecteurs de poids. La création du modèle MSV suit comme étapes essentielles la représentation des documents texte en vecteurs de mots, et la conversion en un format numérique afin d'appliquer ultérieurement toutes les tâches d'exploration de données. Par exemple, nous fournissons d'abord un exemple simple dans le tableau 1 pour illustrer comment créer le vecteur d'index du fichier de données, le vecteur de requête à partir de deux documents D1 et D2.

Tableau 1: Exemple de représentation vectorielle des mots.

Documents	Vecteurs
D1 = (w_1, w_3, w_4)	I1 = (1, 0, 1, 1)
D2 = (w_1, w_2, w_3)	I_2 = (1, 1, 1, 0)
Q = Requête = (w_1, w_2)	Q = (1, 1, 0, 0)

Pour faciliter la compréhension, nous supposons qu'il existe deux documents D_1 (contenant trois mots-clés, w_1, w_3 et w_4) et D_2 (contenant trois mots-clés, w_1, w_2 et w_3), qui sont notés par : $D_1 = (w_1, w_3, w_4)$ et $D_2 = (w_1, w_2, w_3)$.

Pour créer des vecteurs binaires, par exemple, pour D_1 et D_2, nous générons d'abord le dictionnaire de mots-clés dimensionnels constitué de n = 4 mots-clés ici sont dict = (w_1, w_2, w_3, w_4). L'espace vectoriel I de chaque document D peut être créé selon l'ordre du dictionnaire dict en respectant les règles suivantes : $\{\text{si } w_i \in \text{dict} \rightarrow w_i = 1 \text{ sinon } w_i = 0 \mid 1 \leq i \leq 5\}$. Ainsi, les vecteurs de D_1 et D_2 peuvent être notés respectivement I_1 = 1, 0, 1, 1 et I_2 = 1, 1, 1, 0. Ensuite, pour la requête de l'utilisateur, nous adoptons la même technique pour la transformer en vecteur qui dénoté comme Q = 1, 1, 0, 0. Enfin, nous calculons le produit interne entre le vecteur de requête et chacun des vecteurs D_1 et D_2. Dans cet exemple, $I_1Q = 1$ et $I_2Q = 2$, ce qui indique que D_2 est plus pertinent que D_1. Plus loin, le vecteur binaire s'est transformé en vecteurs des

poids par l'adaptation des techniques de calcule des poids, pour chaque terme du document, comme nous verrons dans les sections suivantes.

Il convient de noter que le modèle probabiliste est plus efficace que le modèle booléen mais moins efficace que le modèle vectoriel, c'est pourquoi nous choisissons le modèle de représentation vectorielle dans nos systèmes.

2.2.1 Modèles vectoriels fréquentiels

Comme mentionné dessus, un modèle vectoriel est une méthode algébrique utilisée pour représenter un document dans un espace vectoriel multidimensionnel. Or, les coordonnées d'un document vectoriel, représentent les poids des termes correspondants [58]. En effet, La représentation vectorielle fréquentielle est basée sur la loi de Zipf [59] et la conjecture de Luhn [60]. Dans la loi de Zipf, les termes sont représentés par leurs fréquences d'occurrence et sont classés inversement à leur rang.

La conjecture de Luhn sert à la recherche d'une qualité des termes de sens, en utilisant leur fréquence ; elle considère trois types de descripteurs : les descripteurs non pertinents (faible rang et grande fréquence), les descripteurs moins pertinents et les descripteurs pertinents (rangs intermédiaires). Cette conjecture est utilisée pour réduire la taille des termes du document. De plus, la conjecture de Luhn utilise deux seuils fréquence (seuil maximum et seuil minimum), qui sont définis pour supprimer les termes dont le contenu informatif est considéré faible. Seuls les mots situés entre ces deux seuils représentent les documents pertinents ;

Parmi les méthodes fréquentiel les plus célèbre, qui se base sur les deux principes Zipf et Luhn, nous trouvons la formule TF-IDF (qui signifié : fréquence du terme-fréquence inverse du terme) [51] [12]:

La pondération des termes TF-IDF associe des poids aux termes importants dans le document, qui ont un grand degré d'informativité, elle combine deux facteurs :

- *Term Frequency* (TF) ou pondération locale où le terme, qui a une o haute fréquence est important de décrire le document.
- *Inverse Term Frequency* (IDF) est la Fréquence Inverse des Documents (IDF), ce facteur mesure l'importance d'un terme dans l'ensemble de la collection. Il vise à donner plus de poids aux termes moins fréquents. Un terme qui apparaît souvent dans la base de données documentaire ne devrait pas avoir le même impact qu'une fréquentes. IDF s'exprime généralement comme (équation 1) :

$$\text{IDF} = \log\left(\frac{N - n_i}{N}\right) \quad (1)$$

Où :

- n_i : le nombre de documents contenant le terme t_i,
- N : nombre total de documents.

Généralement ***tf*idf*** est la multiplication de deux termes TF et IDF. Ainsi, l'équation 3 indique que le terme i présent dans le document j a une fréquence tf_{ij} et le poids TF-IDF de ce terme est calculer par la fonction normalisée suivante :

$$tf*idf = (0.5 + (\frac{0.5*tf_{ij}}{\max(tf_{ij})})) * \text{Log}\,(\frac{N-n_i}{N}) \quad (2)$$

Cette mesure associe au terme un score important s'il apparaît fréquemment dans peu de documents et à des scores faibles si ce terme apparaît rarement. D'autres versions normalisées sont également développées.

En effet, différentes versions de TF-IDF ont été adoptées dans ce qui suit, et le probabiliste TF-IDF a été mentionné dans plusieurs travaux [69] [70], ce qui a marqué une amélioration significative dans le domaine de la classification des textes. Comme un prolongement récent, nous retrouvons le TF-IGM [21], qui intègre un nouveau modèle statistique pour mesurer précisément le pouvoir de distinction de classe. Cependant, la méthode présentée n'est pas une méthode de calcul standard qui dépend uniquement du système de classification. La version floue du TF-IDF a été mise en place pour un système de recherche d'informations, comme introduit dans [29].

2.2.2 Modèles neuronaux vectoriels (Word Embedding)

a) Word2vector

Récemment, les méthodes Word Embedding (Plongement de mots en français) [22], sont très recommandées pour les représentations vectorielles de mots. Les modèles du plongement de mots comme FastText [71], Glove [23], et Word2vec [72] sont utilisés pour distribuer la représentation du document. FastText traite chaque mot comme composé des n-grams de caractères, ce qui prend beaucoup du temps pour générer des incorporations de texte plus rapidement que word2vec. Par conséquent, plus que la taille du corpus augmente, les besoins en mémoire augmentent. D'autre part, le modèle Glove est formé sur la matrice de cooccurrence des mots, ce qui nécessite une grande mémoire pour le stockage. En particulier, si vous modifiez les hyperparamètres liés à la matrice de cooccurrence, vous devez reconstruire la matrice à nouveau, ce qui prend beaucoup du temps.

À propos du modèle neuronal word2vec est généralement connu par ces deux types d'architectures à savoir :

- Le premier type du Word2vec est le CBOW [72], qui prédit un mot cible à partir d'un contexte donné, comme le montre la figure 3(a).
- Le second type est présent dans la figure 3(b), appelé Skip-gram [72], et prédit le contexte d'un mot donné.

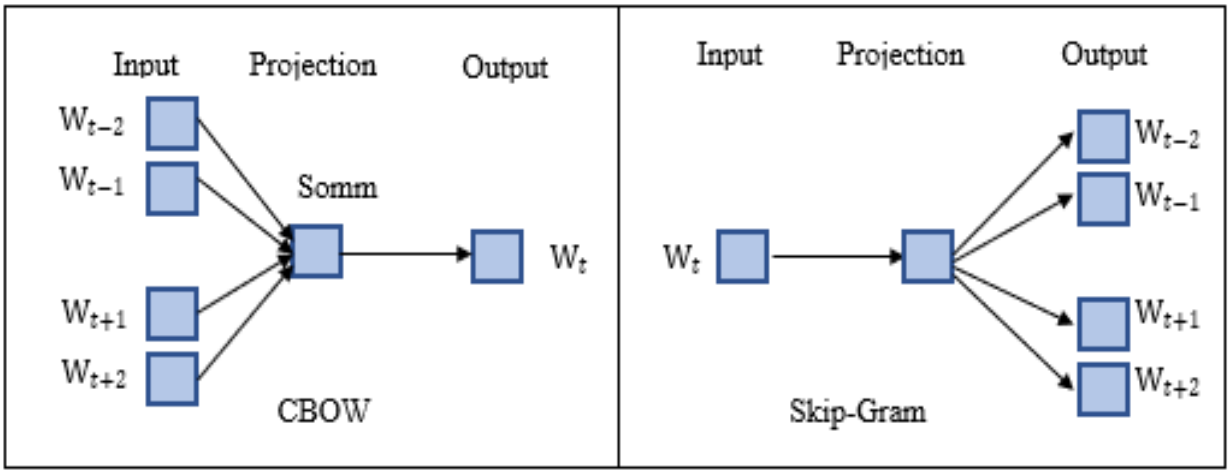

Figure 3: Architectures du modèle word2vec (a) CBOW, (b) Skip-Gram.

Globalement, les modèles cités donnent une grande taille de descripteur, où chaque terme du document est associé à un vecteur. Pour pallier cette lacune, paragraph2vec ou doc2vec a été implémenté pour générer un vecteur représentatif, pour un document complet [28].

a) Document2vector

La motivation initiale derrière l'utilisation du doc2vec était la nature non structurée des documents par rapport aux mots individuels employés. Doc2vec a été créé par Mikilov et Le en 2014. Mikolov était également l'un des auteurs de la recherche originale word2vec, qui est un autre indicateur sur lequel l'architecture doc2vec s'appuie. En outre, le réseau de neurones non supervisé, Doc2vec, permet de produire un vecteur d'entrées pour les modèles décisionnels ML. En se basant sur le principe de word2vec, Doc2vec a deux versions, qui sont la version Distributed Bag of Words Version of Vector (PV-DBOW) [28] et la version Distributed Memory of Paragraph Vector (PV-DM) [28].

Les architectures du Doc2vec sont très avantageux car :

- Elles sont apprises à partir de données non étiquetées et peuvent donc bien fonctionner pour les tâches qui n'ont pas suffisamment de données étiquetées.
- Elles abordent également certaines des principales faiblesses des modèles de sacs de mots. Premièrement, ils héritent d'une propriété importante des vecteurs de mots : la sémantique des mots.
- Elles prennent en considération l'ordre des mots, au moins dans un petit contexte, de la même manière qu'un modèle n-gramme avec un grand n le ferait. Ceci est important, car le modèle n-gram préserve beaucoup d'informations sur le paragraphe, y compris l'ordre des mots. Cela dit, notre modèle est peut-être meilleur qu'un modèle de sac de n grammes car un sac de modèle de n grammes créerait une très haute dimensionnalité.

Contrairement à TF-IDF, Word Embedding donne un vecteur unique pour chaque mot en fonction des mots qui apparaissent autour du mot en question. Mais la TF-IDF reste favorable, vu qu'elle peut être utilisée soit pour assigner des vecteurs à des mots, soit à des documents, comme elle ignore l'ordre des mots. En raison de sa flexibilité et la possibilité de la développer

en douceur, la TF-IDF reste l'intérêt des chercheurs, afin de découvrir des versions sémantiques de cette technique.

2.3 Réduction de la dimensionnalité

La réduction des dimensions est une technique assez employée pour manipuler le fléau de la dimension, vue que les données représentées dans un espace de grande dimension ne sont pas distribuées uniformément, et déforme les résultats de la problématique traitée (comme a mentionné [74] dans son cours. Particulièrement dans le Text Mining, il est essentiel de pratiquer l'étape de la réduction de la dimensionnalité. Il s'agit, alors, d'un ensemble de méthodes, qui propose un recueil des termes élus d'êtres les plus représentatifs du contenu du corpus. Les méthodes de sélection des variables, par exemple, sont souvent utilisées avec plusieurs classifieurs pour améliorer leurs performances et produire les meilleurs résultats de classification [75], en créant un sous ensemble, de toutes les variables du corpus, permet de garder le minimum des variables pertinentes selon certain critère de performance. En général, les méthodes de sélection de caractéristiques sont classées en 3 catégories : les méthodes de filtrage (Filter), les méthodes enveloppes (Wrapper) et les méthodes intégrées (Embedded) [75] [24] [25]. Durant notre étude des systèmes de classification nous avons utilisés les méthodes suivantes :

- ***Sélection de caractéristiques basée sur la corrélation (CFS)*** fait partie de la famille des méthodes de filtrage, qui permis d'éliminer les données non pertinentes et redondantes et, dans de nombreux cas, améliore les performances des algorithmes d'apprentissage [76]. En revanche, cet algorithme vise à trouver l'ensemble de caractéristique individuellement bien corréler avec la classe, mais qui ont une petite inter corrélation entre eux. La technique produit également des résultats comparables à ceux d'un sélecteur de caractéristiques de pointe tiré de la littérature, mais nécessite beaucoup moins de calculs.

- ***Relief*** est parmi les méthodes de filtrage, qui vise à sélectionner des caractéristiques. Il calcule un score de caractéristique, pour chaque caractéristique, qui peut ensuite être appliqué pour classer et sélectionner les caractéristiques les plus performantes [77]. Précisément, Relief essaye de préciser les plus proches voisins d'un certain nombre d'échantillons, sélectionnés au hasard, à partir du jeu des données. Pour chaque échantillon sélectionné, les scores des caractéristiques sont comparés à ceux des voisins les plus proches et les scores pour chaque caractéristique sont mis à jour. L'idée est d'estimer la qualité des attributs en fonction de la qualité de leurs scores et faire la distinction entre des échantillons proches les uns des autres.

- ***Le gain d'information (IG ou* IG*)*** *leur* base est la fonction d'entropie, qui correspond à la quantité d'informations contenues ou délivrées par une source de connaissances [78]. Par conséquent, il est nécessaire de calculer l'entropie des exemples d'apprentissage avant le calcul du gain final. Ainsi, la formule d'entropie a été définie comme :

$$\text{Entropie } (S_A) = \sum_{i=1}^{c} (-p_i)\ log_2(p_i) \quad (3)$$

Où :
S : les exemples de formation ;
A : l'attribut à tester ;
c : le nombre de valeurs possibles pour la fonction ciblée ;
p_i: la proportion des exemples dans S qui ont i comme valeur pour la fonction ciblée.

En se basant sur la formule d'entropie prédéfinie (équation 3), l'équation de gain d'information a été définie comme :

$$\text{Gain } (S, A) = \text{Entropy } (S) - \frac{\sum_{v \epsilon V(A)} |S_v|}{|S|} \text{ Entropy } (S_v) \quad (4)$$

Avec :
V(A) : les valeurs possibles de l'attribut A.
S_v : le sous-ensemble de S qui contient les exemples qui ont la valeur v pour l'attribut A.

Ensuite, les caractéristiques extraites sont classées en respectant les scores IG déterminés par l'équation 5. De plus, le sous-ensemble de caractéristiques le plus pertinent est accepté comme étant des entrées pertinentes pour les classificateurs adoptés.
Autres méthodes de sélection des variables sont également connues dans la littérature à l'image de la corrélation de Person [79]. Généralement la découverte de ce processus et le développement des méthodes de sélection ont participé à l'évolution des descripteurs représentatives des textes, et à l'amélioration de la performance des classifieurs.

2.4 Applications du Text Mining

Les utilisations et les domaines de l'exploration de texte diffèrent, tout comme les méthodes et les outils utilisés pour le succès des systèmes liés aux applications de ce domaine. Le diagramme de Venn [7] représente l'intersection du Text Mining avec les six champs, représentés par des ovales, associés comme les statistiques et linguistique computationnelle [7]. En effet, les sept domaines de pratique du Text Mining existent aux principales intersections de Text Mining avec ses six domaines connexes, comme montra la figure au-dessous (figure 5).

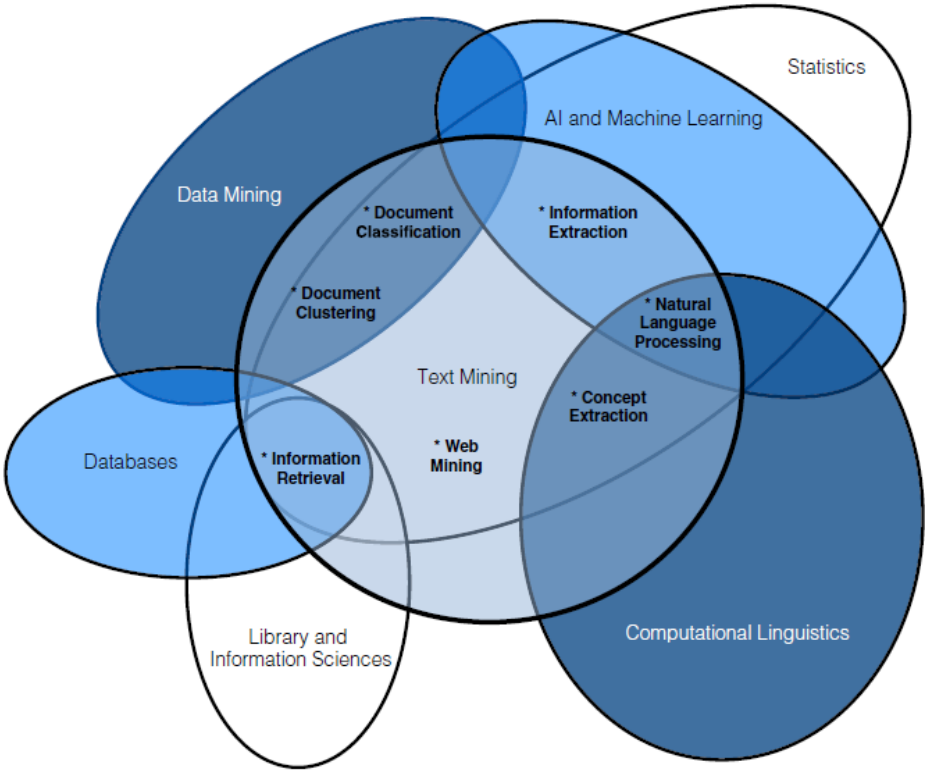

Figure 4: Diagramme de Venn de l'intersection du Text Mining et des six champs associés [7].

Généralement, après l'extraction des caractéristiques les plus représentatives, formant ainsi un vecteur descripteur, ou une forme numériques lisible par les modèles d'apprentissage artificiel qui appartient aussi au champ statistique, pour chaque texte de caractères exprimé en langage naturel, une phase de décision est requise pour déterminer la sortie souhaitée du caractère. Parmi les décisions pris par les systèmes d'application du Text Mining, nous trouvons : La catégorisation multi-label des documents [80], l'analyse des sentiments (inclus dans les applications du web Mining cité dans le diagramme Venn) [7], les systèmes de recommandation [81] et les systèmes de recherche d'information en général [82].

2.4.1 Classification des données non structurées

La classification supervisée est l'une des tâches les plus importantes dans les systèmes intelligents, dont le but est de construire un modèle capable de séparer entre les différentes classes du problème en se basant sur un ensemble de données, appelé ensemble d'apprentissage. Les exemples ou les instances de cet ensemble sont définis sur un espace de caractéristiques bien déterminé. Ensuite, ce modèle doit être capable d'attribuer une nouvelle instance définie, sur le même espace de caractéristiques, à une classe parmi les prédéfinies [83]. Cette tâche facilite la simulation des actes important dans la vie pratique d'un certain nombre de secteurs, qui ont besoin d'un stockage et d'une organisation efficaces des informations, afin d'en faciliter l'accès ultérieurement. Parmi les tâches qui exploite la catégorisation des données (non structurées) nous trouvons : l'analyse des sentiments, la recherche de l'information et les

systèmes de recommandation et notre travail se focalise, principalement, sur le développement des systèmes de classification des textes.

L'architecture initiale de la classification regroupe un ensemble de processus que l'architecture de la Figure 5 montre.

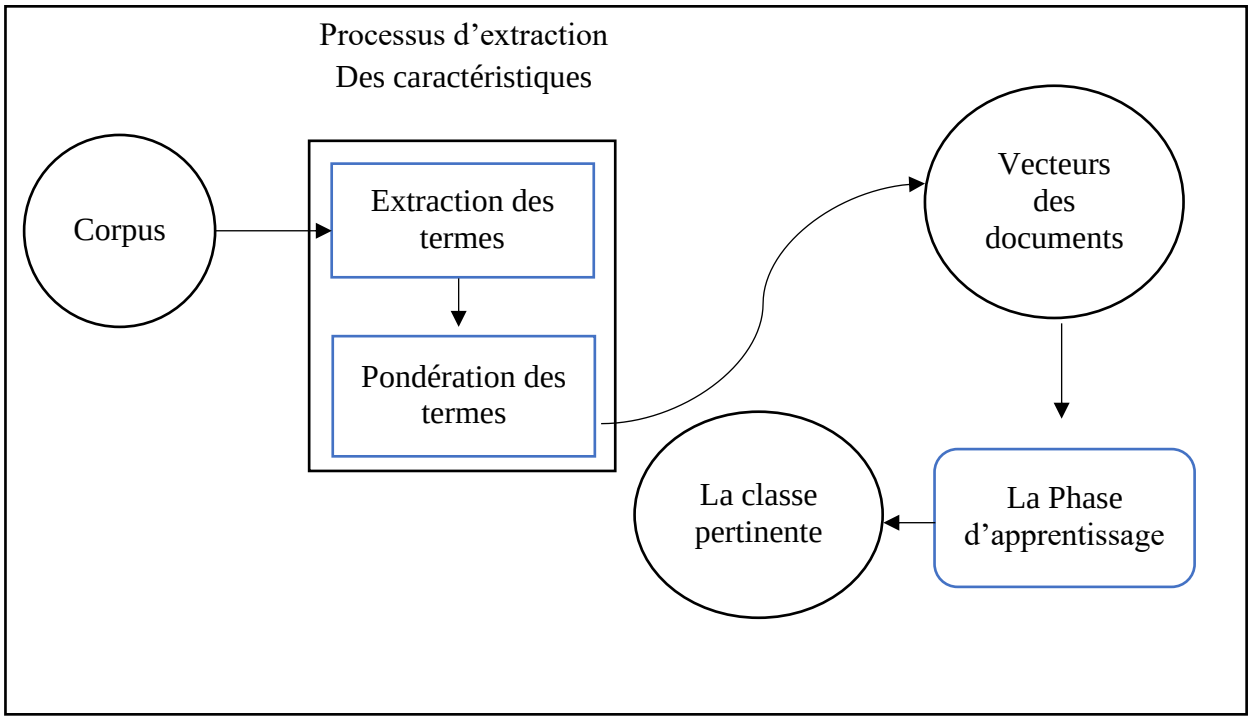

Figure 5: Architecture des systèmes classique de la classification des textes.

Les étapes de la classification d'un corpus ou l'insertion d'une requête (autant que texte brut) dans des catégories prédéfinis sont alors :

(a) L'extraction des termes (Extraction des termes) : dans cette procédure trois phase doivent être réaliser, dont la première phase consiste à remplacer les textes, en langage naturel, par une séquence linéaire de caractères. Deuxièmement, les mots vides sont éliminés à l'aide d'une liste connue par *Stop List*. Finalement, des algorithmes de *Stemming ou lemmatisation* sont nécessaires pour trouver les radicaux de la liste obtenue. Cette procédure est utile dans de nombreux domaines de la linguistique informatique et dans la recherche d'informations.

(b) La représentation numérique/vectorielle des données (Pondération des termes) : Dans cette phase, les termes sont pondérés et transformés en vecteurs. À cet égard, la représentation vectorielle constitue le processus d'extraction des caractéristiques, et plusieurs méthodes ont été proposées dans la littérature [84].

(c) La phase d'apprentissage : (Les modèles artificiels sont employés dans ce stade) où la sortie obtenue, de la phase (b), est divisée en trois ensembles, pour connaître l'ensemble d'apprentissage, l'ensemble de validation et l'ensemble de test. Le premier et le deuxième ensemble sont utilisés pour sélectionner le modèle approprié ; puis, les deux

ensembles sont utilisés pour éduquer le meilleur modèle. Les performances du modèle obtenu sont testées sur la base du troisième ensemble de test.

(d) Les classifieurs qui se base sur le traitement décrit dans la phase (c) décide de la classe pertinente. Employant l'ensembles des méthodes de classification supervisée, présentes dans ce chapitre, l'objectif prochain sera la conception des systèmes de catégorisation binaire et multi classe des données textes.

2.4.2 Analyse des sentiments

L'analyse des sentiments (Ou Opinion Mining) est un sous-domaine du Texte Mining [85], qui essaye de traiter et analyser des opinions (sentiments et attitudes présents dans un texte) afin d'améliorer la qualité des applications, du e-services par exemple. Ces opinions peuvent être classées d'une manière supervisée ou non supervisée, comme ils peuvent avoir une catégorisation binaire ou multi-classes, selon le type des données étudiées [85]. L'objectif des systèmes d'analyse de sentiments et presque similaire à celui de la recherche d'information, où le but principal est de livrer l'information convenable aux utilisateurs du web.

L'ensembles des applications du Text Mining cité dans ce passage repose sur les algorithmes d'apprentissage automatique, qui permettent l'automatisation, la rapidité et la fiabilité de ces systèmes, ainsi que nos systèmes de catégorisation se base principalement sur ces algorithmes la deuxième partie de ce chapitre détaillera les principaux concepts et algorithmes que nous avons adoptés durant notre étude.

III. Généralité sur l'apprentissage statistique

L'apprentissage statistique (sous le nom Apprentissage Automatique (AA), Machine Learning (ML) en anglais) est une technique d'intelligence artificielle, qui se fonde sur des approches mathématiques et statistiques.

Le concept ML a été développés pour que les ordinateurs soient capables à apprendre, à partir d'un jeu de données, pour des fins prévisionnelles et/ou décisionnelles [86]. L'apprentissage automatique permet d'améliorer les performances de la machine pour résoudre des tâches sans être explicitement programmés. En effet, il concerne la conception, l'analyse et l'optimisation, de telles méthodes.

L'évolution relativement récente de la théorie de l'apprentissage s'explique par le développement de moyens de stockage d'information et l'émergence de jeux de données volumineuses. L'apprentissage automatique comporte généralement deux phases :

- La phase d'apprentissage (ou d'entraînement, training en anglais) est généralement réalisée préalablement à l'utilisation pratique du modèle. Elle consiste à estimer un modèle à partir de données d'apprentissage, appelées observations, qui sont disponibles et en nombre fini, lors de la phase de conception du système [87].
- La phase du test correspond à la mise en production : le modèle étant déterminé, de nouvelles données, qui n'apparaissaient pas dans l'ensemble d'observation, peuvent alors être soumises afin d'obtenir le résultat correspondant à la tâche souhaitée [87]. En pratique, certains systèmes peuvent poursuivre leur apprentissage une fois en

production, pour peu qu'ils aient un moyen d'obtenir un retour sur la qualité des résultats produits.

Selon les informations disponibles durant la phase d'apprentissage, l'apprentissage est qualifié de différentes manières : l'apprentissage supervisé [87], l'apprentissage non supervisé [87], l'apprentissage semi-supervisé [88] et l'apprentissage par renforcement [87], etc….

Les statistiques récentes de 2020 à 2021 illustrés dans la figure 6 extraite du site web (https://www.statista.com/statistics/1111204/machine-learning-use-case-frequency),l'apprentissage artificiel a connu un grand emploi dans des différents domaines, dont les applications du Tex Mining appartiennent.

Dès 1980 et au début des années 1990, les deux composants : L'apprentissage statistique et le Text Mining ont été combinés dans le but de développer des systèmes modernes. Les méthodes de l'apprentissage statistique sont appliquées dans la seconde étape du Text Mining, l'*interprétation de l'analyse*, qui permet de réaliser certaines tâches sur le texte. Parmi les anciennes applications des techniques d'apprentissage automatique est l'exploration de données pour la catégorisation des textes.

Généralement la classification ou la catégorisation des données non structurés sont inclus dans le type d'apprentissage supervisé., où les algorithmes de l'apprentissage exigent une transformation des données textuelles brute à une forme numérique qui sera l'entré adéquate au classifieur choisi. Aussi les sorties désirées sont prédéfinies comme les classes, les catégories ou les étiquettes d'où appartient un document.

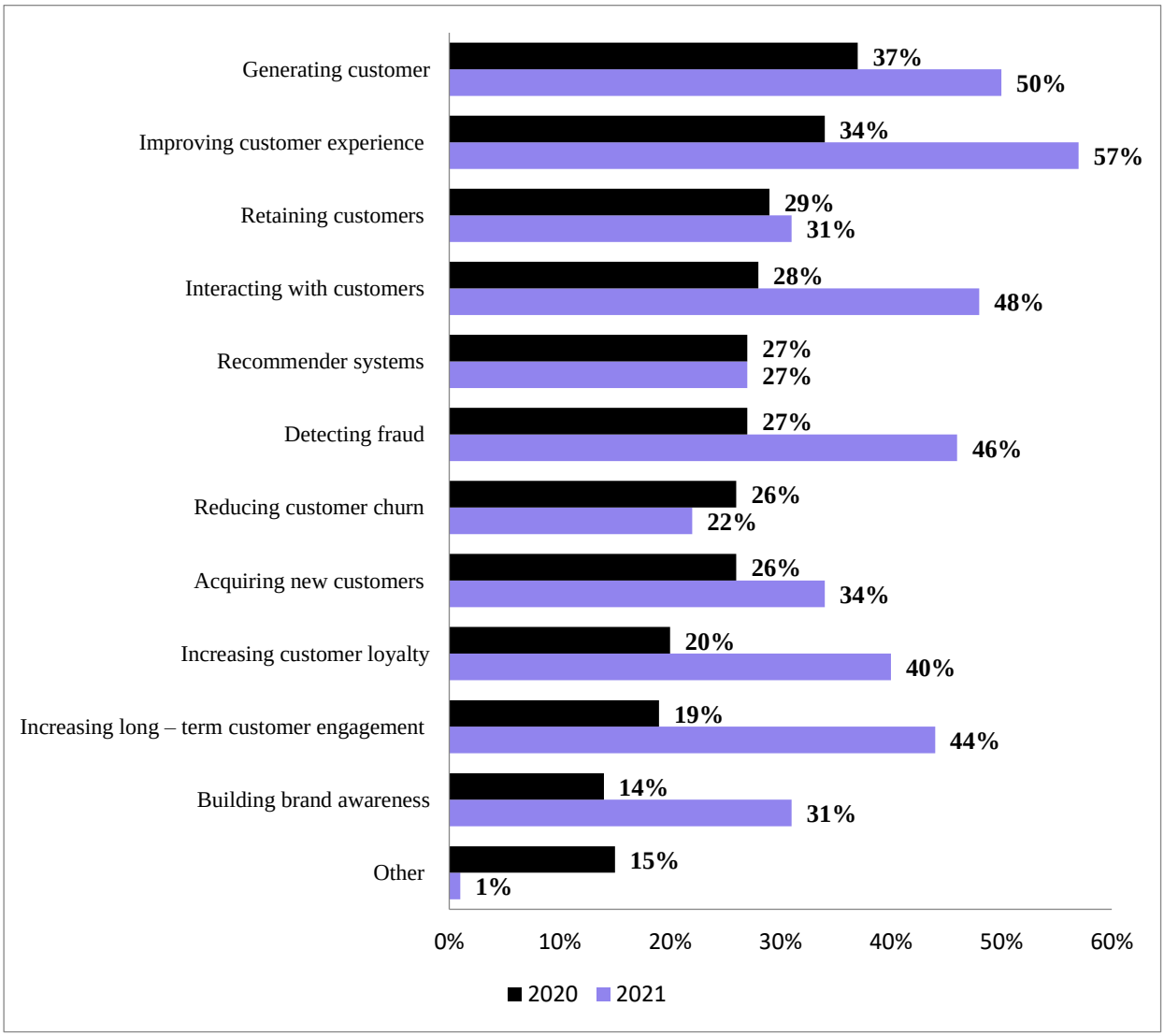

Figure 6: Cas d'utilisation de l'intelligence artificielle et de l'apprentissage automatique pour les entreprises du monde entier.

Le problème d'apprentissage consiste à déduire la fonction qui mappe entre l'entrée et la sortie, de sorte que la fonction apprise peut être utilisée pour prédire la sortie d'une nouvelle entrée. Plusieurs algorithmes ont été discuté dans la littérature, dans ce chapitre nous citerons ceux qui sont largement utilisés et qui se basent sur différentes techniques qui peuvent être probabiliste, structurelles, neuronales, etc.

Ce chapitre présente aussi les démarches communément adoptées pour évaluer les algorithmes de classification et leurs combinaisons afin de renforcer la précision des systèmes de classification.

1. Algorithmes d'apprentissage supervisé pour la catégorisation des Textes

Tout en travaillant avec des problèmes de classification dans l'apprentissage automatique (spécialement l'apprentissage supervisé), divers algorithmes de classification automatique (appelé classifieurs), natifs entrent en jeu. Les classifieurs linéaires en tant que variété

d'algorithmes populaires dans la classification automatique, sont employés par les développeurs des systèmes intelligents qui cherchent souvent des programmes à une faible complexité ce qui accélère le temps de réponse et évite une lourde implémentation.

Par conséquent les classifieurs linéaires restent les plus simples et les plus rapide à utiliser pour des objectifs décisionnels. Aussi, ils assurent de bons résultats lorsque le nombre de dimension de l'espace des observations est grand, comme dans le Text Mining (https://tel.archives-ouvertes.fr/tel-00541059).

Pour la catégorisation des documents, nous pouvons distinguer deux grandes familles de méthodes de classification linéaire comme montrera la figure 7.

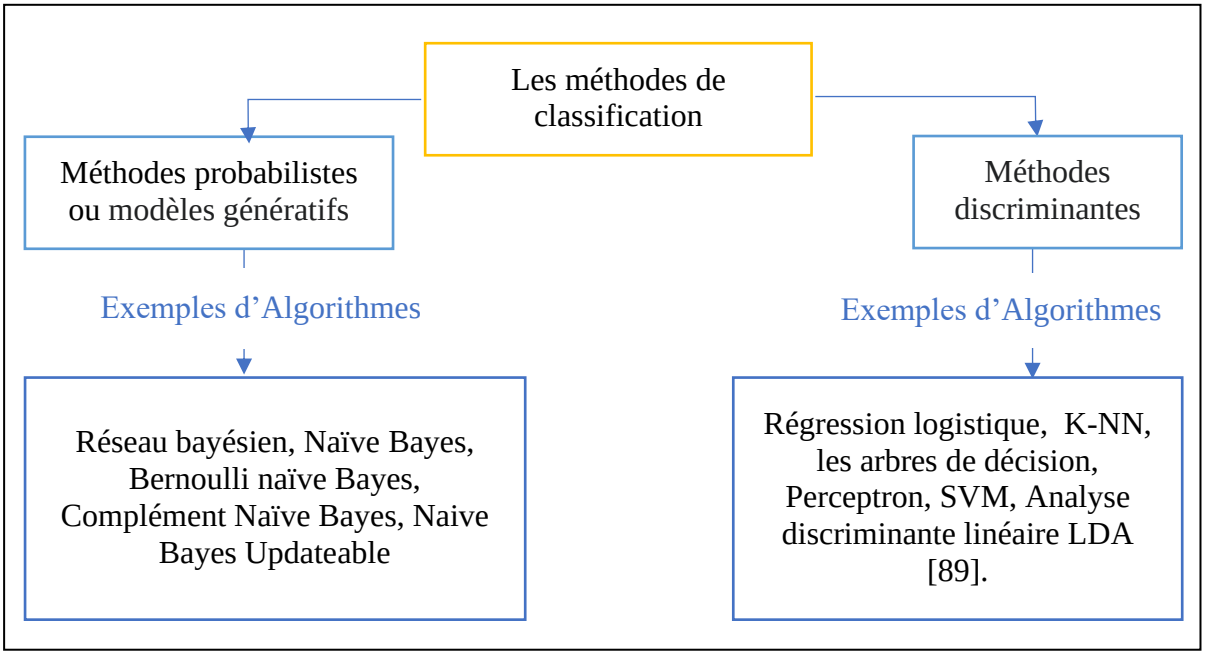

Figure 7: Types des méthodes de classification des textes.

Nous enchainerons par la découverte et la description de chaque catégorie des méthodes de classification (probabiliste et discriminantes), plus les algorithmes de chaque type que nous pouvons l'employé dans des systèmes de catégorisation des Textes.

1.1 Méthodes de classification Probabilistes

Les méthodes de la classification Probabiliste (ou modèles génératifs) cherchent à modéliser la distribution jointe des entrées et des sorties, et reposent sur la *Théorie de la Décision Statistique* (*TDS*). Généralement la *TDS* considère un ensemble de critères d'optimalité qui maximise la probabilité qu'un caractère d'entrée appartient à une classe donnée parmi les classes prédéfinies du problème. Comme Arica décris dans [90] l'efficacité des techniques statistiques dépond principalement de :

- La distribution de l'ensemble de caractéristiques (d'où il est nécessaire d'être gaussienne ou, dans les pires des cas, uniforme).
- La disponibilité d'exemples adéquate pour chaque classe.

- L’extraction, à partir d'un corpus, d’un ensemble de caractéristiques qui représente distinctement chaque classe de caractères.

Dans les approches probabilistes, chaque caractère est représenté en termes de caractéristiques ou de mesures dans un espace de dimension N. Le but est de choisir les caractéristiques qui permettent aux vecteurs des caractères appartenant à différentes catégories d'occuper des régions compactes et disjointes dans l’espace de caractéristiques n-dimensionnel (autrement dit une loi de probabilité jointe est définie pour toutes les variables possibles). Étant donné un ensemble de modèles d'entraînement de chaque classe, l'objectif est d'établir des limites de décision dans l'espace des caractéristiques qui séparent les caractères appartenant aux différentes classes, les limites de décision sont déterminées par des distributions de probabilité qui doivent être spécifiées ou apprises [94].

Comme la figure 7 a montré, les classifieurs bayésiens font partie des méthodes probabilistes, ils sont aussi largement utilisés pour la classification des données non-structurées.

1.1.1 Réseaux Bayésiens

Les réseaux bayésiens (RB) sont des graphes probabilistes acycliques, comme illustra la figure 8.

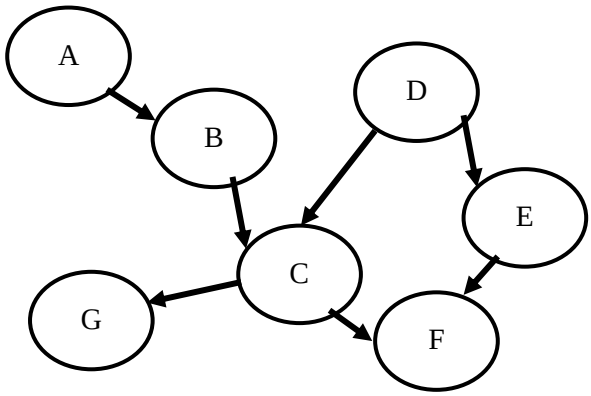

Figure 8: Modèle de graphe probabiliste acyclique.

Les nœuds du graphe bayésien représentent les variables aléatoires et la structure du réseau définit leurs dépendances conditionnelles. Pour utiliser le RB comme outil de classification il faut procéder les étapes suivantes [95] :

- Modélisation du problème : pour n termes représentons un corpus par exemple nous leurs associes un ensemble de variables aléatoires noté par : $X = \{X_1, X_2, \ldots, X_n\}$. Dans le cas de classification, les n variables aléatoires représentent l'ensemble des fonctionnalités, et une variable aléatoire supplémentaire pour la classe.
- Choisir une architecture réseau adéquate parmi un ensemble d’architectures présentes dans la littérature [96].
- Construire la matrice de probabilités conditionnelles du nœud i, connaissant l'état de ses parents « Pa » :

$$\theta_i = P\,(X_i / Pa\,(X_i)). \quad (5)$$

Nous pouvons employer l'estimation du maximum a posteriori [97] pour estimer P(y) et P (x_i / y) directement à partir des données.

- Interférant avec la réponse à une demande donnée, une telle tâche est basée sur la distribution conjointe suivante :

$$P(X_1, X_2, \ldots, X_n) = \prod P(X_i / Pa(X_i)). \quad (6)$$

Étant donné une variable de classe y et un vecteur de caractéristiques de X_1 à X_n, via la relation énoncée par le théorème de Bayes [95] :

$$P(y / x_1, x_2, \ldots, x_n) = (P(y) P(x_1, x_2, \ldots, x_n / y)) / (P(x_1, x_2, \ldots, x_n)) \quad (7)$$

La décomposition de la distribution conjointe, équation 2, permet d'avoir des puissants algorithmes d'inférence qui rendent les réseaux bayésiens très utiles pour modéliser et raisonner lorsque les situations sont incertaines ou si les données sont incomplètes. Ils sont aussi utiles pour les problèmes de classification où les interactions entre les différentes caractéristiques ou variables peuvent être modélisées par des relations de probabilités conditionnelles [98].

Pour les approches d'inférence, deux types qui existent :

- Exactes (inférence par énumération, algorithme d'élimination des variables et algorithme de regroupement) ;
- Approchées (Méthodes d'échantillonnage direct, pondération par la vraisemblance et inférence par simulation des chaines de Markov Monte-Carlo) [99].

En outre, deux problèmes s'imposent lors de l'utilisation des réseaux bayésiens :

- *Le choix de la structure du graphe* où dans certains cas, la structure du réseau bayésien est fournie a priori par un expert, cependant, la détermination automatique de cette structure à partir de l'ensemble d'apprentissage est un problème NP-difficile, les chercheurs font recours aux méthodes évolutionnistes (Bayésien naïf, Hill Climbing, algorithme K2, recherche gloutonne, algorithme génétique, etc.).
- *L'estimation de ses distributions de probabilités* : Pour ce genre de problème, l'apprentissage par maximum de vraisemblance, l'estimation bayésienne, l'algorithme EM, ou autres [100] sont employés.

Les chercheurs font recours aux méthodes évolutionnistes afin de simplifier est renforcer l'utilisation des RB dans des problèmes plus complexes. L'intervention des chercheuses, et les modifications apportées sur l'architecture RB ont donné naissance aux autres classifieurs comme : le classifieur Naïf bayésien et le modèle d'événement multivarié Bernoulli qui seront détaillés par la suite.

1.1.2 Naïve Bayes

La classification naïve bayésienne (NB) est une des méthodes linéaires génératives employée dans l'apprentissage supervisé des données textuelles. Le NB se base sur l'application du théorème de Bayes avec l'hypothèse « naïve » d'indépendance fort entre chaque paire de caractéristiques [101]. Sur la base de cette hypothèse naïve, l'équation (8) est simplifiée en :

$$P\ (y / X_1, X_2, \ldots, X_n) = (P\ (y) \prod P\ (X_i / y)) / P\ (X_1, X_2, \ldots, X_n) \quad (8)$$

Puisque P (X_1, X_2, …, X_n) est constant compte tenu de l'entrée, nous pouvons utiliser la règle de classification suivante :

$$Y = \text{Arg. max}_y\ P\ (y) \prod P\ (X_i / y) \quad (9)$$

En exclusive le classifieur bayésien naïf ne nécessite pas une grande quantité de données d'entraînement pour estimer les paramètres nécessaires (moyennes et variances des différentes variables). Malgré la simplicité du modèle de conception « naïf » et ses hypothèses, les classifieurs bayésiens naïfs ont fait preuve d'une efficacité inattendue et plus que suffisante dans beaucoup de situations réelles complexes.

1.1.3 Bernoulli naïve Bayes

Bernoulli naïve baye ou modèle Bernoulli d'événement multivarié (NBMU) traite les entités comme des entrées de variables binaires indépendantes. Ce modèle est largement utilisé pour les tâches de classification de documents où des caractéristiques d'occurrence de terme binaire sont utilisées plutôt que des fréquences de terme [102].

Soit Xi variable exprimant l'occurrence ou l'absence du i -ème terme du vocabulaire. Étant donné une classe, la probabilité d'un document est mesurée par :

$$P\ (X/\ C_k) = \prod_i^n P_{k_I}^{x_i} \left(1 - P_{k_i}\right)^{(1-x_i)} \quad (10)$$

Où : P_{k_i} est la probabilité que la classe C_k génère le terme i.

Ce modèle d'événement est recommandé pour les textes courts, et il a l'avantage de modéliser explicitement l'absence des termes, mais le NBMU ne peut pas résoudre efficacement les problèmes de classification des textes volumineux [102].

1.1.4 Naïve Bayes Complémentaire

Naïve Bayes Complémentaire (Complement Naive Bayes en anglais (CNB)) est une autre version avancée de la classification naïve bayésienne, qui est souvent utilisée comme base de classification de texte car elle est rapide et facile à mettre en œuvre. Pour estimer les poids dans les Bayes naïfs, nous utilisons des données d'apprentissage d'une seule classe C. En revanche, CNB estime les paramètres en utilisant les données de toutes les classes sauf c [103]. En bref, il estime les probabilités de caractéristiques pour chaque classe c en fonction du complément de c, c'est-à-dire sur les échantillons de toutes les autres classes, plutôt que sur les échantillons d'apprentissage de la classe c elle-même.

L'estimation de CNB est exprimée comme :

$$\hat{\theta}_{\bar{c}i} = \frac{N_{\bar{c}i} + \alpha_i}{N_{\bar{c}} + \alpha} \quad (11)$$

Où :

$\boldsymbol{N_{\bar{c}i}}$ **:** Le nombre de fois où le mot i est apparu dans des documents de classes différentes que c.

$\boldsymbol{N_{\bar{c}}}$ **:** Le nombre total des occurrences du mot dans les classes autres que c ;

αi et α sont des paramètres de normalisation.

Les règles de classification propre à la CNB sont présentes dans [103]. De plus cette extension bayésienne montre des résultats agréables lorsqu'il s'agit d'une classification multi-label [104] and [105]. Parmi les avantages d'un modèle génératif l'estimation de la confiance d'une prédiction et donc de formuler un rejet d'une prédiction, ce qui est impossible dans un modèle discriminatif. De plus la probabilité conditionnelle employée dans ces modèles permet de générer des nouvelles données.

Également, la combinaison de différents modèles par des fonctions linéaires de leurs prédictions, permet une accumulation continue des connaissances.

Puisque la grande quantité des données traitées rend les calculs probabiliste conditionnels couteux, l'utilisation des autres méthodes est nécessaire.

1.2 Méthodes de classification discriminantes

La seconde famille des modèles associés à la classification supervisée est les modèles discriminants (MD) ou conditionnels. Ce type de modèle repose sur l'analyse discriminante comme technique statistique qui vise à décrire, expliquer et prévoir l'appartenance d'un assortiment d'observations à des classes prédéfinies, à partir d'une série de variables prédictives (descripteurs). Différemment aux modèles génératifs, les MD consistent à modéliser les relations qui lient les entrées aux sorties du système, avec un minimum d'hypothèses sur la structure des données d'entrée. Ces méthodes répondent directement à l'objectif de l'utilisateur en se focalisant sur la règle de décision plutôt que sur l'interprétation de cette décision. Outre les MD cherchent d'abord à maximiser la qualité de la classification sur un jeu de test, et utilisent la fonction du coût afin de réaliser l'adaptation du modèle de classification terminale [106][89]. Même si les MD nécessitent plusieurs techniques d'optimisation numériques, ils sont favorisés par :

- La meilleure qualité de précision, qui implique un meilleur résultat d'apprentissage.
- La flexibilité des MD quelle que soit son application
- La prise en compte de toutes les données, et l'utilisation de moins d'échantillons de formation différemment aux autres modèles.

En effet les MD sont utilisés dans le contexte Texte Mining, et ils ont démontré leur efficacité pour la catégorisation multi-label data ; pour cela un ensemble des méthodes de référence : SVM, simple logistique, les forêts aléatoires, et les réseaux de neurones seront discutés par la suite.

1.2.1 Régression Logistique Simple

La Régression Logistique Simple (RLS) est un algorithme supervisé de classification, populaire en apprentissage statistique, aussi classé parmi les modèles discriminants linéaires. Joseph Berkson est le premier qui a mentionné le modèle Logit en 1944. Et en 1989 Palma et Thisse ont annoncé le modèle complet. En estimant les probabilités à l'aide d'une fonction logistique, qui est la distribution logistique cumulative, la régression logistique mesure la relation entre la variable dépendante catégorielle et une ou plusieurs variables indépendantes. Afin de booster le modèle de régression logistique linéaire et de produire le nouveau classifieur RLS l'algorithme Logit Boost, qui est défini par Friedman [107], est inséré avec des fonctions de régression. Le nombre optimal d'itérations Logit Boost à effectuer est validé de manière croisée, ce qui conduit à une sélection automatique d'attributs. L'avantage de la RLS est qu'il a une sélection d'attributs intégrée, si vous appliquez des paramètres par défaut, il arrête d'ajouter des modèles de régression linéaire simples lorsque l'erreur de classification validée croisée ne diminue plus [108]. Le cadre démonstratif et l'algorithme complet de la RLS sont détaillés dans le site : http://mason.gmu.edu/~ddebarr/Logistic_Regression_and_Logit_Boost.pdf.

1.2.2 Machine à vecteurs supports SVM

Les machines à vecteurs de supports connus aussi par l'abréviation SVM sont parmi les approches les plus utilisées pour la catégorisation des textes. Ces classifieurs linéaires sont séduisants grâce à ses caractéristiques principales [107], où les SVMs appliquent une frontière de décision, en utilisant un séparateur à marge maximale, comme illustré dans la figure 9, avec la plus grande distance possible entre les points. En revanche Les SVM sont des méthodes non paramétriques qui résistantes aux problèmes de sur-apprentissage, et qui sont capables de représenter les fonctions complexes.

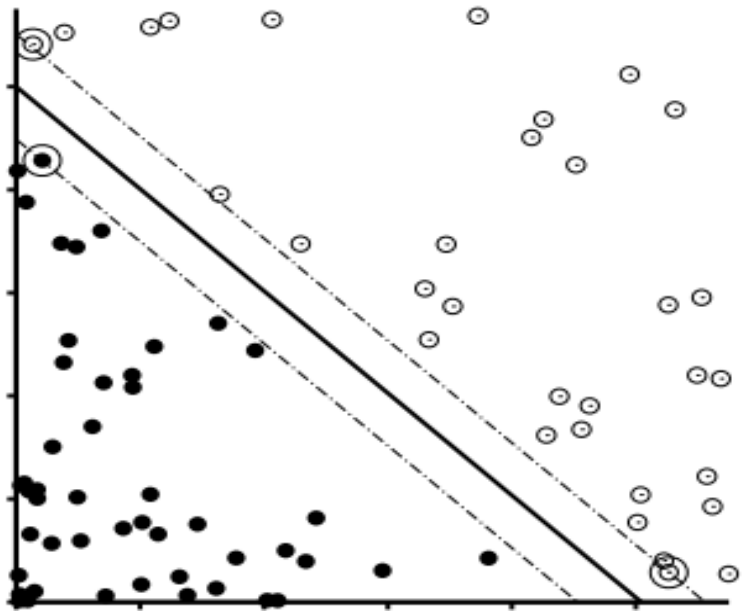

Figure 9: Exemple de séparateur à marges pour une séparation binaire [107].

Aussi parmi les avantages SVM, le traitement des données qui ne sont pas linéairement séparables, par l'explosion les données dans un espace de dimensionnalité supérieure en utilisant des fonctions noyaux pour trouver un séparateur linéaire.

Pour $A=(x^1, y^1),\ldots,(x^N, y^N)$ un ensemble d'apprentissage issu d'un problème de classification binaire tel que les exemples des deux classes sont soit étiquetés par +1 soit par -1.

Les poids *w* qui définissent l'hyperplan séparateur des SVM dans l'espace φ sont donnés Par [107] :

$$w= \sum_{j=1}^{N} \alpha_j y^j \varphi\left(x^j\right) \tag{12}$$

Tel que les α_j sont les coordonnées de la solution du problème dual suivant :

$$(D)\begin{cases} Min\ \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\alpha_j \mathrm{y}^{\mathrm{i}}\mathrm{y}^{\mathrm{j}}\varphi\left(\mathrm{x}^{\mathrm{i}}\right)\varphi\left(\mathrm{x}^{\mathrm{j}}\right) \\ SC: \\ \sum_{i=1}^{n} \alpha_i \mathrm{y}^{\mathrm{i}} = 0 \\ \alpha_i > 0 \end{cases} \tag{13}$$

La classe d'un nouvel objet x est estimée par l'équation :

$$y= signe\ (w.\varphi(x)) \tag{14}$$

Puisqu'il est difficile de connaitre l'espace des caractéristique, certains chercheurs ont proposé de remplacer le terme $\varphi(x^i)\varphi(x^j)$ par $K(x^i, x^j)$, où :

K est une fonction suffisamment complexe pour pouvoir séparer linéairement les données.

Ce genre de fonctions sont appelées fonctions noyaux [109], une variétés de fonctions noyaux sont disponibles dans la littérature, nous citons entre autres : fonction linéaire, fonction gaussienne RBF, fonction polynomiale, fonction sigmoïde [109].

Actuellement, les SVM ont été ajuster pour répondre aux problèmes multi classes au lieu de concevoir seulement les problèmes de classification binaire. Pourtant, il existe deux approches pour adapter aux difficultés multi-classes où la première considère toutes les classes dans la formulation du problème d'optimisation, et la deuxième approche consiste à combiner un ensemble de classifieurs binaires [110].

1.2.3 Réseaux de Neurones Artificiels (RNA) et le Perceptron Multi Couches (PMC)

a) Réseaux de neurones artificiels

L'objectif original the cette contribution est d'inventer des machines intelligentes de perception en simulant la structure physique des cerveaux humains [111]. Les RNA sont comme des systèmes de calcul massivement parallèles constitués d'un très grand nombre de processeurs simples avec de nombreuses interconnexions. Les modèles de réseaux neuronaux tentent d'utiliser certains principes organisationnels (apprentissage, généralisation, adaptabilité,

représentation distribuée, etc.) dans un réseau de graphes orientés pondérés dans lesquels les nœuds sont des neurones artificiels organisés en couches et les arrêtes sont les connexions entre les différentes couches du réseau. Les réseaux de neurones ont la capacité d'apprendre des relations non linéaires complexes entre les entrées et les sorties du problème, d'utiliser des procédures séquentielles d'apprentissage et de s'adapter aux données [94]. Généralement, les RNA sont composés de trois couches principales, c'est-à-dire la couche d'entrée, la couche cachée et la couche de sortie. La première couche est toujours associée aux entrées système. Pour la couche intermédiaire, c'est-à-dire, la couche cachée peut contenir elle-même plusieurs couches, y compris les neurones du kit. En outre, la performance d'une couche, à ce stade, est efficace pour résoudre un problème complexe. La dernière couche ou la couche de décision correspond aux sorties du système et qui produit des classes adéquates pour la tâche de classification. Les neurones entièrement composites sont connectés par des connexions pondérées, qui régissent le processus du réseau. En outre, la classification RNA a besoin de l'inférence, en employant certains algorithmes, afin de réaliser les sorties souhaitées.

De nombreuses architectures de réseaux de neurones ont été proposées dans la littérature, et le Réseau Feed-Forward (ou Feed-Forward Neural Network (FNN)) est un type de structure bien utilisée, sans connexion cyclique entre les couches de réseau, généralement appliquée comme une technique d'apprentissage supervisé dans l'apprentissage automatique [112]. Aussi le perceptron multicouche (PMC) est un des familles de réseaux neuronaux les plus couramment utilisées pour les tâches du Text Mining.

b) Perceptron MultiCouches (PMC)

L'inspiration originale derrière cette technique vient des réseaux bioélectriques du cerveau humain formé par les neurones et leurs synapses. De même, un modèle neuronal de base appelé perceptron possède un ensemble de connexions pondérées (semblables aux synapses dans les neurones biologiques), une unité de sommation et une fonction d'activation (comme la figure 11 montres).

La sortie de l'unité de sommation et une combinaison linéaire des entrées dont les coefficients sont les poids. La fonction d'activation peut être linéaire ou non-linéaire, la fonction d'activation sigmoïde est la plus utilisée [111].

Le PMC est un réseau de neurones statique qui comporte trois types de couches : la couche d'entrée qui présente les données au réseau, la couche de sortie qui sert à la décision, et en fin, les couches cachées qui effectuent le traitement (voir la figure 12).

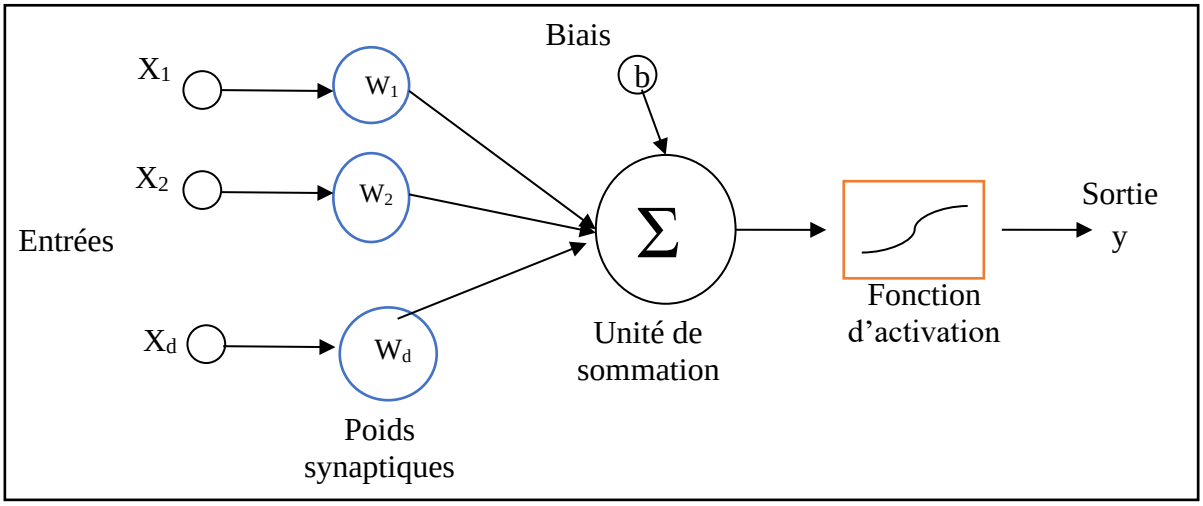

Figure 10: Modèle et fonctionnement d'un réseau de neurones [111].

L'apprentissage supervisé de ces réseaux se fait via l'algorithme de rétropropagation du gradient [113], qui opère en quatre étapes :

- Etape 1 : initialisation arbitraire des poids ;
- Etape 2 : propagation avant d'un exemple de l'apprentissage et calcul de sa sortie ;
- Etape 3 : propagation arrière de l'exemple en fonction de sa sortie désirée ;
- Etape 4 : la mise à jour des poids de connexion.

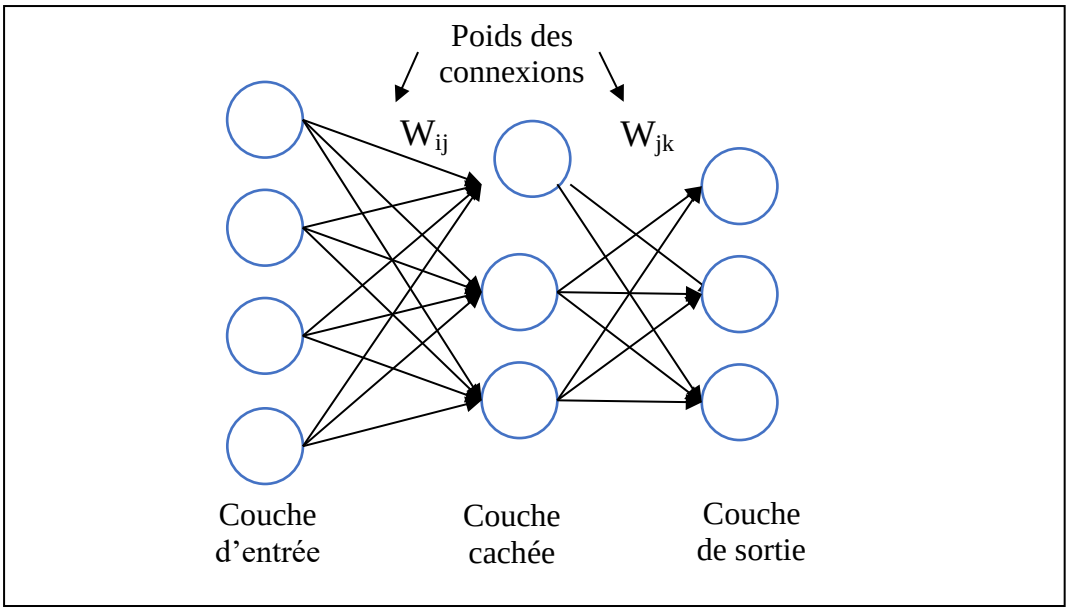

Figure 11: Exemple d'un perceptron à une seule couche cachée [114].

Les trois dernières étapes s'effectuent jusqu'à ce que l'erreur sur les neurones de sorties devienne suffisamment petite ou après un certain nombre d'itérations déterminé à l'avance.

Le PMC peut avoir n'importe quel nombre de couches cachées et n'importe quel nombre de neurones par couche cachée, néanmoins, l'utilisation d'une seule couche cachée est suffisante pour résoudre un problème complexe non linéaire. Le choix du nombre de couches cachées et du nombre de neurones de chaque couche est toujours un défi [115]. Dans la majorité des cas, l'architecture du PMC est déterminée expérimentalement.

1.2.4 Forêts d'arbres décisionnels

Les forêts aléatoires (ou forêts d'arbres décisionnels, et Random Forest classifier) (en anglais)) font partie de la famille des arbres de classification qui consiste à prévoir ou expliquer les réponses d'une variable dépendante catégorielle. Les arbres de classification font partie des modèles discriminantes.

La première apparition des arbres été en 1984, et par la suite des approches récentes comme les arbres renforcés, et les forêts aléatoires, ont montré une grande efficacité, et concurrent la performance des algorithmes de base [116]. Les techniques de ce type d'arbres, les forêts aléatoires, sont assez proches des méthodes plus traditionnelles d'Analyse Discriminante, de Classification, de Tests Non-Paramétriques et d'Estimation Non-Linéaire. Généralement les arbres de classification sont connus par leurs flexibilité, et une option analytique très attrayante, mais qui ne doit toutefois pas remplacer complètement les méthodes traditionnelles.

L'implémentation des FA repose sur la méthodologie de l'arbre des décisions, où l'objectif principal est la correction des lacunes de la méthode CART initiale [117], comme la sensibilité des arbres uniques à l'ordre des prédicteurs, en calculant un ensemble de K arbres partiellement indépendants.

Les étapes poursuivies pour implémenter les FA sont ordonnées comme suit :

- La première phase consiste à créer k nouveaux ensemble d'apprentissage par un double processus d'échantillonnage, où :
  - ✓ La technique *Bootstrap* [118] est la première à réaliser, et qui permet de créer un échantillonnage sur les observations, en utilisant un tirage avec remise d'un nombre N d'observations identique à celui des données d'origine.
  - ✓ Par la suite processus d'échantillonnage sur les p prédicteurs est essentiel, en n'en retenant qu'un échantillon de cardinal m$\sqrt{}$(p).
- Dans La deuxième phase nous entraînons un arbre de décision, sur chaque échantillon, selon une des techniques connues, et en limitant sa croissance par validation croisée.
- La troisième étape consiste à stocker les k prédictions de la variable d'intérêt pour chaque observation d'origine.
- Finalement l'appel du vote majoritaire, décrit dans la section suivante, est essentiel pour La prédiction de la forêt aléatoire.

Les corrections portées sur les arbres des décisions naïves perdent l'aspect visuel des arbres aux FA, ce qui représente un inconvénient à ce type d'arbre, mais n'empêche pas sa large utilisation dans le domaine de la fouille des donnes et son efficacité à empêcher l'enjeu du sur-apprentissage [115]. Les forêts aléatoires sont également très difficiles à battre en termes de

performances, et dans l'ensemble, elles forment un outil (principalement) rapide, simple et flexible, mais avec certaines limitations [119] .

2. Classification hybride

Les techniques hybrides (ou la combinaison de classifieurs) sont liées au principe Boosting [120], qui se base se base sur des modèles qui permettent la liaison entre les méthodes ML : discriminatives et génératifs. Généralement les techniques hybrides sont souvent utilisées pour améliorer les performances des classifieurs instables ou faibles. Plusieurs algorithmes stimulant les performances de la catégorisation ont été proposés dans la littérature, et selon le type d'information de sortie des classifieurs, nous choisissons le type du combinateur, et vu la flexibilité et les meilleurs résultats obtenus dans notre domaine d'étude, nous soulignons les techniques du Vote. [121] défini un ensemble des règles qui permettent la combinaison réussite des classifieurs choisis, comme *: la somme, le produit, la règle du maximum, la règle du minimum et la moyenne*. Ces règles renforcent la sortie des classifieurs en votant sur la classe la plus adéquate.

Le combinateur Vote Majoritaire (VM) est une des techniques du vote les plus optimal, caractérisé par son haut niveau de précision et de robustesse [27]. La méthode du VM trouve la sortie de classe de chaque classificateur et la comptabilise comme un vote pour cette classe, et affecte le modèle d'entrée aux classes avec vote majoritaire, c.à.d., la classe qui reçoit le plus grand nombre de vote est sélectionnée comme décision majoritaire.

En effet, supposons pour *C* classifieurs à combiner pour décider dans un problème de classification à *R* classes, le vote majoritaire a une formule mathématiquement comme suit [122] :

Assigner X_i à la classe C_j si :

$$\sum_{i=1}^{C} \Delta_{ji} = \max_{k=1,..,m} \sum_{i=1}^{C} \Delta_{ki} \tag{15}$$

Où :

$$\Delta_{ki} = \begin{cases} 1 \quad si \quad P(c_k \mid x_i) = \max\limits_{j=1,...,m} P(c_j \mid x_i) \\ 0 \; sinon \end{cases} \tag{16}$$

Une variante entraînable est le vote à la majorité pondérée. Les votes sont multipliés par un poids qui est souvent obtenu en estimant les précisions des classificateurs sur un ensemble de validation. Une sélection possible est :

$$W_i = \text{Log}\,(p_i/\,(1 - p_i)) \tag{17}$$

Où : p_i est la précision du i^{ème} classifieur.

Cette sélection de poids garantit une erreur minimale pour le vote majoritaire pondéré lorsque les sorties sont indépendantes.

Pour juger la qualité des méthodes de classification, quelle que soit leurs types, une variété des mesures sont réservé pour l'évaluation.

3. Évaluation des classifieurs

Selon la taille des données employés, deux manières pour évaluer la performance des classifieurs sont disponibles :

- ✓ La première est appliquée en cas de données en masse, consiste à réserver 80% de l'ensemble de données pour l'apprentissage et les 20% restantes pour le test.
- ✓ La deuxième manière est connue par la validation croisée (ou k- validation croisée) [123], qui est souvent utilisée en cas de données de petite taille, elle consiste à diviser l'ensemble de données original en k parties, une partie est réservée au test et les k-1 parties restantes effectuent l'apprentissage. L'opération se répète k fois de telle sorte que chaque partie a servi exactement une fois comme ensemble de test. Ainsi, La performance du classifieur est la moyenne des performances obtenues dans les k exécutions.

3.1 Matrice de confusion

Après avoir choisir la manière d'apprentissage et du test du classifieur, nous faisons appel à aux mesures de la qualité des classifieurs. La plupart des mesures de performance de la classification sont construites à partir de la matrice de confusion qui reporte les prédictions correctes et incorrectes des exemples de chaque classe. Le tableau à deux dimension, tableau 2, présente la matrice de confusion pour une classification binaire.

Tableau 2: Modèle binaire de la Matrice de confusion.

		Données prédites	
		Positif	**Négatif**
Données observées	**Positif**	**True Positive (TP)**	**False Negative (FN)**
	Négatif	**False Positive (FP)**	**TrueNegative (TN)**

Où :

TP (True Positive) : le nombre d'instances de la classe 1 correctement classifiées ;
FN (False Negative) : le nombre d'instances de la classe 1 incorrectement classifiées ;
FP (False Positive) : le nombre d'instances de la classe 2 incorrectement classifiées ;
TN (TrueNegative) : le nombre d'instances de la classe 2 correctement classifiées ;

Il est fondamental de savoir que plus que la Matrice de Confusion (MC) est une matrice creuse plus que le classifieur est efficace. En revanche les éléments qui composent la MC produisent

autres mesures qualitatives des systèmes de classification, comme présentera la section suivante.

3.2 Formules des mesures d'évaluation des classifieurs

Un ensemble des mesures classique, d'évaluation des classifieurs des documents, permettent aux développeurs de visualiser la qualité des systèmes développés. Dans cette partie on mentionne :

- La précision : est le nombre de documents correctement attribués à la classe mère C_i rapporté au nombre de documents total proposé pour la classe C_i. Cette mesure définie la puissance prédictive des classifieurs en employant la formule suivante :

$$\text{Precision} = \text{PPV} = \frac{TP}{TP+FP} \tag{18}$$

- Sensibilité/spécificité évalue l'efficacité des classifieurs sur une seule classe en estimant la probabilité que les prédictions positives/négatives soient vraies :

$$\text{Rappel} = \text{Sensibilité} = \frac{TN}{TP+FN} \tag{19}$$

$$\text{Spécificité} = \frac{TN}{TP+FP} \tag{20}$$

- La F-Mesure est une mesure composite combine la précision et le rappel privilégiant une par rapport à l'autre en paramétrant le réel $\boldsymbol{\beta} > 0$. Une valeur inférieure à 1 favorise le rappel et vice versa.

$$\text{F-Mesure} = \frac{(\beta^2+1)*\text{Precision}*\text{Recall}}{\beta^2*\text{Precision}+\text{Recall}} \tag{21}$$

- La performance est la mesure la plus utilisée pour évaluer la performance globale des classifieurs

$$\text{Performance} = \frac{TP+TN}{TP+FP+TN+FN} \tag{22}$$

3.3 Fonction d'efficacité du récepteur (Courbe ROC)

La courbe ROC (Receiver operating characteristic) dite aussi caractéristique de performance ou courbe sensibilité /spécificité, défini comme fonction d'efficacité des classifieurs dans le cas de classification. Le graphique de la fonction ROC est représenté souvent une courbe qui donne le taux de vrais positifs (fraction des positifs qui sont effectivement détectés) en fonction du taux de faux positifs (fraction des négatifs qui sont incorrectement détectés) afin de visualiser les performances d'un modèle de classification pour tous les seuils de classification. L'allure de

la courbe ressemble à celle de la figure 13 et chaque forme représente l'état du test des classifieurs, où plus que la courbe a des valeurs élevées, plus le classifieur fait moins de fautes.

Autre mesure dérivée de la courbe ROC a prouvé son profit pour l'évaluation de la performance d'un système de classification et qui est connu par l'AUC (aire sous la courbe ROC). Les valeurs d'AUC sont comprises dans une plage de 0 à 1. Un modèle dont 100 % des prédictions sont erronées à un AUC de valeur 0. Si toutes ses prédictions sont correctes, son AUC est de 1.

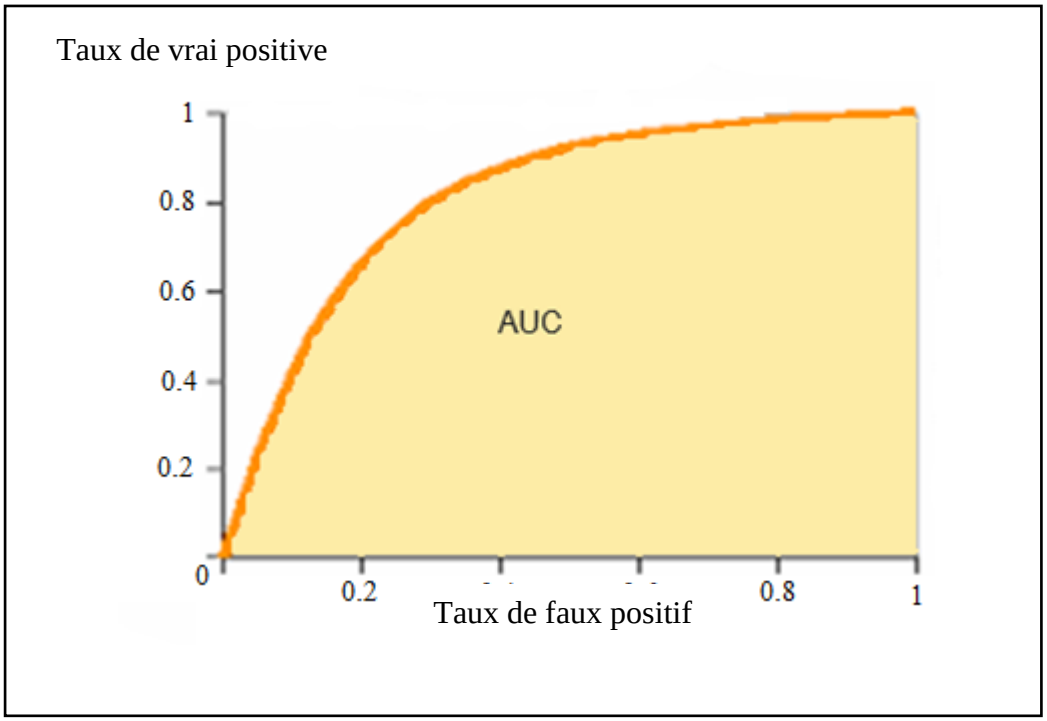

Figure 12: Exemple de la courbe ROC [124].

En outre le score AUC mesure la qualité du classement des prédictions, plutôt que leurs valeurs absolues quel que soit le seuil de classification sélectionné. Donc elle est meilleure pour l'estimation de la capacité de rejet du système, et tant que l'aire sous la courbe est grande, l'allure ROC converge vers le parfait ce qui signifier un excellent fonctionnement du système adopté pour la classification.

L'ensemble des algorithmes de classification et les moyens d'évaluation, discutés au-dessus, sont utiles pour la proposition et la construction des architectures pour des systèmes de classification robustes, comme nous verrons dans les sections suivantes.

IV. Conclusion

Ce chapitre présente un aperçu global sur le contexte général de notre thèse. De plus, l'état de l'art proposée dévoile l'usage Text Mining avec les bases de sa fondation, l'ensembles de ses processus, ainsi que ses applications innovantes, afin que la machine traite d'une manière automatique la langue naturelle, en effet, les applications du Text Mining se déversent et

généralement nous devons distinguer entre le traitement du langage naturel, la recherche et la récupération de l'informations (IR), le regroupement de documents, la classification des documents, le Web Mining, l'extraction de l'information, et l'extraction des concepts, comme domaines majeurs du Text Mining. Aussi l'apprentissage artificiel a participé à l'évolution des disciplines du traitement de l'information électronique, et particulièrement au développement des applications du texte Mining.

En effet, notre objective est de proposer des systèmes de classification qui permettent de classifier selon des classes multiples un corpus des données textuelles non structurées, en employant l'ensemble des classifieurs présenté dans ce chapitre. Pour cela, dans ce chapitre nous avons présenté l'architecture générale des systèmes de classification populaire, qui se base principalement sur les processus du Texte Mining, présentés dans la première partie du chapitre, et dont le dernier processus de prise de décision utilise les classifieurs d'AA. Aussi, nous avons exposé un ensemble des mesures de performances sur lesquels nous pouvons juger la performance des classifieurs, or le choix du meilleur classifieur dépend du type de l'application ainsi que le type des données traité. En revanche, pour avoir les meilleures performances des systèmes de classification et pour assurer la fiabilité et la sécurité de ces systèmes, il est recommandé d'utiliser les techniques d'hybridation des classifieurs ML. Généralement, la classification hybride a un ensemble des caractéristiques, comme il est discuté dans ce chapitre et le Vote majoritaire est une des techniques préconisées.

Dans les chapitres suivants nous proposons une collection des systèmes de classification des données de type non structurées, qui suivent les processus du Texte Mining et où le vote d'un ensemble des classifieurs est appliqué pour la tâche de catégorisation. En outre, notre objectif est de réaliser une étude comparative pratique sur ces systèmes, et de conclure l'impact de chaque processus de la classification textuelle sur sa performance.

Chapitre 2 : Systèmes de catégorisation multi-classes, comparaison et approche probabiliste.

Chapitre 2 : Systèmes de catégorisation multi-classes, comparaison et approche probabiliste

I. Introduction

La croissance importante des données électroniques, sur le web, rend l'accès à l'information de plus en plus difficile et la recherche des connaissances pertinentes, contenues dans une base de données documentaire, plus couteux. Par conséquent, des systèmes robustes tels que les systèmes d'analyse des sentiments, et la recherche d'information, ont réussi à résoudre ce problème. En outre, la tâche de classification a été officiellement identifiée comme l'une des meilleures solutions pour analyser le contenu utile à partir des documents et développer les systèmes d'extraction des connaissances [125].

En effet, un ensemble de processus contrôle l'efficacité des systèmes de classification, comme le prétraitement [126] et la représentation numérique du corpus [127]. Le premier processus permet une génération du vocabulaire sur lequel nous nous basons pour produire l'une des types de représentations existantes des descripteurs, quoi que ça soit du type booléen [18], vectoriel [20] ou probabiliste [19].

Pourtant, l'objective des contributions, présentés dans ce chapitre, est de conclure l'impact de génération qualitative des vecteurs descripteurs sur les applications Text Mining. Dernièrement, les méthodes du plongement lexical, par exemple, Word2Vec [72] et Glove [23] (décrites dans le premier chapitre), sont récemment suggérés comme représentations vectorielles des mots, appliqué pour distribuer la vectorisation des documents. Comme indiqué dans le premier chapitre de cette thèse, le Word2vec ou tout autre modèle similaire présente chaque terme du document par un vecteur, ce qui produit des descripteurs de grande taille.

Ainsi, pour résoudre le problème du descripteur de taille massive, le modèle neuronal paragraph2vec (ou doc2vec) a été implémenté pour générer un vecteur représentatif pour le texte complet [28]. Cette contribution a marqué un progrès significatif dans le domaine du traitement automatique des Textes, et notre contribution porte sur la visualisation de l'impact des paramètres neuronaux sur la qualité de la vectorisation ainsi que la classification automatique (et multi-classe) des données non-structurées.

Avant de discuter l'impact de la vectorisation sur la performance de la catégorisation multiple des textes, nous proposons une étude comparative entre différents systèmes de classification des textes. En effet, nous comparons un ensemble des méthodes et des algorithmes utilisés pour les différents processus qui composent l'architecture globale des systèmes de catégorisation multiple. À base des expérimentations et des résultats obtenus, nous avons pu extraire un ensemble de remarques et des obstacles qu'un système de catégorisation peut affronter. Aussi, nous avons constaté que la vectorisation des grandes masses des données et la génération des descripteurs pertinents et de taille optimale, pour des classifieurs sensible aux entrées (comme le PMC), reste toujours un conflit qui nécessite de nouvelles approches. Pour cela, notre dernière contribution, présentés dans ce chapitre, porte sur le développent d'un nouveau système de classification neuronale, qui intègre une nouvelle approche de pondération probabiliste, afin de composer un descripteur de tailles optimal et pratique pour un certain type de classifieurs.

Le reste de ce chapitre est constitué de trois grandes parties cohérentes, où nous commençons par l'analyse et la comparaison des systèmes de la recherche et de la catégorisation de l'information, afin de présenter l'ensemble des facteurs qui aident à atteindre des systèmes robustes. Par la suite, nous nous focalisons sur l'impact de la vectorisation neuronal et l'ajustement de ces paramètres neuronaux, sur la précision de la catégorisation. Finalement, nous suggérons une partie qui décrit la nouvelle approche probabiliste pour la vectorisation et la catégorisation neuronale des données non-structurées.

II. Systèmes de la recherche et d'analyse de l'information textuelle basés sur l'apprentissage artificiel analyse et comparaison

Actuellement plusieurs recherches ont été élaborées dans le domaine de la recherche d'informations (RI) pour construire des systèmes de décision très performants, afin de faciliter l'accès à l'information. En revanche, une des tâches d'exploration de texte pertinent est la classification des documents, où un contrôle de catégorisation de contenu reste utile dans de nombreuses applications du Text Mining (voir chapitre 1). Dans ce contexte, deux problèmes majeurs se posent :

- La détermination des informations pertinentes, à savoir la sélection des caractéristiques représentatives dans les documents.
- La sélection des meilleurs classifieurs parmi les plus connus.

Dans cette partie, nous nous focalisons sur la présentation et la comparaison des systèmes d'analyse et d'apprentissage statistique des données textuelles brutes afin de déterminer ses lacunes et améliorer les performances de la tâche de la catégorisation des textes. Cela a, aussi, un impact direct sur la libération de l'information pertinente.

1. Utilité des systèmes de classification

L'architecture proposé ressemble à un certain point aux architectures des systèmes de la recherche de l'information. Les systèmes de recherche d'information, en général, consiste à extraire l'information pertinente demandé à travers une requête/ demande exprimer en langage naturel par exemple la figure 14. Lorsque la base des connaissances, d'où en extrait la réponse de la requête, et bien organisé et classé la réponse est devenue de plus en plus pertinente. Dans le cas traditionnel le processus Matching est réalisé grâce aux formules de similarité [128].

Le pré-traitement des données est aussi nécessaire pour la tâche de RI, et il subit le même traitement dont nous avons parlé dans les processus Text Mining (dans le premier chapitre de cette thèse). Aussi la représentation vectorielle des deux composante connaissance (= corpus) et la demande (requête) est nécessaire comme la Figure 13 indique.

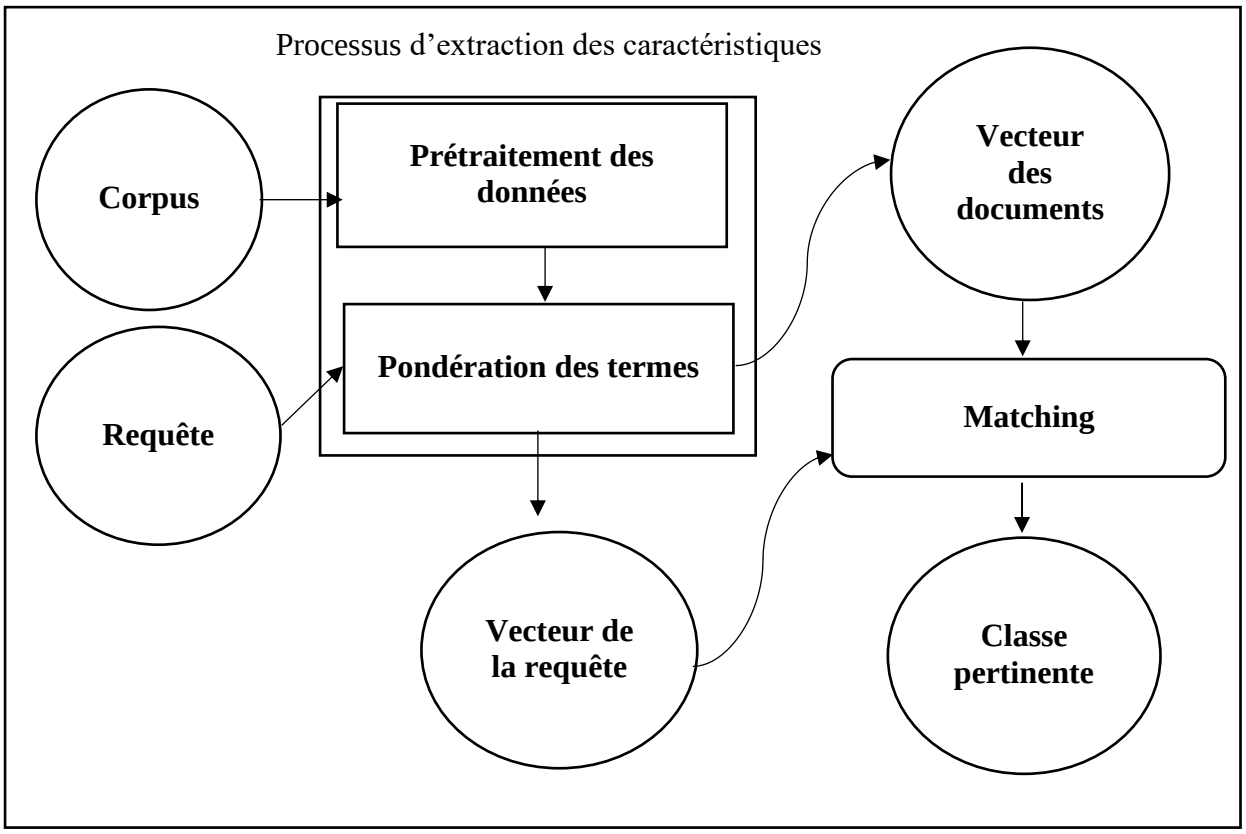

Figure 13: Architecture des systèmes classique de la recherche de l'information.

Dans la nouvelle architecture, proposé dans la Figure 14, une fois la demande reçue, nous appelons le processus d'extraction des caractéristiques. Ce dernier est représenté, comme entrée du classifieur qui donne la classe la plus appropriée comme réponse. Alors le Matching dans la nouvelle proposition est réalisé grâce aux algorithmes de l'apprentissage statique.

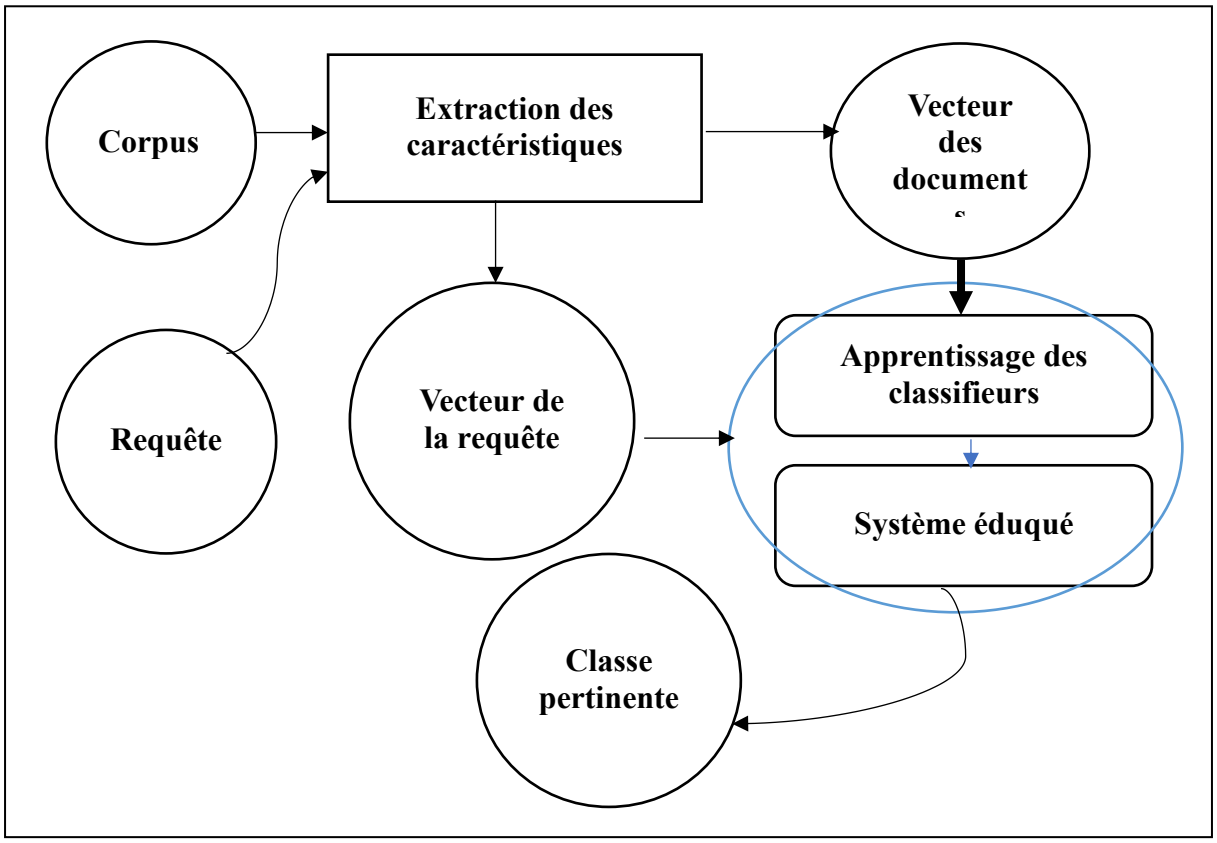

Figure 15: : Architecture des systèmes classique de la recherche de l'information adoptée.

En revanche, nous bénéficions des deux tâches décisionnelles, en appliquant bien sûr le processus Text Mining, pour classifier des documents bruts et extraire simultanément l'information pertinente à travers la classification de la requête. La requête, dans ce système, est inclus dans les données réservées au test des classifieurs, ce qui permet de visualiser son appartenance à la classe réponse adéquate.

Pour avoir les meilleures méthodes du classification garanti l'efficacité du système de recherche d'information (RI) dans des contextes textuels. Pour cela une étude comparative entre l'utilisation des différentes composantes du système de catégorisation a été mis an œuvres. En outre l'ensemble des résultats et remarques obtenus seront afficher et discuter par la suite.

2. Comparaison des systèmes d'analyse et de catégorisation multi-classes des textes

Le but de cette comparaison est de choisir les meilleurs algorithmes pour certain processus de l'architecture, présenté dans la figure 5 du chapitre 1, qui serve à la catégorisation des textes [129] [16].

2.1 Base des données

La classification selon des étiquettes multiple, spécialement lorsque le nombre des classes est supérieur à 3, organise et différencie bien les données, ce qui facilite l'accès aux informations correctes et évite la confusion dans la plupart des cas. Parmi les données qui doivent être

distinguées selon plusieurs catégories nous trouvons les données médias (les nouvelles ou les news), où une meilleure classification de ce type de donnes facilite l'archivage, l'accès à l'information pertinent, et une bonne recommandation des nouvelles. L'entreprise British Broadcasting Corporation (BBC) mis en disposition des bases des nouvelles exprimées en langage nature, classifiés selon N-classe, ce qui peut servir aux tests des systèmes de catégorisation multiple des textes.

La première base que nous avons exploitée, pour l'analyse comparative de nos systèmes, est appelé BBCNews [108] et qui contient 2225 documents classifiés selon 5 classes (business, Entertainment, politiques, sport, et technologie).

La deuxième base de référence, nous employons la multi- label data BBCSport [108], qui comprend 737 documents, également organisés selon 5 classes prédictives (athlétique, cricket, football, rugby, tennis).

2.2 Prétraitement des données

Le prétraitement des données cherche à éliminer les données bruyantes, qui n'ont pas d'utilité et qui occupent la mémoire. D'ailleurs cette phase assure la conservation minimale des donnes sans perdre de l'information, et parmi ses outils la radicalisation des termes présentait dans le chapitre 1. Afin de choisir l'algorithme optimal, nous comparons un ensemble des modèles Stemmer comme : *Lovins Stemmer*, *Iterated Lovins Stemmer*, *Snowball et Snowball Stemmer*, utilisant les deux bases citées précédemment, et nous démontrons leurs influences sur la qualité de la classification.

2.3 Outils de la classification

Pour comparer entre différents systèmes de classification basés sur différents classifieurs, nous avons utilisé comme matériel : un PC Dell compatible, Intel (R) Core i5-CPU 2,50 GHz et 4 Go de RAM. Nous indiquons que notre implémentation est basée sur le langage portable java.

Les classifieurs employés pour la conception des systèmes sont : SVM, les Forêts Aléatoires (FA), Naïve Bayes complémentaire (NBMU), Naïve Bayes (BN), et la régression logistique (RLS). Afin de renforcer la performance de la classification, nous proposons par la suite une combinaison adéquate de l'ensemble de ces classifieurs par l'utilisation de la fonction Vote. Il convient de noter que 80% des données employés seront réserver pour l'apprentissage des classifieurs et 20% pour le test. En revanche, les mesures de performance qualifiée pour juger les systèmes sont : la précision, le rappel (Recall), la performance (Accuracy), et la courbe ROC. (Voir le descriptive de chaque algorithme et les mesures de performance dans le chapitre 2).

2.4 Résultats de l'étude comparative et analyse

Le tableau 3 présente les résultats de classification des données BBCNews, basé sur un ensemble des classifieurs avec le changement des méthodes de radicalisation (à savoir : Lovins Stemmer, Iterated Lovins Stemmer et SnowBall stemmer) pour chaque classifieur. Nous déclarons aussi que la représentation vectorielle utilisée, dans ce cas, est la TF-IDF.

Tableau 3: Comparaison des différents systèmes de classification en employant différents stemmers et classifieurs.

Stemmers	**Lovins**	**Iterated Lovins**	**snowball**
Classifieurs	**performance (%)**	**performance (%)**	**performance (%)**
NB	96.3	95.95	96.71
FA	95.77	95.82	96.31
MNBU	97.43	97.30	97.57
RLS	96.94	97.03	96.04
SVM	97.84	97.70	97.66
Vote	97.97	97.90	97.52

Comme prévu, les systèmes basés sur le vote ont les meilleurs taux de reconnaissance. Ainsi, nous comparons l'efficacité des stemmers utilisées, en se basant uniquement sur la dernière ligne du tableau 3, nous remarquons que le meilleur système de classification est celui qui se base sur : Lovin (comme stemmer) + le Vote (comme classifieur), avec un taux de **97,97%.**

Une autre façon d'évaluer la performance des systèmes de catégorisation est la matrice de confusion, qui enregistre les classes des données observées et prédites. A cet égard, la matrice de confusions du système basé sur le stemmer Lovin + le classifieur Vote est donnée par le tableau 4.

Tableau 4: Matrice de confusion du système de classification basé sur le stemmer Lovin et le classifieur Vote.

Classes	**Business**	**Entertainment**	**Politics**	**Sport**	**Tech**
Business	495	2	7	0	6
Entertainment	2	374	4	0	6
Politics	6	3	406	1	1
Sport	1	0	1	509	0
Technology	1	3	1	0	369

Compte tenu de la grande taille de la base de données, nous pouvons dire que cette matrice de confusion est une matrice creuse. En fait, le pourcentage de documents mal classés est inférieur à 2,5%.

Passant à la deuxième base du test, qui est la BBCSport, le tableau 5 montre une comparaison entre différents classifieurs, cités par avant, combiné avec l'ensemble des caractéristiques obtenues par les 3 stemmers.

Tableau 5: Comparaison des différents systèmes de classification en employant BBCSport data, différents Stemmers et classifieurs.

Stemmers	**Lovins**	**Iterated Lovins**	**SnowBall**
Classifieurs	**Performance (%)**	**Performance (%)**	**Performance (%)**
NB	97.82	97.96	98.10
FA	93.35	98.23	94.02
MNBU	98.77	98.37	99.32
RLS	97.96	98.23	97.96
SVM	98.77	98.64	98.50
Vote	99.18	99.05	99.32

Aussi, les systèmes basés sur le classifieur vote ont les meilleurs taux de reconnaissance. Ainsi, nous comparons l'efficacité des stemmers utilisées, en se basant uniquement sur la dernière ligne du tableau 5. En déduit alors que le meilleur système de classification est celui qui emploi comme stemmer Snow Ball et le classifieur Vote, où la performance de ce système est de 99,32%. De plus, le tableau 6 donne la matrice de confusion du système élu performant.

Tableau 6: Matrice de confusion du système de classification basé sur le stemmer Snow Ball et le classifieur Vote.

Classes	**athletics**	**Cricket**	**Football**	**Rugby**	**tennis**
athletics	100	0	1	0	0
cricket	0	122	2	0	0
Football	0	1	264	0	0
Rugby	0	0	0	147	0
tennis	0	0	1	0	99

Nous voyons que la matrice de confusion est une matrice creuse, où le nombre des « 0 » indique la qualité de la classification.

Aussi les courbes ROC générés pour cette étude (en employant les deux bases des données) montre une grande efficacité des classifieurs choisis pour l'insertion des textes dans des catégories prédéfinies. Or, la figure 16 illustre l'allure de la courbe ROC pour le système de classification qui utilise le vote comme classifieur, le stemmer Snow Ball et la base des données BBCNews. Ainsi, le reste des courbes pour les autre systèmes sont présents dans notre article [129].

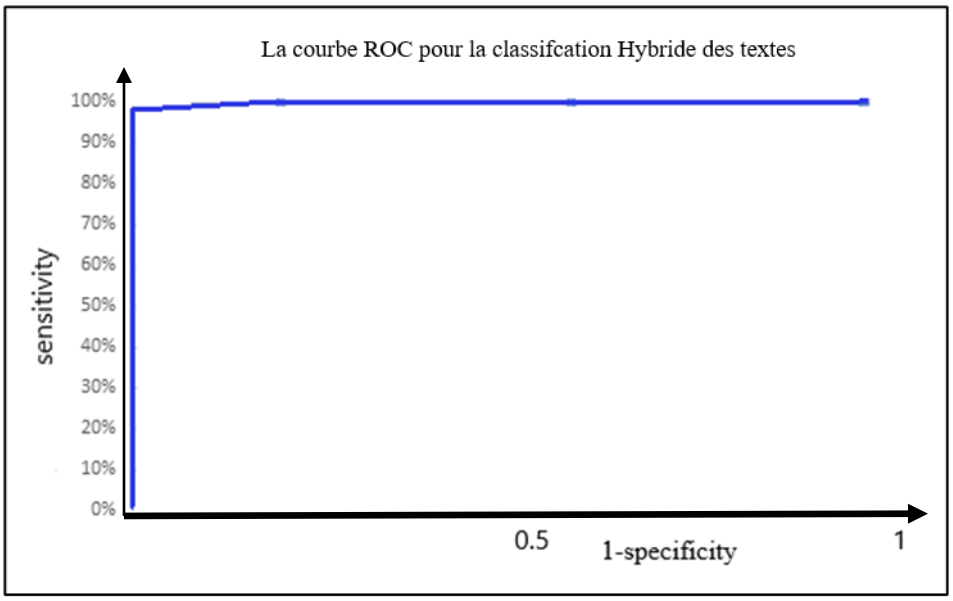

Figure 16 : Courbe ROC du classifieur Vote.

Dans cette première contribution, nous avons comparé plusieurs systèmes de classification des données non-structurées, qui peuvent conduire à un système de recherche d'informations dans des textes, c'est à dire, si nous considérons la requête comme fichier texte, parmi les données du test des classifieurs, la réponse sera la classe pertinente. Certainement un posttraitement et recommandé afin de livrer la réponse la plus précise.

D'ailleurs l'examen comparatif proposé relève l'importance de chaque processus de la classification des données textes en général. Alors, nous pouvons penser que le processus de représentation aura aussi un impact sur la performance de la catégorisation. Pour prouver cette influence notre prochaine suggestion confirme l'impact de la représentation vectorielle sur la catégorisation des données non-structurées.

III. Impact de la vectorisation sur la précision de la catégorisation des textes

L'étude comparative, réalisé dans la première partie de ce chapitre, démontre l'importance de chaque processus des systèmes d'analyse, afin d'avoir des performances excellentes de ces systèmes. Alors, après le prétraitement des corpus et la sélection d'un indexe représentative, nous devons sélectionner la meilleure technique de vectorisation et de sélection de variables pertinentes, en analysant l'impact de cette dernière sur la précision de la catégorisation souhaitée. Dans cette section, nous intéresserons à la méthode neuronale du plongement lexical (Word Embedding) que nous avons bien défini par avant. Nous proposons une comparaison structurée entre les modèles de la vectorisation neuronale (Doc2vec) et leurs impacts sur la classification textuelle. En effet, nos études indiquent que de nombreuses caractéristiques neuronales influencent la performance du système de classification [202]. En outre, nous essayons de prouver que les paramètres du réseau de neurones (Doc2vec), comme le nombre d'époque et la taille du vecteur de mot, doivent être ajustés pour améliorer les performances de

la représentation la Mémoire Distribuée du Vecteur de Paragraphe (PV-DM) (détaillée dans le chapitre Etat de l'art sur le Text Mining).

Après la génération de la matrice descriptive, en pratiquant la variation PV-DM du modèle Doc2vec, nous appelons des classifieurs spécifiques de l'AA tels que l'SVM, Logistic Function et le réseau de neurones Feed-Forward supervisé, sans oublier la combinaison de ces classifieur par le Vote majoritaire [202].

1. Plongement Lexical (PL) analyse et comparaison

1.1 Architecture du Plongement Lexical adoptée

Comme mentionner dans le premier chapitre, les méthodes du Plongement Lexical (PL), par exemple, Word2Vec, FastText [130] et Glove, sont largement suggérées pour les représentations vectorielles de mots, appliqués pour distribuer la représentation du document. Word2Vec ou tout autre modèle similaire du Plongement Lexical (PL) présente chaque terme du document par un vecteur, ce qui produit des descripteurs de grande taille. Ainsi, pour résoudre le problème du descripteur de taille massive, Paragraph2Vec ou Doc2Vec a été implémenté pour générer un vecteur représentatif pour un texte complet [28]. En effet, Doc2vec est un réseau neuronal artificiel, contrôlé par un ensemble de paramètres qui régit les caractéristiques du vecteur de sortie.

La figure 17 présente la version du doc2vec adopté dans notre contribution qui est la Mémoire Distribuée du Vecteur de Paragraphe (PV-DM). Cette version s'appuie sur l'architecture CBOW du word2vec ; mais au lieu d'utiliser uniquement des mots proches pour prédire le mot, ils ont également ajouté un autre vecteur de caractéristiques (noté par paragraphe ID), qui est unique au document. Les entrées du réseau de neurone PV-DM sont constituées de :

- Vecteurs de mots W, où chaque mot a un vecteur représentative unique de dimension 1*V ;
- Vecteurs d'identification de document D, où le vecteur d'identification de document a une dimension de 1*C ; avec : C = le nombre total de documents.

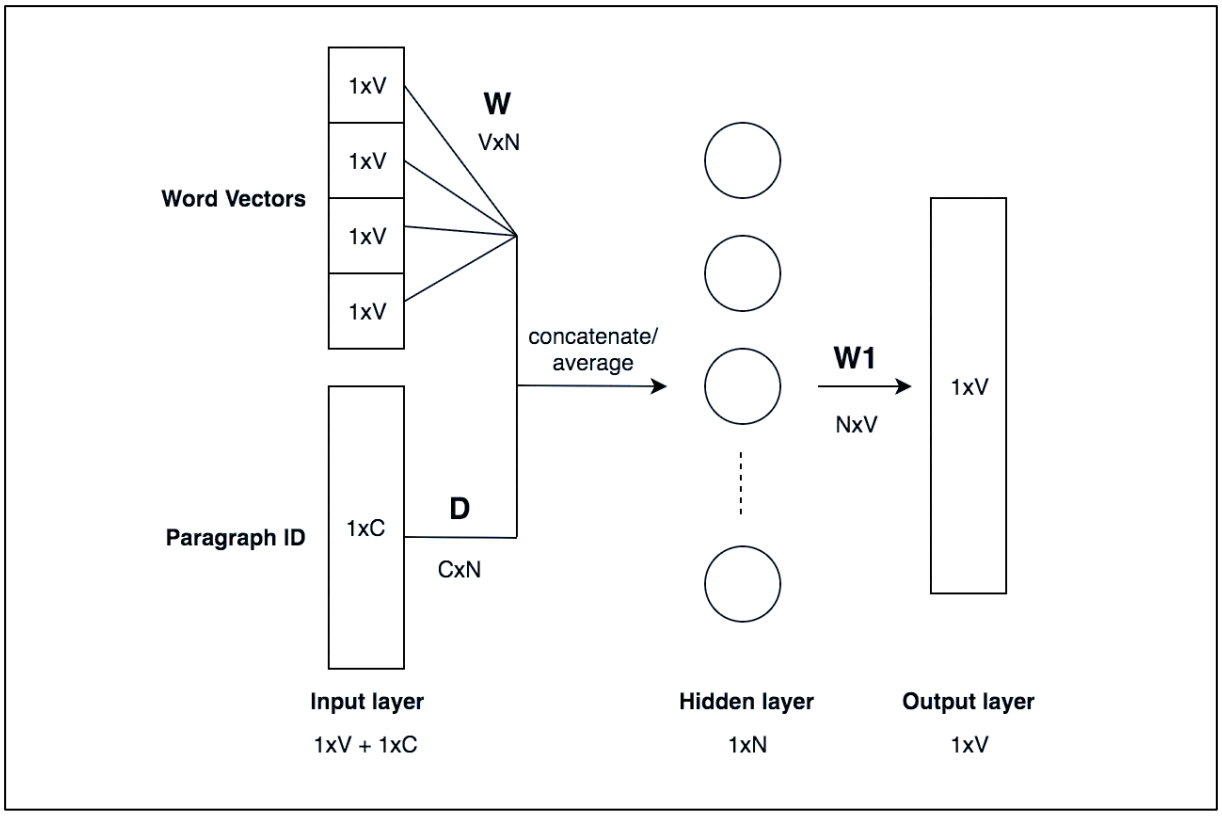

Figure 17: Architecture Doc2vec adoptée (PV-DM).

Dans la couche cachée la matrice de poids W est de dimension V*N et la matrice des poids D est de dimension C*N. Lors de l'apprentissage des vecteurs de mots W, le vecteur de document D est également entraîné et, à la fin de l'apprentissage, nous aurons une représentation numérique du document complet. Comme le PV-DM est un réseau de neurone, alors il existe différents paramètres qui le contrôle et dans ce qui suit, nous citons et nous analysons ces paramètres.

1.2 Analyse des paramètres neuronaux du Doc2vec

Comme mentionner par avant, Doc2vec a deux architectures, c'est-à-dire, la version Sac de Mots distribués par Vecteur de Paragraphe (PV-DBOW) et la version de mémoire distribuée par vecteur de paragraphe (PV-DM) [28]. En raison de son efficacité dans le contexte connexe, nos systèmes comparatifs utilisent la représentation neuronale PV-DM [131]. Généralement, le fonctionnement du modèle PV-DM nécessite l'ajout d'un vecteur de document d'identification combiné à un vecteur de mot pour chaque mot du paragraphe. Ainsi, pour combiner des vecteurs, il est essentiel d'appliquer une de ces méthodes : l'addition, la concaténation ou la méthode standard [28]. Le choix du processus du fusionnement affecte la qualité de la représentation et de la classification, comme montrera la partie expérimentation de cette partie. Également, les performances de la représentation PV-DM dépendent de trois paramètres principaux :

- La méthode de combinaison utilisée, dans la couche de projection du modèle,

- Le nombre d'Epoch, où un Epoch correspond à un apprentissage sur toutes les données, plus ce nombre est grand plus nous obtenons les bonnes précisions, mais cela tarde le temps de repense des systèmes.
- La taille du vecteur.

Le choix des trois paramètres est utile pour diminuer la complexité, minimiser le temps de réponse du système et la gestion de la mémoire [202].

Notre objective est de trouver une représentation neuronale pour analyser les avis des clients du site commercial Amazon et classifier les news de la base des données BBCNews. Le vote majoritaire sera le classifieur adopté afin de produire un système sécurisé et fiable.

2. Comparaison des systèmes de classification multi-classes basée sur PL

Dans cette contribution, l'architecture du système de classification est présente dans la figure 19, où le processus flache à discuter est celui de la représentation vectorielle qui permet de fournir l'entrée convenable (Matrix Embedding) à un ensemble des classifieurs du ML.

Nous rappelons que l'objectif du travail est de vérifier la performance des classifieurs artificiel, en changeant les paramètres du réseau de neurone Doc2vec cités par avant. Dans les sous paragraphes suivants, nous décrivons en détail les composantes du système adopté.

2.1 Système de classification adopté

Nous commençons notre expérimentation par l'étape de prétraitement des données. Le système présent dans la figure 18 applique le même processus prétraitement de la contribution précédente, où nous avons : éliminé les mots vides, et utilisé le stemmer Snow Ball qui est vérifié performant dans la première partie du chapitre. Ce procédé permet une amélioration considérable de la phase de classification. Ensuite, la matrice des poids (Embedding Matrice) présente des entrées d'une sélection de classifieurs d'apprentissage automatique.

Chaque classifieur a ses paramètres, qui permettent l'amélioration de la performance de la classification. Dans notre cas nous utilisons le noyau polynomial comme fonction de noyau pour le modèle SVM.

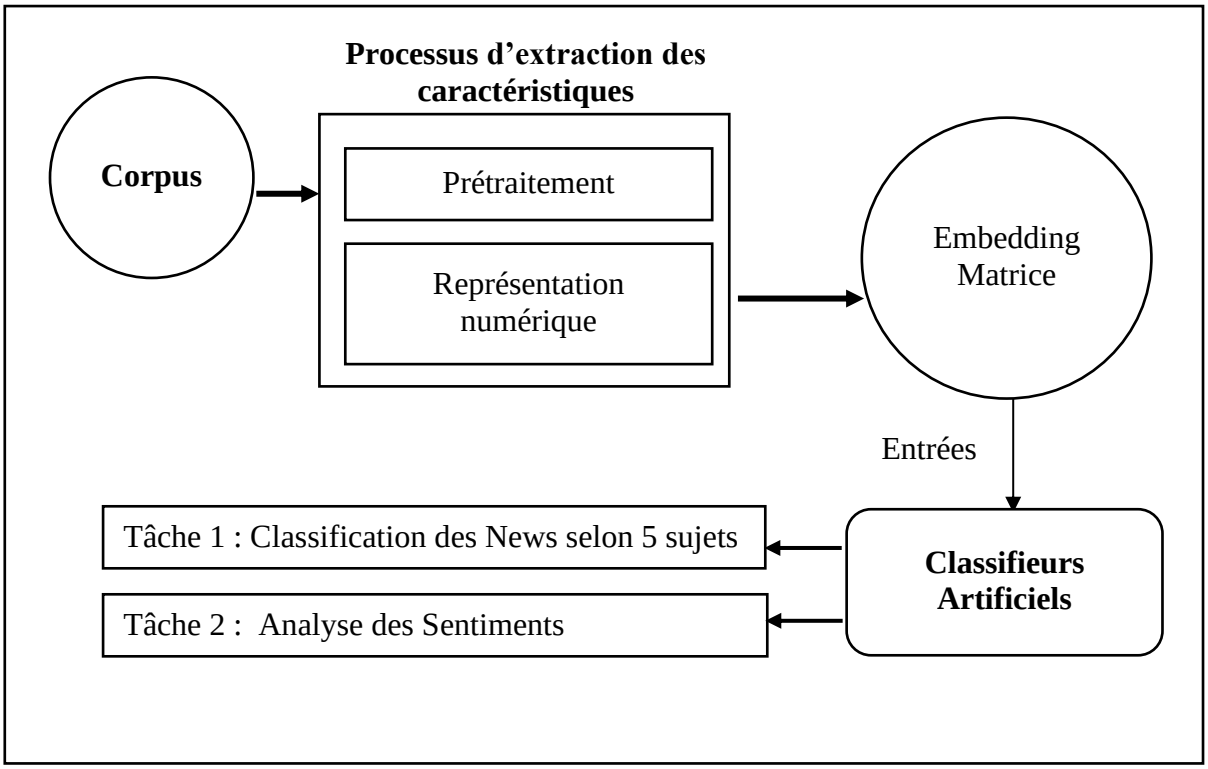

Figure 18: Architecture du système de catégorisation adoptée.

Pour classer avec le réseau de neurones Feed-Forward (FNN), nous testons avec la bibliothèque DL4J, où le Soft Max est la fonction d'activation et le numéro des Epoch est fixé sur 10. Afin de combiner l'ensemble des classifieurs utilisés, nous utilisons comme hybridation des classifieurs artificiel la technique du vote avec les fonctions combinatoires majoritaires.

Base des données

Les deux bases de données de références que nous avons utilisées pour réaliser ce travail sont :

- BBCSport en tant que données multi-étiquetées, qui contient des actualités sportives classées dans cinq classes prédéfinies. Le fichier CSV, associé à BBCSport, est composé de deux colonnes (News et Classes).
- Pour l'analyse des sentiments commercial, nous utilisons les données d'Amazon (disponibles sur le site Web de Kaggle), qui contient 4002 avis des clients d'Amazon, classées selon les critiques positives ou négatives.

2.2 Résultats de l'étude comparative

La validation croisée à k blocs (K-cross validation) est employée pour le préciser les blocs des données qui seront employer pour le test et l'entrainement des modèles ML. Cette technique permet d'éviter un ensemble des problèmes d'apprentissage des classifieur comme le problème de surapprentissage [132].

Dans notre cas, nous utilisons 10-validation croisée pour apprendre et tester les classifieurs utilisés lors de la phase de classification de nos systèmes.

2.3 Classification des données à étiquettes multiples Résultats et discussion

En utilisant les données d'actualité (BBCSport), les tableaux (7, 8, 9 et 10) affichent les résultats des systèmes de classification basés sur la représentation PV-DM, et un ensemble de classifieurs (SVM, FNN et fonction logistique FL), combinées, dans ce qui suit, par le vote majoritaire. La comparaison est basée sur la variation des composantes des trois paramètres primaires :

- Le type du modèle employé dans la couche de projection du réseau PV-DM. (Ajouter, Concaténer ou calculer la moyenne).
- Le nombre d'époque.
- La taille vectorielle du mot.

Tableau 7: Résultats de classification utilisant la représentation PV-DM (avec numéro d'époque = 1 et taille du vecteur= 100), un ensemble de méthodes de classifieurs.

	Epoch Number=1 et taille du vecteur=100								
	Doc2vec_Average			Doc2vec_Add			Doc2vec_concat		
	P%	R%	A %	P %	R%	P%	P %	R%	P%
SVM	90.8	90.8	90.77	91.0	90.6	90.6	75	74.4	74.3
LF	91.9	91.7	92.1	92.1	92.1	76	75.1	75.3	75.3
FNN	89.4	89.1	98.1	90.3	90.4	90.3	75.6	75.4	75
VoteMaj	90	90	90	94	94	94	76.7	75.3	75.3

Tableau 8: Résultats de classification utilisant la représentation PV-DM (avec numéro d'époque = 5 et taille du vecteur= 100) et un ensemble des classifieurs.

	Epoch Number=5 et taille du vecteur =100								
	Doc2vec_Average			Doc2vec_Add			Doc2vec_concat		
	P%	R%	A%	P%	R%	A %	P%	R%	A%
SVM	97.5	97.4	97.4	96.4	96.2	96.2	91.5	91.4	91.4
LF	96.1	96.1	96	91.2	91.2	91.1	82.3	82.9	82.9
FNN	97.3	97.3	97.2	97.2	97.1	97.1	90.6	90.7	90.6
Maj vote	96.2	96.2	96.2	92.4	92.3	92.2	89.9	89	89

Tableau 9: Résultats de classification utilisant la représentation PV-DM (avec numéro d'époque = 1 et taille du vecteur= 300), un ensemble de classifieurs.

	Epoch Number= 1 et Taille du vecteur= 300								
	Doc2vec_Average			Doc2vec_Add			Doc2vec_concat		
	P%	R%	A%	P%	R%	A%	P%	R%	A%
SVM	89.1	89	89	86.4	86.4	86.4	75 .1	74.9	74.9
LF	91.9	91.7	92.1	92.1	92.1	76	75.1	75.3	75.3
FNN	88.3	88.2	88.1	86	85.9	85.5	74.2	74.4	74.2
Majvote	94.3	94.3	92.4	92.4	92.3	92.3	78	78	78

Tableau 10: Résultats de classification utilisant la représentation PV-DM (avec numéro d'époque = 5 et taille du vecteur= 300), un ensemble de classifieurs.

	Epoch Number= 5 et Taille du vecteur= 300								
	Doc2vec_Average			Doc2vec_Add			Doc2vec_concat		
	P %	R%	A %	P%	R%	A %	P%	R%	A%
SVM	97.7	97.7	97.6	97	97	97	91	90	90
LF	98.1	98.1	98.1	97.7	97.6	97.6	79.6	79.6	80
FNN	98.1	98.1	98.1	98.1	96	96	90	90	90
Maj vote	98.1	98.1	98.1	97	97	97	91	90	90

Les paramètres mentionnés, au-dessus ont un impact significatif sur la dimension des entrées des classifieurs et du temps de réponse des systèmes. Les mesures de performance employée sont la précision (P), Rappel (R) et la performance (A).

Nous remarquons, à partir des résultats affichés dans les tableaux ci-dessus, que la modification des paramètres de la méthode du PL modifier les performances de la classification. Les tableaux et la figure 19 montrent la faiblesse de la méthode de concaténation à présenter les meilleures performances de la classification. Pour cette raison elle est éliminée dans cette comparaison. Par ailleurs, l'architecture PV-DM, qui utilise la méthode de la moyenne en phase de projection, donne des résultats plus intéressants. En outre, un nombre d'époque optimal permet de réduire le nombre d'itérations, ce qui produit une petite complexité d'algorithme.

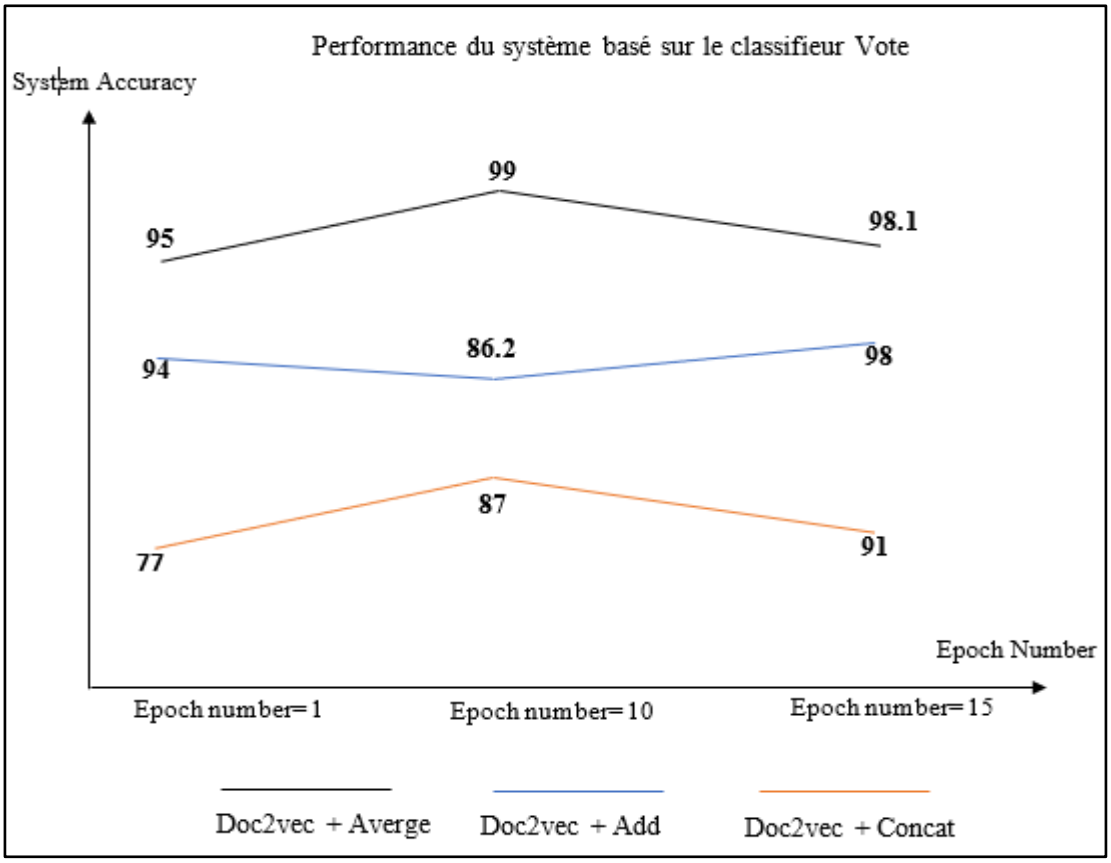

Figure 19: Précision du vote basée sur la variation des paramètres PV-DM (avec une taille du vecteurfixe = 100, et une modification du nombre d'époques).

La figure 19 montre que le nombre d'époques a une influence importante sur les performances de la représentation neuronal et la classification. Les matrices de confusion présentée, dans les tableaux 11 et 12 prouvent qu'un bon choix du numéro d'époque, réduit le taux de faux positifs, de 0,025 à 0,002, dans la phase de classification.

Tableau 11 : Matrice de confusion du système basé sur la représentation PV-DM avec un bon choix du nombre des Epoch, en employant la base des données BBCSport.

C 1	C 2	C 3	C 4	C 5	
101	0	0	0	0	**Class 1**
0	123	1	0	0	**Class 2**
2	0	263	0	0	**Class 3**
1	0	1	145	0	**Class 4**
0	0	0	0	100	**Class 5**

Tableau 12:Matrice de confusion du système basé sur la représentation PV-DM avec un mauvais choix du nombre Epoch pour la base des données BBCSport

C 1	C 2	C 3	C 4	C 5	
99	1	0	0	1	**Class 1**
1	116	1	6	0	**Class 2**
3	1	247	8	6	**Class 3**
2	6	13	123	3	**Class 4**
5	1	2	2	90	**Class5**

Vue les résultats présentées au-dessous, nous somme focaliser sur l'utilisation de la méthode de la moyenne dans la phase de projection du réseau PV-DM. Par la suite nous proposons de visualiser l'importance du choix optimal des paramètres nombre d'Epoch et la taille du vecteur, afin de présenter un système de catégorisation d'une complexité minimal. Pour cela, le tableau 13 résume les résultats de la classification des données multi-étiquetées à l'aide du classifieur vote et la représentation PV-DM, en variant ces paramètres (la taille du vecteur et nombre d'Epoch).

Comme indiqué, le système basé sur :

- ✓ La représentation PV-DM avec :
- Méthode de concaténation= la moyenne,
- Une taille du vecteur= 100
- Et un numéro d'époque = 10
- ✓ Le vote majoritaire,

Pour classifier des données multi étiquetés, présente une excellente précision égale à **99%.**

Tableau 13 : Résumé des résultats de la classification des données multi-étiquetées à l'aide du classifieur vote et en variant les paramètres de la PV-DM.

Paramètres	**Les performances des systèmes basés sur le classifieur Vote**
Taille du vecteur=100 et Nombre des Epochs=1	93%
Taille du vecteur= 100 et Nombre des Epochs= 10	99%
Taille du vecteur= 100 et Nombre des Epochs=15	98.1%
Taille du vecteur=300 et Nombre des Epochs=5	97.6%
Taille du vecteur= 300et Nombre des Epochs=10	98.8%
Taille du vecteur= 300et Nombre des Epochs=15	99.1%
Taille du vecteur= 500et Nombre des Epochs=6	97.6%

En revanche, le choix de la taille du vecteur = 100 permet un temps de réponse optimal, par rapport au choix de la taille = 300. En outre, une taille du vecteur minimal assure une réduction significative de la dimensionnalité des caractéristiques.

2.4 Classification des données de sentiment commercial d'Amazon

Le célèbre concours « Kaggle » a lancé le défi d'analyser un million des avis des clients d'Amazon, comme contenu pertinent pour développer les demandes du commerce électronique. Pour confirmer l'impact des paramètres neuronaux sur la représentation et la classification des opinions commercial, nous sommes limités à 4002 commentaires de la base de données Amazon. En pratiquant toujours la méthode de la moyenne, pour la version PV-DM du doc2vec, nous suggérons une variété de valeurs pour le reste des paramètres, c'est-à-dire la taille du vecteur et le numéro d'époque.

Les résultats affichés sur le tableau 14 prouve que la classification ou l'analyse des avis des clients d'Amazon, doit suivre l'architecture du système de classification standard. Ainsi, ils prouvent qu'un bon choix des paramètres du réseau neuronal PV-DM influence la précision de l'analyse des sentiments.

Tableau 14: Résultats de classification des données Amazon à l'aide du classifieur Vote, changeant les paramètres de la PV-DM

Paramètres	Les performances des systèmes basés sur le classifieur Vote
Taille du vecteur=1000 et Nombre des Epochs=10	90%
Taille du vecteur= 1000et Nombre des Epochs=15	82%
Taille du vecteur=300 et Nombre des Epochs=10	90%
Taille du vecteur=100 et Nombre des Epochs=5	94%
Taille du vecteur=100 et Nombre des Epochs=10	92%
Taille du vecteur=500 et Nombre des Epochs=10	82%

Tableau 15: Matrice de confusion avec un bon choix des paramètres du Doc2vec.

Positif	**Negatif**	
1803	195	**Positif**
192	1808	**Negatif**

Tableau 16: Matrice de confusion avec un mauvais choix des paramètres du Doc2vec.

Positif	**Negatif**	
1685	316	**Positf**
312	1689	**Negatif**

En incluant les matrices de confusion (Tableaux 15 et 16), nous démontrons la transition des avis clients vers la classe appropriée, ce qui réduit le taux de faux positifs lorsque, par exemple, le choix du membre d'époque est réussi. Aussi la courbe ci-dessous (voir la figure 20) résume

l’évolution du système de classification basé sur le vote lorsque le choix des paramètres de la représentation neuronal est parfait.

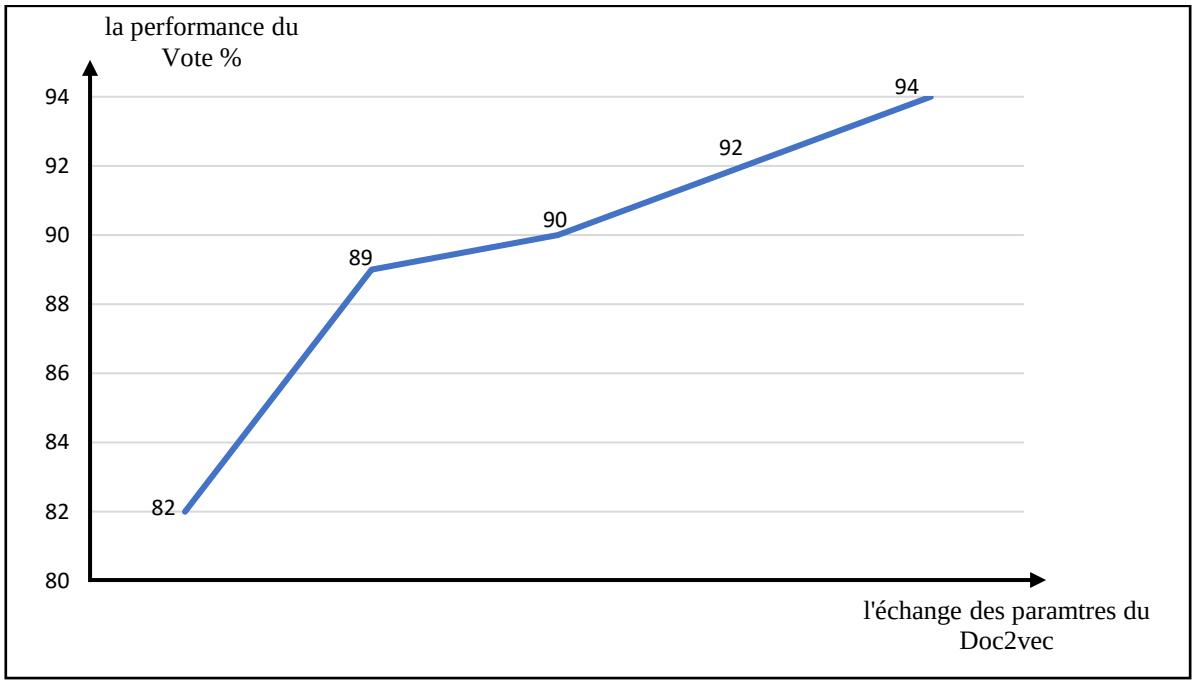

Figure 20: Courbe d'évolution de la performance du vote, appliquant plusieurs paramètres neuronaux PV-DM.

À partir des exemples donnés, nous avons illustré l'impact des caractéristiques neuronales de doc2vec sur la qualité de la classification des documents, en tant qu’application du Text Mining. De plus, une sélection optimale de ces paramètres diminue la complexité des algorithmes utilisés, le temps de réponse du système et de déploiement ainsi que l'optimisation de la mémoire. Comme illustre le tableau 14, 100 vecteurs sont suffisants pour présenter un corpus de 4000 instances.

Il convient de noter qu'il n'y a pas de cas général pour le choix des paramètres ; chaque base de données nécessite sa propre étude.

Les deux premières parties de ce chapitre démontrent que les classifieurs artificiels exigent des descripteurs de haute qualité pour présenter des hautes performances, au niveau du traitement des données non-structurées. D’ailleurs, la vectorisation des données exprimées en langage naturel et la réduction de dimensionnalité des vecteurs descripteurs doivent respectés certains critères, afin d’empêcher la perte de l’information, que les méthodes de sélection des variables peuvent provoquer, et de surmonter le problème du surapprentissage des classifieurs. Comme exemple de classifieur qui nécessite un descripteur performant et de taille optimale est le perceptron multi couche, où les résultats de classification de ce classifieur sont absents dans l'étude comparative citée par avant ; ainsi que les approches classiques ne répondent pas aux exigences de la catégorisation neuronale des textes. Par conséquent, nous proposons une nouvelle approche, dans la partie suivante qui serve ce type de systèmes de catégorisation.

IV. Approche probabiliste pour la vectorisation et la catégorisation neuronale

Parmi les limites détectées, dans le Text Mining, est la masse des bases des données textuelles et leur représentation dans des grands espaces. Cela, empêche l'automatisation du traitement données volumineuse et l'extraction des connaissances. À cet égard les méthodes et les techniques de l'exploration de textes sont souvent utilisés, afin de relever les défis de la haute dimensionnalité [76]. La méthode de pondération TF-IDF (fréquence du terme-la fréquence inverse du terme) est l'approche la plus requise pour représenter le document. Malheureusement, la TF-IDF produit des descripteurs de grande taille, ce qui nécessite des modèles d'une grande complexité [133]. Par conséquent, les systèmes de classification des textes basés sur les modèles ML souffrent du phénomène de surapprentissage et deviennent très lents. Du coup, pour surmonter ces problèmes, nous utilisons les méthodes de sélection des attributs, comme phase importante, dans le processus du système adopté. Pourtant, en donnant l'aspect déterministe de cette dernière, nous risquons une perte énorme de l'information [25] [134].

1. Préambule

Notre proposition consiste à introduire un système de classification, testé sur une base des données multi classes, en employant l'architecture présente dans la figure 21. L'architecture adopté suit les processus majeurs de la fouille des textes, afin de contribuer à résoudre le problème provoqué par les méthodes de sélection des attributs, et à améliorer la performance du système de catégorisation, en appliquant les réseaux de neurones artificiels dans la phase décisionnel.

Figure 21: Les processus des applications du Text Mining.

Comme première étape du système, nous votons pour un bon choix du type de descripteur, où le mot n'est pas la seule façon de décrire un document. Il existe de nombreuses formes de descripteurs sous forme des radicaux et des n-gramme [73]. Généralement, le choix du descripteur dépend de la nature de la langue traitée, et il a un rôle crucial dans la pertinence des résultats [95] [24].

Le nouveau système se base sur un bon choix des vecteur descripteurs, après un bon prétraitement des données, afin de relever les défis de la haute dimensionnalité. Pour cela, les méthodes de sélection des attributs sont employées pour avoir des descripteurs de taille optimale, en identifiant les attributs pertinents et informatifs ; ainsi que l'amélioration de la précision des algorithmes de classification.

Pour récupérer l'information perdues, que les méthodes de la sélection des attributs peuvent provoquer, nous proposons une nouvelle vectorisation probabiliste des documents, à base des termes sélectionnés pertinents. Les nouveaux vecteurs sont composés des poids calculés par une fonction probabiliste extraite du modèle BM25 [84]. Généralement, la BM25 est une fonction qui se base sur des évènements probabilistes pour réaliser le processus d'appariement dans les systèmes de la recherche d'information. Ce modèle permet de définir un score de pertinence pour un document D ' S(D)' par rapport à une requête. Ce score est le rapport des deux probabilités :

(1) P(R/D) probabilité de trouver une information pertinente,

(2) P(NR/D) probabilité de trouver une information non pertinente, en employant les termes d'une requête exprimer en langage naturel.

Autrement dit la BM25 permet de calculer la probabilité d'importance des termes requête dans les documents d'un corpus, afin d'afficher les documents les plus pertinents pour la requête.

Projetant la démarche BM25 sur notre cas d'étude, la fonction utilisée pour calculer les nouveaux poids des termes sélectionnés pertinent Ti, par la méthode de sélection employé, détermine l'importance de chaque terme Ti dans le document Dj à base des deux probabilités :

- P_{ti} (R/Dj) : la probabilité pour que le terme Ti soit pertinent (R) ou important pour représenter le document Dj
- et P_{ti} (NR/Dj) la probabilité pour que le terme Ti ne soit pas pertinent (NR).

Le modèle 2-Poisson appliqué en BM25 permet les transformations nécessaires des deux probabilités (1) et (2) afin de générer la fonction score de pertinence du document D. Également dans notre cas la loi 2- Poisson [136] est appliqué dans le but de représenter la distribution des termes dans les documents comme un mélange de deux distributions de Poisson : l'une représentant la fréquence des termes pertinents pour décrire le document, l'autre celle des termes non-pertinents.

La fonction du poids W_{ij}, qui sera présentée dans la section méthodologie, calcule le poids composé de coefficients probabilistes locaux et globaux de chaque terme t_i élu pertinent et permet de composé le vecteur descripteur de chaque document D_j du corpus.

Plus spécifiquement et précisément, les formules des composants sont composées par la fréquence de chaque descripteur, la longueur de chaque document et la taille du corpus.

Généralement, les probabilités sont un outil naturel pour essayer de quantifier l'incertitude. Par conséquent, le choix du modèle probabiliste revient à sa capacité d'étudier l'information incertaine qu'on la touche dans la représentation d'un document, ou dans la représentation des caractéristiques.

Pour montrer la performance de ce traitement, nous proposons des études comparatives entre la représentation TF-IDF et la nouvelle représentation probabiliste, pour classer un corpus de référence de multi-classes.

De plus, dans la phase de classification, nous utilisons plusieurs versions du réseau bayésien et le perceptron multicouches, et en général les résultats obtenus, en termes de performance, sont excellentes.

Les prochaines sections de ce chapitre sont organisées comme suivant : nous commençons par la présentation de la méthodologie qui contiendra l'approche populaire afin de comprendre la différence avec l'approche adoptée. Les processus et les outils de notre système seront discutés aussi. Aussi, les résultats des systèmes comparés seront affichés par la suite et discutés, avant de conclure.

2. Méthodologie et algorithme de pondération probabiliste

2.1 Approche probabiliste pour la catégorisation neuronale

Comme il était évoqué dans la dernière partie du deuxième chapitre, la catégorisation des textes est une des tâches d'exploration des données, dont ses processus sont notamment :

(1) Le prétraitement,
(2) La représentation numérique des données et la génération de la matrice descriptrice,
(3) La réduction de la dimensionnalité si nécessaire,
(4) Et finalement l'appel des classifieurs.

Dans l'approche populaire, la pondération TF-IDF est utilisée pour pondérer et générer des vecteurs représentatifs de grande taille. Cela influence le rendement de certains classifieurs, même si nous faisons appel aux méthodes de sélection des attributs [69].

En revanche, les résultats affichés, dans l'analyse comparative présente dans la première partie de ce chapitre, exclus les réseaux de neurones PMC vue les résultats médiocres que nous avons constatés lors de nos expérimentations. Suite aussi à l'impact de la vectorisation des données sur la classification, en général, notre nouvelle proposition porte également sur la génération des nouveaux poids aux termes, élus pertinents, pour un document. Cependant, nous avons proposé que la préparation de notre corpus pour réaliser une catégorisation neuronale, doit avoir un traitement plus puissant et efficace.

Dans cette partie, nous proposons une nouvelle approche pour sélectionner les caractéristiques d'un document donné en tenant compte du contexte. Notre approche sélectionne un ensemble de caractéristiques probabilistes, dont les composantes sont des scores probabilistes. En fait, nous allons suivre trois principales étapes :

- Dans la première étape, nous utilisons la TF-IDF pour transformer le corpus en termes de bases matricielles.
- Dans la deuxième étape, la sélection d'un ensemble de termes pertinents notés B est réalisé par les méthodes SA.
- Enfin, pour chaque document, nous calculons les poids probabilistes pour chaque élément parmi B.

L'architecture du système adopté sera détaillée dans la section suivante.

2.2 Architecture du système de catégorisation basée sur la nouvelle vectorisation probabiliste

Afin de réaliser une catégorisation multiple des documents, à base de l'approche probabiliste pour la génération du descripteur d'entrée des algorithmes artificiels employés, nous soulignons un ensemble de processus figuré dans l'architecture illustrée dans la figure 22.

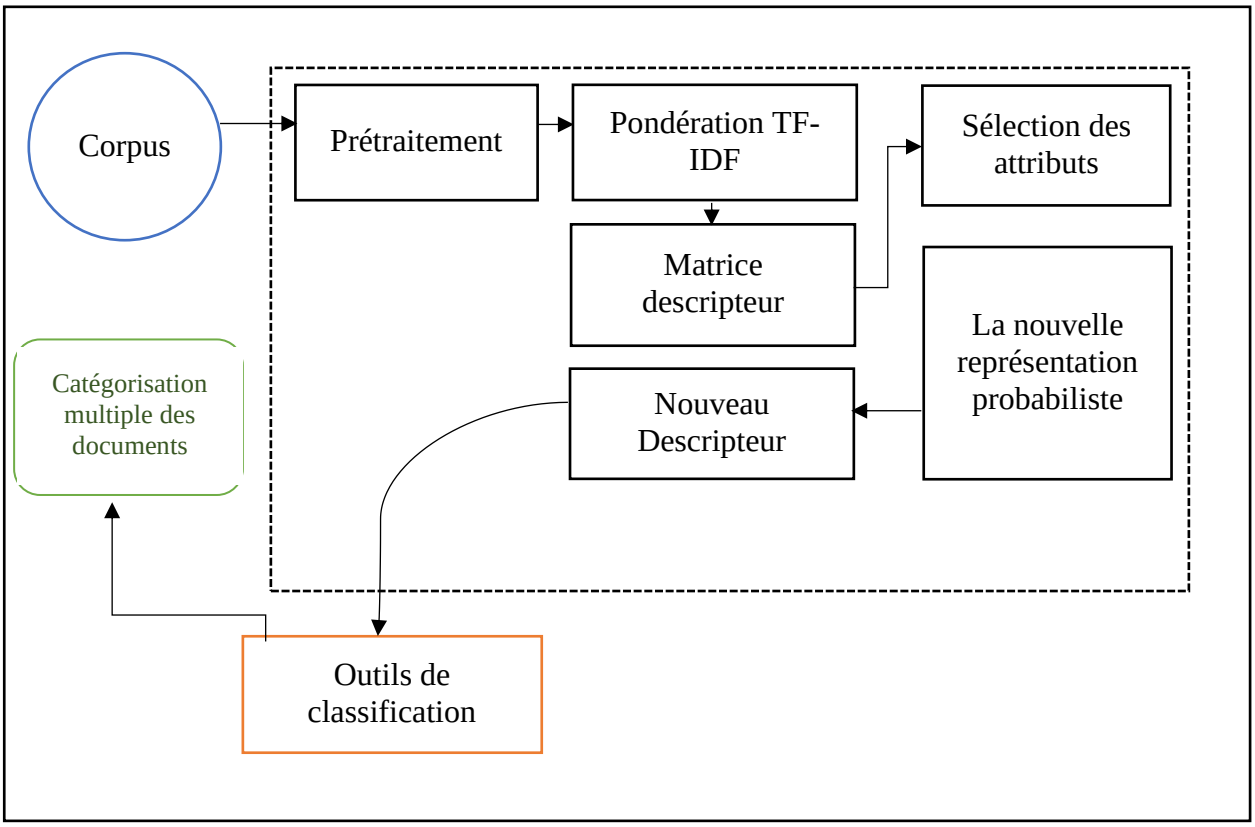

Figure 22: Système adopté pour la catégorisation multiple des textes en employant la nouvelle vectorisation probabiliste.

2.2.1 Prétraitement

La première phase du système adopté, est le prétraitement des données, où nous avons :

- Éliminé les mots vides, en employant une liste des mots vides.
- Gardé les racines des mots, en appliquant le Lovins comme algorithme de racinisation (qui a montré son efficacité dans l'étude comparative des systèmes de classification citée dans la première partie du chapitre).

Cette étape aidera à la libération de la mémoire et à la sélection des éléments pertinents du corpus.

2.2.2 Sélection des Attributs (SA)

Un problème central de l'apprentissage machine consisté à identifier un ensemble significatif de caractéristiques, à partir desquelles nous pouvons construire un modèle de classification pour une tâche particulière. Dans notre système deux tâches fondamentales ont été réalisés :

- Dans un premier temps la représentation vectorielle du corpus a été mis en œuvre, en employant la pondération TF-IDF. En revanche, la matrice descriptrice de base (notée par M), est généré avec un grand nombre d'attributs ce qui empêchera le fonctionnement du classifieur.
- La deuxième tâche consiste à sélectionner un nombre optimal des attributs, qui seront les plus importants pour représenter le contexte de la base étudié ; cependant une méthode de

sélection des attributs est appliquée. Dans notre cas nous avons utilisé la méthode populaire qui sélectionne les sous-ensembles de caractéristiques basée sur la corrélation (CFS).

La CFS calcule la corrélation entre chaque terme de l'ensemble des termes obtenus en utilisant TF-IDF. La section (II. 3.3.3) du premier chapitre décrit le fonctionnement de la méthode CFS. La CFS est connue par son efficacité, sa simplicité, et sa rapidité d'exécution [76] pour cela elle est choisie.

2.2.3 Algorithme de pondération probabiliste proposée

Compte tenu de l'impact significatif de la représentation vectorielle sur la qualité de la catégorisation des textes bruts, relevé par l'étude comparative citée dans les premières parties de ce chapitre, une nouvelle version de la RV probabiliste a été produite [69]. La méthode introduite permet de créer des descripteurs pertinents et de fonder un système de catégorisation neuronale de documents robuste.

Le modèle probabiliste est la base de notre nouvelle représentation vectorielle probabiliste, où les poids qui composent les vecteurs descripteurs sont calculés par une fonction de racine probabilistes. Alors, pour pondérer les termes sélectionnés pertinents (par les algorithmes de SA) nous nous basons sur le traitement de l'informations incertaine. Les méthodes SA peuvent provoquer une perte d'information, dont le choix du meilleur attribut est infructueux. Cependant, la nouvelle pondération calcule le rapport des probabilités :

- P_{ti} (R/ D_j), que chaque terme sélectionné soit représentatif, important, ou pertinent pour un document.
- P_{ti} (NR/ D_j), que le terme sélectionné ne soit pas important pour représenter un document ;

Et produit la fonction du poids W_{ij} exprimé dans l'équation 24. Cette équation indique le poids du terme élu pertinent t_i dans le document d_j, prenant en considération : la longueur du document, la longueur moyenne des documents et des constantes BM25 d'ajustement [19].

Soit T = $\{t_1, t_2 \dots t_n\}$ l'ensemble des termes obtenus, par l'application de la CFS. Pour chaque élément Dj de notre corpus et t_i de T, un vecteur $W_{i,j} = \{W_{1,j}, \dots, W_{n,j}\}$ est calculé par la formule :

$$W_{i,j} = \mathrm{Log}\left(\frac{N}{\mathrm{df_{ij}} + \varepsilon}\right) * \frac{(\mathrm{K}+1)\mathrm{df_{ij}}}{\frac{\mathrm{k}(1-\mathrm{b})+\mathrm{b}*\mathrm{k}*\mathrm{dl_j}}{\mathrm{avgl}}} + \mathrm{tf_{ij}} \qquad (23)$$

La fonction du poids présentée est composée de deux types de paramètres :

- Paramètres locaux

➢ $\mathrm{df_{ij}}$: Fréquence du descripteur i dans le document j ; pour éviter le calcul de redondance, nous conservons le df_{ij} calculé par TF-IDF ;

➢ La fréquence $\mathrm{df_{ij}}$ est initialisée par $0 < \varepsilon << 1$ pour tenir compte du cas dfij=0 ;

- dl_j : Nombre des descripteurs dans le document j.
    - Paramètres globaux
- N : Nombre de documents dans la collection ;
- avgl : Nombre moyen de descripteurs dans le document j ;
- Constantes K généralement fixé autour de 1,2 à 2
- b = 1 est la fréquence relative (pleine échelle en fonction de la longueur du document); elle est généralement fixée autour de 0.75.

Algorithme : Pondération Probabiliste

```
-Entrées : Fichier de la Matrice des fréquences des termes sélectionnés par CFS
-Sorties : Fichier de la Matrice des poids probabiliste
Variables
/* Dlj Nombre des descripteurs dans le document j */
/*N : Nombre de documents dans la collection*/
N : réel
Dlj : réel
K : réel
B : réel
W : réel
/*Nombre moyen de descripteurs dans le document j*/
avgl : réel
/* nombre des i = n, j= m dans la matrice d'entrée Matrice [j][i]*/
Début
b ← 1
K← 1.2
Tant que (non fin fichier) faire
Pour I←1 à n Faire
 Pour J←1 à m Faire
Lire (Matrice (I, J))
/* */
Dfij ← valMatrice (I,J)
x←N/ (dfij+1)
R ← Log (x)
F← ((K+1)*df_ij)(( (K+1)*df_ij)/avgl)) + df_ij
Wij ← R*F
/*insérer dans la nouvelle matrice probabiliste les nouveaux poids Wij*/
Matrice probabiliste (I,J) ← Wij
Fin pour
```

Fin pour
Retourne (Matrice probabiliste (I,J))
Fin tant que
Fin Pondération Probabiliste

L'ensemble des termes obtenus, appelés caractéristiques (features), est représenté comme entrée de plusieurs outils de classification, afin de classer chaque document du corpus dans la classe correcte.

2.2.4 Outils de catégorisation neuronal

Après le processus d'extraction des caractéristiques, nous avons obtenu un nouveau descripteur du corpus, qui est divisé en trois ensembles, qui sont, l'ensemble d'apprentissage, l'ensemble de validation et l'ensemble de test. Le premier et le deuxième ensemble sont utilisés pour sélectionner le modèle approprié. La performance du modèle obtenu est testée sur la base du troisième ensemble de test. Dans ce travail, nous utilisons le réseau bayésien et le PMC comme classifieurs.

a) Classification bayésienne

Notre objectif global est d'associer une ou plusieurs classes à un document, ce qui en facilite l'exploitation. Étant donné qu'il est rapide et facile à mettre en œuvre, la classification bayésienne est souvent utilisée et recommandée pour la classification des textes.

En revanche, un classifieur naïf bayésien, comme mentionner dans le chapitre apprentissage artificiel, est un type de classification linéaire qui peut être définie comme une simplification des réseaux bayésiens, et cette dernière a de nombreux dérivés tels que le complément naïf Bayésien. Dans cet ouvrage, nous comparons les trois méthodes de recherche locales qui permettent de sélectionner l'architecture du réseau bayésien et qui ont donné une meilleure classification, à savoir :

- **Hill Climbing (HC)** : consiste à ajouter, supprimer et inverser des arcs, où dans chaque itération, HC conserve la meilleure architecture [19].
- **K2** : Contrairement à la méthode HC, l'algorithme d'apprentissage K2 du réseau Bayes utilise un ordre fixe de variables pour ajouter des arcs. Cependant, le réseau commence comme un réseau de Bayes naïf et continue à partir de là. Il permet également à l'utilisateur de désigner le nombre maximum de parents que chaque nœud peut avoir [19].
- **TabuSearch** : définit la notion de voisinage et initialise la recherche par un ensemble de solutions. Ainsi, pour éviter la production d'un même ensemble de solutions, cette méthode utilise un ensemble de mouvements interdits [19].

b) Catégorisation neuronale à base du PMC

Le PMC est un réseau neuronal artificiel orienté, avec apprentissage supervisé, utilisé pour modéliser n fonctions non linéaires, où pour toute fonction f : IR^n -> IR^p non linéaire est

modélisée à l'aide du PMC. Pour classifier notre corpus, nous devons déterminer trois types de couches :

a. La couche d'entrée : elle est composée de n neurones et c'est une couche virtuelle associée à l'entrée du système. Cela signifie que le nombre de termes sélectionnés est de 68, et que le nombre de neurones de la couche d'entrée est de 68.
b. Couche cachée : un perceptron multicouche peut avoir un nombre quelconque de couches cachées et un nombre quelconque de neurones par couche cachée, mais il convient de noter que l'utilisation d'une couche cachée est suffisante pour résoudre un problème complexe non linéaire et le choix du nombre de couches cachées est toujours un problème difficile. Différentes approches pour la sélection de neurones cachés est proposée dans la littérature [137] [138]. Comme la taille du corpus utilisé, BBCSPORT, est 731, nous utilisons la méthode de validation simple pour sélectionner le nombre de neurones de la couche cachée.
c. Couche de sortie : cette couche est appelée couche de décision et contient p neurones. Comme le corpus BBCSPORT est composé de 5 classes prédictives, d'où le nombre de neurones de décision est 5. Il est essentiel de noter que les neurones du PMC sont reliés entre eux par des connexions. La figure 23 donne l'architecture du perceptron multicouche adopté.

Les poids de ces connexions régissent le fonctionnement du réseau et programment une application de l'espace d'entrée à l'espace de sortie par une transformation non linéaire.

La création d'un perceptron multicouche pour résoudre un problème donné nécessite l'inférence de la meilleure application possible, telle que définie par un ensemble de données d'entraînement constitué des paires de vecteurs d'entrée et de sorties. L'algorithme appelé *propagation inverse* peut réaliser cette inférence [139].

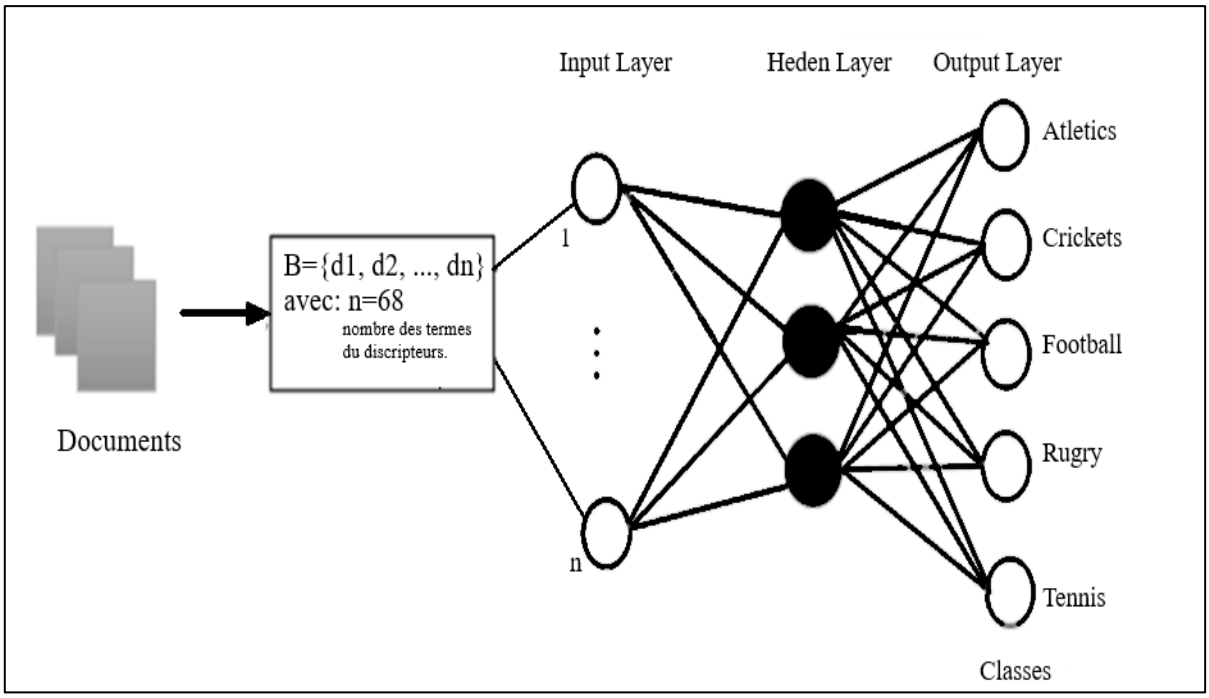

Figure 23: Architecture du système neuronal adopté pour la classification multiple des textes.

3. Expérimentation et résultats des systèmes adoptés

Afin de prouver l'efficacité des descripteurs proposés, sélectionnés et représentés par les méthodes citées ci-dessus, nous proposons une classification textuelle basée sur un ensemble de classifieurs.

Base des données

Nous utilisons un ensemble de données publiques de la BBC composé de 773 articles, chacun classé dans l'une des 5 catégories suivantes : athlétisme, cricket, football, rugby, tennis. La figure 24 visualise la distribution des textes selon la classe d'appartenance.

Paramètres de la catégorisation

Il convient de noter que 80% des données sont réservées à l'apprentissage des classificateurs et 20 % pour le test, en utilisant la fonction split [140]. Pour l'évaluation du système de la classification, nous avons employé comme mesures : la performance, la précision, le rappel, et la F-mesure.

Pour sélectionner l'architecture Bayes Net satisfaisante, nous avons testé avec différentes méthodes de recherche (Hill Climber, TabuSearch et K2) ; les résultats obtenus sont notés dans le tableau 17.

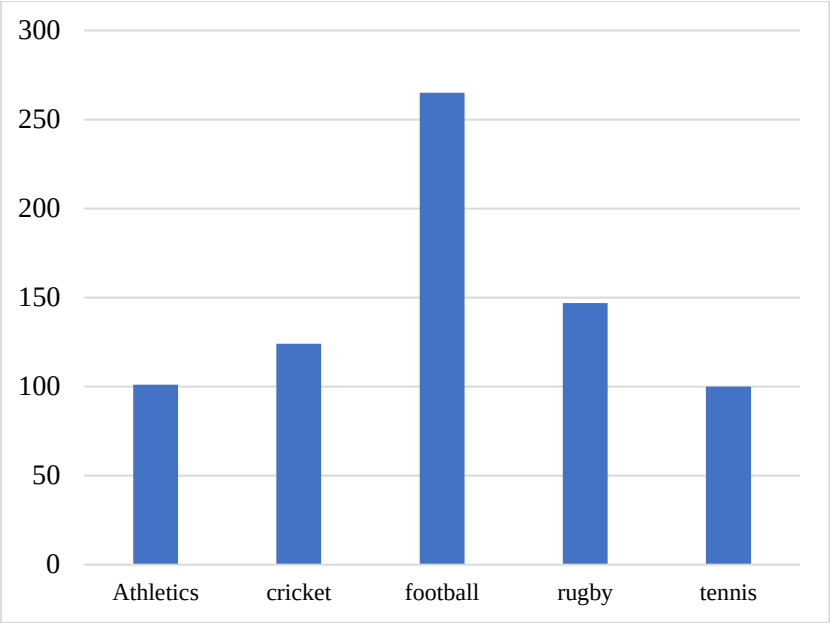

Figure 24: Les 5 classes de la base de test BBCSport.

Tableau 17: Résultats pour différents algorithmes d'apprentissage de la structure du réseau à l'aide de la nouvelle méthode probabiliste.

Structure of Bayesian Network	Accuracy on test set (%)	Recall (%)	F-Measure (%)
Tabu search	98.29	98.30	98.30
K2 Algorithm	98.00	98.01	98.01
Hill Climber Algorithm	98.29	98.30	98.30

Tableau 18: Résultats pour un nombre différent de nœuds masqués.

	Représentation Probabiliste	TF-IDF
Nombre des nœuds dans la couche cachée	La performance de l'ensemble du Test (%)	La performance de l'ensemble du Test (%)
2	88.03	85.56
3	100	95.85
5	100	94.67
10	100	94.67
50	100	94.67

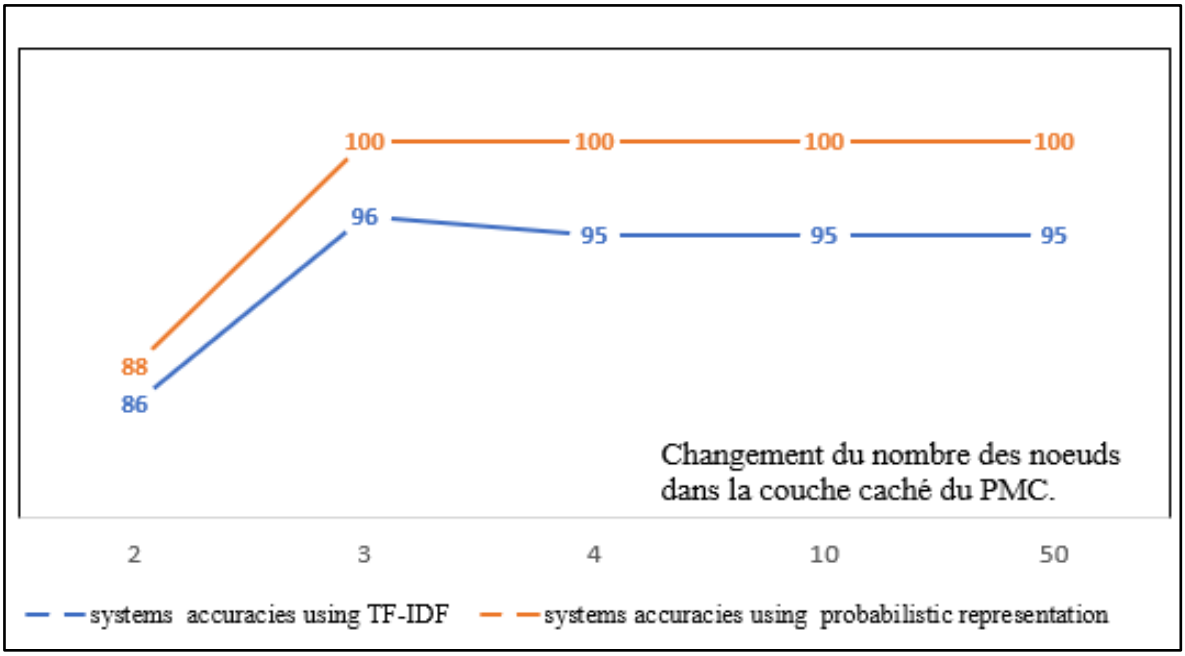

Figure 25: Changement de la performance des systèmes de catégorisation en fonction du nombre de nœuds dans la couche cachée du MLP utilisant les deux représentations numérique (TF-IDF, et la nouvelle pondération probabiliste.

Pour le perceptron multicouche, nous avons essayé d'augmenter le nombre de neurones afin de sélectionner les meilleurs résultats de classification. Le nombre de neurones sélectionné doit être suffisamment élevé pour modéliser le problème, mais pas très élevé pour éviter le surapprentissage du classifieur. Pour cela, nous somme limiter aux 3 couches, où les résultats, des deux systèmes, sont maximisés et stabilisés. (Consultez le tableau 18 et la figure 25).

3.3 Résultats et Discussion

A travers les résultats présents dans cette partie nous comparons entre deux sortes de systèmes :

Le premier système (repose sur l'approche classique décrite dans le paragraphe II. 1) consiste à un prétraitement des textes et une pondération TF-IDF, afin de générer une matrice descriptrice de l'ensemble des termes du corpus. Par la suite, la phase de la SA est suggérée pour avoir un descripteur de taille optimal, qui sera l'entrée des classifieurs réservés à la phase décisionnelle du système.

Le deuxième système est identique au premier, mais les termes élus pertinents, par l'utilisation de la méthode de SA, sont repondérer par la méthode probabiliste proposé, avant d'appeler les classifieurs du ML.

Le tableau 19 présente une comparaison entre les deux systèmes, avec les différents classifieurs utilisés, combinés avec l'ensemble des caractéristiques proposées.

Tableau 19: Résultat de classification en employant les classifieurs bayésiens et le Perceptron multi couches.

	Representation Probabiliste			TF-IDF		
Classifieurs	**Precision (P)**	**Recall (R)**	**Performance (A)**	**P**	**R**	**A**
Bayes Net	98.30%	98.3%	98.29%	94.30%	92.9%	92.89%
NB	98.4%	98.3%	98.29%	94.10%	92.3%	92.30%
PMC	100%	100%	100%	94.80%	94.7%	94.67%

Dans notre premiere contribution [16] la classification bayesienne a montré un excellent score de performance supérieure à **97%**. L'étude comparative realiser dans cette contribution est effectué entre différents systèmes qui utilisent des méthodes de pre-traitement, de décision différentes, et où l'utilisation des algorithmes de selection d'attributs est absente.

Le score indiqué dans le tableau 19 prouve que les méthodes de sélection deminuent la performance du classifieur bayésien. En contrepartie la méthode de représentation probabiliste proposée permet de récupérer les pertes et améliorer le score de performance des réseaux bayésiens, où le taux de reconnaissance obtenu est de **98.3%.**

Pour le PMC l'utilisation de la matrice descriptrice, generé par la TF-IDF, sans optimasation ou reduction de ses dimentionnalité, était presque imposible. Pour cela, il était nécessaire d'inclure les moyens de selection des attributs afin de minimiser les entrées du PMC. L'utilisation des processus du deuxieme système, avec le classifieur PMC était plus reussite, et notre methode a prouvé son efficacitée et sa force devant l'approche populaire, où le score obtenue en adoptant le deuxieme système est de 100% , qui un score de classification parfait.

Matrice de confusion

Une autre façon d'évaluer l'amélioration des résultats de la classification est la matrice de confusion. Tableaux : 20, 21, 22 et 23 donnent la matrice de confusion des classifieurs bayesiens et du PMC, en utilisant la représentation probabiliste et TF-IDF comme méthodes de représentation des descripteurs obtenus par la recherche Best-first comme méthode de sélection des attributs.

Les matrices de confusion montrent la convergence des textes vers leurs classes adéquates lorsqu'on utilise le deuxième système qui notre système proposé. Et tant que le score de l'approche probabiliste, pour la catégorisation neuronale des documents, est de 100% la matrice de confusion convenable (tableau 23) est une matrice creuse, où le nombre des textes pour chaque classe est concentré sur le diagonal de la matrice.

Tableau 20: Matrice de confusion de la classification bayésien, utilisant la représentation TF-IDF.

Class	A	C	F	R	T
Athletics(A)	17	0	0	0	0
Cricket (C)	0	20	0	0	0
Football (F)	0	0	37	0	0
Rugby (R)	0	0	1	21	0
Tennis (T)	0	0	0	0	20

Tableau 21: Matrice de confusion de la classification bayésien, utilisant la nouvelle représentation probabiliste.

Class	A	C	F	R	T
Athletics	17	0	0	0	0
Cricket	0	20	0	0	0
Football	0	0	38	0	0
Rugby	0	1	1	20	0
Tennis	0	0	0	0	20

Tableau 22: Matrice de confusion de la classification neuronale, utilisant le classifieur PMC et la représentation TF-IDF.

Class	A	C	F	R	T
Athletics	17	0	0	0	0
Cricket	1	13	1	5	0
Football	0	1	36	1	0
Rugby	0	0	0	22	0
Tennis	0	0	1	0	19

Tableau 23: Matrice de confusion de la classification neuronale, utilisant le classifieur PMC et la nouvelle représentation probabiliste.

Class	A	C	F	R	T
Athletics	16	0	0	0	1
Cricket	0	20	0	0	0
Football	0	0	38	0	0
Rugby	0	0	0	22	0
Tennis	0	0	0	0	20

Tableau 24: Comparaison de l'approche de classification neuronale, employant la base des données BBCSport, avec autres systèmes de la littérature.

Systèmes de classification de la base BBCSport	**Performance %**
[4]	98.9 ± 0.7
[141]	97
Notre systèmes	100

Notre système présente une amélioration de 3% en comparant avec autres systèmes qui adoptent aussi la classification de la base des données BBCSport. Comme le tableau 24 montre, les performances du système de classification neuronal, adoptant la nouvelle approche de la vectorisation probabiliste présente un excellent score de 100%.

V. Conclusion

Grâce aux études comparative présentées dans ce chapitre, nous avons prouvé l'impact de la représentation de l'information, la vectorisation et la génération des descripteurs de taille optimale et pertinents, sur la qualité de la catégorisation des textes.

Ainsi, l'ensemble des résultats montrent qu'une bonne représentation numérique des corpus influence positivement la performance des classifieurs de l'apprentissage art. Aussi, en se basant sur l'hypothèse de perte d'information durant le processus de la réduction des dimensionnalités et les faiblesses de la TF-IDF, nous avons pu confirmer la supposition, à travers l'étude comparative présente dans ce chapitre, et proposer des solutions correctives. La contribution présentée, permet de repondérer les termes sélectionnés pertinents pour représenter le corpus, afin de redéfinir leurs importances dans chaque document. Notre nouvelle représentation génère des vecteurs descripteurs probabilistes avec une dimension optimale. En outre, le descripteur proposé montre son efficacité avec les réseaux bayésiens et le PMC, comme outils de classification sensibles aux entrées. Les performances des systèmes que nous avons proposés ont été excellentes, où la classification bayésienne présente un score de classification qui est égale à 98%, et un score plus que parfait qui atteint 100% comme performance du classifieur PMC.

Partant, les résultats obtenus sont stimulants, et nous encouragent à tester le système avec d'autres bases des données de grande taille et d'aller vers un système scalable. En effet le système proposé est interopérable, car son intégrité avec autres systèmes fonctionnels est possible, et peut présenter des résultats prometteurs. Comme autres propositions qui permettent l'amélioration des vecteurs descripteurs afin de progresser et renforcer les systèmes de classification des données non-structurées, nous présentons dans le chapitre suivant des nouvelles contributions, qui se basent principalement sur la logique Floue et des SIFs améliorés.

Chapitre 3 : Contributions au moteur d'inférence et à la représentation vectorielle flous pour le renforcement des systèmes de classification.

Chapitre 3 : Contributions au moteur d'inférence et à la représentation vectorielle flous pour le renforcement des systèmes de classification.

I. Introduction

L'automatisation des systèmes experts est le défi dans plusieurs domaines [142] [143]. Notamment, dans le contexte de l'intelligence artificielle appliquée aux données textuelles, l'objectif principal est de représenter, gérer et présenter des informations pertinentes, en utilisant des moyens intelligents. Pour cette raison, les systèmes d'inférence floue (SIF) [144] sont pratiqués par des experts pour développer un ensemble des systèmes avantageux. Sauve, l'automatisation reste limitée pour identifier les règles d'inférence, où l'intervention des experts est nécessaire, et la modélisation des données extensives sont encore une restriction du SIF. Cependant, la sélection manuelle des règles d'inférence est insuffisante, spécialement, lorsque la quantité des données est grande ou l'intervalle des connaissances est vaste. Parmi, les applications populaires des SIFs, nous citons la classification des données, où plusieurs systèmes sont connus par des classifieurs flous, et qui ont connu un grand succès. Également, pour la pondération des termes, comme processus important du Text Mining, nous appliquons le raisonnement flou pour extraire le poids floue, noté par la TF-IDF floue (où Fuzzy TF-IDF, FTF-IDF). Une simple application du FIS, spécialement la manière manuelle de génération des règles d'inférences, pour produire la matrice descripteur FTF-IDF, occupe plus du temps lors de la définition des règles d'inférence, et aussi les réglementations pertinentes seront difficiles à établir. De même, la grande taille des données d'entrée nécessite une génération automatisée de règles d'inférence. Dans ce chapitre, nous proposons les modèles d'association [145] comme une technique calculée pour résoudre le problème présenté. Ainsi, l'utilisation de la nouvelle méthode automatique permet de sélectionner les meilleures règles en fonction de quatre étapes clés :

- Prétraitement (réduction des fonctionnalités),
- Détermination de la fonction d'appartenance,
- Représentation nominale des entrées, où nous appelons les modèles d'association,
- Post-traitement des règles acceptées.

Généralement, le prétraitement des données effectue le fonctionnement du système, où les données bruyantes sont éliminées et les informations pertinentes sont sauvegardées. Le prétraitement est conseillé dans notre pratique et les expressions mathématiques pour modéliser le processus sont indispensables. Ensuite, dans l'approche proposée, le deuxième processus est basé sur la fonction d'appartenance donnée par le SIF, où chaque entrée et sortie ont ses valeurs d'appartenance, ce qui permet la transaction vers la représentation nominale.

En effet, l'intérêt des modèles d'association est de sélectionner les meilleures règles en se basant sur le niveau de confiance [146] , où les règles avec un score de confiance supérieur à 90% sont préférées. Parmi les méthodes d'association proposées dans la littérature, nous avons choisi des méthodes simples comme : l'algorithme Apriori et le filtre d'association [147]. Finalement, le post-traitement des résultats élus est une étape cruciale pour bénéficier des trois variétés de règles générées.

Différemment à la manière traditionnelle, l'approche proposée accorde une réponse temporelle optimale et consolide les règles, où les règles fournies sont plus précises et pertinentes.

Ce chapitre présente deux sortes d'application qui incluent l'automatisation des règles floues, et qui sont : la classification floue des donnes structurés ainsi que la pondération floue des termes en langage naturel. Les expérimentations, des applications adoptés, démontrent que notre technique affecte les performances des systèmes développés, où par exemple l'insertion des modèles d'association améliore les performances FTF-IDF par rapport à la représentation régulière, en utilisant le SIF traditionnel, et d'autres méthodes existantes comme : la TF-IDF, la fréquence locale (TF) et la fréquence globale du terme.

Les principales contributions de ce chapitre sont les suivantes : l'automatisation de la sélection des règles d'inférence (RI), qui contrôle les sorties floues, et la visualisation de l'impact de la modification, de ce processus du SIF, sur deux types d'applications : la catégorisation floue ainsi que la catégorisation à base des algorithmes de l'apprentissage automatique. Le reste du chapitre est organisé comme suivant : d'abord nous décrivons les bases de la LF et les SIF. Par la suite l'approche proposée ainsi que les méthodes employées seront discutées. La représentation floue FTF-IDF, les applications sur lesquelles nous avons appliqué la nouvelle approche, ainsi que les résultats obtenus seront détaillés et discutés avant de conclure le chapitre.

II. Logique floue et systèmes d'inférences flous

1. Logique Floue (FL)

Les fonctions d'appartenances classiques ont été évaluées comme une logique booléenne, variant dans la gamme {Vrai, Faux} ou {0, 1}, par conséquent, leurs applications dans des domaines imprécis et incertains sont très limitées [18]. Ainsi, la logique floue (LF), inventée par Zadeh [148], est l'extension de la logique classique dont les fonctions d'appartenance ont des valeurs dans l'intervalle [0, 1]. Sur la base de cette conversion, la LF attire toujours l'attention des chercheurs et des industriels pour le développement des différentes applications y compris les applications web.

En revanche, la logique floue est une logique polyvalente qui repose sur la théorie mathématique, à savoir la théorie des ensembles flous, dont l'objectif principal est la manipulation des notions incertaines du langage naturel. Dans ce contexte, une nouvelle fonction d'appartenance a été définie : μ_A : X -> [0, 1], qui implique que x appartient à l'ensemble flou A, avec un degré de vérité (/ou d'appartenance) égal à $\mu_A(x)$.

En général, la théorie de l'ensemble flou peut être utilisée dans un large éventail de domaines où les informations sont incomplètes ou imprécises, tels que la classification, la recherche opérationnelle, l'aide à la décision, la reconnaissance de formes, l'exploration de données, les systèmes d'information et de nombreux autres domaines de l'intelligence artificielle.

En outre, le traitement des demandes de classification, par exemple, la fonction d'appartenance $\mu_c(x)$ détermine le degré d'appartenance pour chaque variable linguistique x, qui est un réel entre 0 et 1 à la classe convenable c.

Plusieurs travaux, dans la littérature, adoptent le principe flou afin de réaliser des systèmes décisionnels tel que l'organisation ou la recherche de l'information pertinente. Ces systèmes

suivent les processus majeurs des systèmes appelé les systèmes d'inférence floues (SIFs) décrites dans la section suivante.

2. Systèmes d'Inférences Flous (SIFs)

Les systèmes d'inférences flou se base sur la théorie de la logique floue et servent à transformer les données d'entrée en données de sortie à partir de l'évaluation d'un ensemble des règles. Ce pendant les SIFs sont constitués de trois processus majeurs [148] illustrés dans la figure 27.

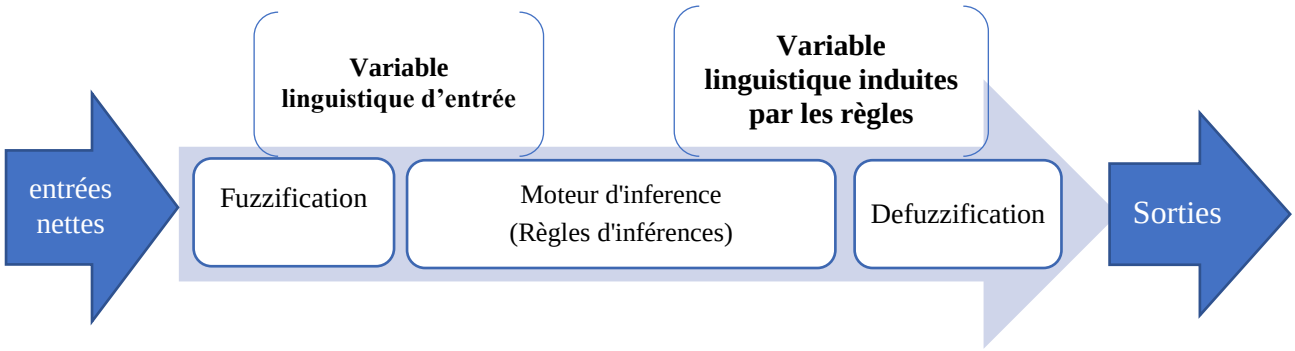

Figure 26: Processus des systèmes d'inférence flous.

Le premier processus est celui de la ***Fuzzification*** [148], qui consiste à caractériser les variables linguistiques utilisées par le système, par la transformation d'une donnée numérique en variable linguistique. Pourtant, la fonction d'appartenance permet de définir le degré d'appartenance d'une donnée numérique à une variable linguistique. Plusieurs formes de ces fonctions sont présentes dans la littérature, comme : la fonction trapézoïdale, fonction gaussienne, et la fonction triangulaire. L'utilisation des fonction simples favorise la précision de la conception, cependant les fonctions complexes n'apporte pas de précision.

La deuxième phase du SIF est la phase d'inférence [148], où ***le moteur d'inférence*** est appliqué pour la condensation d'un système basé sur un ensemble de règles, exprimé par des experts, et qui doit bien définir le problème.

La phase finale est la phase ***défuzzification*** [148] qui permet de déduire une sortie unique, pour chaque entrée, en se basant sur l'ensemble flou de sortie agrégée. De nombreuse méthodes de défuzzification sont cités dans la littérature, et la fonction centroïde reste la plus populaire.

III. Automatisation des Règles d'Inférences (RI) floues

1. Règles d'inférences classiques

Pour prendre une décision, en employant les SIF, plusieurs valeurs de variables linguistiques sont convenablement définies par les fonctions d'appartenance, et sont aussi liées entre elles par des règles.

Chaque règle donne lieu à une conclusion partielle, qui est agrégée aux autres règles pour fournir une conclusion (agrégation). En général, pour déterminer les règles qui régissent les

SIF, associés à un problème donné, il est nécessaire de contacter un expert [151]. En effet, il est important de noter que cette tâche devient plus difficile si le nombre des entrées et/ou des sorties est très large ou l'univers du discours est très large. La solution proposée, pour l'automatisation du processus d'inférence, se base sur l'application des algorithmes d'AA prédictives, qui sont les modèles d'association.

2. Nouvelle approche pour l'automatisation des règles d'inférence floues

Les règles d'association est une technique populaire, étudiée d'une manière approfondie, dans l'objectif de prédire des relations significatives entre deux ou plusieurs variables pour un jeux des données. En effet, l'intérêt principal de l'approche proposé est d'intégrer les modèles d'association afin de générer d'une manière automatique les RI d'un système décisionnel basé sur le raisonnement flou. Par conséquent, une fois que les caractéristiques sont représentées en entrée du SIF, il est nécessaire de faire appel à certains modèles d'association (MA) pour sélectionner automatiquement les meilleures règles d'inférence qui permettra d'améliorer la qualité de la décision désirée. L'ensemble des modèles utilisés, dans la contribution proposée, sont détaillés la suite.

2.1 Modèles d'associations

Le sujet des modèles d'association est considéré comme partie des approches d'apprentissage symbolique non supervisées, souvent utilisées dans le domaine de l'exploration des données (Data Mining) [30].

En règle générale, une règle d'association (A→B) est composée d'un ensemble d'éléments appelés items ou objets I {i_1, i_2, i_3, etc.}, et d'un ensemble d'opérations (transactions) T {t_1, t_2, t_3, t_4, etc.} correspond à un ensemble d'apprentissage qui sera utilisé pour déterminer les règles d'association, dans l'étape suivante du modèle. Le volume de la transaction est le nombre d'éléments contenus dans la transaction. Une notion importante pour un ensemble d'items est son support σ, qui fait référence au nombre de transactions observées qui le contiennent. Le support (S) et la confiance (C) sont les mesures de performance d'une règle association, définies, par les équations 25, 26 et 27 comme suit :

$$\sigma (X)= \text{Carte } \{ti /X \subseteq ti, ti \in T\} \quad (24)$$

$$S (A \rightarrow B) = \sigma (A \cup B) /N \quad (25)$$

Où :

Support (S) : est l'occurrence de la règle dans l'ensemble des données utilisées.

σ (A U B) : support des éléments {AUB}

N : le nombre total de transactions.

Confiance : mesure de la validité de la règle (pourcentage des exemples qui vérifient la conclusion).

$$C (A \rightarrow B) = \sigma (A \cup B) / \sigma(A) \quad (26)$$

Il convient de mentionner qu'une règle avec un faible support peut être observée seulement par hasard. Pour un ensemble de transactions, nous pouvons générer des règles en fondant toutes les règles de l'association avec le support ≥ min support et confiance ≥ min confiance, où min support et min confiance sont les seuils du support et de la confiance.

Plusieurs algorithmes, pour l'extraction d'items fréquents (règles d'association), sont proposés dans la littérature, tels que l'algorithme d'Apriori [147], Close [149], OCD [150], et autres. Dans l'approche proposée, nous employons comme MA : l'algorithme Apriori, en raison de leur simplicité, et aussi le filtre association afin de sélectionner les meilleures règles.

2.1.1 Algorithme apriori

Algorithme apriori est un algorithme de règle d'association populaire [147] qui inclut des étapes précises pour tenir compte de la croissance du nombre d'ensembles d'éléments candidats. Le processus de l'algorithme est résumé en trois étapes :

1) Générer des ensembles items ;
2) Calcul de la fréquence des ensembles items ;
3) Conserver les ensembles d'items avec un support minimal, ce qui représente l'ensemble des items fréquents.

L'algorithme Apriori présente certaines limites, car le déroulement des données initiales est récurrent, et où le calcul des supports et les générations des règles sont très coûteux en termes de temps.

2.1.2 Filtre d'association

Pour faire fonctionner le filtre d'association, une association arbitraire, comme Apriori, est appliquée sur des données qui ont été passées par un filtre arbitraire. Pour le filtre d'association, la structure du filtre est basée exclusivement sur les données de formation et le filtre traitera les instances de test sans modifier leur structure [147]. Le Filtre d'association est un algorithme plus efficace que l'algorithme Apriori basé sur les deux (Nombre de cycles effectués, grands ensembles d'items) car l'algorithme d'Apriori génère le nombre de cycles effectués et génère un ensemble d'éléments importants ce qui dégrade la performance de l'algorithme.

2.2 Nouvelle approche de sélection automatique des Règles d'Inférences

Les concepteurs du système définissent l'ensemble des règles en se basant sur leurs connaissances, une fois les variables linguistiques définies, comme entrée dans le système d'inférence flou. Dans notre approche, nous choisissons automatiquement les règles appropriées, et nous pouvons résumer notre approche en quatre étapes principales, comme montre la figure suivante :

Figure 27: Processus de la génération automatique des règles d'inférence.

a) Prétraitement des entrées

Dans l'étape du prétraitement, nous considérons l'ensemble des données D = $\{x^1 ... x^k ... x^N\} \subset IR^n$ en décomposant l'intervalle $[\min_k x_i^k, \max_k x_i^k]$ avec $\forall i \in [1, n]$. Puis nous définissons des fonctions de membre appropriées. Le choix de la fonction l'appartenance n'est pas une pupille droite et l'expérience uniquement peut aider le concepteur. Aussi, l'utilisation des fonction simples favorise la précision de la conception, cependant les fonctions complexes n'apporte pas de précision.

b) Transformation de données numériques en données nominales.

Le deuxième processus concerne la représentation nominale de l'ensemble de données, où pour chaque simple attribut d, nous déterminons les fonctions d'appartenance floues convenable. Par la suite la composante de d est remplacée par ces ensembles flous en utilisant les opérateurs logiques 'ET' et 'OU'. Étant donné les données numériques $x^k = [x_1^k ... x_i^k ... x_R^k]$. Si $\forall k \in [1, N]$ $\forall i \in [1, R]$ x_i^k est A_i^k et B_i^k , la représentation nominal de x^k est $\bar{x}^k = [A_1^k, B_1^k, \ldots, A_i^k, B_i^k, \ldots, A_R^k, B_R^k]$. Dans ce sens $D = \{ \bar{x}^1, \ldots, \bar{x}^k, \ldots, \bar{x}^R \} \subset IR^{2n}$ où R est la dimension des données après la réduction (step1). La table 25 présente quelques exemples de la transformation numérique, des dégrées d'appartenance d'un élément d des données iris, à une forme nominale.

Tableau 25: Représentation nominale de l'ensemble de données d'iris.

The nominal representation of :	**Is :**
[d=[3.3,1.0], Iris-versicolor]	[low, low, Iris-versicolor]
[d=[5.1,1.8], Iris-virginica]	[mean AND great, mean AND great, Irisvirginica]
[d=[5.6,2.4], Iris- virginica]	[mean AND great, great, Iris- virginica]
[d=[6.7,2.0], Iris- virginica]	[great, great, Iris- virginica]
[d=[5.0,1.7], Iris-versicolor]	[mean AND great, mean AND great, Iris Versicolor]

c) Application des règles d'association

Modèles d'association pour sélectionner les meilleures règles en fonction du niveau de confiance. Dans la partie précédente nous avons décrit les modèles d'association, tels que l'association Apriori et Filtre, employés pour sélectionner les meilleures règles.

d) Post-traitement

Post-traitement des règles sélectionnées. Les modèles d'association produisent trois types de règles, à savoir :

- Règles rejetées (RR), celles-ci ne donnent aucune information sur les sorties ;
- Règles explicites (RE), celles-ci ne donnent que des informations sur les sorties attendues comme conséquence ;
- Règles implicites (IR), celles-ci présentent des informations sur les classes mais pas en conséquence notées IR.

Le premier type des règles est supprimé du fichier des règles sélectionnées. Le deuxième type de règles est maintenu sans aucune modification. Pour transformer le troisième type de règles en règles explicites, nous utilisons les résultats algébriques suivants :

$$A \vee (\mathrm{B} \wedge \mathrm{C}) \cong (A \vee B) \wedge (\mathrm{A} \vee \mathrm{C}) \quad (27)$$

$$A \wedge (B \vee C) \cong (A \wedge B) \vee (A \wedge C) \quad (28)$$

$$(\overline{\mathrm{B} \vee \mathrm{C}}) \cong \bar{B} \wedge \bar{C} \quad (29)$$

$$(\overline{\mathrm{B} \wedge \mathrm{C}}) \cong \bar{B} \vee \bar{C} \quad (30)$$

$$(A \vee B \to C) \cong (A \to C) \vee (B \to C) \quad (31)$$

$$(A \to C \vee B) \cong (A \wedge \bar{\mathrm{B}} \to C) \quad (32)$$

$$A \to C \vee B \cong \bar{B} \wedge \bar{C} \to \bar{A} \quad (33)$$

En utilisant l'ensemble de données d'iris, pour l'objectif de classification, nous remarquons quelques exemples pour les trois types des règles :

- Pour un RR :***la largeur des pétales est faible -> la longueur des pétales est faible***.
- En tant qu'IR, nous avons trouvé : ***classe IS Iris-versicolor -> la longueur du pétale est faible ET pétale la largeur est faible***. Ce dernier est transformé en ER à l'aide de résultats algébriques.
- Le troisième exemple est le ER :***la longueur des pétales est faible ET la largeur des pétales est faible -> classe IS Iris-versicolore***, où la sélection de la classe finale est raisonnable.

IV. Applications de la logique floue et l'automatisation des règles d'inférence à la classification.

Cette partie de ce chapitre se focalise sur l'application de l'approche de l'automatisation des règles d'inférence dans des systèmes décisionnels comme : la classification floue des données structurés, ainsi que la représentation vectorielle floue pour la catégorisation automatique des données non structurées.

Dans un premier temps, nous illustrons l'enchainement de la sélection automatique des règles d'inférence, pour réaliser la classification floue des données iris. Par la suite, notre objective est de visualiser l'impact de l'approche indiquée sur la représentation TF-IDF floue, qui influence la performance des classifieurs du ML.

1. Application de la sélection automatique des règles d'inférence à la classification floue des données iris

Généralement la classification floue suit les processus canoniques des SIF, décrites précédemment, afin de mapper des entités en tant qu'entrées, noté par X, vers des classes de sorties C= $\{c_1, c_2, \ldots, c_n\}$. Le problème est alors de déterminer pour chaque élément x, avec $x \in X$, le degré $\mu_c(x)$ auquel l'objet x appartient à la classe $c \in C$. Cependant, la fonction d'appartenance $\mu_c(x) : X \to [0,1]$ a été définie pour chaque classe $c \in C$ [10]. En revanche, les systèmes de classification flou (SCF) utilisent des connaissances informelles sur le domaine du

problème traité, et nécessite l'intervention des experts afin de les définir [151]. En général, un SCF est un classifieur qui utilise :

- Des ensembles flous pendant l'entraînement ou pendant son fonctionnement.
- Un système d'inférence flou, basé sur des règles d'inférence de type *si-alors*, qui produit une étiquette de classe c pour chaque entrée x.

Dans notre cas, les règles d'inférence, pour la classification floue des données iris, ont été générer d'une manière automatique, grâce à l'approche proposé, et le reste de cette section présente les expérimentations et les résultats obtenues.

Base des données

Les données de fleurs d'iris sont les plus connues dans le domaine de la classification [152], sont utilisées dans la partie expérimentation, pour prouver l'efficacité de l'approche adoptée.

Résultats et discussion

Dans cette partie, nous utilisons la méthode proposée pour classer les données iris avec deux dimensions [petallenght et petalwith] et deux classes [Iris-versicolor et Irisvirginica]. Nous utilisons des formes trapézoïdales comme fonctions d'appartenance.

La fonction d'appartenance en sortie est une fonction binaire 0 pour la classe Iris-versicolor et 1 pour la classe Iris-virginica. Le tableau 26 donne l'ensemble des règles produites par les modèles d'association : Apriori et le filtre d'association.

Tableau 26: Description des règles sélectionnées par type et mesure de confiance.

Selected rules		Type	Confidence
IF	**THEN**		
Petallength IS low AND petalwith IS Low	Class IS Iris-versicolor	ER	1
Petalwith IS low AND class IS Iris-versicolor	petallength IS low	IR	0.98
Petallength IS low AND class IS Iris-versicolor	Petalwith IS low	IR	0.98
Class IS Iris-versicolor	Petallength IS low	IR	0.96
Petalwith IS great	Class IS Iris'virginica	ER	0.96
Class IS Iris-versicolor	Petallength IS low AND petalwith IS low	IR	0.94
Class IS Iris-versicolor	Petallength IS low AND petalwith IS low	IR	0.92
Petelwidth IS low	Class IS Iris-versicolor	ER	0.92
Class IS Iris-versicolor	Petalwith IS low	IR	0.96
Petalwith IS low	Petallength IS low	RR	0.90

Après avoir transformé les règles implicites en règles explicites, nous obtenons l'ensemble des règles présentées dans le tableau 26. Dans ce contexte, les modèles de filtre d'associations et d'apriori conduisent à un certain ensemble de règles.

Il convient de noter que la sélection des meilleures règles par l'un des modèles d'association est basée sur une confiance supérieure à 0,9. Cette valeur reste un seuil adopté par l'utilisateur confirmant qu'une règle ait lieu sauf s'elle revient 90 fois sur 100 fois observés. Un tel choix assure un degré très élevé de sécurité de notre système. Mais il peut causer une perte fatale de l'information puisqu'elle sanctionne les règles ayant une fréquence inférieure à 90% ; en outre, il n'y a pas de raison d'uniformiser ce choix pour différents corpus.

Une façon pour remédier à ce problème consiste à mettre en place une stratégie génétique pour sélection un seuil optimal**.**

Tableau 27: Transformation des règles implicites aux règles explicites.

Rule	Active	IF petal Length	And petalwith	The Class
1	+	Low	Low	0
2	+	Mean	Low	1
3	+	Great	Low	1
4	+		Great	1
5	+		Low	0
6	+	Low	Great	1

Concernant la tâche d'inférence, nous avons testé trois méthodes : le minimum, le maximum, le produit, et le meilleur système est celui basé sur le minimum. Le tableau 27 donne la matrice de confusion associée au système proposé. À cet égard, deux données ont été rejetées en raison de l'absence de règle appropriée.

Tableau 28: Matrice de confusion pour la classification floue des données iris, basée sur la sélectionné automatique des règles d'inférence floues.

	Iris-versicolor	Iris-virginica
Iris-versicolor	45	3
Iris-virginica	0	50

Pour améliorer la qualité de notre classificateur, nous pouvons sélectionner d'autres règles tout en diminuant la valeur de confiance à 0,8.

Le tableau 29 présente les performances du système proposé pour la classification, et qui donne un taux de reconnaissance de 96,00%.

Tableau 29: Performances du système de classification floue adopté, basées sur les modèles d'association.

	TP	FP	Precision	Recall	F-measure	class
	0.90	0.00	1.00	0.90	0.95	**Iris-versicolor**
	1.00	0.06	0.94	1.00	0.97	**Iris- Virginica**
Mean	0.95	0.03	0.97	0.95	0.96	

Le rappel du système de classification Floue proposé est de 96%, il est possible de jouer sur un bon choix de fonctions d'appartenance d'entrée et de sortie pour améliorer le taux de reconnaissance.

Contrairement à la technique traditionnelle, pour sélectionner des règles d'inférence pour le classifieur flou qui se base sur l'intervention des experts, la nouvelle technique se concentre sur l'automatisation du choix des règles. Par conséquent, notre approche utilise les modèles

d’association les plus populaires, à savoir l'Apriori et le Filtre d’association. De plus, le nouveau système de classification floue donne des résultats satisfaisants, où le taux de reconnaissance est de 96%. Même si nous testons sur un ensemble des données moyen, notre approche encourage les praticiens du domaine du contrôle des systèmes intelligents à l'utiliser pour des problèmes plus complexes. Cependant, la nouvelle application proposée s’intéresse à l’amélioration de la pondération floue des données non structurées.

2. Application de la sélection automatique des règles d’inférence à la vectorisation floue et son impact sur la classification automatique

Dans cette section, nous présentons des nouvelles contributions, où la première consiste à l’insertion de la pondération floue, FTF-IDF, dans le processus de la catégorisation automatique des données textuelles, en utilisant les classifieurs ML. La deuxième intervention visualise l’impact de l’automatisation des règles d’inférence sur la qualité de la représentation vectorielle floue ainsi que la performance des classifieurs du ML [153] [154].

2.1 Classification des textes basé sur la pondération floue et les modèles ML

Le modèle de classification que nous avons suivi, se résume en 4 processus majeurs, qui sont l’analyse des entrées et la création des indexes, la pondération des termes et la génération des descripteurs, la réduction de la dimensionnalité des vecteurs, et finalement la classification automatique. Pour le processus de pondération des termes, nous proposons dans ce chapitre, l’utilisation de la version floue de la populaire TF-IDF, noté par Fuzzy TF-IDF ou FTF-IDF. La section suivante décrit la méthode de pondération floue, qui suit généralement le processus SIF.

2.2 Pondération TF-IDF floue (FTF-IDF)

La FTF-IDF est une des méthodes de pondération proposées, où la base de son calcul est la logique floue. Pour déterminer le poids FTF-IDF, l’organigramme présent dans la figure 28 montre qu'il est nécessaire de suivre le processus du système d'inférence floue. En outre, les valeurs : la fréquence du terme (TF), la fréquence inverse du document (IDF) et la longueur du document (N) sont traitées comme des variables d'entrée pour le SIF.

Le calcul de la FTF-IDF est divisée en trois étapes, à savoir la fuzzification, les règles d’inférence et la défuzzification, et la figure 28 illustre les étapes du SIF suivit afin de déterminer les poids flous FTF-IDF [154].

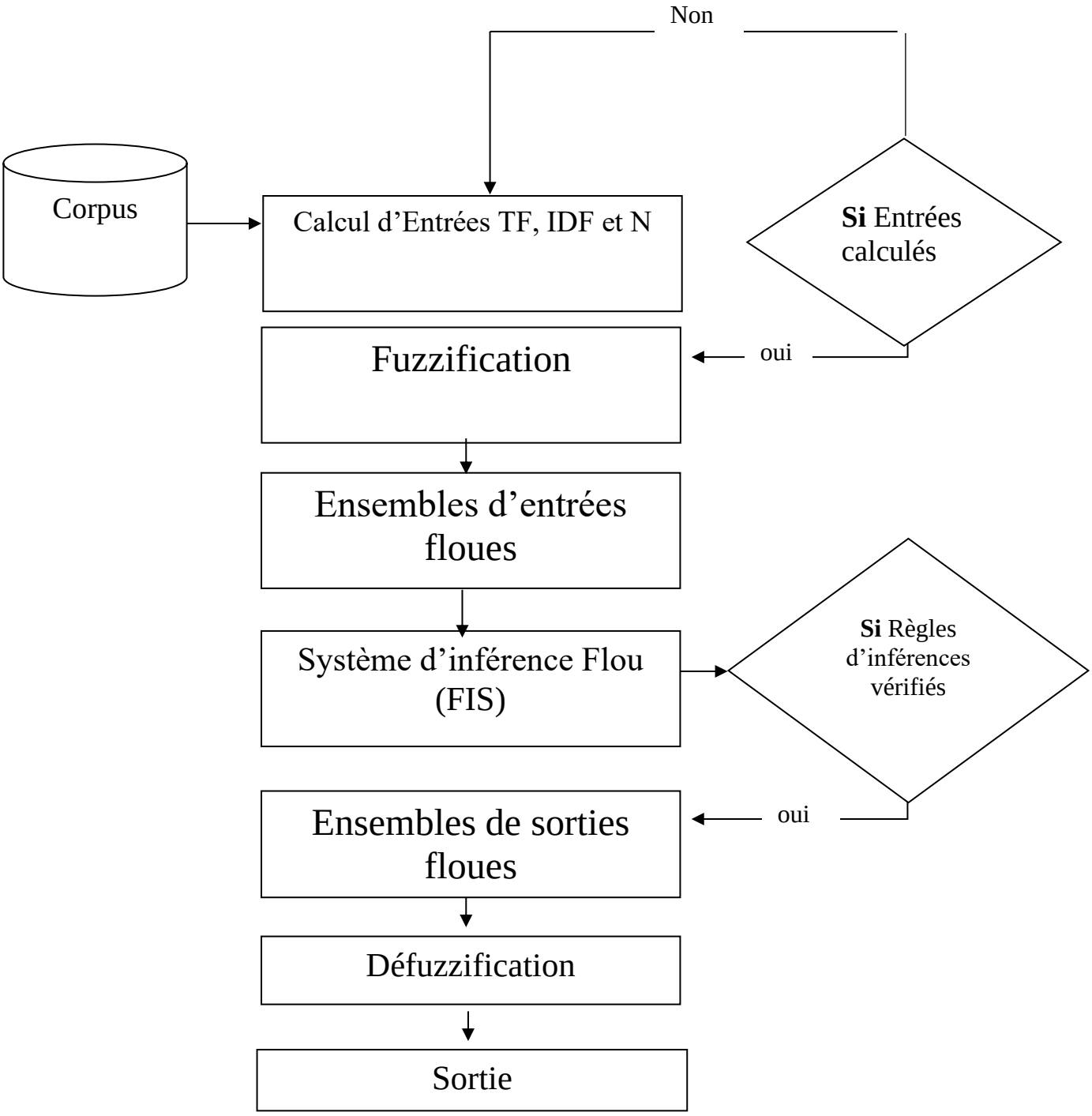

Figure 28: Organigramme du système d'inférence floue pour déterminer la pondérée floue FTF-IDF.

Fuzzification : dans ce processus, les valeurs d'entrée nettes sont converties en degré d'appartenance grâce à la fonction d'appartenance $\mu_A(x) : X \rightarrow [0,1]$. En revanche, le choix de la fonction triangulaire est convenable pour modéliser les fréquences des termes et les sorties désirées, où la nature de cette fonction est simplement définie par une limite inférieure 'a', une limite supérieure 'b', une valeur 'm'' où 'a < m < b', et la fonction d'appartenance $\mu_A(x)$ est définie par :

$$\mu_A(x) = \begin{cases} 0, & x < a \\ \frac{x-a}{m-a} & a<x<b \\ \frac{b-x}{b-m} & m<x<b \\ 0, & x>b \end{cases} \quad (35)$$

La fonction FTF-IDF utilise des termes linguistiques pour représenter toutes les variables d'entrée et de sortie. En outre, les entrées et sorties linguistiques sont définies comme indiqué dans la table 30.

Tableau 30: Les composantes du système d'inférence FTF-IDF.

	Linguistic Variables	Linguistic Values	Linguistic Values Intervals
Inputs	TF: la fréquence du terme (term frequency.)	Very Low Low Medium Hight Very Hight	[0, 0.1] [0.05, 0.45] [0.3, 0.7] [0.5, 0.9] [0.8, 1]
	IDF: la fréquence inverse du terme (inverse term frequency)	Very Low Low Medium Hight Very Hight	[0, 1] [0.5, 4] [2.5, 6.5] [5, 9] [8, 10]
	N: la longueur du document (Length of the document)	Very Low Low Medium Hight Very Hight	[0, 0.1] [0.5, 0.25] [0.2, 0.4] [0.3, 0.7] [0.5, 1]
Output	Fuzzy Weight: (poid flou) FTF- IDF scores	Low Medium Hight	[0, 0.4] [0.25, 0.75] [0.6, 1]

Ces termes linguistiques sont représentés par des fonctions d'appartenance, obtenues à partir du domaine de connaissance, comme il est montré dans la figure 29. La plage des variables d'entrée, TF (fig 29 (a)), IDF (fig 29 (b)) et N (fig 29 (c)), pour le SIF sont représentés par : très élevé (VH), élevé (H), moyen (M), faible (L) et très faible (VL). La plage des poids est représentée sous les variables linguistiques : élevée (H), moyenne (M), et faible (L).

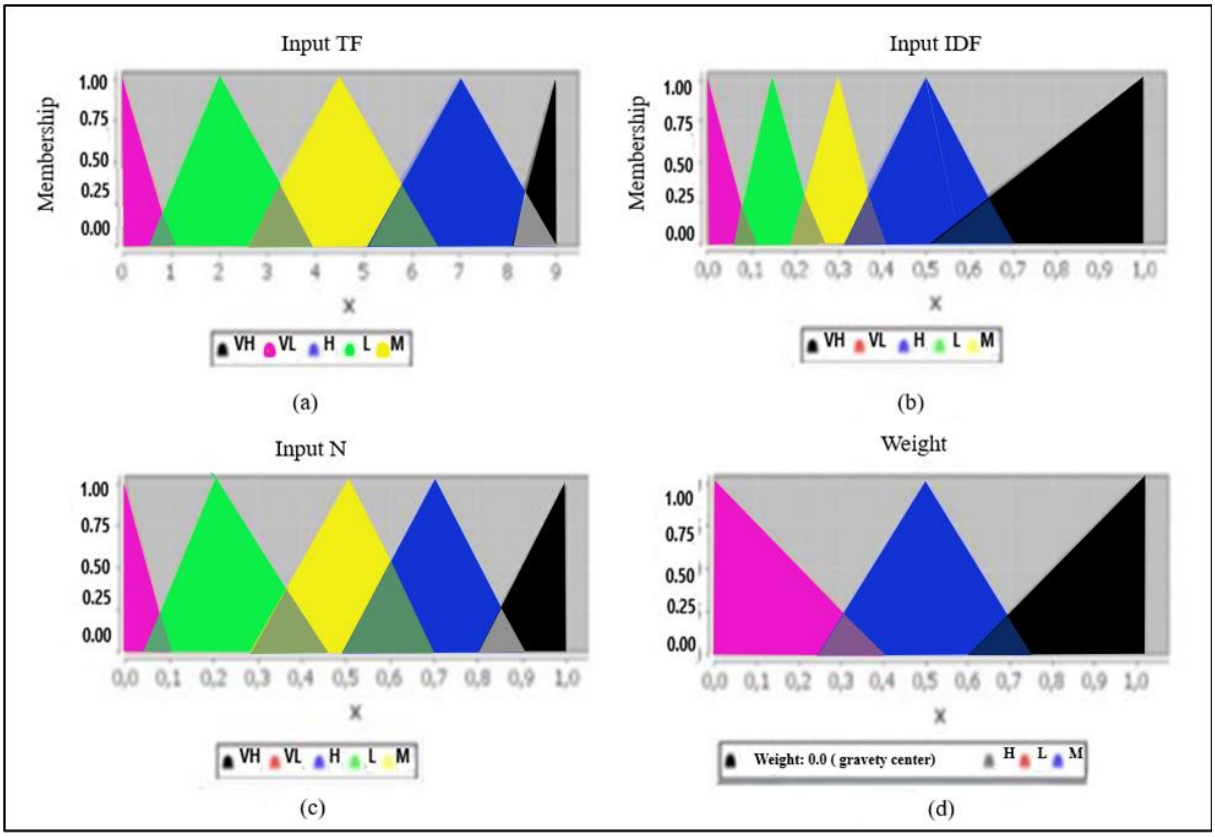

Figure 29: Les degrés d'appartenances des entrées FTF-IDF : (a) TF, (b) IDF, (c) N et (d) la sortie Wd,t.

- **Les Règles d'inférence** : sont utilisées comme outil pour déduire le poids final de chaque terme, en fonction des variables d'entrée : TF, IDF et N. les règles étaient exprimées sous forme de règle ***si-alors***; dans la mise en œuvre du système, il s'exprime comme suit : «*SI (TF est VH) ET (IDF est VH) ET (N est VH) ALORS (Le poids est H)*»;
- **Défuzzification** : Dans notre cas, nous utilisons la méthode populaire du centroïde. Cette dernière permet de définir le poids, une seule sortie, pour chaque indice de terme, en se basant sur l'ensemble flou de sortie agrégé.

2.3 Impact de la TF-IDF floue sur la performance des classifieurs automatique

L'utilisation de la FTF-IDF était, toujours, intégrée dans la conception des systèmes décisionnel flous. Dans ce chapitre nous essayerons d'adopter la pondération floue afin de générer des vecteurs descripteurs, d'un ensemble des données textuelles non structurées, et de présenter ces descripteurs comme entrées des classifieurs du ML. Le but principal est de visualiser l'impact de cette pondération sur la classification supervisée, afin de corriger les lacunes de la fonction populaire TF-IDF. Pour ce faire on s'est basé sur un corpus des données multi-classes, un ensemble de méthodes de réduction de la dimensionnalité pour le descripteur obtenu, et un groupe de classifieurs ML sensible aux entrées délivrés.

a) Base des données et processus prétraitement

Pour tester les performances du descripteur flou, nous proposons BBCNEWS et BBCSPORT [108] comme corpus de 2962 News. La bases des données se compose de 5 classes prédictives. Les corpus utilisés sont subits le prétraitement standard des données, décrit dans le premier chapitre de cette thèse.

b) Outils du processus de la classification

Après avoir générer le descripteur flou, des bases des données employées, et pour améliorer l'efficacité des classifieurs utilisés, il est nécessaire d'intégrer des méthodes de réduction de la dimensionnalité. Plusieurs études ont prouvé l'efficacité de méthodes de sélection d'attributs dans différents domaines. Dans ce travail, nous comparons trois méthodes qui sont : gain d'informations, Relief et CFS. Aussi, nous comparons plusieurs algorithmes d'apprentissage automatique tels que Naïve Bayes et ses dérivés, SVM et le classifieur forêt aléatoire.

L'évaluation des systèmes de catégorisation se base sur : Rappel, précision, F-Mesure, la performance et l'AUC. Les méthodes de classification, ainsi que les mesures de performance sont bien définies dans le chapitre état d'art.

c) Résultats et discussion

Nous notons que 80% des données sont réservées à l'apprentissage des classifieurs, et 20% pour le test. Pour sélectionner les résultats satisfaisants, nous avons testé avec différents modèles.

Les tableaux 31, 32 et 33 représentent les résultats du système basé sur la pondération FTF-IDF ; CFS, Relief et IG en tant que méthodes de sélection d'attributs. Dans cette partie, nous testons sur les données BBCSPORTS.

Tableau 31 : Résultats du système de classification, des BBCSport data, basés sur la pondération FTF-IDF et CFS comme méthode de sélection d'attributs.

	Precision %	Recall %	F-measure %	Accuracy %	AUC%
Naïve Bay	97	97	96.8	96.95	100
Bayes net	96	96	96	95.91	99.07
Bayes Update	97	96.6	96	96.5	100
RandomForest	96.2	96	95.8	95.91	98.5
SVM	97.4	97.3	97.3	97.27	99.9

Table 32: Résultats du système de classification, des BBCSport data, basés sur la pondération FTF-IDF et Relief comme méthode de sélection d'attributs.

	Precision %	Recall %	F-measure %	Accuracy %	AUC%
Naïve Bay	92	90	90	90	98
Bayes network	95	95	95	95	99.9
Bayes Update	98	98	98	98	99
Random F	98	98	98	98	99
SVM	93	93	93	93	95

Tableau 33: Résultats du système de classification, des BBCSport data, basés sur la pondération FTF-IDF et IG comme méthode de sélection d'attributs.

	Precision %	Recall %	F-measure %	Accuracy %	AUC%
Naïve Bay	98.7	98.6	98.6	98.63	99.5
Bayesian net	95.2	95	95	95	99.9
Bay Update	98.7	98.6	98.6	98.6	99.5
RandomForest	93	92	92	92.13	99.5
SVM	93	93	92.4	92.5	95

Comme prévu, les systèmes basés sur les classifieurs bayésiens ont les meilleurs taux de reconnaissance. Dans ce contexte, le meilleur système de catégorisation pour cette base des données est le classifieur Bayes + gain d'informations avec un Taux = 98,7%.

Utilisant la base des données BBCNEWS, les tableaux 34 et 35 montrent une comparaison globale des différents classifieurs utilisés, combinés avec l'ensemble des caractéristiques, sélectionnées par les méthodes de sélection des attributs, et la représentation vectorielle FTF-IDF. La comparaison conserve que les meilleures performances.

Tableau 34: Résultats du système de classification, des BBCNews data, basés sur la pondération FTF-IDF et IG comme méthode de sélection d'attributs.

	Precision %	Recall %	F-measure %	Accuracy %	AUC%
Naïve Bay	93	93	93	93.03	97.4
Bay Update	93	93	93	93.03	97.4
Random Forest	80	80	79.6	80	94.3
SVM	71	71.15	71.4	71.5	87.3

Tableau 35: Résultats du système de classification, des BBCNews data, basés sur la pondération FTF-IDF et Relief comme méthode de sélection d'attributs.

	Precision %	Recall %	F-measure%	Accuracy %	AUC%
Naïve Bay	94	94	94	94	98
Bayesian net	94	94	94	94	98
Bay Update	92.6	92.6	92.6	92.62	97.2
Random Forest	83	82	82	82	95
SVM	72	72	72	72	87.3

Les résultats obtenus prouvent que la pondération floue des termes, FTF-IDF, est en concurrence avec les méthodes de pondération existantes dans la littérature. Les modèles d'apprentissage automatique proposés pour la classification, obtiennent des résultats satisfaisants, même si la taille des données augmente.

Les méthodes de réduction ou la sélection des caractéristiques ont influencé le rendement des classifieur. Également, Les résultats donnés prouvent que la reconnaissance le taux de classificateurs évolue parallèlement à l'échange des méthodes de sélection des features.

La qualité de la pondération peut être améliorée en améliorant les règles d'inférence et ça sera l'hypothèse sur laquelle est fondé la prochaine contribution qui permet d'avoir une bonne prédiction des poids ainsi qu'un descripteur vectoriel flou pertinent.

2.4 Application de l'automatisation des règles d'inférence au processus FTF-IDF

Avant de détailler les composants des systèmes adoptés, nous défendons les raisons de pratiquer cette approche. Tout d'abord, la représentation FTF-IDF dite optimale est l'une des représentations vectorielles la plus recommandées dans la littérature. Aussi la FTF-IDF est souvent utilisé dans des systèmes décisionnels flous robustes, par exemple, les systèmes de recommandation, de classification, et de la recherche d'informations, et où la représentation des données a une influence significative. En outre, les systèmes donnés utilisent une grande masse de données, ce qui nécessite un traitement précis et raffiné. Pour cela, l'utilisation du SIF doit être correctement automatisée pour gagner en temps de calcul.

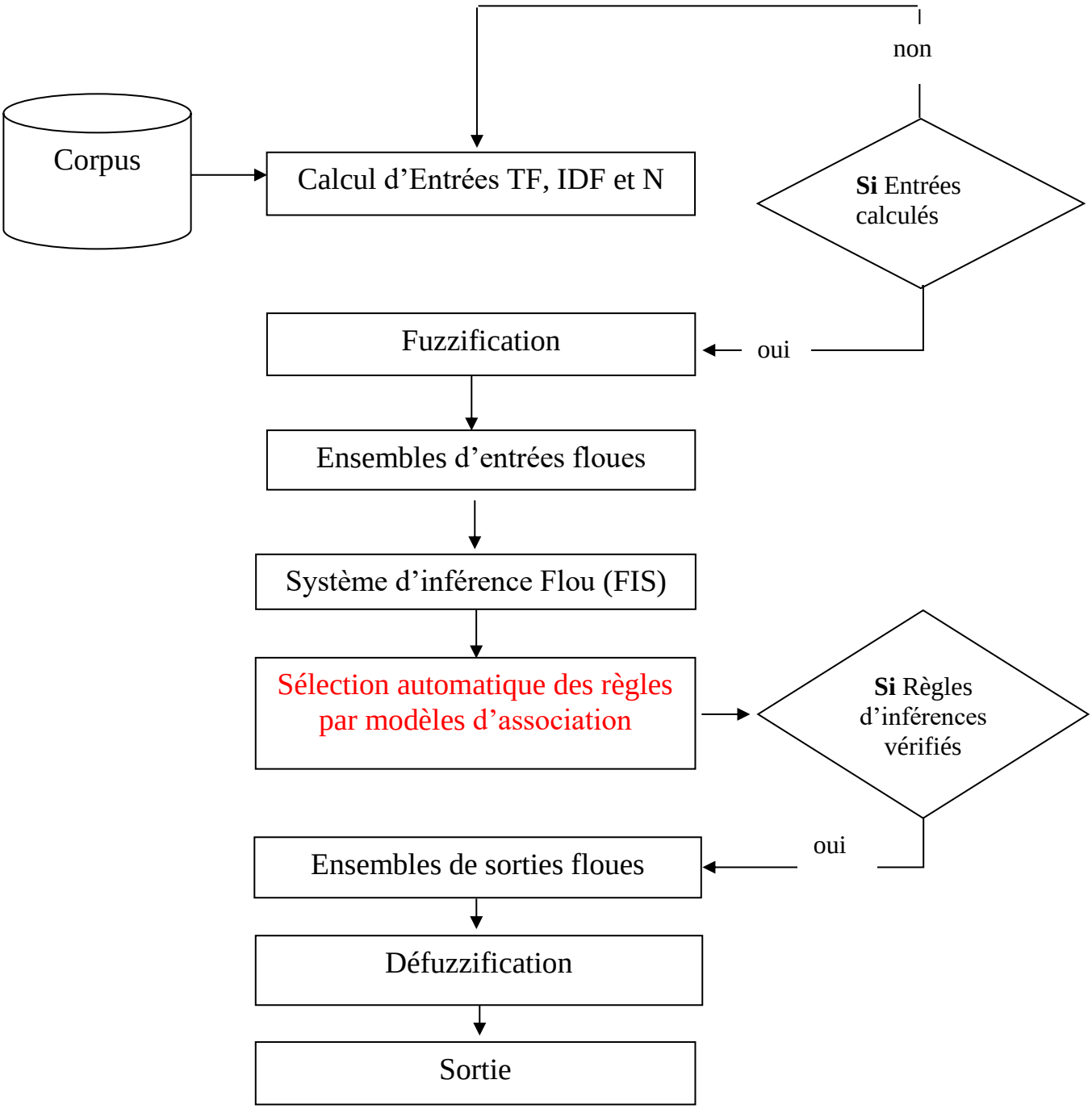

Figure 30 : Organigramme du système d'inférence floue pour déterminer la pondérée floue FTF-IDF optimale.

C'est pourquoi notre contribution a pris en compte la minimisation du temps de traitement des données, ainsi que la précision des sorties du SIF. Par d'un ensemble des tests, nous avons pu constater l'importance des règles d'association, qui nous permettent de générer, à partir d'un ensemble de données, des règles d'inférence pertinentes qui améliorent les sorties du SIF et les poids des données textuelles [153].

L'organigramme présent dans la figure 30, organise les étapes pour la conception de la pondération floue FTF-IDF optimale.

Pour tester le descripteurs FTF-IDF générés par le SIF automatisé, nous proposons de l'intégrer dans un système de classification, qui se base sur certains classifieurs d'apprentissage automatique (comme les réseaux bayésiens). Le choix des réseaux bayésiens revient aussi à leur réputation pour la classification des corpus de textes et leurs obligations de précision des descripteurs d'entrée. La non-existence des sources qui regroupent les composantes de notre approche, nous a poussés à réaliser cette contribution, qui pourrait aussi être une base solide pour la logique floue et les utilisateurs du ML.

Les paragraphes suivants décrivent les améliorations proposées sur la sélection des règles floues pour améliorer les sorties FTF-IDF.

2.4.1 Option automatique des règles d'inférence pour la FTF-IDF

La génération des poids flous, cette fois ci, suit les processus du SIF dont le processus de génération des RI est automatisé. Nous utilisons les règles floues de type **SI-ALORS** générées automatiquement par les modèles d'association afin de produire des descripteurs flous. Comme indiqué dans la partie III de ce chapitre, nous commençons par le prétraitement des données de la pondération floue (FTF-IDF décrite dans la partie IV de ce chapitre spécialement le section 2.1.1). Puis la représentation nominale des entrées (TF, IDF, N) et sortie (Weight W) de la représentation floue est nécessaire pour appeler les modèles d'association qui permettent la sélection des RI pertinentes. Nous employons dans cette approche les mêmes algorithmes d'association, à savoir l'Apriori et le filtre d'association, qui permettent de générer les trois types des règles mentionnés par avant (IR, ER, et RR). De même les règles de type RR seront rejeter, cependant nous gardons les règles explicites (ER) et un poste traitement est recommandé pour les règles implicites (IR). Les résultats algébriques du post traitement sont expliqué dans la section III. 2.2 de ce chapitre).

Comme mesure de performance des règles sélectionnés pertinente, nous utilisons la mesure confiance, gardant les règles qui ont une confiance supérieure à 90%.

L'option automatique des règles d'inférence pour la FTF-IDF, aura surement un impact sur la qualité des descripteurs flous, ainsi que la performance de la classification automatique. Pour cela, nous proposons dans la section suivante, de démontrer que la nouvelle approche FTF-IDF présente des scores de performance significative pour un ensemble des classifieur de l'AA, en comparant avec la FTF-IDF et autres techniques populaires.

2.4.2 Impact de la nouvelle FTF-IDF sur la performance des classifieurs ML

Pour tester les performances de la nouvelle technique, qui produit un ensemble des caractéristiques à partir des données textuelles, nous proposons classer les données d'actualités de la BBC en utilisant :

- La représentation FTF-IDF générée avec le nouveau SIF
- Un ensemble de classifieurs bayésiens.

Les résultats de la comparaison réalisée sont présents dans cette section, pour prouver l'efficacité de l'automatisation du processus du contrôleur flou.

La K-validation croisée avec k = 10, est appliquée comme moyen pour prédire l'efficacité du classifieur. De plus, pour sélectionner les résultats satisfaisants, nous avons employé différents modèles bayésiens supervisés, pour classer l'actualité de la BBC électronique, les résultats obtenus sont présentés ci-dessous.

Dans cette partie, nous comparons l'efficacité de la technique suggérée pour créer une représentation textuelle floue (FTF-IDF). Ainsi, nous proposons une étude comparative utilisant deux types de FTF-IDF (simple FTF-IDF et améliorer FTF-IDF) et d'autres méthodes comme TF-IDF, la fréquence du terme et la fréquence du terme inverse.

Tableau 36 : Les résultats de la classification bayésienne pour BBCSport Data, en utilisant la représentation FTF-IDF simple et améliorée.

	With simple FTF-IDF			**With the ameliorate FTF-IDF**		
	Recall %	**Precision %**	**Accuracy %**	**Recall %**	**Precision %**	**Accuracy %**
Bayes Net	95	95	95	97.1	97	97
Bayes Naïve	92 - 98.6	90 - 98.7	90 - 98	95 - 98.9	95 - 99	95- 98.9
Bays Update	98.7	98.7	98.6	99.3	99.2	99.1

En effet, la méthode proposée, qui est la FTF-IDF améliorée, présente des résultats de classification satisfaisants, où la classification de la base des données BBCSport à l'aide des classifieurs bayésiens, comme indiqué dans le tableau 36, note des progrès significatifs par rapport aux résultats FTF-IDF simples.

La précision donnée qui est égale à 99%, dans le tableau 36, prouve que l'automatisation des règles pour produire le poids FTF-IDF a un excellent impact sur la vectorisation textuelle et la décision supervisée.

D'autre part, nous recommandons de tester l'effet de la nouvelle représentation sur la tâche de classification de texte, en utilisant BBCNews comme base de données massive. Le tableau 37 présente la progression des résultats de classification en utilisant la nouvelle technique pour générer une représentation FTF-IDF pour les termes de la base de données donnée.

Tableau 37: Les résultats de la classification bayésienne pour BBCNews Data, en utilisant la représentation FTF-IDF simple et améliorée.

	With simple FTF-IDF			With the ameliorate FTF-IDF		
	Recall %	**Precision %**	**Accuracy %**	**Recall %**	**Precision %**	**Accuracy %**
Bayes Net	93 - 94	93 – 94	93 - 94	96	96	96
Bayes Naïve	94	94	94	97	97	97
Bays Update	93	93	93	96	96	96

Par conséquent, la nouvelle méthode améliore les performances de la représentation FTF-IDF, où la classification statistique utilisant comme entrées les vecteurs flous, prouve une amélioration significative de la précision comme indiqué dans les résultats des tableaux.

De plus, au cours des expérimentations, nous avons marqué un gain significatif en temps, en règles de précision et en qualité de pondération. En effet, la technique proposée peut produire des effets satisfaisants dans différents domaines.

Pour confirmer l'efficacité de l'approche adoptée, c'est-à-dire la pondération FTF-IDF amélioré, nous proposons une étude comparative avec d'autres mesures qui servent à déterminer les poids des termes en langage naturel. Nous contribuons en tant que méthodes populaires :

- Le schéma TF-IDF : calcule le poids de chaque terme dans le corpus en utilisant la formule TF- IDF expliquée dans le premier chapitre.
- La fréquence locale (TF) : fréquence du terme uniquement sur le document.
- L'IDF : fréquence globale du terme.
- La simple FTF-IDF.
- La FTF-IDF développé.

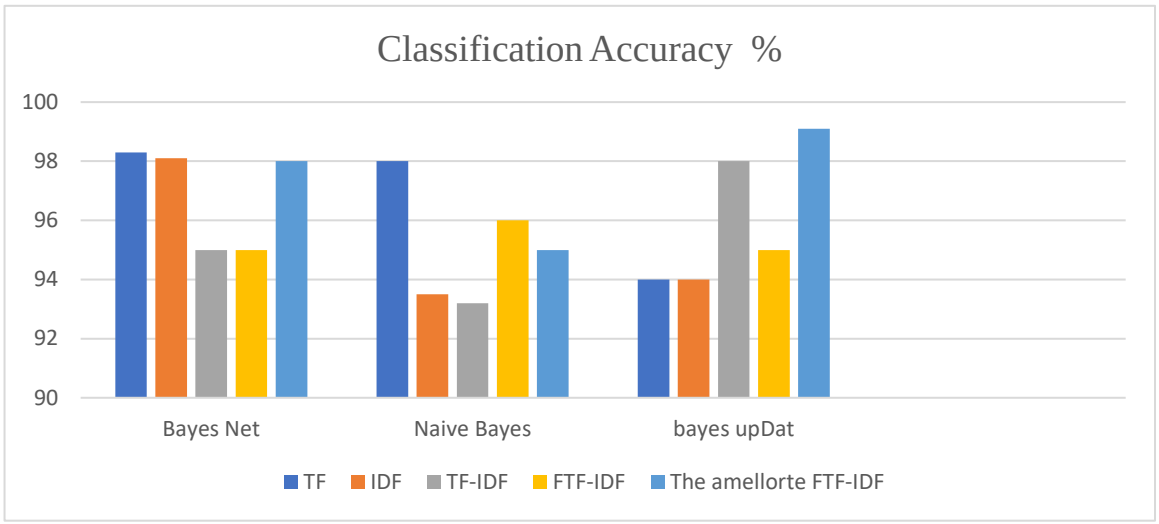

Figure 32: Résultats de la classification utilisant les classificateurs Bayésiens, une collection des méthodes de pondération et les données BBCSPORT.

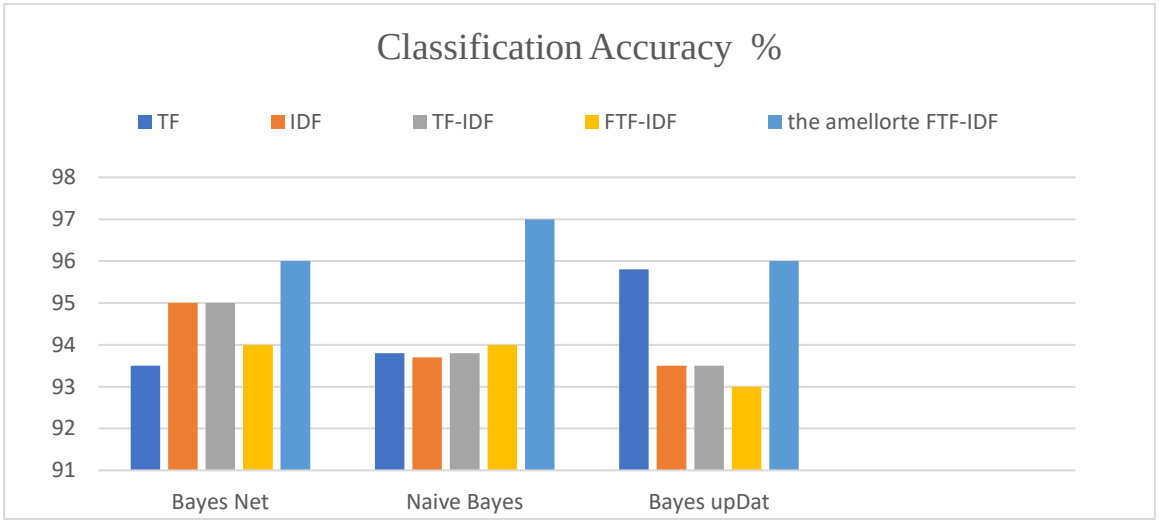

Figure 31: Résultats de la classification utilisant les classificateurs Bayésiens, une collection des méthodes de pondération et les données BBCNEWS.

Les figures 31 et 32 illustrent la précision de la classification à l'aide des méthodes de pondération des données textuelles, des classifieurs bayésiens et des ensembles de données BBC. Encore une fois, les résultats prouvent l'efficacité de l'approche adoptée par rapport aux méthodes populaires (en se basant sur les résultats des figures 31 et 32) pour calculer les poids flous et la puissance du FTF-IDF avec le réseau bayésien comme classifieur du ML.

V. Conclusion

Contrairement à la technique traditionnelle de sélection des RI pour un SIF, qui se base principalement sur l'intervention des experts, la nouvelle intervention se concentre sur l'automatisation du choix de ces règles. L'approche proposé se base, particulièrement, sur l'intégration des modèles d'association (comme le modèle Apriori et Filtre d'association) accompagne d'un ensemble de processus (à savoir le prétraitement et le posttraitement) afin de des règles de contrôle pertinentes.

Dans ce chapitre nous avons présenté deux applications de l'approche de sélection automatique des règles d'inférence, dont :

- La première est la classification floue des données iris, où la précision du système adopté est de 97%.
- La deuxième contribution porte sur l'amélioration de la qualité de la pondération floue (FTF-IDF) pour améliorer les performances du système de classification automatique. Effectivement, les résultats de classification ont été excellent, où la classification des bases des données de références à l'aide du classificateur bayésien atteint une précision de 99%.

En outre, en comparant avec un ensemble de techniques de pondération populaires, notre approche donne généralement les meilleurs scores. Généralement, les contributions présentes dans ce chapitre encouragent les praticiens dans le domaine du contrôle des systèmes intelligents à les utilisées pour des problèmes plus complexes.

Durant l'implémentation de la pondération floue, nous avons rencontré quelques lacunes de la LF en générale, qui empêche, à un certain niveau, l'exactitude des descripteurs flous. À cet égard, dans le prochain chapitre nous allons découvrir la Logique Neutrosophique, ses avantages ainsi que son utilité pour une représentation vectorielle plus puissante.

Chapitre 4 : Nouveau modèle neutrosophique avancé pour la rénovation de la vectorisation floue des données textuelles.

Chapitre 4 : Nouveau modèle neutrosophique avancé pour la rénovation de la vectorisation floue des données textuelles

I. Introduction

Étant donné que l'appariement flou des entrées et le traitement flou qui en résulte est un domaine de recherche actif, il a été appliqué avec succès pour traiter et analyser des données textuelles à l'aide d'un ensemble de processus constituant le système d'inférence floue (FIS) [155]. Cependant, la LF ne pourrait pas simuler des faits qui sont neutres ou qui nécessitent un examen attentif des degrés d'incertitude, d'ambiguïté et d'indétermination. Pour cela, Florentin Smarandache a inventé une nouvelle logique appelée 'Neutrosophic Logic' (Logique Neutrosophique LN) [156], et qui a compensé le défaut de la LF [155].

Récemment, un ensemble de systèmes dans le contexte Text Mining a été développé en adoptant la nouvelle logique neutrosophique, et cette mise à jour comprenait des systèmes de classification et de recherche [155] [157] [158]. Néanmoins, cette rénovation n'a introduit que les dernières étapes de mise en œuvre et n'a pas atteint l'étape la plus critique du système d'analyse de texte, qui est l'étape de pondération des termes ou de représentation des caractéristiques [159].

En général, la représentation et la sélection des caractéristiques affectent la réalisation des tâches du Text Mining [160], par exemple, une vectorisation précieuse des textes a un impact positif sur la performance des systèmes de la classification des textes [16]. L'une des méthodes de pondération les plus populaires est la TF-IDF, et le SIF a été utilisée pour calculer la TF-IDF floue (FTF-IDF), où les ensembles flous et les degrés d'appartenance sont appliqués pour définir les poids flous des termes du document [154] [29]. Le poids final dépend uniquement de l'occurrence du mot sans tenir compte de sa valeur de pertinence ou du degré d'ambiguïté.

Dans ce chapitre, nous proposons la neutrosophique TF-IDF (NTF-ID) basée sur le Système d'Inférence Neutrosophique (SIN) et les ensembles neutrosophique [161] (Smarandache). Notre proposition est une version étendue de la pondération fréquentielle floue et une nouvelle technique de pondération, pour améliorer la précision de la FTF-IDF [162].

La contribution proposée vise à :

- Produire des vecteurs descripteurs pour la tâche de classification des textes,
- Calculer les poids en fonction de leurs fréquences locales et globales [145]
- Considérer le degré d'imprécision de chaque entrée pour les SIN.

L'objectif est de présenter plus d'informations sur les termes qui représentent un document en incluant l'ambiguïté du terme donné. Il est nécessaire de savoir qu'un petit degré d'ambiguïté implique un degré de pertinence élevé des mots. En outre, la sélection des termes ambigus réduit la taille finale des descripteurs et renforce leur efficacité. Ainsi, les résultats de la SIN sont des pondérations informatives basées sur le degré absolu de pertinence et d'ambiguïté [163].

Comme la NTF-IDF considère trois zones de décision [156] à savoir : vrai, intermédiaire et faux, cela permet de faire progresser la qualité de la pondération des termes en résolvant l'ambiguïté présentée dans le chevauchement entre les ensembles flous et représentant les fréquences de termes ambigus. Notre nouvelle méthode [162] utilise comme entrées les

composantes FTF-IDF : la fréquence des termes TF, la fréquence inverse des termes IDF, et la longueur du document N, exprimée dans l'espace neutrosophique 3D [156]. En effet, la valeur associée à chaque composant des ensembles neutrosophiques est distinguée, différemment de la FTF-IDF, où le chevauchement entre les ensembles flous représente l'appartenance de fréquence ambiguë qui définit l'ambiguïté du mot. Le SIN utilisé donne, en sortie, deux types de poids : Le poids de vérité (W_t), c'est-à-dire, le poids pertinent et le poids ambigu (Wi). Tant que la valeur Wi n'est pas supérieure à W_t, le terme concerné est ambigu et n'est pas essentiel pour représenter un document.

En revanche, la fonction de combinaison Zhang [164] est appliquée pour combiner les sorties des poids neutrosophiques, et présenter le poids final du terme, inséré dans le vecteur représentatif du document. La partie expérimentale prouve la performance de la version étendue de FTF-IDF, c'est-à-dire NTF-IDF ; où les résultats obtenus montrent l'amélioration des scores de classification, en utilisant un ensemble de données de références et une variété des classifieurs du ML : SVM et Feed-Forward Network.

Le reste de ce chapitre est structuré comme suit : après avoir une idée sur la LN et les SINs, nous présentons les observations et les motivations derrière le développement de la nouvelle méthode de pondération. Ensuite, nous expliquerons la démarche de pondération NTF-IDF (la théorie de NTF-IDF et son fonctionnement). Avant de conclure le chapitre nous présenterons les résultats des systèmes de catégorisation des textes, basés sur la nouvelle représentation neutrosophique et autres mécanismes.

II. Généralité sur la logique Neutrosophique

1. Logique Boolean, floue et neutrosophique

Un des outils de l'intelligence artificielle est la logique floue inventait par Zadeh comme logique évolutionniste de la logique Boolean. Pour développer l'algèbre de Boole, la LF remplace la valeur de vérité d'une proposition limitée au choix {vrai, faux} ou {0,1} par un degré d'appartenance, qui appartient à l'intervalle [0, 1]. Par la suite une nouvelle version intuitionniste de la LF a été développée, l'idée de base de cette nouvelle logique est de caractériser chaque affirmation logique dans un espace 2D, où chaque dimension de l'espace représente, respectivement, le degré de vérité (T%) et le faux degré (F%) de la proposition étudiée, avec T%, F% sont les sous-ensembles réels [165].

Ainsi pour élargir l'espace de représentation des degrés d'appartenance, et en se basant sur les cas d'études traités par la LF, Smarandache a introduit la logique neutrosophic. Cette logique permet de projeter les degrés d'appartenance dans un espace de 3D pour une bonne simulation réaliste des phénomènes. Le tableau 38 résume la description des trois logiques citées, c.à.d., logique Boolean, floue et neutrosophic afin de distinguer entre eux.

Tableau 38: Généralités sur la logique Boolean, floue et neutrosophique.

Logique	Description
Logique Boolean (LB)	Les fonctions d'appartenance classiques varient dans la plage {True, False} ou {0, 1}. Par conséquent, les applications BL dans des domaines imprécis et incertains sont limitées. De cette manière, une représentation binaire des données textuelles a été adoptée par [166] mais les faiblesses de cette approche, par exemple la perte d'informations, l'ont rendue exclue et inefficace.
Logique Floue (LF)	La LF se base sur la définition de la valeur degré d'appartenance (T) d'une variable par un nombre qui varie dans l'intervalle [0, 1] progressivement, contrairement à la logique classique binaire qui manipule seulement les valeurs 0 et 1. Aussi la théorie des sous ensemble floues adopté par la LF est diffèrent à la théorie Boolean des ensembles.
Logique Neutrosophic (LN)	L'idée de base de la LN est de caractériser chaque variable logique dans un espace Neutrosophique en 3D, où chaque dimension de l'espace représente, respectivement, le degré vérité (T%), le faux degré (F%) et le degré d'indétermination (I%) de la variable en considération. Différemment à la LF les T%, I%, F% sont les sous-ensembles réels standards ou non-standards de $]^{-}0, 1^{+}[$, considérés aussi comme composants indépendantes.

La logique neutrosophic est la logique généraliste de la LF et la version LF intuitionniste. Vue les performances de la LN à étudier des phénomènes indéterministe, compatible au traitement suivit par les systèmes d'analyse des données en langage naturel, notre intérêt se focalise sur cette logique afin d'inventer des nouvelles applications au domaine du TALN.

2. Logique Neutrosophique (LN)

La LN a été proposée par Florentine Smarandache qui est basée sur l'analyse non standard donnée par Abraham Robinson (2016) [166] dans les années 1960. Tous les énoncés de la logique neutrosophique simulent la pensée humaine, car il est très difficile de conclure à cause de l'imprécision des systèmes humains qui pourrait être due à l'imperfection des connaissances que l'homme reçoit (observation) du monde extérieur [155]. Généralement, la logique neutrosophique a été développée pour représenter un modèle mathématique d'incertitude, d'imprécision, d'ambiguïté, d'incomplétude, d'incohérence, de redondance et de contradiction [155]

2.1 Sous-ensembles neutrosophique

Chaque raisonnement, floue, intuitionniste et neutrosophique, adopte la notion sous-ensembles (SE) afin de simuler, concevoir et modéliser le phénomène étudié. La définition des

SE diffère selon la logique adoptée et ses caractéristiques, ainsi que la table suivante (tableau 39) propose le descriptif principal de l'ensemble et du SE qui dépendent à chaque logique.

Tableau 39: Définitions des sous-ensembles pour l'ensemble des logiques : floue, intuitionniste et neutrosophique.

Sous-ensemble Floue	Sous-ensemble intuitionniste	Sous-ensemble Neutrosophique
• Introduit par : L. Zadeh • Ensembles flous est une généralisation de la notion de l'ensemble classique. • Soit X un ensemble de référence, *un sous ensemble flou* A de X est l'ensemble des tuplet : • $A= \{<x, T_A(x)> :x \epsilon X\}$ • Avec $T_A(x) \epsilon [0, 1]$ est le degré d'appartenance de x à A.	• Introduit par : K.T.Atanassov • Ensembles Flous Intuitionniste est une généralisation de la notion de l'ensemble flou. • Soit X un ensemble de référence, *un sous ensemble flou intuitioniste* A de X est l'ensemble des tuplet : • $A= \{<x, T_A(x), F_A(x)> :x \epsilon X\}$ • Avec, $T_A(x)$, $F_A(x)$ sont des applications de X dans [0, 1] et qui vérifient la condition : $0 \leq T_A(x)+F_A(x) \leq 1$ • $T_A(x)$: est le degré d'appartenance de x à A. • $F_A(x)$: est le degré de non-appartenance de x à A.	• Introduit par : F. Smarandache • Ensembles Neutrosophique est une généralisation de la notion de l'ensemble flou et l'ensemble floue intuitionniste. • Soit X un ensemble de référence, *un sous ensemble neutrosophique A* de X est l'ensemble des tuplet : • $A= \{<x, T_A(x), I_A(x), F_A(x)> :x \epsilon X\}$ • Avec $T_A(x)$, $I_A(x)$, $F_A(x)$ sont des applications de X et qui vérifient la condition : $0 \leq T_A(x)+ IA(x)+ F_A(x) \leq 3$ • $T_A(x)$: est le degré d'appartenance de x à A. • $I_A(x)$: est le degré d'indétermination de x à A. • $F_A(x)$: est le degré de non-appartenance de x à A.

Différemment aux autres raisonnements cités dans la table ci-dessous, en raisonnement Neutrosophique chaque proposition est estimée avoir le pourcentage de vérité dans un sous-ensemble T, le pourcentage d'indétermination dans un sous-ensemble I et le pourcentage de fausseté dans un sous-ensemble F, où T, I, F sont standard ou non standard sous-ensembles réels de $]^{-}0,1^{+}[$ et qui sont notamment indépendantes [166].

En revanche, pour un espace de points X, avec un élément noté x dans X, l'ensemble neutrosophique A dans X est caractérisé par une fonction d'appartenance de vérité T_A, une fonction d'appartenance d'indétermination I_A et une fonction d'appartenance de fausseté F_A. T_A (x), I_A (x) et F_A (x) sont des sous-ensembles réels standard ou non standard dans l'intervalle $]^{-}0,1^{+}[$, où :

$$T_A : X \rightarrow]^{-}0, 1^{+}[, I_A : X \rightarrow]^{-}0, 1^{+}[, F_A : X \rightarrow]^{-}0, 1^{+}[$$

Généralement, il n'y a aucune restriction sur la somme des sous-ensembles T_A (x), I_A (x) et F_A (x), d'où :

$$^{-}0 = \sup T_A(x) + \sup I_A(x) + \sup F_A(x) = 3^{+}$$

En effet, l'idée neutrosophique permet de refléter la dynamique des choses et des idées étudiés à titre d'exemple la proposition estimée sur le succès d'un film n'a pas de valeur fixe, où la valeur de vérité de la proposition peut changer d'une paye à autre par exemple.

En revanche, la proposition que le film serait échoué peut donner des composantes neutrosophiques du sort : T= 0% (vrai), I= 0% (indéterminé) et F= 100% (faux). Par conséquence la LN permet également de modifier les valeurs par rapport à l'observateur, où par exemple : la proposition que le film réussirait, peut produire des composants neutrosophiques (t = 0,50, i = 0,40, f = 0,30) ; comme il peut avoir autre score selon les juges.

Généralement, le concept de sous ensemble permet des graduations dans l'appartenance d'un élément à une classe selon la logique employée. Par conséquent, les sous-ensembles adoptés dans le concept neutrosophique permet sa flexibilité afin de modéliser les imperfections des données ce que le rend plus proche au raisonnement humain.

2.2 Opérateurs Neutrosophiques

Chaque logique a ses propres opérations sur les ensembles et les sous-ensembles associés. Afin d'exploiter le raisonnement neutrosophique, par exemple, pour l'inférence des systèmes d'expert neutropéniques, il est essentiel de connaitre et de définir les opérations ou les opérateurs de la LN.

Dans le tableau 40 présente les connecteurs logiques qui se diffèrent en fonction du problème à résoudre. La table contient les connecteurs flous comme sort de comparaison afin de distinguer les opérateurs de base neutrosophique, qui seront l'objectif de notre contribution.

Tableau 40: Les opérateurs des logiques neutrosophique et floue.

La logique Neutrosophique:	La logique Floue:
• Soit U un univers de discours, et M un ensemble inclus dans U. Un élément x de U est noté par rapport à l'ensemble M comme (T (x), I (x), F (x)) et appartient à M de la manière suivante : il est t% vrai dans l'ensemble, i% indéterminé (inconnu s'il l'est) dans l'ensemble, et f% faux, où t varie en T, i varie en I, f varie en F.	• Soit U un univers de discours, et M un ensemble inclus dans U. la fonction d'appartenance d'un élément x de U est notée par rapport à l'ensemble M comme $\mu_{M(x)} \in [0, 1]$
• Pour la logique neutrosophique à valeur unique (t, i, f), la somme des composantes est : $0 \leq t+i+f \leq 3$ lorsque les trois composants sont indépendants [164] ;	• Pour un ensemble des propositions, l'intervalle unitaire classique [0, 1] est utilisé. • Nous pouvons également définir les partitions floues (M_1, M_2, ..., M_j, ..., M_N) où : $\forall x \in X \sum \mu_{Mj}(x)=1$ with$Mj \neq \emptyset$ and $Mj \neq X \forall\ 1< j <N$

$0 \leq t+i+f \leq 2$ lorsque deux composants sont dépendants [164], tandis que le troisième est indépendant d'eux ; $0 \leq t+i+f \leq 1$ lorsque les trois composants sont dépendants [164]. • Les *opérateurs neutrosophiques*, Pour les propositions A et B notées respectivement par LN (A) = (T_1, I_1, F_1) et LN (B) = (T, I, F), sont présentés comme suit : *<u>*Union*</u> NL(AUB) = $(\boldsymbol{T_1} \oplus \boldsymbol{T} \ominus \boldsymbol{T_1} \odot \boldsymbol{T}, \boldsymbol{I_1} \oplus \boldsymbol{I} \ominus \boldsymbol{I_1} \odot \boldsymbol{I}, \boldsymbol{F_1} \oplus \boldsymbol{F} \ominus \boldsymbol{F_1} \odot \boldsymbol{F})$ *<u>*Intersection*</u> NL(A∩B) = ($T_1 \odot$T, $I_1 \odot$I, $F_1 \odot$F) *<u>*Complément*</u> LN ($\overline{\boldsymbol{A}}$)= ({1+} ⊖ T1, {1+} ⊖ I1, {1+} ⊖ F1) *<u>*Equivalence*</u> NL(A<->B) = (({1+} ⊖$T_1 \oplus \boldsymbol{T1} \odot \mathbf{T}$) ⊙ ({1+} ⊖T⊕ $\boldsymbol{T1} \odot \mathbf{T}$)), (({1+} ⊖I1⊕ $\boldsymbol{I1} \odot \mathbf{I}$) ⊙({1+} ⊖I⊕ $\boldsymbol{I1} \odot \mathbf{I}$)), (({1+} ⊖F1⊕ $\boldsymbol{F1} \odot \mathbf{F}$) ⊙ ({1+} ⊖F⊕ $\boldsymbol{F1} \odot \mathbf{F}$))	• Soit A et B deux sous-ensembles flous de X définis par les fonctions d'apprentissage, *les opérateurs* flous sont présentés sous forme de flux : * <u>*Égalité*</u> A=B only if ∀x∈ X / $\mu_{A(x)} = \mu_{B(x)}$ *<u>*Inclusion*</u> A⊂B only if ∀x∈ X/$\mu_{A(x)} < \mu_{B(x)}$ *<u>*Intersection*</u> ∀x∈ X/$\mu_{A\cap B(x)} = \min(\mu_{A(x)}, \mu_{B(x)})$ *<u>*Union*</u> ∀x∈ X/$\mu_{AUB(x)} = \max(\mu_{A(x)}, \mu_{B(x)})$ * <u>*Complément*</u> $\mu_{\bar{A}(x)} = 1-\mu_{A(x)} \forall x \in X$

III. Systèmes d'Inférence Neutrosophique (SIN)

Les ensembles, les sous-ensembles et les opérateurs neutropéniques sont les composantes d'un système expert neutrosophique. En revanche, pour prendre une décision neutrosophique, un système d'inférence neutrosophique (NIS) a été proposé dans la littérature, où les principaux processus sont présentés par la figure 33.

Figure 33: Processus du système d'inférence Neutrosophique (SIN).

Le système d'inférence neutrosophique consiste en une étape de Neutrosophication qui attribue chaque entré (crispe input) à la fonction d'appartenance appropriée. La base de connaissances neutrosophique lie l'entrée à la variable de sortie et l'étape de Deneutrosophication relie la

valeur neutrosophique à la valeur de sortie nette (crispe output) (comme la figure 33 illustre). Généralement, nous pouvons définir chaque processus par :

- Processus de Neutrosophication : Les mesures acquises par le système neutrosophique sont ensuite converties en ensembles neutrosophiques appropriés pour capturer les mesures de vérité, de fausseté et d'indétermination, en utilisant respectivement les fonctions de vérité, de fausseté et d'indétermination.
- Moteur d'inférence neutrosophique : dans ce processus, les mesures générées par l'étape de Neutrosophication sont ensuite utilisées par le moteur d'inférence pour évaluer les règles de contrôle. En effet, les règles sont placées dans la base des règles neutrosophiques, où la Création de règles d'ensemble neutrosophiques pour les trois bases de connaissances : vrai, faux et indétermination. Cette évaluation conduira à un ensemble neutrosophique ou à plusieurs ensembles neutrosophiques qui seraient sur l'univers des actions possibles.
- Processus de Deneutrosophication : le dernier processus du SIN, où la sortie nette pour chaque entrée est générée compte tenu de leurs degrés : de vérité, d'indétermination et de fausseté.

Vu que le modèle neutrosophique à la capacité à exprimer divers types de données d'incertitude spécialement l'ambiguïté, la LN et le SIN sera la base de notre contribution sur la vectorisation neutrosophique des données textuelles non-structurées, comme montreront les sections suivantes [168].

IV. Application du raisonnement neutrosophique pour la rénovation de la vectorisation floue des données textuelles.

Regardant toutes les propositions et les versions de la pondération TF-IDF, nous remarquons que la notion degré d'ambiguïté est absente, et elle dépend toujours de la fréquence du mot [21], du raisonnement mathématique complexe ou du but de classification [21]. Comme il était mentionné la LN prend en considération l'ambigüité et l'incertitude des données, pour cela elle était adoptée avec les SIN pour le développement de plusieurs systèmes d'experts, comme les systèmes de classification qui ont subi des transformations favorables [155], et sera adopté de notre part pour le développement d'une pondération avancé des termes, en langage naturel, qui prend en considération le degré d'ambiguïté pour le calcul des poids plus précis de ces termes.

En résumant, cette section propose une nouvelle théorie de la pondération des mots, prenant son degré d'ambiguïté pour représenter un document via l'extension de la représentation floue FTF-IDF par raisonnement neutrosophique.

1. Principales observations et motivations

Comme mentionné dans le chapitre précédent, le SIF est souvent utilisé pour le développement des systèmes des experts, où la pluralité de ces systèmes ont été améliorés grâce à l'emploi des SIN qui ont démontré une précision compétitive. Dans le contexte Text Mining, la pondération des termes d'index, FTF-IDF, appliquée dans les systèmes TALN flous, néglige l'ambiguïté des mots malgré sa forte présence en utilisant les ensembles flous. Par conséquent,

il est urgent de surmonter cet obstacle en utilisant la Neutrosophique pondération TF-IDF (NTF-IDF), qui donne l'opportunité de relier la pondération fréquentielle avec le degré d'ambiguïté du terme. Cependant, l'exploitation de la LN pour la production des poids des termes en langage naturel est absente dans la littérature. En revanche, la projection des imputes FTF-IDF dans l'espace neutrosophique 3D permet d'intégrer le degré d'ambigüité d'un terme qui représente un document, et de marquer une précision considérable pour les poids des termes. Plus expressif, il faut savoir qu'un petit degré d'ambiguïté signifie un degré de pertinence élevé pour chaque mot. De plus, la sélection de termes ambigus réduit la taille finale des descripteurs et renforce leur efficacité.

Pour valider les caractéristiques fournies, nous proposons d'impliquer la NTF-IDF, dans un système de classification, avec les méthodes de sélection des variables pour réduire la dimensionnalité des entrées des classifieurs ML et pour éviter la perte de l'information pertinente. De plus, les résultats de la classification textuelle, utilisant des classifieurs d'apprentissage automatique sensibles aux entrées fournies, pourraient montrer que notre proposition est utile et efficace.

2. Pondération Floue FTF-IDF et ses lacunes

Nous rappelons que pour déterminer les poids FTF-IDF, il est nécessaire de suivre le processus du système d'inférence floue, où les valeurs de la fameuse pondération TF-IDF [12], c.à.d., TF, IDF et N sont traitées comme des variables d'entrée pour la phase de fuzzification, avant de déterminer l'univers du discours.

Généralement, la première étape de Fuzzification convertit les valeurs d'entrée nettes en degrés d'appartenance à l'aide du formulaire de fonction d'appartenance. La fonction FTF-IDF utilise des termes linguistiques pour représenter toutes les variables d'entrée et de sortie. Ces termes linguistiques sont représentés par des fonctions d'appartenance, obtenues à partir du domaine de la connaissance. La fonction d'appartenance du type triangulaire est utilisée pour modéliser les degrés d'appartenance de toutes les variables du SIF composite. La plage des variables d'entrée TF, IDF et N pour le SIF sont représentées comme très élevée (VH), élevée (H), moyenne (M), faible (L) et très faible (VL). La plage de poids est représentée par des fonctions d'appartenance élevée (H), moyenne (M) et faible (L). Ensuite, la phase des règles floues déduit le poids final de chaque terme, en fonction des variables d'entrée : TF, IDF et N. Lors de la dernière étape de défuzzification, nous utilisons le fameux calcul centroïde, qui permet de définir le poids, une seule sortie, pour chaque indice de terme, en se basant sur l'ensemble flou de sortie agrégée. Il est à noter que les autres techniques donnent les mêmes résultats dans notre cas.

La FTF-IDF néglige le degré d'informativité du terme pour un document donné, et donc l'imprécision du poids généré est absente. En effet, durant, la phase fuzzification nous remarquons le chevauchement entre les ensembles qui modélisent les variables fréquentielles d'entrées (TF, IDF) ce qui empêche une claire appartenance à la classe convenable de la fréquence du terme. Cette imprécision produit une ambiguïté au niveau d'informativité du terme d'un document, également le LF n'est pas capable d'étudier le phénomène d'incertitude

et d’ambigüité et qui sollicite le déplacement vers la modélisation neutrosophique pour une pondération des termes plus précise.

3. Nouvelle pondération neutrosophique NTF-IDF

3.1 Architecture du SIN pour la déduction de la NTF-IDF

Généralement, l'intervalle de neutralité, donné par le raisonnement neutrosophique, est absent dans FL et d'autres logiques alliées. De plus, la LF est préoccupée par l'appartenance et la non-appartenance, pour un élément, à un état spécifique et ne traite pas de la nature ambiguë des données [156]. Notre approche proposée se concentre sur l'amélioration de la qualité de la pondération du terme en résolvant l'ambiguïté présentée dans le chevauchement entre les ensembles flous, comme le montre l'exemple de la figure 34.

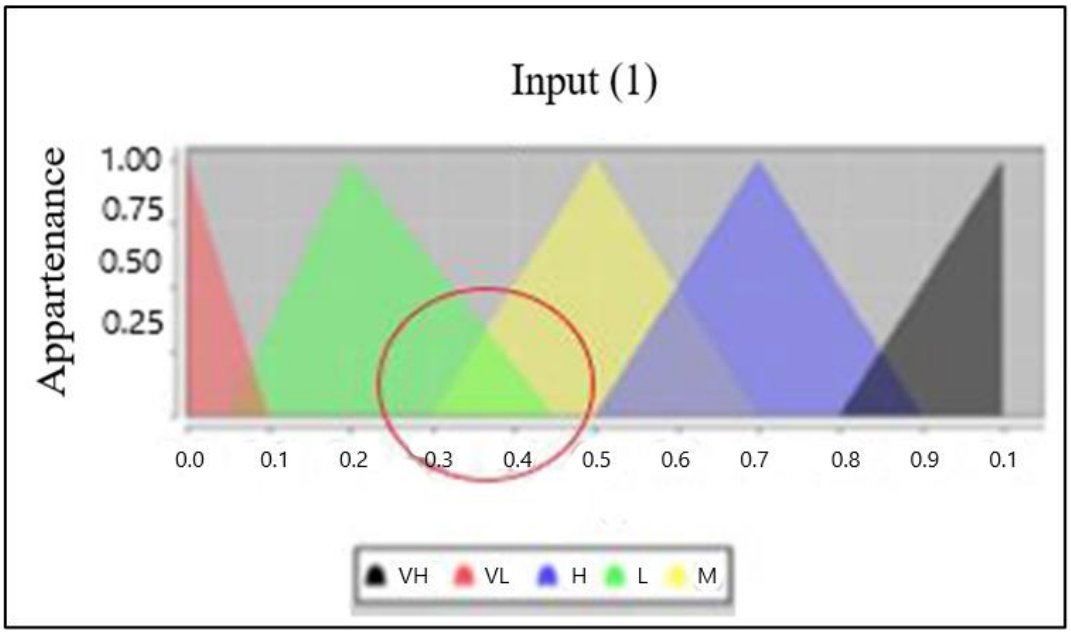

Figure 34: Exemple de chevauchement entre des ensembles flous pour une entrée donnée.

Par ailleurs, les fréquences et les occurrences des mots, qui appartiennent à l'intervalle du chevauchement seront imprécises et ne nous permettront pas de savoir s’ils sont fortement représentatifs pour un document. Par conséquence, chaque mot aura un degré élevé d'ambiguïté pour être pertinent pour le contenu textuel. Cette ambiguïté empêche l'efficacité des pondérations des termes, ainsi que leur degré d'informativité concernant un document donné, qui devient peu fiable, comme nous l'explorons ci-après.

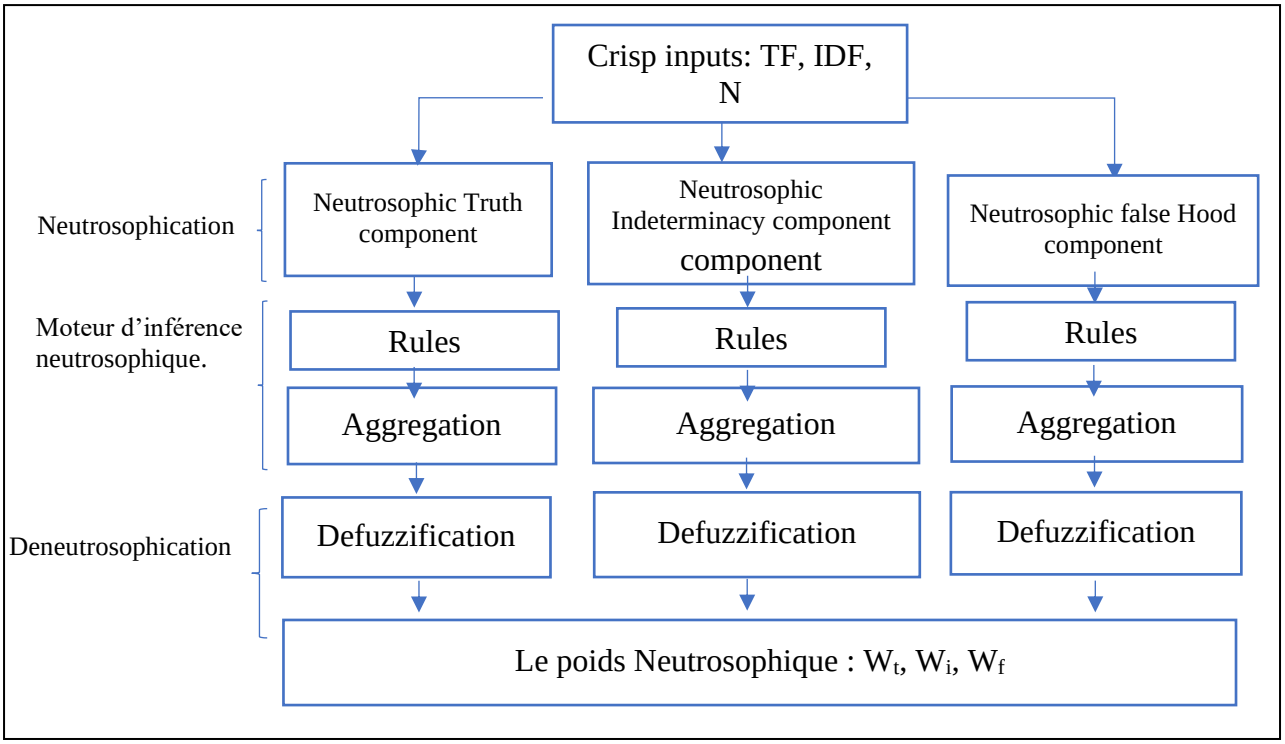

Figure 35: Système d'inférence neutrosophique pour déduire les poids NTF-IDF.

La figure 35 présente l'architecture du SIN pour la déduction de la NTF-IDF, où l'ensembles des processus d'un SIN, présenté dans les paragraphes précédents, sont correctement appliqués. En effet, la définition des fonctions d'appartenance et des variables linguistes pour les entrées de notre méthode de pondération des termes est la tâche principale de l'étape de Neutrosophication dans le SIN adopté. Les fonctions d'appartenance Truth, qui définissent le degré pertinent d'un terme, ont été conçues de manière qu'il n'y ait aucun chevauchement, pour les plages où le chevauchement a été conçu à l'aide du FIS. Les régions qui se chevauchent, présentées par la figure 34, sont enregistrées dans le SIF conventionnel, ont été capturées par indétermination neutrosophique. En tant que processus suivant pour générer NTF -IDF est le contrôleur neutrosophique, où le travail principal dans ce processus est de définir les règles neutrosophiques. Le NTF-IDF incorpore une approche simple, basée sur des règles neutrosophiques de cette forme :

SI X ***et*** Y ***ALORS*** Z.

En cas de rectification des règles d'inférence neutrosophiques, la NTF-IDF présentera une méthode précise de pondération vectorielle.

Dans la dernière étape de Deneutrosophication, nous utilisons la méthode de défuzzification centroïde pour générer les sorties vectorielles.

3.2 Corrélation et critères de conception des fonctions d'appartenances neutrosophiques pour les variables NTF-IDF

Comme mentionné ci-dessus, lorsque nous définissons une appartenance exacte des fréquences du terme, c'est-à-dire son occurrence locale TF et sa fréquence globale IDF, aux ensembles ne pose pas de problème. D'un autre côté, si les valeurs de fréquence du terme sont incluses dans les intervalles de chevauchement des ensembles, comme nous le voyons dans la figure 36 (a), cela influence le degré d'informativité du mot, et son degré d'ambiguïté devient plus critique.

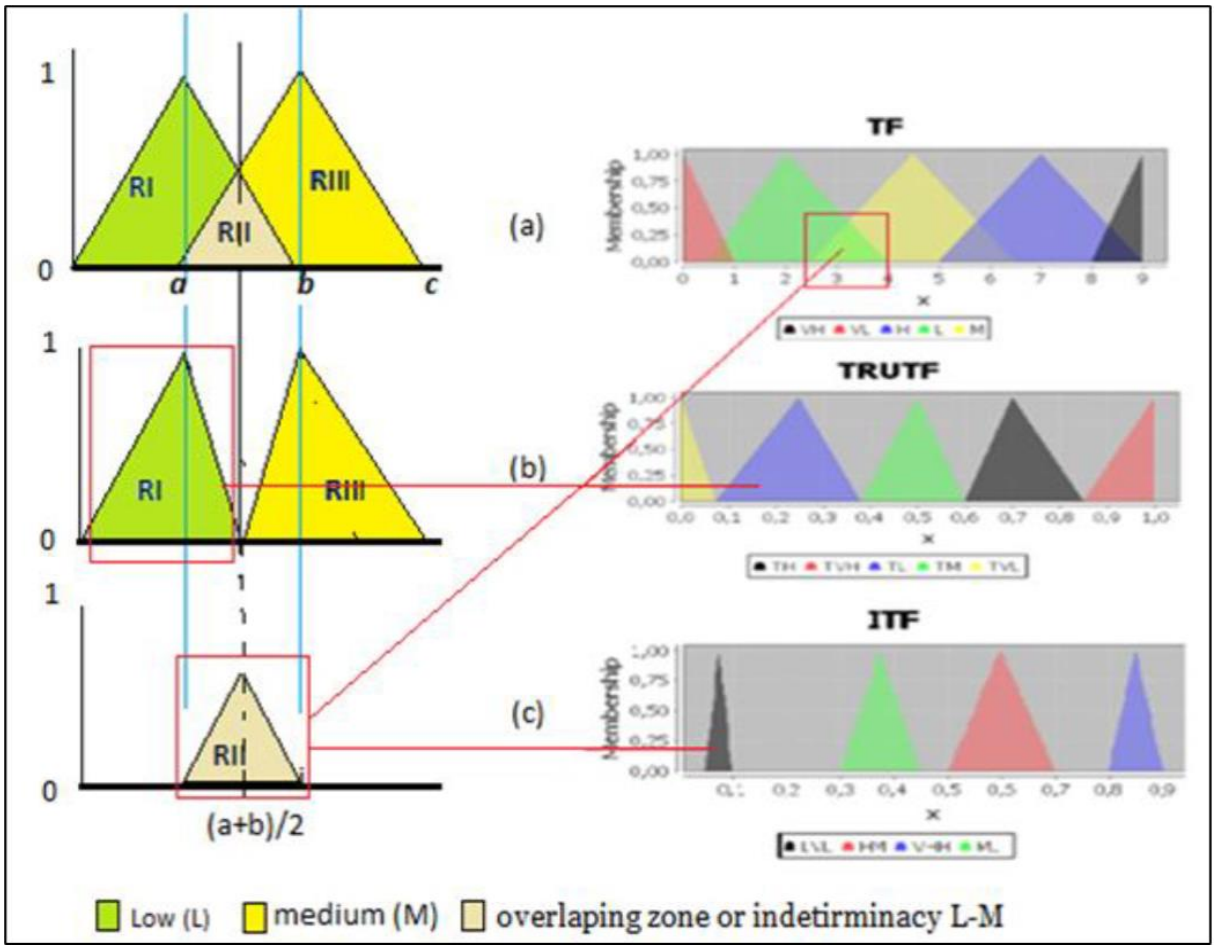

Figure 36: Corrélation et critères de conception des fonctions d'appartenances pour (a) la fréquence d'attribut flou TF, (b) la composante de vérité neutrosophique et (c) la composante d'indétermination neutrosophique.

Dans ce stade, les principales questions qui se posent sont : Comment déterminer les fonctions d'appartenance neutrosophique ? Et comment utiliser des ensembles flous qui se chevauchent pour déterminer les degrés d'ambiguïté des termes en fonction de leurs fréquences ? Pour y répondre, nous proposons la figure 36, au-dessus, pour illustrer la corrélation et les critères de conception des fonctions d'appartenance (MF) pour le SIN adopté. Principalement, nous spécifions la composante neutrosophique de vérité absolue (pertinence) et ambiguë (indétermination).

Précisément, les figures 36 (a) et (b) donnent la correspondance et les règles pour concevoir respectivement les fonctions d'appartenance pour les composants de vérité SIF et

neutrosophique. De plus, l'entrée TF (terme fréquence) (figure 36 a) est utilisée pour démontrer le fonctionnement des fonctions d'appartenance neutrosophique. Ici, nous choisirons deux variables linguistiques comme exemple de traitement : les ensembles bas et moyen.

Pour définir les degrés d'appartenance neutrosophique, pour une entrée, nous divisons la figure 36 (a), pour les deux ensembles, par exemple, moyen (M) et faible (L), en trois régions RI, RII et RIII associées respectivement aux plages [0 - (a + b) / 2], [a-b] et [(a + b) / 2 - c]. Le principal de chaque région est spécifié comme suit :

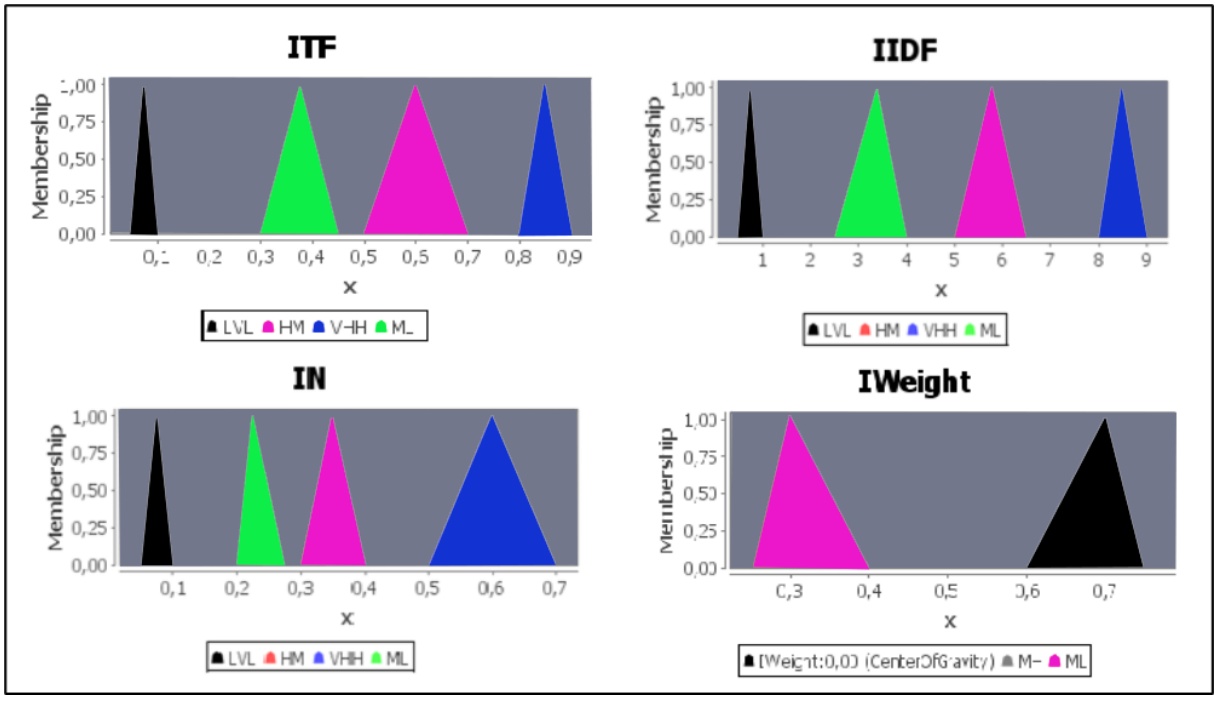

Figure 37: Fonctions d'appartenance ambiguës pour les valeurs de TF, IDF, N et le poids indéterminé.

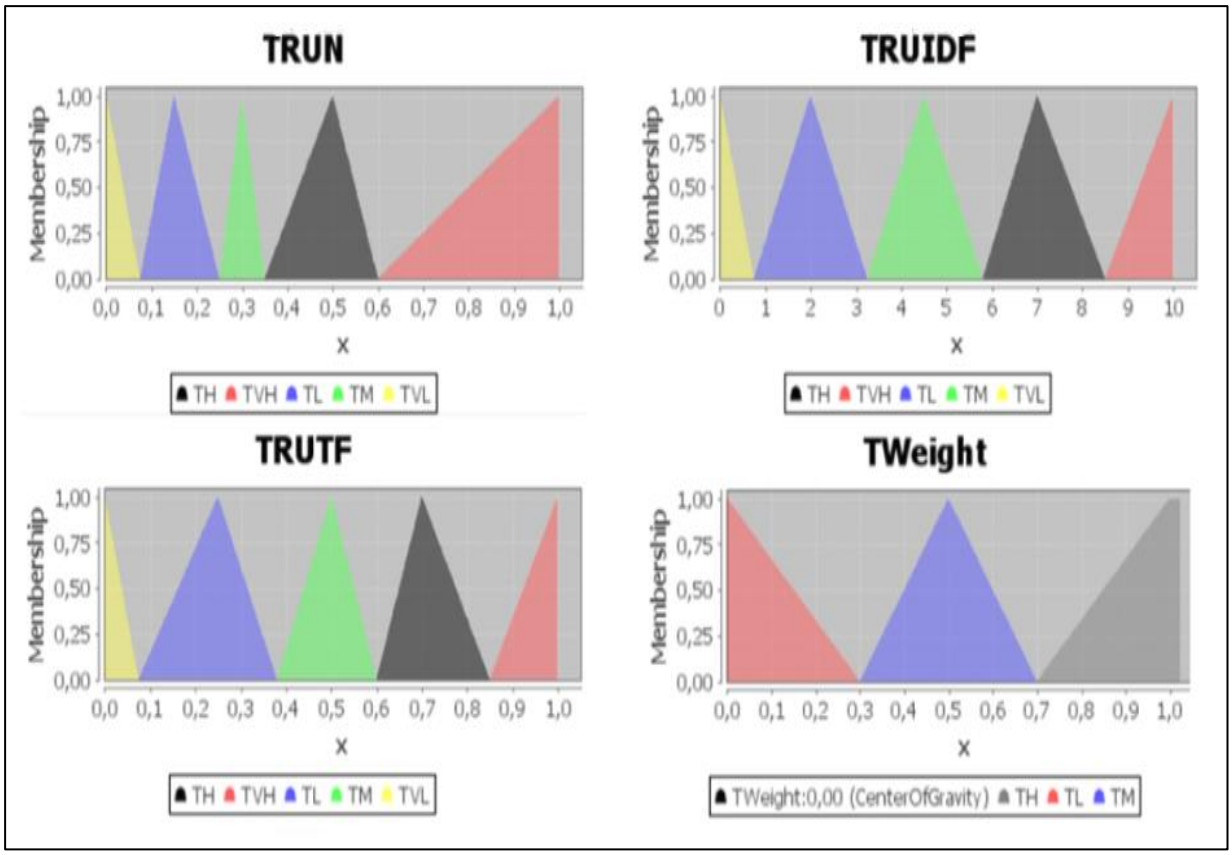

Figure 38: Fonctions d'appartenance de vérité pour les valeurs de TF, IDF, N et le poids Wt.

Pour mieux comprendre, nous proposons l'exemple suivant, qui montre la transformation du poids en passant du raisonnement flou au raisonnement neutrosophique.

Nous pratiquons l'exemple du mot « cérémonie » avec une fréquence locale TF = 3,2 dans un document de classe A, un taux global IDF = 0,62, avec une longueur du document N = 0,6. Nous tenons compte du fait que le terme à basse fréquence locale ne représente pas le contenu du document. Aussi, nous pouvons garder la parole avec un médium TF, comme nous pouvons considérer sa présence comme pertinente.

Ensuite, pour définir le degré d'appartenance du mot cérémonie, nous remarquons que les fréquences du terme donné appartiennent à l'intervalle du chevauchement des ensembles flous (voir Figure 36), c'est-à-dire le TF ϵ [2.5, 4], où nous ne pouvons pas précisément voir si le mot a une occurrence faible ou moyenne. Cette ambiguïté quant à l'appartenance de la fréquence du terme et influence son degré de pertinence pour représenter un document, où les termes ayant une valeur d'appartenance à un ensemble (bas-moyen) élevée présentent la catégorie des termes ambiguës. La séparation proposée permet de définir le degré d'ambiguïté selon la fréquence du terme. De plus, cela nous aidera à justifier que le mot reste un mot important pour représenter le document, par exemple, si le degré d'appartenance à l'ensemble (bas-moyen) est plus élevé cela signifier que l'ambiguïté du terme est également significative et le terme n'est pas pertinent, et vice versa. Le raisonnement est valable pour les autres entrées utilisées pour calculer le poids

NTF-IDF. De plus, nous visualisons l'importance d'appartenir à un ensemble précis dans la phase de prédiction des règles qui juge les sorties du système.

À propos des poids de sortie pour le terme cérémonie, ont les valeurs du degré d'appartenance respectivement aux ensembles vérité et indétermination : 0,4 et 0,6, où nous pouvons remarquer, en regardant les figures 37 et 38, le poids d'ambiguïté est supérieur au poids de vérité, ce qui montre que le degré de représenter le document est médiocre. De plus, si nous trouvons un mot avec Wt supérieur à Wi, alors son poids aura une grande valeur, ce qui signifie que le terme est pertinent pour représenter le document.

3.3 Calcul du poids final NTF-IDF

Pour utiliser une valeur de poids unique, dans le document vectoriel, nous proposons l'une des fonctions de score [169] [164], pour combiner les deux poids de sortie (Wi à Wt) afin de générer un poids de terme unique. Dans notre cas nous avons choisi la fonction Zhang [169], qui permet de définir le score et l'ordre associé entre les ensembles neutrosophiques à valeur unique (SVN) comme suit :

Soit A = (T, I, F) un SVN, ensuite une fonction de score S est définie comme suit :

$$S_{zhanG}(A) = \frac{(2 + T - I - F)}{3} \tag{35}$$

Où:

T, I, F représentent le degré de valeur d'appartenance de vérité, la valeur d'appartenance d'indétermination et les valeurs d'appartenance de fausseté de A.

Le système d'inférence neutrosophique génère un poids de score 3D, où pour chaque terme (t) a un poids ($W_{d,t}$) dans le document d est qui est la composition des poids suivantes :

(Poids-Vérité $W_{d,t}$ ->W_t, Poid-Indétermination $W_{d,t}$ -> W_i, Poids-Faux $Wd_{,t}$ -> W_F).

Dans notre cas, nous ne traitons que les valeurs de vérité $W_{d,t}$ (W_t) et ambiguës $W_{d,t}$ (W_i), où nous n'utilisons que les fonctions de vérité et ambiguës pour résoudre le problème d'ambiguë de la logique floue. Par conséquent, pour avoir un score représentatif, nous utilisons l'équation 36 pour combiner les composantes de sortie neutrosophiques. La valeur obtenue représente le score neutrosophique (NTF-IDF) associé au terme t dans un document d. finalement, cette phase correspond au processus d'extraction des caractéristiques, où nous générons le descripteur neutrosophique sous forme des vecteurs et qui sera l'entrée d'un ensemble des classifieurs souvent utilisés dans les systèmes de classification de texte.

4. Expérimentation et résultats

4.1 Matériels

Pour prouver l'efficacité du descripteur suggéré, basé sur la nouvelle méthode citée ci-dessus, nous proposons une classification des données textuelles avec le SVM et le classifieur

artificiel neuronal. Notant que l'objet principal de notre contribution est d'introduire la nouvelle représentation neutrosophique NTF-IDF. Ainsi, toute amélioration de cette étape peut améliorer la précision des systèmes adoptés. Comme premier test de NTF-IDF, nous suggérons la

classification des nouvelles, en utilisant les classifieurs les plus sensibles aux entrées. Par ailleurs, nous comparons avec le FTF-IDF classique pour confirmer les avancées de la nouvelle méthode NTF-IDF. L'ensemble des algorithmes a été implémenté avec le langage Java [170].

4.2 Bases des données

Le premier élément de notre étude est le choix de la base des données, où nous avons choisi quatre bases des données pour les classer selon leurs domaines. Ainsi, les méthodes adoptées ont été testées sur deux types de données, à savoir le type de données multi-étiquetées et le types des données qui sert à la classification binaire. Pour le premier type nous avons utilisé comme bases des données :

- La base de données BBCSPORTS contient 737 articles et cinq étiquettes de classe (athlétisme, cricket, football, rugby, tennis) [108].
- La base de données BBCNews comprend 2 225 documents des nouvelles de la BBC et cinq étiquettes de classe (affaires, divertissement, politique, sport et technologie) [108].
- L'ensemble de données de 20newsgroups comprend environ 18000 groupes de discussion indexés sur 20 sujets comme il est indiqué sur le site web :

  (https://www.kaggle.com/crawford/20newsgroups)

  Dans ce travail, nous utilisons 50% du corpus 20Newsgroupes pour illustrer les performances de NTF-IDF pour la classification de données multi-étiquetées.

Et pour le deuxième type nous avons employé :

- Les avis des clients d'Amazon et qui sont groupés dans deux classes. Cette base des données est réservée pour l'analyse des sentiments de type commerciales, (disponibles sur le site Web de Kaggle). Nous avons exploité 4002 avis des clients Amazon (2001 avis positifs et 2001 avis négatifs) pour confirmer l'impact de NTF-IDF sur la représentation et la classification des textes qui représentent les sentiments de ces clients.

4.3 Paramètres de la classification

a) Prétraitement des données

Nous rappelons que le processus de prétraitement est un ensemble de mesures proposées pour nettoyer les données textuelles avant de passer à la représentation numérique des corpus [83]. Alors, comme première étape du prétraitement, nous transformons les textes d'entré en une séquence de caractères. Par la suite, nous attribuons une fonction qui convertit le texte en minuscules et supprime les ponctuations et les symboles. Aussi, les mots vides ont été éliminés à l'aide d'une liste des mots vides. Enfin, les algorithmes de Stemming doivent déterminer les radicaux des mots, où nous utilisons l'algorithme de Stemming de Lovin [145]. Globalement, le prétraitement de texte permet d'économiser la mémoire et de prédire plus de résultats en utilisant, seulement, les données pertinentes.

Après l'étape de prétraitement, pour le système de classification adopté, nous proposons de pondérer les termes réservés en utilisant la nouvelle méthode NTF-IDF, qui prend en compte le degré d'ambiguïté du mot pour représenter le document, sans aucune possibilité de perte

d'information. Comme présenté dans ce qui suit, la précision des valeurs des poids calculés par le NFT-IDF se reflète sur la précision d'un ensemble de classifieurs.

b) Paramètres des classifieurs

Comme mentionné ci-dessus, une fois que les documents sont représentés en termes de vecteurs, en utilisant la méthode de pondération NTF-IF, il est nécessaire de faire appel à un classifieur adéquat. La tâche de prédiction consiste à affecter le bon document à la bonne classe. Dans notre proposition, nous comparons l'impact des représentations vectorielles NTF-IDF et du FTF-IDF en utilisant le SVM et les réseaux de neurones (FNN) comme classifieurs artificiel.

Chaque classifieur a ses paramètres qui permettent d'améliorer les performances en fonction du problème. Ainsi, pour le Support Vector Machine (SVM), le type du noyau caractérise le modèle SVM et peut modifier les résultats de la classification. Dans notre cas nous appliquons le noyau polynomial [200]. D'autre part, nous utilisons la fonction IG comme méthode de sélection d'attribut pour réduire la dimensionnalité des descripteurs.

Pour le Réseau de Neurones Artificiels (FNN), nous avons effectué une série des exécutions en faisant varier les paramètres du réseau à l'aide de la bibliothèque d'apprentissage profond de la JVM (DL4J). Plusieurs paramètres régissent l'efficacité de la classification, par exemple le nombre de couches cachées et la fonction d'activation. L'architecture FNN employée affiche deux couches cachées, le nombre de nœuds pour l'entrée FNN dépend du nombre des termes représentant le corpus et le nombre des nœuds pour la couche de sortie dépend du nombre des classes. Ainsi, nous utilisons la fonction Soft Max comme fonction d'activation, le nombre d'époque fixé à 10 et la 10-validation croisée pour apprendre et tester les classifieurs utilisés dans la phase de classification.

c) Mesures de performance

Il existe plusieurs mesures pour évaluer les performances d'un classifieur donné, comme il était mentionné dans le deuxième chapitre de cette thèse. Particulièrement, nous employons comme mesures d'évaluation : la précision, le rappel, la performance, la F-mesure, la courbe ROC et la valeur AUC.

4.4 Résultats et discussions

Notre intérêt dans cette section est de montrer l'impact de la nouvelle pondération NTF-IDF et ses nouvelles caractéristiques sur la classification des textes.

Tableau 41: Résultats de la classification en employant la FTF-IDF dans la phase de pondération.

	BBCSPORT				*BBCNews*				*Amazon Data*				*20 newsgroups*			
	P (%)	R (%)	F-M (%)	A (%)	P (%)	R (%)	F-M (%)	A (%)	P (%)	R (%)	F-M (%)	A (%)	P (%)	R (%)	F-M (%)	A (%)
SVM	92	92	92	92	73	73	73	72.8	85	84.6	85	85	80	80	80	80
SVM + IG	92.4	92.3	92.3	92.3	73	73	73.8	72.76	86	86.5	86	86	81.2	81.6	81.6	82
ANN	87	87	87	86.4	78.8	78.7	78.8	78.74	82	82	82	82	82	82	82	82

Tableau 42: Résultats de la classification, en employant la NTF-IDF dans la phase de pondération.

	BBCSPORT				*BBCNews*				*Amazon Data*				*20 newsgroups*			
	P (%)	R (%)	F-M (%)	A (%)	P (%)	R (%)	F-M (%)	A (%)	P (%)	R (%)	F-M (%)	A (%)	P (%)	R (%)	F-M (%)	A (%)
SVM	90	90	90	90	83.24	83.6	83.4	83.4	90	90.1	90	90.1	83	83	83	83
SVM + IG	91	91.5	91.3	92.3	90.6	90.5	90.5	90.5	90.5	90.6	90.54	90.5	84	84.7	85.5	84.5
ANN	89	89.6	89.3	89.4	88	88.2	88	88	91	91.2	91	91.2	83.5	83.6	84	84

Nous avons choisi une variante de corpus pour prouver et valider les performances de notre méthode de pondération par rapport à la vectorisation FTF-IDF traditionnelle. Les tableaux 41 et 42 montrent les performances de la classification en utilisant la méthode floue FTF-IDF et la nouvelle vectorisation neutrosophique NTF-IDF dans la phase de pondération des termes. Généralement, notre approche présente des performances supérieures à ceux de l'approche floue, que ça soit pour la classification multiple (pour les data sets BBCSport, BBCNews et 20Newsgroups) ou l'analyse des sentiments (pour la base Amazone) et la différence entre les performances dépasse 10%.

De manière uniforme, la méthode de sélection des attributs joue un rôle essentiel dans l'amélioration des résultats de classification et les résultats affichées prouvent l''impact la méthode IG, en tant que méthode de sélection de caractéristiques, sur les systèmes de de classification des textes. Par conséquent, les résultats présentés prouvent l'amélioration de la performance de la classification SVM en utilisant l'IG et le nouveau descripteur neutrosophique qui emploi les NTF-IDF où la différence entre les scores de performance peut être égale à 7%.

Une autre façon, qui démontre l'efficacité de la nouvelle représentation, est la forme de la courbe ROC ainsi que la zone sous la courbe ROC (AUC) donne une estimation raisonnable de la capacité du rejet du système, c'est-à-dire de sa capacité à rejeter des modèles mal classés sans rejeter les instances bien classées.

Or, les courbes données pour les méthodes de pondération comparées, utilisant le classifieur SVM et que la figure 39 illustre, montrent que les performances du système de classification basé sur NTF-IDF ont une performance satisfaisante et un excellent score AUC. Aussi, l'analyse de la figure 39 montre que la trajectoire ROC (b) est loin d'être qualifiée, cependant la deuxième

courbe (a) qui a une forme de courbe parfaite, démontre l'efficacité du système proposé avec un score AUC supérieure à 96%.

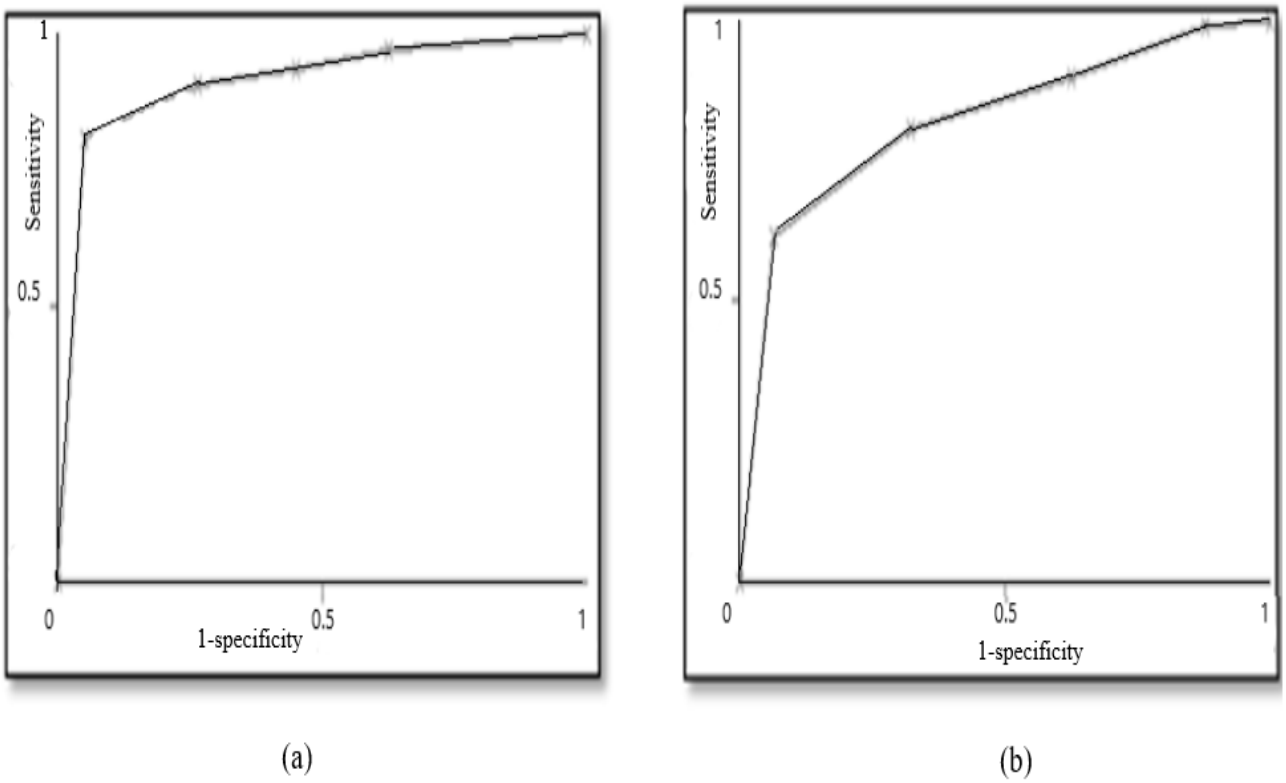

Figure 39: Courbes ROC pour la classification SVM en utilisant : NFT-IDF (a) et FTF-IDF (b) comme méthode de représentation des caractéristiques.

Notre nouveau descripteur NTF-IDF influence positivement le comportement des classifieurs, où nous remarquons une évolution notable des scores AUC [83], en comparant avec la FTF-IDF et l'ensembles des exemples cités dans la figure 40.

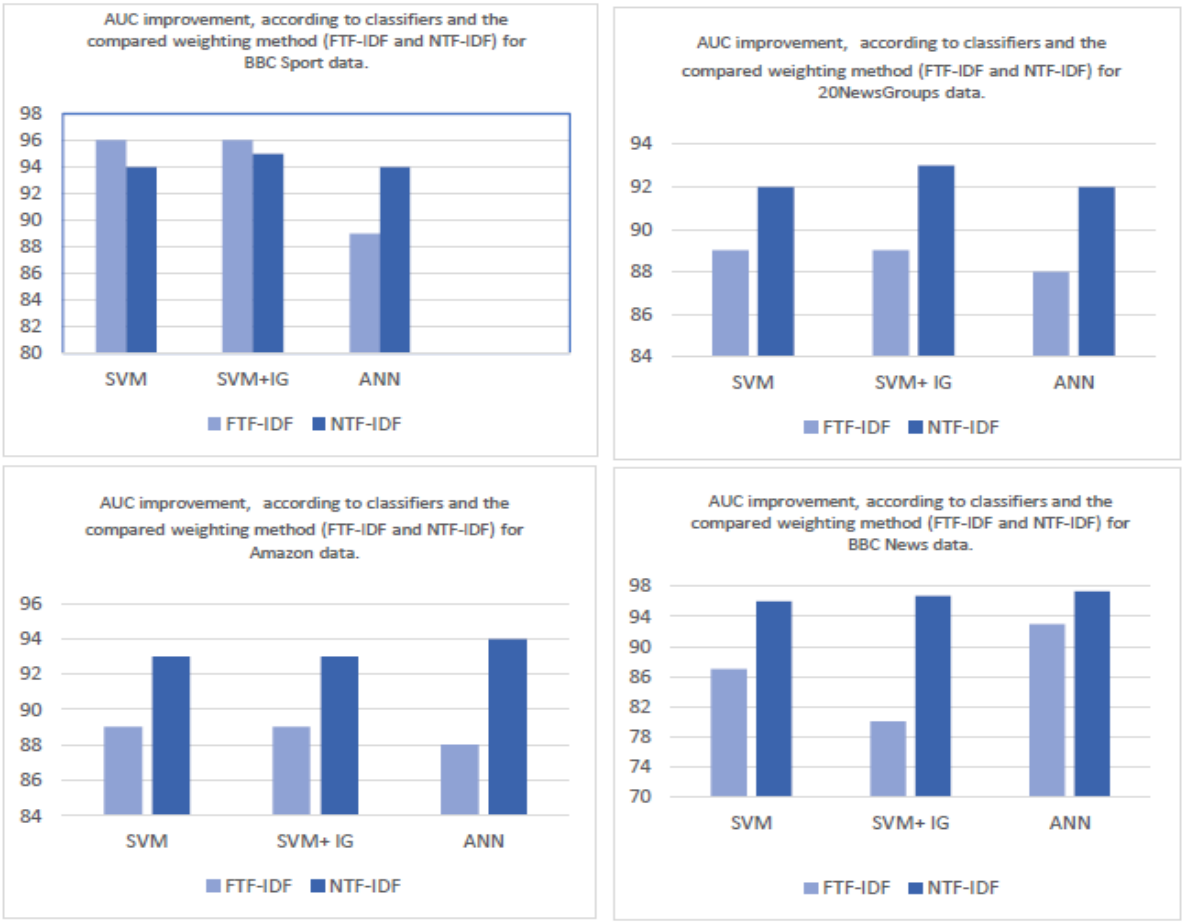

Figure 40: Amélioration, selon les classifieurs et la méthode de pondération comparée (FTF-IDF et NTF-IDF) pour différents ensembles de données.

V. Conclusion

Pour le chapitre quatre nous nous sommes intéressés à introduire un nouveau modèle de pondération (NTF-IDF), qui intègre la LN pour améliorer la pondération floue, FTF-IDF, des termes représentant le texte exprimer en langage naturel. L'approche NTF-IDF est très différente des approches populaires de la vectorisation des textes. De même, la stratégie utilisée par NTF-IDF (en comparant avec la FTF-IDF) est très différente de la stratégie et les techniques actuelles de la pondération fréquentielle.

Ce nouveau modèle combine les avantages du raisonnement neutrosophique, spécialement la fonction d'ambiguïté, pour déterminer un poids informatif et précis pour chaque terme du corpus. Notre méthode peut être une extension du modèle populaire FTF-IDF, qui produit des descripteurs pertinents pour les applications du Text Mining comme : la classification multi-libellés des textes ou l'analyse des sentiments.

Les résultats des expérimentations indiquent que l'utilisation de la NTF-IDF, dans le processus d'extraction des caractéristiques, influence positivement la performance des classifieurs. La comparaison de la nouvelle pondération NTF-IDF avec la FTF-IDF en utilisant des bases des données de référence montre une grande précision en utilisant notre nouvelle méthode. Le taux

de reconnaissance du système adopté est supérieur à 90% ; un score de précision non atteint avec la FTF-IDF. De même, la précision du poids neutrosophique a un impact significatif sur les méthodes de sélection des attributs et aboutit à une précision élevée.

Dans l'ensemble, le modèle est bien mis en œuvre pour corriger les lacunes de la représentation FTF-IDF. Ainsi, nous avons démontré que la transformation du FTF-IDF par le NL produit une méthode de pondération des termes efficace et utile. En effet, nous avons pu définir davantage un degré d'ambiguïté du terme en fonction de ses fréquences populaires et générer des poids informatifs, qui servaient la pertinence des vecteurs représentatifs pour les corpus de langage naturel. Le modèle peut avoir d'autre amélioration pour améliorer la performance des descripteurs neutrosophiques, où nous pouvons avoir une excellente pertinence et une dimension optimale malgré les grandes quantités des données traitées.

Conclusion générale et Perspectives

Notre conclusion résume les quatre chapitres précédents de la thèse et décrit un certain nombre de directions de recherche pour les travaux futurs. Dans un premier lieu nous mettons en évidence nos contributions, qui traitent les différents défis abordés dans cette mémoire. Ensuite, nous discutons des orientations futures afin d'étendre et d'accroitre la recherche menée dans cette thèse.

Résumé des contributions

Cette thèse a présenté des contributions variées au problème de la classification automatique (et multi-classe) des textes bruts et l'analyse des avis des utilisateurs du web. En effet, en premier lieu, nous avons comparais entre différentes méthodes employées dans les processus du Text Mining, afin de visualiser l'impact de chaque processus sur la performance des systèmes de classification. Effectivement, nous avons prouvé l'influence du processus d'extraction des caractéristiques sur la précision de ces systèmes.

Notre but général était de concevoir des descripteurs robustes visant à fournir une description discriminante des données textes, lors de l'extraction des caractéristiques. Ce descripteur se base sur un bon prétraitement des entrées, une vectorisation et calcule des poids précis, ainsi qu'une sélection optimale et pertinente des caractéristiques.

Dans ce cadre, notre première contribution a consisté à l'introduction de la notion de l'apprentissage statistique pour le Text Mining, et ses services à l'exploitation des connaissances du web, ainsi que l'ensembles des processus et matériaux tendances, réserver pour la réalisation de nos contributions.

Également, à travers cette comparaison, nous avons démontré, l'impact du processus de la représentation et la vectorisation des textes sur la tâche de catégorisation binaire et multiple des documents. en effet, une meilleure représentation reflète une meilleure performance des systèmes d'analyse des textes. Cette remarque a conduit à la proposition des nouvelles approches de représentation et de vectorisation à base du processus classique, et à la création des systèmes performants.

Notre première approche de vectorisation des données non structurés, est une nouvelle version probabiliste vectorielle qui fournit des vecteurs d'entrées au classifieur PMC. Cela permet la génération de descripteur de taille optimale avec une sélection adéquate des caractéristiques pertinentes, qui représentes mieux un corpus. En effet, nous avons prouvé son efficacité pour la conception des systèmes de classification neuronale multiple, où le taux de reconnaissance du système adopté, pour la classification d'une base des nouvelles de référence, est de **100%**, tandis que les méthodes de vectorisation classiques n'ont pas montré une performance convaincante.

En outre, notre recherche nous a poussé vers le raisonnement flou qui est largement utiliser dans le domaine TALN, où l'approche floue de pondération des termes FTF-IDF est souvent employée dans les applications du Text Mining. Dans un premier temps, nous avons réussi à résoudre le problème de sélection manuelles des règles d'inférences, comme processus important dans les systèmes experts flous. Par l'automatisation de ce processus à l'aide de notre approche proposé et qui se base principalement sur les modèles d'association, nous avons pu

progresser les performances et diminuer la complexité d'un certain nombre d'application des SIFs, comme les systèmes de classification floues. De plus, grâce aux améliorations apportés sur la prédiction des règles d'inférence, la qualité de la FTF-IDF et les descripteurs d'entré pour des classifieurs ML ont présenté une excellente précision pour les différents types des systèmes de catégorisation.

Dans le même contexte, la NTF-IDF a été mis en œuvre comme une version étendue de la Fuzzy TF-IDF (FTF-IDF) qui vise également à corriger les lacunes de cette dernière.

En général, la nouveauté de cette contribution réside dans deux aspects : premièrement, une nouvelle méthode de pondération des termes, qui utilise les fréquences, local et globale, du terme comme composantes principales pour définir la pertinence et l'ambiguïté du terme ; deuxièmement, l'application de la LN pour déduire des poids est considérée comme un modèle original dans cette participation. En outre, grâce à la modélisation 3D de la LN et en pratiquant les caractéristiques du SIN, nous avons pu calculer des poids précises. Les nouveaux poids neutrosophiques prennent en considération les valeurs : degré d'ambiguïté et degré de pertinence de chaque terme, ce qui influence la pertinence globale du descripteur représentatif pour un corpus donnée. De plus, la technique introduite a été combinée avec différents modèles ML pour améliorer la précision et la pertinence des vecteurs de caractéristiques obtenus pour alimenter le mécanisme de classification. En effet, la partie expérimentation de notre étude illustre que la nouvelle méthode a un impact positif sur la catégorisation et l'analyse des textes. En outre, le taux de reconnaissance du système adopté est supérieur à **91%,** un score de précision non atteint avec la Fuzzy TF-IDF, employant certains classifieurs. Aussi, la comparaison des systèmes qui emploient :

- Un ensemble des bases des données de référence, dans différents domaines du Text Mining,
- La sélection des attributs,
- Des classifieurs d'apprentissage automatique, à savoir SVM et Feed-Forward Network,
- Le nouveau descripteur neutrosophique (NTF -IDF),

Présente une amélioration sur le score de performance de 10%, en employant différentes méthodes de pondération populaires. Globalement, notre proposition encourage les praticiens des systèmes de contrôle intelligents à utiliser notre contribution même pour des problèmes plus complexes.

Perspectives

Cette section discute les orientations possibles pour les travaux futurs, qui feraient progresser nos recherches et fourniraient une meilleure étude. Les axes de travail futurs incluent l'amélioration des process du Text Mining, comme :

- Le choix optimal du vocabulaire d'un corpus à l'aide des algorithmes d'apprentissage, afin de gérer la dimensionnalité des descripteurs représentatives des grandes bases des données ;
- L'adaptation des méthodes de sélection et d'extraction des variables pertinentes avec les corpus volumineux ;

- L’intégration de la tâche de classification, présentés dans cette thèse, dans des systèmes d’exploration des données avancés.

L’amélioration dans les processus Text Mining conduira au renforcement d’un ensembles des systèmes qui traite d’une manière automatique et intelligente les données exprimés en langage naturel. Cela facilite l’utilisation et l’exploitation du web pour les simples utilisateurs ainsi que les investisseurs du e-service. Dans le même contexte nous proposons de progresser de plus les fonctionnalités de la vectorisation fréquentielle des données textuelles, en utilisant les avantages et les caractéristiques de la LN. En effet, similairement à l’idée d’automatisation du processus de génération des règles d’inférence floues, nous proposons d’automatiser également la génération des règles d’inférence neutrosophique, d'autant que leur nombre dépasse le nombre nécessaire pour un inférence flou.

Idem, nous recommandons et soutenons l'idée d'exploiter le taux d'ambiguïté, de pertinence et de non-pertinence d’un terme, fournit par la représentation neutrosophique. Cela aidera à sélectionner un vocabulaire qui contient l’ensemble des termes pertinents pour représentation des corpus. Cette tâche permettra, aussi, la réduction de l’espace représentatif des données sans intervention des méthodes habituelles de sélection de variable. Comme elle pourra produire des descripteurs efficaces et puissants pour le traitement et l’analyse des grandes masse des données, ainsi les utilisés comme entrés compatible aux algorithmes du Deep Learning.

L’emplois de la LN sera adéquate au perfectionnement des applications du web y contient le web sémantique. Au fur et à mesure des travaux futurs, nous suggérons d’exploiter cette logique ainsi que les SINs pour améliorer les applications et les systèmes d’e-services, à savoir : les systèmes de recommandation, les systèmes et les plateformes connues par l’Opinion Mining, les systèmes de recherche par mots clés, etc…

Bibliographies

[1] Ben-Dov, M., & Feldman, R. (2009). Text mining and information extraction. In Data mining and knowledge discovery handbook (pp. 809-835). Springer, Boston, MA.

[2] Ahmad, S., & Varma, R. (2018). Information extraction from text messages using data mining techniques. Malaya Journal of Matematik, (1), 26-29.

[3] Xie, X., Fu, Y., Jin, H., Zhao, Y., & Cao, W. (2020). A novel text mining approach for scholar information extraction from web content in Chinese. Future Generation Computer Systems, 111, 859-872.

[4] Trieu, L. Q., Tran, H. Q., & Tran, M. T. (2017, December). News classification from social media using twitter-based doc2vec model and automatic query expansion. In Proceedings of the Eighth International Symposium on Information and Communication Technology (pp. 460-467).

[5] Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L., & Brown, D. (2019). Text classification algorithms: A survey. Information, 10(4), 150.

[6] Aggarwal, C. C., & Zhai, C. (2012). A survey of text classification algorithms. In Mining text data (pp. 163-222). Springer, Boston.

[7] Miner, G., Elder IV, J., Fast, A., Hill, T., Nisbet, R., & Delen, D. (2012). Practical text mining and statistical analysis for non-structured text data applications. Academic Press.

[8] Salloum, S. A., Al-Emran, M., Monem, A. A., & Shaalan, K. (2018). Using text mining techniques for extracting information from research articles. In Intelligent natural language processing: Trends and Applications (pp. 373-397). Springer, Cham..

[9] Shah, K., Patel, H., Sanghvi, D., & Shah, M. (2020). A comparative analysis of logistic regression, random forest and KNN models for the text classification. Augmented Human Research, 5(1), 1-16.

[10] Chamoso, P., Rivas, A., Rodríguez, S., & Bajo, J. (2018). Relationship recommender system in a business and employment-oriented social network. Information sciences, 433, 204-220.

[11] Gelbukh, A. (Ed.). (2018). Computational Linguistics and Intelligent Text Processing: 18th International Conference, CICLing 2017, Budapest, Hungary, April 17–23, 2017, Revised Selected Papers, Part II (Vol. 10762). Springer.

[12] Bounabi, M., El Moutaouakil, K., & Satori, K. (2018, April). A probabilistic vector representation and neural network for text classification. In International Conference on Big Data, Cloud and Applications (pp. 343-355). Springer, Cham..

[13] Qaiser, S., & Ali, R. (2018). Text mining: use of TF-IDF to examine the relevance of words to documents. International Journal of Computer Applications, 181(1), 25-29..

[14] Uysal, A. K., & Gunal, S. (2014). The impact of preprocessing on text classification. Information processing & management, 50(1), 104-112..

[15] Rani, S. R., Ramesh, B., Anusha, M., & Sathiaseelan, J. G. R. (2015). Evaluation of stemming techniques for text classification. International Journal of Computer Science and Mobile Computing, 4(3), 165-171..

[16] Bounabi, M., Moutaouakil, K. E., & Satori, K. (2019). A comparison of text classification methods using different stemming techniques. International Journal of Computer Applications in Technology, 60(4), 298-306..

[17] Lemaire, B. (2008, March). Limites de la lemmatisation pour l'extraction de significations. In 9e Journées internationales d'Analyse Statistique des Données

Textuelles (pp. 725-732).

[18] Fox, E. A. (1983). Extending the Boolean and vector space models of information retrieval with p-norm queries and multiple concept types (Doctoral dissertation, Cornell University).

[19] Kadhim, A. I. (2019, April). Term weighting for feature extraction on Twitter: A comparison between BM25 and TF-IDF. In 2019 international conference on advanced science and engineering (ICOASE) (pp. 124-128). IEEE.

[20] Joachims, T. (1996). A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization. Carnegie-mellon univ pittsburgh pa dept of computer science.

[21] Chen, K., Zhang, Z., Long, J., & Zhang, H. (2016). Turning from TF-IDF to TF-IGM for term weighting in text classification. Expert Systems with Applications, 66, 245-260.

[22] Vukotić, V., Claveau, V., & Raymond, C. (2015, June). IRISA at DeFT 2015: supervised and unsupervised methods in sentiment analysis. In DeFT, Défi Fouille de Texte, joint à la conférence TALN 2015.

[23] Pennington, J., Socher, R., & Manning, C. D. (2014, October). Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543)..

[24] Miao, J., & Niu, L. (2016). A survey on feature selection. Procedia Computer Science, 91, 919-926.

[25] Solorio-Fernández, S., Carrasco-Ochoa, J. A., & Martínez-Trinidad, J. F. (2020). A review of unsupervised feature selection methods. Artificial Intelligence Review, 53(2), 907-948.

[26] Vora, S., & Yang, H. (2017, July). A comprehensive study of eleven feature selection algorithms and their impact on text classification. In 2017 Computing Conference (pp. 440-449). IEEE.

[27] Rahman, A. F. R., Alam, H., & Fairhurst, M. C. (2002, August). Multiple classifier combination for character recognition: Revisiting the majority voting system and its variations. In International Workshop on Document Analysis Systems (pp. 167-178). Springer, Berlin, Heidelberg.

[28] Le, Q., & Mikolov, T. (2014, June). Distributed representations of sentences and documents. In International conference on machine learning (pp. 1188-1196). PMLR..

[29] Gupta, Y., Saini, A., & Saxena, A. K. (2015). A new fuzzy logic based ranking function for efficient information retrieval system. Expert Systems with Applications, 42(3), 1223-1234.

[30] Zhang, C., & Zhang, S. (2003). Association rule mining: models and algorithms (Vol. 2307). Springer.

[31] Chowdhury, G. G. (2003). Natural language processing. Annual review of information science and technology, 37(1), 51-89.

[32] Vaidya, P. (2015). Opinion Mining and Sentiment Analysis in Data Mining. Scholars Journal of Engineering and Technology SJET. Sch. J. Eng. Tech. B, 31, 71-75.

[33] Agrawal, R., & Batra, M. (2013). A detailed study on text mining techniques. International Journal of Soft Computing and Engineering, 2(6), 118-121

[34] Witten, I. H., & Frank, E. (2002). Data mining: practical machine learning tools and techniques with Java implementations. Acm Sigmod Record, 31(1), 76-77.

[35] Dhall, D., Kaur, R., & Juneja, M. (2020). Machine learning: a review of the algorithms and its applications. Proceedings of ICRIC 2019, 47-63.

[36] Lamsiyah, S., El Alaoui, S. O., & Espinasse, B. (2018). Résumé automatique guidé de textes: État de l'art et perspectives. In CORIA.

[37] Ronk, J. (2014). Structured, semi-structured and unstructured data. Retrieved 29 July 2015.

[38] Piatetsky-Shapiro, G. (1991). Discovery, analysis, and presentation of strong rules. Knowledge discovery in databases, 229-238.

[39] Lamsiyah, S., El Alaoui, S. O., & Espinasse, B. (2018). Résumé automatique guidé de textes: État de l'art et perspectives (Guided Summarization: State-of-the-art and perspectives). In Actes de la Conférence TALN. Volume 2-Démonstrations, articles des Rencontres Jeunes Chercheurs, ateliers DeFT (pp. 55-72).

[41] Feldman, R., & Dagan, I. (1995, August). Knowledge Discovery in Textual Databases (KDT). In KDD (Vol. 95, pp. 112-117).

[42] Hearst, M. A. (1999, June). Untangling text data mining. In Proceedings of the 37th Annual meeting of the Association for Computational Linguistics (pp. 3-10).

[43] Kodratoff, Y. (1999, June). Knowledge discovery in texts: a definition, and applications. In International Symposium on Methodologies for Intelligent Systems (pp. 16-29). Springer, Berlin, Heidelberg.

[44] Frederic, E. (2007). Text Mining applied to SPAM detection. Presentation given at University of Geneva.

[45] Nguyen, A., O'Dwyer, J., Vu, T., Webb, P. M., Johnatty, S. E., & Spurdle, A. B. (2020). Generating high-quality data abstractions from scanned clinical records: text-mining-assisted extraction of endometrial carcinoma pathology features as proof of principle. BMJ open, 10(6), e037740.

[46] Öztürk, N., & Ayvaz, S. (2018). Sentiment analysis on Twitter: A text mining approach to the Syrian refugee crisis. Telematics and Informatics, 35(1), 136-147.

[47] Greco, F., & Polli, A. (2020). Emotional Text Mining: Customer profiling in brand management. International Journal of Information Management, 51, 101934.

[48] Shardlow, M. (2014). A survey of automated text simplification. International Journal of Advanced Computer Science and Applications, 4(1), 58-70.

[49] Kim, Y. M., & Delen, D. (2018). Medical informatics research trend analysis: A text mining approach. Health informatics journal, 24(4), 432-452.

[50] Yadav, A. K. S., & Sora, M. (2021). Fraud detection in financial statements using text mining methods: A review. In IOP Conference Series: Materials Science and Engineering (Vol. 1020, No. 1, p. 012012). IOP Publishing..

[51] DALBERA, J. Le corpus entre données, analyse et théorie. Corpus, 2002, no 1.

[52] Lovins, J. B. (1968). Development of a stemming algorithm. Mech. Transl. Comput. Linguistics, 11(1-2), 22-31.

[53] Kessler, R., Torres-Moreno, J. M., & El-Bèze, M. (2006). Classification automatique de courriers électroniques par des méthodes mixtes d'apprentissage. INGENIERIE DES SYSTEMES D INFORMATION, 11(2), 93.

[54]www.researchgate.net/publication/283784302_Un_Nouvel_Algorithme_de_Stemmatisation_pour_l'Indexation_Automatique_de_Documents_non-structures_Stemmer_SAID.

[55] Jagić, T., & Brkić, L. (2020). Hot Topic Detection Using Twitter Streaming Data. In 2020 43rd International Convention on Information, Communication and Electronic Technology (MIPRO) (pp. 1730-1735). IEEE.

[56] Sueno, H. T., Gerardo, B. D., & Medina, R. P. (2020). Converting Text to Numerical

Representation using Modified Bayesian Vectorization Technique for Multi-Class Classification. International Journal, 9(4)..

[57] Robertson, S. E., & Walker, S. (1997, July). On relevance weights with little relevance information. In Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 16-24).

[58] Salton, G., Wong, A., & Yang, C. S. (1975). A vector space model for automatic indexing. Communications of the ACM, 18(11), 613-620..

[59] Kluckhohn, C. (1950). Human behavior and the principle of least effort.

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

[69] Bounabi, M., El Moutaouakil, K., & Satori, K. (2018, April). A probabilistic vector representation and neural network for text classification. In International Conference on Big Data, Cloud and Applications (pp. 343-355). Springer, Cham.

[70] Aizawa, A. (2003). An information-theoretic perspective of tf–idf measures. Information Processing & Management, 39(1), 45-65.

[71] Stein, R. A., Jaques, P. A., & Valiati, J. F. (2019). An analysis of hierarchical text classification using word embeddings. Information Sciences, 471, 216-232.

[72] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems (pp. 3111-3119).

[73] Schonlau, M., & Guenther, N. (2017). Text mining using n-grams. Schonlau, M., Guenther, N. Sucholutsky, I. Text mining using n-gram variables. The Stata Journal, 17(4), 866-881.

[74] Mcheick, H., Saleh, L., Ajami, H., & Mili, H. (2017). Context relevant prediction model for COPD domain using bayesian belief network. Sensors, 17(7), 1486.

[75] Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. Journal of machine learning research, 3(Mar), 1157-1182.

[76] Hall, M. A. (1999). Correlation-based feature selection for machine learning..

[77] Urbanowicz, R. J., Meeker, M., La Cava, W., Olson, R. S., & Moore, J. H. (2018). Relief-based feature selection: Introduction and review. Journal of biomedical informatics, 85, 189-203.

[78] Blessie, E. C., & Karthikeyan, E. (2012). Sigmis: A feature selection algorithm using correlation based method. Journal of Algorithms & Computational Technology, 6(3), 385-394.

[79] Rangkuti, F. R. S., Fauzi, M. A., Sari, Y. A., & Sari, E. D. L. (2018, November). Sentiment analysis on movie reviews using ensemble features and pearson correlation based feature selection. In 2018 International Conference on Sustainable Information Engineering and Technology (SIET) (pp. 88-91). IEEE..

[80] Korde, V., & Mahender, C. N. (2012). Text classification and classifiers: A survey. International Journal of Artificial Intelligence & Applications, 3(2), 85.

[81] Betancourt, Y., & Ilarri, S. (2020). Use of Text Mining Techniques for Recommender Systems. In ICEIS (1) (pp. 780-787)..

[82] Rosso, P., Ferretti, E., Jiménez, D., & Vidal, V. (2004). Text categorization and information retrieval using wordnet senses. In The Second Global Wordnet Conference GWC (pp. 299-304).

[83] Aharrane, N., El Moutaouakil, K., & Satori, K. (2015, March). A comparison of

supervised classification methods for a statistical set of features: Application: Amazigh OCR. In 2015 Intelligent Systems and Computer Vision (ISCV) (pp. 1-8). IEEE..

[84] Whissell, J. S., & Clarke, C. L. (2011). Improving document clustering using Okapi BM25 feature weighting. Information retrieval, 14(5), 466-487.

[85] Liu, B., & Zhang, L. (2012). A survey of opinion mining and sentiment analysis. In Mining text data (pp. 415-463). Springer, Boston, MA..

[86] Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E. D., Gutierrez, J. B., & Kochut, K. (2017). A brief survey of text mining: Classification, clustering and extraction techniques. arXiv preprint arXiv:1707.02919.

[87] Bonaccorso, G. (2017). Machine learning algorithms. Packt Publishing Ltd..

[88] Kessler, R., Torres-Moreno, J. M., & El-Bèze, M. (2004). Classification thématique de courriels avec apprentissage supervisé, semi-supervisé et non supervisé. les actes de VSST, 493-504..

[89] Aggarwal, C. C., & Zhai, C. (2012). An introduction to text mining. In Mining text data (pp. 1-10). Springer, Boston, MA..

[90] Koukourikos, A., Stoitsis, G., & Karampiperis, P. (2012, September). Sentiment Analysis: A tool for Rating Attribution to Content in Recommender Systems. In RecSysTEL@ EC-TEL (pp. 61-70)..

[91] Artemenko, O., Pasichnyk, V., Kunanets, N., & Shunevych, K. (2020). Using sentiment text analysis of user reviews in social media for e-tourism mobile recommender systems. In COLINS (pp. 259-271).

[92] Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E. D., Gutierrez, J. B., & Kochut, K. (2017). A brief survey of text mining: Classification, clustering and extraction techniques. arXiv preprint arXiv:1707.02919.

[93] Arica, N., & Yarman-Vural, F. T. (2001). An overview of character recognition focused on off-line handwriting. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 31(2), 216-233..

[94] Jain, A. K., Duin, R. P. W., & Mao, J. (2000). Statistical pattern recognition: A review. IEEE Transactions on pattern analysis and machine intelligence, 22(1), 4-37.

[95] Aharrane, N., Dahmouni, A., El Moutaouakil, K., & Satori, K. (2017). A robust statistical set of features for Amazigh handwritten characters. Pattern Recognition and Image Analysis, 27(1), 41-52.

[96] Stephenson, T. A. (2000). An introduction to Bayesian network theory and usage (No. REP_WORK). IDIAP..

[97] Zhu, Q. Y. (1991). Modèles bayésiens et application à l'estimation des caractéristiques de produits finis et au contrôle de la qualité (Doctoral dissertation, Ecole Nationale des Ponts et Chaussées).

[98] Patrick, N. A. I. M., Wuillemin, P. H., Leray, P., Pourret, O., & Becker, A. (2007). Réseaux bayésiens, 3e éd..

[99] Russell, S., & Norvig, P. (2010). Intelligence artificielle: Avec plus de 500 exercices. Pearson Education France.

[100] Heckerman, D. (2008). A tutorial on learning with Bayesian networks. Innovations in Bayesian networks, 33-82.

[101] Raschka, S. (2014). Naive bayes and text classification i-introduction and theory. arXiv preprint arXiv:1410.5329.

[102] McCallum, A., & Nigam, K. (1998, July). A comparison of event models for naive bayes

text classification. In AAAI-98 workshop on learning for text categorization (Vol. 752, No. 1, pp. 41-48).

[103] Rennie, J. D., Shih, L., Teevan, J., & Karger, D. R. (2003). Tackling the poor assumptions of naive bayes text classifiers. In Proceedings of the 20th international conference on machine learning (ICML-03) (pp. 616-623).

[104] Berger, A. (1999, January). Error-correcting output coding for text classification. In IJCAI-99: Workshop on machine learning for information filtering.

[105] Zhang, T., & Oles, F. J. (2001). Text categorization based on regularized linear classification methods. Information retrieval, 4(1), 5-31..

[106] Rakotomalala, Ricco. Pratique de l'Analyse Discriminante Linéaire.

[107] Friedman, J., Hastie, T., & Tibshirani, R. (2000). Special invited paper. additive logistic regression: A statistical view of boosting. Annals of statistics, 337-374.

[108] Greene, D., & Cunningham, P. (2006, June). Practical solutions to the problem of diagonal dominance in kernel document clustering. In Proceedings of the 23rd international conference on Machine learning (pp. 377-384)..

[109] Hsu, C. W., Chang, C. C., & Lin, C. J. (2003). A practical guide to support vector classification.

[110] Hsu, C. W., & Lin, C. J. (2002). A comparison of methods for multiclass support vector machines. IEEE transactions on Neural Networks, 13(2), 415-425.

[111] Cheriet, M., Kharma, N., Liu, C. L., & Suen, C. (2007). Character recognition systems: a guide for students and practitioners. John Wiley & Sons.

[112] Setiono, R. (2001). Feedforward neural network construction using cross validation. Neural Computation, 13(12), 2865-2877..

[113] De Villiers, J., & Barnard, E. (1993). Backpropagation neural nets with one and two hidden layers. IEEE transactions on neural networks, 4(1), 136-141.

[114] Kotsiantis, S. B., Kanellopoulos, D., & Pintelas, P. E. (2006). Data preprocessing for supervised leaning. International journal of computer science, 1(2), 111-117.

[115] Karsoliya, S. (2012). Approximating number of hidden layer neurons in multiple hidden layer BPNN architecture. International Journal of Engineering Trends and Technology, 3(6), 714-717.

[116] Genuer, R., & Poggi, J. M. (2017). Arbres CART et Forêts aléatoires, Importance et sélection de variables..

[117] Genuer, R. (2021). Contributions to Random Forests Methods for several Data Analysis Problems (Doctoral dissertation, Université de Bordeaux).

[118] Rigatti, S. J. (2017). Random forest. Journal of Insurance Medicine, 47(1), 31-39..

[119] Montillo, A. A. (2009). Random forests. Lecture in Statistical Foundations of Data Analysis.

[120] Ferreira, A. J., & Figueiredo, M. A. (2012). Boosting algorithms: A review of methods, theory, and applications. Ensemble machine learning, 35-85..

[121] Van Erp, M., Vuurpijl, L., & Schomaker, L. (2002, August). An overview and comparison of voting methods for pattern recognition. In Proceedings Eighth International Workshop on Frontiers in Handwriting Recognition (pp. 195-200). IEEE.

[122] Kittler, J. (1998). Combining classifiers: A theoretical framework. Pattern analysis and Applications, 1(1), 18-27.

[123] Berrar, Daniel. "Cross-Validation." (2019): 542-545..

[124] Mouchère, H. (2007). Étude des mécanismes d'adaptation et de rejet pour l'optimisation

de classifieurs: Application à la reconnaissance de l'écriture manuscrite en-ligne (Doctoral dissertation, INSA de Rennes)..

[125] Sinoara, R. A., Camacho-Collados, J., Rossi, R. G., Navigli, R., & Rezende, S. O. (2019). Knowledge-enhanced document embeddings for text classification. Knowledge-Based Systems, 163, 955-971.

[126] Ali, C. B., Mulki, H., & Haddad, H. (2018). Impact du Prétraitement Linguistique sur l'Analyse de Sentiment du Dialecte Tunisien (). In Actes de la Conférence TALN. Volume 1-Articles longs, articles courts de TALN (pp. 383-392).

[127] Singh, K., Devi, H. M., & Mahanta, A. K. (2017). Document representation techniques and their effect on the document Clustering and Classification: A Review. International Journal of Advanced Research in Computer Science, 8(5).

[128] Buscaldi, D., Felhi, G., Ghoul, D., Le Roux, J., Lejeune, G., & Zhang, X. (2020). Calcul de similarité entre phrases: quelles mesures et quels descripteurs?. In Actes de la 6e conférence conjointe Journées d'Études sur la Parole (JEP, 33e édition), Traitement Automatique des Langues Naturelles (TALN, 27e édition), Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition). Atelier DÉfi Fouille de Textes (pp. 14-25).

[129] Bounabi, M., El Moutaouakil, K., & Satori, K. (2017, March). A comparison of Text Classification methods Method of weighted terms selected by different Stemming Techniques. In Proceedings of the 2nd international Conference on Big Data, Cloud and Applications (pp. 1-9).

[130] Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics, 5, 135-146.

[131] Hong, S. (2016). Improving Paragraph2Vec. In Report.

[132] Feng, X., Liang, Y., Shi, X., Xu, D., Wang, X., & Guan, R. (2017). Overfitting reduction of text classification based on AdaBELM. Entropy, 19(7), 330.

[133] Liu, P., Yu, H., Xu, T., & Lan, C. (2017, December). Research on archives text classification based on Naive Bayes. In 2017 IEEE 2nd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC) (pp. 187-190). IEEE.

[134] Bharti, K. K., & Singh, P. K. (2015). Hybrid dimension reduction by integrating feature selection with feature extraction method for text clustering. Expert Systems with Applications, 42(6), 3105-3114.

[136] Hiemstra, D. (2000). A probabilistic justification for using tf× idf term weighting in information retrieval. International Journal on Digital Libraries, 3(2), 131-139..

[137] Panchal, G., Ganatra, A., Kosta, Y. P., & Panchal, D. (2011). Behaviour analysis of multilayer perceptrons with multiple hidden neurons and hidden layers. International Journal of Computer Theory and Engineering, 3(2), 332-337.

[138] Ettaouil, M., & Ghanou, Y. (2009). Neural architectures optimization and Genetic algorithms. Wseas Transactions On Computer, 8(3), 526-537.

[139] Dahmouni, A., El Moutaouakil, K., & Satori, K. (2016). Robust face recognition using local gradient probabilistic pattern (LGPP). In Proceedings of the Mediterranean Conference on Information & Communication Technologies 2015 (pp. 277-286). Springer, Cham.

[140] Dobbin, K. K., & Simon, R. M. (2011). Optimally splitting cases for training and testing high dimensional classifiers. BMC medical genomics, 4(1), 1-8.

[141] Wu, X., & Li, H. (2017, October). Topic mover's distance based document classification. In 2017 IEEE 17th International Conference on Communication Technology (ICCT) (pp. 1998-2002). IEEE.

[142] Kirk, R. E., & Othmer, D. F. (Eds.). (1947). Encyclopedia of Chemical Technology: Stillbite to thermochemistry (Vol. 13). Interscience Encyclopedia, Incorporated..

[143] Unnikrishnan, P., Govindan, V. K., & Kumar, S. M. (2019). Enhanced sparse representation classifier for text classification. Expert Systems with Applications, 129, 260-272.

[144] Miller, T. (2019). Explanation in artificial intelligence: Insights from the social sciences. Artificial intelligence, 267, 1-38.

[145] Agrawal, R., & Srikant, R. (1994, September). Fast algorithms for mining association rules. In Proc. 20th int. conf. very large data bases, VLDB (Vol. 1215, pp. 487-499).

[146] Scheffer, T. (2005). Finding association rules that trade support optimally against confidence. Intelligent Data Analysis, 9(4), 381-395..

[147] Bathla, H., & Kathuria, K. (2015). Apriori algorithm and filtered associator in association rule mining. International Journal of Computer Science and Mobile Computing, 4(6), 299-306.

[148] Tanaka, K. (1996). An introduction to fuzzy logic for practical applications. philpapers.org

[149] Pasquier, N. (2000, May). Extraction de Bases pour les Règles d'Association à partir des Itemsets Fermés Fréquents. In Inforsid'2000 Congress (pp. 56-77).

[150] Mannila, H., Toivonen, H., & Verkamo, A. I. (1994, July). E cient algorithms for discovering association rules. In KDD-94: AAAI workshop on Knowledge Discovery in Databases (pp. 181-192).

[151] Bounabi, M., El Moutaouakil, K., & Satori, K. (2020). Association models to select the best rules for fuzzy inference system. Adv. Intell. Syst. Comput, 1076, 349-357.

[152] Bezdek, J. C. (2013). Pattern recognition with fuzzy objective function algorithms. Springer Science & Business Media.

[153] Bounabi, M., Moutaouakil, K. E., & Satori, K. (2020, December). The Automatic option of inference rules for the fuzzy TF-IDF. In 2020 IEEE 2nd International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS) (pp. 1-6). IEEE..

[154] Bounabi, M., El Moutaouakil, K., & Satori, K. (2019, October). Text classification using Fuzzy TF-IDF and Machine Learning Models. In Proceedings of the 4th International Conference on Big Data and Internet of Things (pp. 1-6)..

[155] Ansari, A. Q., Biswas, R., & Aggarwal, S. (2013). Neutrosophic classifier: An extension of fuzzy classifer. Applied soft computing, 13(1), 563-573..

[156] Smarandache, F. (2003, September). Definiton of neutrosophic logic-a generalization of the intuitionistic fuzzy logic. In EUSFLAT Conf. (pp. 141-146).

[157] Kandasamy, I., Vasantha, W. B., Obbineni, J. M., & Smarandache, F. (2020). Sentiment analysis of tweets using refined neutrosophic sets. Computers in Industry, 115, 103180.

[158] Akhtar, N., Qureshi, M. N., & Ahamad, M. V. (2017). An improved clustering method for text documents using neutrosophic logic. In Applications of Soft Computing for the Web (pp. 167-179). Springer, Singapore.

[159] Kou, G., Yang, P., Peng, Y., Xiao, F., Chen, Y., & Alsaadi, F. E. (2020). Evaluation of feature selection methods for text classification with small datasets using multiple criteria decision-making methods. Applied Soft Computing, 86, 105836.

[160] Ge, L., & Moh, T. S. (2017, December). Improving text classification with word embedding. In 2017 IEEE International Conference on Big Data (Big Data) (pp. 1796-1805). IEEE..

[161] Wang, H., Smarandache, F., Zhang, Y., & Sunderraman, R. (2010). Single valued neutrosophic sets. Infinite study.

[162] Bounabi, M., Elmoutaouakil, K., & Satori, K. (2021). A new neutrosophic TF-IDF term weighting for text mining tasks: text classification use case. International Journal of Web Information Systems.

[163] Liang, W. E. I. (2017). Exploration on translation of the literary term ambiguity. China Terminology, 19(4), 29.

[164] Zhang, H. Y., Wang, J. Q., & Chen, X. H. (2014). Interval neutrosophic sets and their application in multicriteria decision making problems. The Scientific World Journal, 2014.

[165] Smarandache, F. (2010). Neutrosophic set–a generalization of the intuitionistic fuzzy set. Journal of Defense Resources Management (JoDRM), 1(1), 107-116.

[166] Singh, B., & Mishra, A. K. (2015). Fuzzy logic control system and its applications. Int. Res. J. Eng. Technol, 2(8), 742-746.

[167] Robinson, A. (2016). Non-standard analysis. Princeton University Press.

[168] Radwan, N., Senousy, M. B., & Alaa El Din, M. (2016). Neutrosophic logic approach for evaluating learning management systems. Neutrosophic Sets and Systems, 3, 3-7.

[169] Mullai, M., Broumi, S., & Stephen, A. (2017). Shortest path problem by minimal spanning tree algorithm using bipolar neutrosophic numbers. Infinite Study..

[170] Bounabi, M., Elmoutaouakil, K., & Satori, K. (2021). A new neutrosophic TF-IDF term weighting for text mining tasks: text classification use case. International Journal of Web Information Systems.

Liste des publications

Papiers des Conférences

- **The automatic option of inference rules for the fuzzy TF-IDF**
  Bounabi, M., Moutaouakil, K.E.., Satori, K.
  2020 IEEE 2nd International Conference on Electronics, Control, Optimization and Computer Science, ICECOCS 2020, 2020, 9314404.

- **Neural Embedding Hybrid ML Models for Text Classification**
  Bounabi, M., Moutaouakil, K.E., Satori, K.
  2020 1st International Conference on Innovative Research in Applied Science, Engineering and Technology, IRASET 2020, 2020, 9092230.

- **Association Models to Select the Best Rules for Fuzzy Inference System**
  Bounabi, M., El Moutaouakil, K., Satori, K.
  Advances in Intelligent Systems and Computing, 2020, 1076, pp. 349–357.

- **Text classification using Fuzzy TF-IDF and Machine Learning Models**
  Bounabi, M., El Moutaouakil, K., Satori, K.
  ACM International Conference Proceeding Series, 2019.

- **A comparison of text classification methods method of weighted terms selected by different stemming techniques**
  Bounabi, M., Moutaouakil, K.E., Satori, K.
  ACM International Conference Proceeding Series, 2017, Part F129474, 43

Articles

- Accepted paper entitled "**The Optimal Inference Rules Selection for Unstructured Data Multi-classification**." on the international journal Statistics, Optimization & Information Computing. In October 2021.

- **The impact of neural embedding characteristics on text mining tasks: Document classification use case**
  Bounabi, M., Moutaouakil, K.E., Satori, K.
  International Journal of Advanced Trends in Computer Science and Engineering, 2020, 9(1.5 Special Issue), pp. 103–110, 16.

- **A new neutrosophic TF-IDF term weighting for text mining tasks: text classification use case**
  Bounabi, M., Elmoutaouakil, K., Satori, K.
  International Journal of Web Information Systems, 2021.

- **A comparison of text classification methods using different stemming techniques**
  Bounabi, M., El Moutaouakil, K., Satori, K.
  International Journal of Computer Applications in Technology, 2019, 60(4), pp. 298–306

- **A probabilistic vector representation and neural network for text classification**
  Bounabi, M., Moutaouakil, K.E., Satori, K.
  Communications in Computer and Information Science, 2018, 872, pp. 343–355