REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE MOHAMED BOUDIAF - M'SILA

FACULTE MATHEMATIQUES ET DE L'INFORMATIQUE

DEPARTEMENT D'INFORMATIQUE

N° :……..

DOMAINE : Informatique

FILIERE : Informatique

OPTION : Systèmes d'Informations Avancés

Mémoire présenté pour l'obtention

Du diplôme de Master Académique

Par: Touina Hanane

Intitulé

Classification automatique de textes

Soutenu devant le jury composé de :

……	Université de M'sila Président
M. DABBA Ali	Université de M'sila Rapporteur
M. BOUDAA ABDELGHANI	Université de M'sila Co-Rapporteur
……	Université de M'sila Examinateur

Année universitaire : 2017 /2018

Remerciements

Avant tout,

Je remercie ALLAH qui nous a donné la force,

le courage et l'espoir nécessaire pour accomplir ce travail.

Ce travail aussi modeste.

n'a été rendu possible que grâce aux orientations éclairées de mon encadreur : Monsieur Dabba Ali que nous tenons à lui exprimer ma parfaite gratitude et mes sincères remercîments pour la qualité de son encadrement et pour ses conseils judicieux et avisés.

Je tenons à remercier également les membres de jury pour avoir fait le plaisir d'accepter d'examiner ce travail.

Je dédie ce mémoire à tous ceux qui ont contribué à ma formation et qui m' ont soutenus dans nos études.

Dédicaces

A mes très chers parents qui ont toujours répondu présents dans les moments les plus difficiles et m'ont soutenu et encouragé tout le long de mes études, leurs confiances et leurs Sacrifices qui ont contribués à ma réussite.

A mes chers frères A ma famille

A tous mes amies

Et tous ceux qui par leurs conseils,

leur attention,

leurs encouragements et leur soutien m'ont aidé à réaliser cette œuvre.

Je ne manquerais pas de remercier tous les professeurs qui m'ont suivi pendant mon cursus universitaire.

Résumé

On cherche à présenter le domaine de classification automatique de textes. Le but est de représenter tous les documents sous forme d'une représentation vectorielle dont les composants seront des termes. Notre objectif est d'associer à chaque document non classé sa catégorie en se basant sur un ensemble de textes préalablement classés.

Dans notre approche, nous utilisons l'algorithme des naïves bayes et calculons les métriques de performance et les comparons avec d'autres algorithmes d'apprentissage.

Mots clés : Classification supervisée de textes, filtrage, naïve bayes.

Abstract

We seek to present the field of automatic classification of texts. The goal is to represent all documents in the form of a vector representation whose components will be terms. Our goal is to associate each unclassified document with its category based on a set of previously classified texts.

In our approach we use the naive bayes algorithm and calculate performance metrics and compare them with other learning algorithms.

Keywords: Supervised text classification, filtering, naïve bayes.

ملخص

نحن نسعى لتقديم مجال التصنيف التلقائي للنصوص. الهدف هو تمثيل جميع الوثائق في شكل تمثيل متجه تكون عناصره عبارة عن مصطلحات. هدفنا هو ربط كل وثيقة غير مصنفة بفئتها بناءًا على مجموعة من النصوص المقروءة مسبقاً.

في نهجنا ، نستخدم خوارزمية باييس السذاجة ونحسب مقاييس الأداء ونقارنها بخوارزميات التعلم الأخرى.

كلمات البحث : تصنيف النص تحت الإشراف ، التصفية ، نايف باي

Table des matières

LISTE DES ACRONYMES ..... ***IV***

TABLE DES MATIERES ..... ***I***

LISTE DES FIGURES ..... ***IV***

LISTE DES TABLEAUX ..... ***VI***

INTRODUCTION GENERALE **1**

CHAPITRE 1 : TEXT MINING

1.1. INTRODUCTION ..... ERREUR ! SIGNET NON DEFINI.

1.2. QU'EST CE QUE LE TEXT MINING ? **3**

1.3. CONCEPTS **3**

1.4. OUTILS DE TEXT MINING **4**

1.4.1. OUTILS DE CLASSIFICATION 4
1.4.2. OUTILS DE RESUME AUTOMATIQUE 4
1.4.3. OUTILS D'EXTRACTION DE CONNAISSANCES 5
1.4.4. SUITES LOGICIELLES DE TEXT MINING 5

1.5. PROCESSUS DE LA FOUILLE DE TEXES : **5**

1.6. FONCTIONNALIITES DU TEXT MINING **6**

1.6.1. CLUSTERING 6
1.6.2. ANALYSE DE SENTIMENTS 7
1.6.3. EXTRACTION DE RELATIONS 7

1.7. DES CAS D'UTILISATIONS TRES VARIENT (MARKETING) : **7**

1.7.1. CONNAISSANCE CLIENT : 7
1.7.2. PUBLICITE 8
1.7.3. BANQUES ET ASSURANCES 8
1.7.4. BIOMEDICAL 9

1.8. TECHNIQUES LIEES A LA FOUILLE DE TEXTES **9**

1.8.1. LE TRAITEMENT AUTOMATIQUE DES LANGUES « TAL» 9
1.8.2. LA RECHERCHE D'INFORMATION « RI» 9
1.8.3. L'EXTRACTION D'INFORMATION « EI » 9

1.9. METHODES UTILISEES POUR LA FOUILLE DE TEXTES **10**

1.10. RELATION ENTRE TEXT MINING ET APPRENTISSAGE AUTOMATIQUE 10

1.11. LA MISE EN OEUVRE DU TEXT MINING 10

1.12. CONCLUSION 12

CHAPITRE 2 : METHODES DE CLASSIFICATION

2.1. INTRODUCTION 13

2.2. POURQUOI AUTOMATISER LA CLASSIFICATION? 14

2.3. HISTORIQUE DE LA CATEGORISATION DE TEXTES : 15

2.4. DEFINITION DE LA CLASSIFICATION 16

2.5. LES TYPES DE CLASSIFICATIONS AUTOMATIQUE DE TEXTE : 17

2.5.1. CATEGORISATION (SUPERVISE) 17
2.5.2. CLUSTERING (NON SUPERVISE) 17

2.6. CATEGORISATION DE TEXTE 18

2.6.1. DEFINITION 18
2.6.2. COMMENT CATEGORISER UN TEXTE ? 19
2.6.2.1. L'APPRENTISSAGE 19
2.6.2.2. LE CLASSEMENT 19
2.6.3. PROCESSUS DE LA CATEGORISATION DE TEXTE 20
2.6.4. REPRESENTATION DE TEXTE (CHOIX DES TERMES) 20
2.6.4.1. REPRESENTATION EN SAC DE MOTS (BAG OF WORDS) 21
2.6.4.2. REPRESENTATION AVEC LES RACINES LEXICALES (STEMMING) 21
2.6.4.3. REPRESENTATION AVEC LES LEMMES 22
2.6.4.4. REPRESENTATION AVEC LES N-GRAMMES 22
2.6.4.5. REPRESENTATION CONCEPTUELLE 22
2.6.5. PONDERATION DES TERMES (CODAGE DES TERMES) 22
2.6.5.1. MESURE TF (TERM FREQUENCY) 22
2.6.5.2. MESURE TFIDF (TERM FREQUENCY INVERSE DOCUMENT FREQUENCY) 22

2.7. ALGORITHMES DE CLASSIFICATION AUTOMATIQUE DE TEXTE 23

2.7.1. ALGORITHME DES K-VOISINS LES PLUS PROCHES KNN 23
2.7.1.1. DEFINITION 23
2.7.1.2. PRINCIPE DE FONCTIONNEMENT 24
2.7.1.3. DETAILS D'ALGORITHME DES K-PLUS PROCHES VOISINS KNN 24
2.7.1.4. CRITIQUES DE LA METHODE 25
2.7.1.5. LES DOMAINES D'APPLICATION 25
2.7.2. METHODE DE ROCCHIO 25
2.7.3. LES ARBRES DE DECISION 25
2.7.3.1. DEFINITION 25
2.7.3.2. DETAILS D'ALGORITHME D'ARBRE DE DECISION 26
2.7.3.3. CRITIQUES DE LA METHODE 26

2.7.3.4. LES DOMAINES D'APPLICATION 26
2.7.4. NAÏVE BAYES 27
2.7.5. MACHINES A SUPPORT DE VECTEURS (OU SVM) 27
2.7.6. RESEAUX DE NEURONES 28
2.8. QUEL EST LE MEILLEUR CLASSIFIEUR ? 29
2.9. CRITERES D'EVALUATION DU CLASSIFICATEUR 29
2.10. CONCLUSION 32
CHAPITRE3:EXPERIMENTATION ET IMPLEMENTATION
3.1. INTRODUCTION 33
3.2. LE CLASSIFICATEUR BAYESIEN NAIF (NAIVE BAYESIAN CLASSIFIER) 34
3.2.1. DESCRIPTION DU MODELE BAYESIENNE 35
3.2.2. ESTIMATION DE LA VALEUR DES PARAMETRES 36
3.2.3. CONSTRUIRE UN CLASSIFICATEUR A PARTIR DU MODELE DE PROBABILITES 37
3.2.4. ANALYSE 37
3.2.5. AVANTAGE 37
3.3. PROPOSITION 38
3.4. OUTILS DE DEVELOPPEMENT: 39
3.4.1. LE LANGAGE DE PROGRAMMATION (JAVA) 39
3.4.2. ENVIRONNEMENT DE DEVELOPPEMENT 39
3.4.3. COMPOSANTS DE NETBEANS 40
3.5. PRESENTATION DE LA PLATEFORME WEKA 41
3.5.1. STRUCTURE DE DONNEES 41
3.5.2. CARACTERISTIQUES PRINCIPALES 41
3.6. EXPERIMENTATION 42
3.6.1. PRESENTATION DU CORPUS D'EXPERIMENTATION ET L'APPROCHES UTILISEES 42
3.6.2. PRETRAITEMENTS EFFECTUES SUR LES CORPUS : D'APPRENTISSAGE, DE TEST : 43
3.6.3. APPLICATION DE LE CLASSIFICATEUR BAYESIEN NAÏF EFFECTUES SUR LES CORPUS D'APPRENTISSAGE, DE TEST 44
3.7. DISCUSSION 47
3.8. CONCLUSION: 48
CONCLUSION GENERALE ET PERSPECTIVES 49
REFERENCES BIBLIOGRAPHIQUES 50

Liste des Acronymes

TAL : Le traitement automatique des langues.

RI : La recherche d'information.

EI : L'extraction d'information.

CAH : Classification Ascendante Hiérarchique.

AA : Apprentissage Automatique.

CT : Catégorisation de textes.

KNN : Algorithme des k-voisins les plus proches.

RNA : Les réseaux de neurones.

SVM : Machines à support de vecteurs.

NB: L'algorithme Naïve Bayes.

WEKA: Waikato Environment for Knowledge Analysis.

ARFF : Attribute-Relation File Format.

CSV: Comma-Separated Values.

TF: Term Frequency.

IDF: Inverse Document Frequency.

TFIDF: Term Frequency Inverse Document Frequency.

GUI : Grafical User Interface**.**

API : Application Programming Interface.

EDI : Un environnement de développement intégr

Liste des figures

FIGURE 1.1. SCHEMA GENERAL LA TACHE DE FOUILLE DE TEXTES 6

FIGURE 2.1. PROCESSUS DE CLASSIFICATION ILLUSTRE 16

FIGURE 2.2. SCHEMA DE CLUSTERING 18

FIGURE 2.3. PROCESSUS DE CATEGORISATION DE TEXTES 20

FIGURE 2.4. MATRICE DOCUMENT × TERME 23

FIGURE 2.5. LES VECTEURS A SUPPORT [32] 28

FIGURE 3.1. SCHEMATIC DIAGRAM OF PROPOSED APPROACH 39

FIGURE 3.2. L'INTERFACE GRAPHIQUE DE NETBEANS IDE 8.1 40

FIGURE 3.3. FICHIER .ARFF D'APPRENTISSAGE ET DE TEST UTILISE DANS LES EXPERIMENTATIONS 42

FIGURE 3.4. SELECTION LE FICHIER .ARFF 43

FIGURE 3.5. REPRESENTATION VECTORIEL DE FICHIER .ARFF 43

FIGURE 3.6. NETTOYAGE LE SAC DE MOT 44

FIGURE 3.7. ELIMINATION DES SIGNES DE PONCTUATION 44

FIGURE 3.8. DIVISE DE CORPUS D'APPRENTISSAGE 45

FIGURE 3.9. APPLICATION DE LE CLASSIFICATEUR BAYESIEN NAÏF 45

FIGURE 3.10. RESULTAT DES MESURES DE CORPUS D'APPRENTISSAGE 46

FIGURE 3.11. RESULTAT DES MESURES DE CORPUS DE TESTE 46

Liste des tableaux

TABLEAU 2.1. MATRICE DECISIONNELLE 19

TABLEAU 2.2. EXEMPLE DE LA REPRESENTATION EN « SAC DE MOTS » LES CHIFFRES ET DATES SONT SUPPRIMES DE LA REPRESENTATION 21

TABLEAU 2.3. TABLEAU DE CONTINGENCE 30

TABLEAU 3.1. DESCRIPTION DES DONNEES.....42

TABLEAU 3.2. COMPARAISON LES RESULTATS 47

Introduction générale

La révolution de l'internet a fait exploser les informations textuelles, qui sont un patrimoine vivant des entreprises, des administrations et des particuliers, il est devenu indispensable aux utilisateurs du web de trouver les documents pertinents, pour cette raison il devient de plus en plus important de disposer de solutions efficaces pour conserver, chercher et classer ces informations, afin d'assister les utilisateurs à trouver leurs besoins et faciliter leur travail dans certaines tâches qui sont devenues impossible à traiter manuellement. Donc il est très intéressant de compter sur une application automatique qui est la classification et la catégorisation des textes.

Le but de nos travaux est de développer un modèle fondé sur l'apprentissage automatique pour la catégorisation de textes en utilisant la méthode de Naïve Bayes (NB) donc on peut distinguer deux grandes parties :

- La catégorisation de textes.
- La catégorisation thématique avec Naïve Bayes (NB).

Ce mémoire va être organisé de la façon suivante :

Dans le premier chapitre, nous présentons une introduction à notre domaine d'étude via l'explication de sujet du text Mining.

Dans le deuxième chapitre, nous allons résume d'une façon claire la catégorisation automatique des textes et exposer les différentes algorithmes d'apprentissage utilisé pour cette dernier.

Dans le troisième chapitre, nous met l'accent sur la classification de textes en utilisant l'algorithme Naïve Bayes (NB) et expose l'architecture du logiciel conçu et son fonctionnement, ainsi que son implémentation et quelques exemples de démonstration.

Et nous avons terminé par une conclusion qui nous voir ultérieurement

CHAPITRE 1 : TEXT MINING

1.1 INTRODUCTION

Text Mining est un nouveau domaine en plein essor qui tente de glaner des informations significatives à partir du texte en langage naturel. Il peut être vaguement caractérisé comme le processus d'analyse de texte pour extraire des informations utiles à des fins particulières. Comparé au type de données stockées dans les bases de données, le texte est non structuré, amorphe et difficile à traiter algorithmiquement. Néanmoins, dans la culture moderne, le texte est le véhicule le plus commun pour l'échange formel d'informations. Le domaine de l'exploration de texte traite habituellement des textes dont la fonction est la communication d'informations factuelles ou d'opinions, et la motivation pour essayer d'extraire des informations d'un tel texte est convaincant, même si le succès n'est que partiel. Cette dernière est l'outil le plus communément offert aux utilisateurs pour accéder à de grandes collections de données. C'est un ensemble de systèmes qui s'intéressent à l'analyse et au traitement des informations textuelles. Dans le présent chapitre on va exposer l'ensemble des méthodes et outils liés à la fouille de données textuelles.

1.2.QU'EST CE QUE LE TEXT MINING ?

Le Text Mining est une branche du Data Mining qui se spécialise dans le traitement de corpus de textes pour en analyser le contenu puis en extraire des connaissances. Les principales tâches à accomplir consistent en la reconnaissance de l'information présente dans le document et son interprétation. Tout cela est possible grâce à une recherche sémantique reposant sur l'analyse du langage naturel et la gestion de bases de connaissances spécialisées. On peut par exemple distinguer une plainte d'un client à une demande d'information, ou même un spam d'un message publicitaire, en inspectant la tournure des phrases [26].

L'exploration de texte fait référence à une collection de méthodes utilisées pour trouver des modèles et créer de l'intelligence à partir de données de texte non structurées [31].

L'extraction d'informations implicites, précédemment inconnues et potentiellement utiles à partir d'une grande quantité de ressources textuelles [31].

La fouille de textes ou « l'extraction de connaissances » dans les textes est une spécialisation de la fouille de données et fait partie du domaine de l'intelligence artificielle. Cette technique est souvent désignée sous l'anglicisme text mining [29].

Il désigne un ensemble de traitements informatiques consistant à extraire des connaissances selon un critère de nouveauté ou de similarité dans des textes produits par des humains pour des humains. Dans la pratique, cela revient à mettre en algorithme un modèle simplifié des théories linguistiques dans des systèmes informatiques d'apprentissage et de statistiques [29].

Les disciplines impliquées sont donc la linguistique calculatoire, l'ingénierie des langues, l'apprentissage artificiel, les statistiques et l'informatique [29].

1.3.CONCEPTS

Le Text Mining permet de rechercher une information précise expertisée pour un contexte spécifique, de comparer plusieurs documents et en déduire une opération adaptée. Il permet ensuite la réalisation de tâches telles qu'architecturer l'information extraite de façon à ce qu'elle soit réutilisable rapidement [26].

On fait appel au Text Mining aussi bien pour une base de textes volumineuse que pour un choix réduit d'articles. Dans les deux cas, les tâches réalisées consistent en l'analyse linguistique, statistique et sémantique de documents pour en extraire les informations pertinentes [26].

Il peut enfin servir à enrichir l'index d'un moteur de recherche pour améliorer la consultation des documents grâce à une représentation sémantique des données et aider ainsi à leur classification automatique [26]. Pour l'exploration de texte, le texte sera traité et transformé en représentation numérique [31].

1.4. OUTILS DE TEXT MINING

Les outils de Text Mining sont dotés de fonctionnalités d'extraction et de catégorisation d'informations non structurées [27].

Les outils de Text Mining ont pour objectif de faciliter la découverte de connaissances. En première analyse nous pouvons identifier 4 types d'outils de Text Mining [5] :

- les outils de classification.
- les outils de résumé automatique.
- les outils d'extractions de connaissances.
- les suites logicielles de Text Mining.

1.4.1. Outils de classification

Les outils de classification permettent de réaliser des traitements à haut niveau de valeur ajoutée sur des fonds documentaires. Ils assurent la réalisation des opérations suivantes :

- génération automatique de plans de classement : organisation de façon dynamique et intuitive d'un ensemble non structuré de documents en thèmes et établissement d'une véritable cartographie du fonds documentaire considéré.
- catégorisation automatique : classement par apprentissage des documents dans un plan de classement préexistant, il est possible à ce niveau de catégoriser des fonds documentaires de natures hétérogènes [5].

1.4.2. Outils de résumé automatique

L'objectif d'un outil de résumé automatique est de produire, à partir du contenu d'un document, une représentation condensée dans laquelle les informations importantes du texte original sont préservées tout en tenant compte des besoins de l'utilisateur.

Il existe deux grandes catégories de techniques pour construire un résumé automatique :

- la reformulation, qui s'attache à comprendre le contenu du document de manière à générer un nouveau texte, contenant de nouvelles phrases, différentes du texte original.
- l'extraction, qui repose sur l'extraction d'information. Le résumé obtenu contient les

éléments jugés importants du texte original [5].

1.4.3. Outils d'extraction de connaissances

La vocation des outils d'extraction de connaissances est d'identifier l'information pertinente. Ces outils mettent en œuvre une analyse du texte pour interpréter et construire une représentation formelle qui permettra d'apporter automatiquement des réponses précises à l'utilisateur. Il ne s'agit donc pas simplement de sélectionner un fragment brut du texte, mais de mettre des éléments en relation pour restituer une information complète et structurée à partir d'un patron prédéfini [5].

1.4.4. Suites logicielles de Text Mining

Les suites logicielles de Text Mining sont de véritables boîtes à outils dont la vocation est de faciliter la découverte de connaissances.

Cet ensemble d'outils propose l'ensemble des fonctionnalités qui sont offertes par les différents outils que nous venons de voir [5].

1.5. PROCESSUS DE LA FOUILLE DE TEXES :

Les étapes nécessaires pour effectuer le processus de la fouille de textes sont :

- **Acquisition** : Source de données telle que : corpus textuels, bibliothèques électroniques, Web…etc.
- **Nettoyage des données** : Segmentation du texte, élimination des mots vides, lemmatisation.
- **Filtrage** : Sélection des mots les plus pertinents.
- **Extraction des connaissances** : Application de l'un des algorithmes de la fouille de textes [22].

Dans ce schéma, les données figurent dans des ovales tandis que le programme réalisant la tâche est matérialisé par un rectangle. C'est bien sûr dans les différentes données que la spécificité de la fouille de textes se manifestera : tout ou parties d'entre elles seront de nature textuelle, ou en découleront après un prétraitement. Ce schéma est très simple, mais nous verrons qu'il oblige tout de même à se poser quelques bonnes questions. Par exemple, il n'est pas toujours facile de distinguer ce qui joue le rôle de données d'entrée ou de ressources dans la définition d'une tâche. Un bon critère serait le suivant :

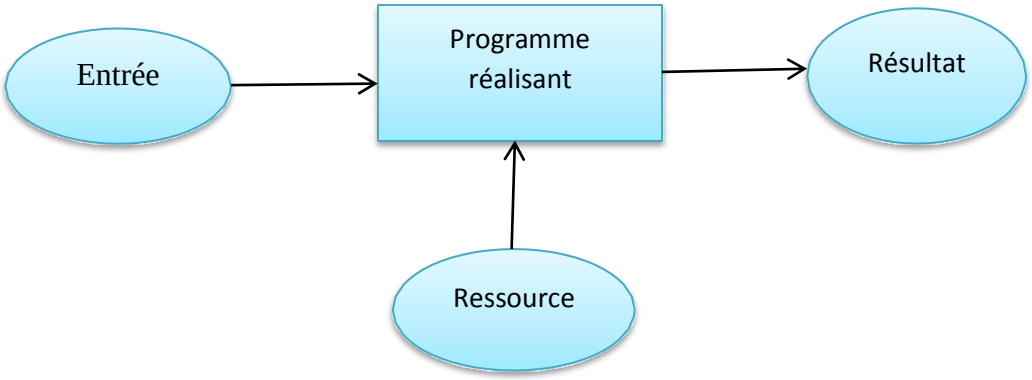


Figure 1.1 : Schéma général la tache de fouille de textes.

Une ressource est une donnée stable, qui n'est pas modifiée d'une instance d'application à une autre, alors que la donnée d'entrée, elle, change chaque fois. Certaines ressources sont obligatoires dans la définition de certaines tâches, d'autres facultatives. C'est souvent par ce biais que des connaissances externes et générales peuvent être intégrées au processus de réalisation de la tâche. Les ressources sont donc un des principaux leviers pour faire rentrer un peu de linguistique dans le domaine de la fouille de textes. C'est le cheval de Troie des linguistes [17].

1.6. FONCTIONNALITES DU TEXT MINING

Le Text Mining apporte de nouvelles fonctionnalités au Data Mining [26], présentées ci-dessous :

1.6.1. Clustering

Un cas usuel d'utilisation Text Mining est la catégorisation de documents de manière non supervisée suite à la collecte de données texte non-structurées. Prenons l'exemple d'un sondage d'opinion public, réalisé par l'entreprise Eaagle aux Etats Unis et qui a posé la question suivante : What should be the priorities for the next US president?

Comme il s'agit d'une question ouverte, il est essentiel d'avoir un outil capable de traiter les réponses textuelles. La première étape consiste alors en la création de catégories « Recherche de catégories par l'algorithmes statistiques », pour trier les réponses, en analysant la fréquence d'apparition des termes ou de groupes de termes. La deuxième étape consiste en la classification les réponses à l'aide d'algorithme de Clustering selon les catégories [26].

Le regroupement fait référence au regroupement d'enregistrements, d'observations ou de cas en classes, chaque thème contient des objets partageant les mêmes propriétés [2].

La différence entre le clustering et la classification est que le clustering ne dépend de classes prédéfinies. Dans la classification, chaque enregistrement est affecté à une classe prédéfinie. En clustering, il n'y a pas de classes prédéfinies ni d'exemples. Les enregistrements sont regroupés en fonction de l'autosimilarité [2].

1.6.2. Analyse de sentiments

Qu'ils soient sur les réseaux sociaux ou sur les sites de ventes en ligne, les utilisateurs sont incités à donner leur avis. Si dans certains cas il est possible de noter un produit ou un film en cochant des étoiles, il s'agit la plupart du temps d'un commentaire écrit par l'internaute, ce qui représente une donnée non-structurée [26].

À partir de modèles pré-entrainés, le Text Mining permet d'évaluer si un commentaire est plutôt positif ou négatif. Les méthodes les plus élémentaires, basées sur des analyses statistiques comme l'attribution de poids aux mots clés à valeur sentimentale, permettent de générer des pourcentages de positivité [26].

Quant aux méthodes actuelles les plus poussées, elles passent par des outils d'analyse du langage naturel afin de détecter le sens de la phrase et d'en estimer le sentiment correspondant, ce qui rend possible la détection de plainte des clients ou encore de demande d'informations [26].

1.6.3. Extraction de relations

Traitement du langage naturel. En effet, elle permet de détecter une relation sémantique entre un ou plusieurs groupes de mots [26].

Par exemple: La petite souris grignote le morceau de fromage dans la cuisine. Les Relations sont entre mangeur et mange et aliment.

Cet outil est notamment utilisé pour déceler des relations précises dans un texte ou pour découvrir de nouvelles données alimentant des bases de connaissances [26].

1.7. DES CAS D'UTILISATIONS TRES VARIENT (MARKETING) :

Le texte non structuré est très commun et peut représenter la majorité des informations disponibles pour un projet de recherche ou d'exploration de données particulier. De ce fait, les connaissances que l'on pourrait en tirer sont la raison de l'expansion des techniques de fouille de texte et leurs applications dans plusieurs disciplines et secteurs d'activité variés [26] :

1.7.1. Connaissance client :

Les données issues du contact avec le client (enquêtes, mails, lettre de réclamation,

retranscription de messages téléphoniques..) constituent une source pertinente d'informations sur le client, venant renseigner notamment sur les besoins et leur adéquation avec les offres de service, ainsi que sur leur satisfaction et leur intention de fidélité. Le volume de ces données est important et ne cesse de croître. Le Text Mining va permettre de faire ressortir l'information pertinente de ces gros corpus de textes, pour une meilleure connaissance client. Par exemple, la suite logicielle TEMIS Insight Discovered est utilisée depuis quelques années par EDF pour parfaire sa connaissance client [26].

1.7.2. Publicité

Les réseaux sociaux ont créé un nouvel univers qui permet de faire communiquer des personnes du monde entier sur des plateformes internet communes. Dans ce contexte, les agences Web, qui cherchent à proposer des annonces publicitaires pour mieux cibler les clients, se retrouvent face à plein de données à exploiter mais sous format non structuré. C'est alors qu'intervient le Text Mining, spécialisé dans ce type de données.

En effet, à l'aide du Text Mining, on peut s'inspirer des habitudes de l'internaute ainsi que de l'analyse sémantique du contenu texte qu'il lit et qu'il écrit, afin que s'affiche sur sa page une publicité suffisamment pertinente pour qu'elle attire son attention et qu'il clique dessus.

Fort de cette tendance, Critéo, une entreprise française fondée en 2005, utilise cette technologie aussi bien sur le web que dans les applications mobiles, ce qui a entrainé des ventes records chez ses clients [26].

1.7.3. Banques et assurances

Les banques et les assurances font de plus en plus appel à des systèmes automatisés pour optimiser leurs tarifs et pour mieux connaître leurs clients. Depuis des années l'usage du Data Mining est devenu coutumier, et, pour compléter cet outil les chercheurs ont mis en place des outils de Text Mining analysant directement les plaintes, constats et factures pour éviter toute analyse manuelle.

Des logiciels comme SAS® Enterprise Miner et Clementine Text Mining sont capables de traiter des données non-structurées en utilisant d'abord les technologies du Text Mining, pour l'analyse des textes, et par la suite des outils de Data Mining plus classiques.

L'enjeu est grand, le but ici est à la fois de tenter de prévoir le nombre d'accidentés pour l'année suivante, d'élaborer des profils précis des assurés, de connaître la satisfaction globale des clients. D'autres outils plus poussés permettent de détecter des constats et factures

frauduleuses [26].

1.7.4. Biomédical

Il existe un intérêt croissant pour la fouille textuelle et les stratégies d'extraction de l'information appliquées à la littérature sur la biologie moléculaire et biomédicale, en raison du nombre croissant de publications électroniques disponibles stockées dans des bases de données telles que PubMed.

« Avec le développement des systèmes biologiques, les chercheurs ont tendance à comprendre les systèmes biomédicaux complexes d'un point de vue systèmes biologiques. Ainsi, la pleine utilisation du Text Mining pour faciliter les systèmes biologiques de recherche sur le cancer est en train de devenir une préoccupation majeure » [26].

1.8. TECHNIQUES LIEES A LA FOUILLE DE TEXTES

La fouille de textes s'apparente à d'autres domaines avec qui elle est très complémentaire traitement automatique des langues (TAL) et la recherche documentaire (RI) et l'extraction de l'information (EI) [19].

1.8.1. Le traitement automatique des langues « TAL»

Depuis une quinzaine d'années, avec la généralisation de l'outil informatique et d'Internet, les applications du TAL au sens large du terme se multiplient dans les disciplines philologiques. Le TAL est une discipline à la frontière de la linguistique et de l'informatique, qui concerne l'application de programmes et techniques informatiques à tous les aspects du langage humain [19].

1.8.2. La recherche d'information « RI»

La recherche d'information « RI » s'intéresse aux documents dans leur globalité et aux thèmes qu'ils abordent, pour comparer les documents et détecter des typologies. Elle cherche à détecter tous les thèmes présents [19].

1.8.3. L'extraction d'information « EI »

L'extraction d'information « EI » recherche des informations précises dans les documents, sans les comparer, en tenant compte de l'ordre et de la proximité des mots pour discriminer des énoncés différents ayant des mots clés identiques. L'extraction d'information consiste en l'alimentation d'une base de données structurée à partir de données exprimées en langage naturel. Il s'agit de détecter dans le texte en langage naturel les mots correspondant à chaque champ de la base de données. L'analyse est locale. L'extraction d'information est plus

complexe, car elle nécessite d'effectuer une analyse lexicale et morphosyntaxique pour reconnaître les constituants du texte (phrases, mots, verbes, adjectifs), leur nature pour détecter les phrases pertinentes et en extraire les informations voulues [19].

1.9. METHODES UTILISEES POUR LA FOUILLE DE TEXTES

La fouille de textes fait appel à l'analyse de données qui se caractérise par deux grandes familles de méthodes :

Les méthodes de classification qui produisent un regroupement d'objets ou d'individus décrits par un certain nombre de variables ou de caractères (classification de type nuées dynamiques, Classification Ascendante Hiérarchique - CAH -).

Les méthodes factorielles qui font essentiellement des représentations graphiques caractérisant les liens entre les différents critères (Analyse en Composantes Principales, Analyse Factorielle des Correspondances, Analyse des Correspondances Multiples) [13].

1.10. RELATION ENTRE TEXT MINING ET APPRENTISSAGE AUTOMATIQUE

Le Text Mining fait appel à diverses méthodes d'analyse, comme la linguistique, la classification automatique ou la catégorisation. L'application de ces méthodes, nécessite en fonction du type d'indicateur que l'on souhaite mettre en place, une plus ou moins grande connaissance formalisée du domaine couvert par les documents à analyser.

Comme le Text Mining cherche des informations cachées et utilise des algorithmes communs d'intelligence artificielle, d'apprentissage automatique et de statistiques [20].

1.11.LA MISE EN OEUVRE DU TEXT MINING

On peut distinguer deux étapes principales dans les traitements mis en place par la fouille de textes :

La première étape, l'analyse, consiste à reconnaître les mots, les phrases, leurs rôles grammaticaux, leurs relations et leur sens. Cette première étape est commune à tous les traitements. Une analyse sans interprétation n'a que peu d'intérêt et les deux sont dépendantes. C'est donc le rôle de la seconde étape d'interpréter cette analyse.

La seconde étape, l'interprétation de l'analyse, permet de sélectionner un texte parmi d'autres. Des exemples d'applications sont la classification de courriers en spam, c'est-à-dire les courriers non sollicités, ou non spam, l'application de requêtes dans un moteur de recherche de documents ou le résumé de texte qui sélectionne les phrases représentatives d'un texte voire les reformule.

Le critère de sélection peut être d'au moins deux types : la nouveauté et la similarité. Celui de la nouveauté d'une connaissance consiste à découvrir des relations, notamment des implications qui n'étaient pas explicites car indirectes ou entre deux éléments éloignés dans le texte. Celui de la similarité ou contradiction par rapport à un autre texte ou encore la réponse à une question spécifique consiste à découvrir des textes qui correspondent le plus à un ensemble de descripteurs dans la requête initiale. Les descripteurs sont par exemple les noms et verbes les plus fréquents d'un texte [30].

1.12.CONCLUSION

La fouille de données textuelle est une discipline née dans la communauté de recherche documentaire et de l'intelligence artificielle dans le but de valoriser les bases de données textuelles. Elle offre des perspectives nouvelles pour la statistique et répond au défi du traitement des grands corpus de textes. Aujourd'hui la fouille de données textuelles est la branche de la statistique exploratoire qui cherche à découvrir des structures inconnues et utiles dans les textes. Dans le chapitre suivant, nous exposons les déférentes méthodes de classifications automatiques de textes.

CHAPITRE 2 : METHODES DE CLASSIFICATION

2.1. INTRODUCTION

La classification de textes a pour objectif de regrouper les textes similaires, c'est à dire thématiquement proches, au sein d'un même ensemble. L'intérêt d'une telle démarche est d'organiser les connaissances de façon à pouvoir effectuer, par la suite, une recherche ou une extraction d'information efficace.

Le volume de documents numériques s'accroissant, des besoins en classification automatique se sont fait ressentir aussi bien sur internet (moteurs de recherche), qu'au sein des entreprises (classement de documents internes, dépêches d'agences, etc.). On distingue dans le domaine de la classification automatique deux types d'approches : la classification supervisée et la classification non supervisée. Ces deux méthodes diffèrent sur la façon dont les classes sont générées. Dans ce qui suit notre travail va être concentré sur la catégorisation de textes (la classification supervisée).

2.2. POURQUOI AUTOMATISER LA CLASSIFICATION?

On assiste aujourd'hui à un accroissement de la quantité d'information textuelle disponible et accessible d'une manière exponentielle. D'après les derniers chiffres, on parle de plus de 200 millions de serveurs hôtes sur Internet et plus de 3 milliards de pages, la taille des corpus tests utilisés est passée de quelques mégaoctets à plusieurs Giga-octets [9].

Dans les années 1996-1997, Reuters a produit un peu plus de 800 000 nouvelles en anglais par année. Si l'on ajoute aux articles écrits par les journalistes de l'agence ceux provenant d'autres sources, on arrive à un total de 5.5 millions de textes anglais par année à catégoriser. À un moment, l'organisation employait 90 personnes dédiées à l'étiquetage de ces documents. Il serait à coup sûr très intéressant de pouvoir déterminer avec précision le coût de classification. De combien de temps a besoin un humain pour associer un texte à une catégorie ? En pratique, il s'agit d'une question difficile à répondre. Assurément, plusieurs variables influencent le phénomène et les lignes qui suivent porteront sur certaines d'entre elles [9].

Certainement une grande partie du temps consommé pour classer un document est employé dans sa lecture, puis éventuellement à sa relecture. On peut aussi imaginer que la longueur des textes à classer est assez déterminante du temps qui va être requis pour cette opération, et sans doute, d'une personne à une autre, la vitesse de lecture varie. Une fois cette étape achevée, il faut trancher à quelle(s) catégorie(s) ce texte appartient. Au temps de réflexion exigé s'ajoute, certainement, le temps de se référer à la description des classes et éventuellement de consulter d'autres textes préalablement associés à certaines classes, pour valider la décision.

D'autres facteurs interviennent également comme, comme par exemple le nombre de classes qui peut faire la différence : plus il y a de classes différentes, autrement dit plus il y a d'étiquettes possibles pour un texte donné, plus il est difficile de faire un choix parmi celles-ci. Aussi, plus la sémantique des catégories est précise, fine, détaillée, plus il faut faire attention avant d'y associer un document. À cet égard, classer des documents appartenant soit à la catégorie «informatique» soit à la catégorie «mathématiques» est vraisemblablement plus aisée que celle de classer des documents appartenant à l'une ou l'autre des catégories «Intelligence artificielle» , «Génie logiciel» et «Système d'information» [9].

En conséquence, nous pouvons résumer les contraintes majeures qui s'opposent au traitement manuel de classification des documents textuels dans les trois points suivants :

- La réalisation manuelle de cette tâche par un expert est extrêmement coûteuse en terme de temps et personnel car il s'agit de lire attentivement chaque texte, au vu de la quantité phénoménale de textes aujourd'hui accessibles (par le biais du réseau Internet en particulier) [9].
- Les traitements manuels sont peu flexibles et leur généralisation à d'autres domaines est quasi impossible; c'est pourquoi on cherche à mettre au point des méthodes automatiques [9].
- Cette opération peut être perçue comme subjective puisque basée sur l'interprétation du document, deux experts peuvent classer différemment un même document, ou encore un même expert peut classer différemment un même document soumis à deux instants différents [9].

Ainsi l'intérêt de la recherche d'automatisation de la classification de textes n'est plus à démontrer, et c'est dans cette perspective que plusieurs travaux de recherche se concentrent ces dernières années [9].

2.3. HISTORIQUE DE LA CATEGORISATION DE TEXTES :

C'est une discipline assez ancienne, en 1627, Gabriel Naudé propose un classement selon cinq grands thèmes : théologie, jurisprudence, histoire, sciences et arts, belles lettres. Le désir de maîtriser l'Univers se fait sentir dans la multiplication des encyclopédies. L'encyclopédie de Diderot (parue entre 1751 et 1772) est organisée selon l'ordre alphabétique avec des renvois associatifs alors que celle de Panckoucke (parue de 1776 à 1780) suit une organisation méthodique selon un ordre arborescent [9].

Le système de classification par thème, apparu dès les débuts de l'écriture et institutionnalisé à Alexandrie conduisit à la création par Dewey, en 1876, d'un système de classification« universel ». Il s'agit d'une classification documentaire de type encyclopédique. Toutefois l'idée d'effectuer la classification de textes par des machines remonte au début des années 60 et qui a connu des progrès considérables à partir des années 90 avec l'apparition d'algorithmes beaucoup plus performants qu'auparavant [9].

Jusqu'au début des années 80, pour construire un classifier, il fallait consacrer d'importantes ressources humaines à cette tâche. Plusieurs experts éditaient des règles manuellement puis les affinaient au fur et à mesure des tests. L'avènement des de l'AA s'est donc traduit par un gain de temps conséquent. Il n'est plus nécessaire par exemple de reconfigurer tout le système en cas de changement d'arborescence [9].

Ces évolutions technologiques et algorithmes avancées font aujourd'hui de la catégorisation un outil fiable [9].

Au début des années 90, les travaux proviennent essentiellement de la communauté de Recherche d'Information (RI). En effet, les méthodes de numérisation, les algorithmes de classification et les méthodologies de test ont été adaptés à la CT en particulier au cours des conférences TREC [9].

La communauté d'Apprentissage Automatique (AA) s'est intéressée elle aussi à ce problème il y a une dizaine d'années en le considérant comme domaine d'application à ces algorithmes de reconnaissance des formes. Actuellement, les méthodes de numérisation de texte restent largement inspirées de la RI alors que les classificateurs les plus performants sont issus de l'AA [9].

Une autre communauté composée essentiellement de statisticiens et de linguistes, traite également le problème de la CT en s'appuyant sur les méthodes d'analyse de données. Le but ici n'est pas de créer un système qui classe automatiquement des documents sans intervention humaine mais d'extraire des informations synthétiques du corpus. Les problématiques traitées ici sont par exemple l'étude des genres littéraires ou la détermination de l'auteur d'un texte [9].

2.4. DEFINITION DE LA CLASSIFICATION

La classification est un processus qui permet d'organiser des données en classes homogènes dont le but est la simplification de la représentation de ces derniers. La classification donc crée une hiérarchie qui améliore la recherche de documents ; elle génère une vue d'ensemble qui favorise la connaissance de l'environnement ciblé. La classification d'un document est réalisée à partir d'une évaluation statistique de ses données. La figure 2.1 illustre ce mécanisme [5].

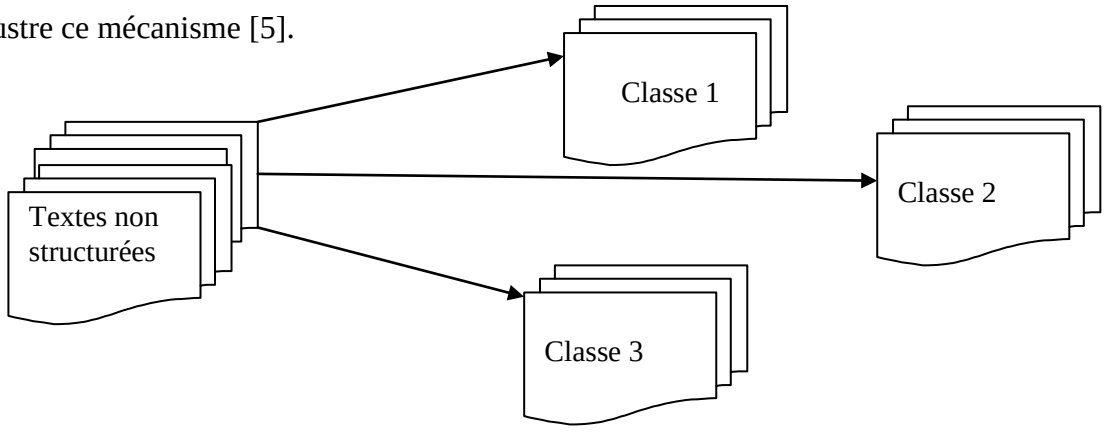


Figure 2.1. Processus de classification illustré.

Prend en entrée des documents sous format vectoriel retourne en sortie des documents classifiés [5].

2.5. LES TYPES DE CLASSIFICATIONS AUTOMATIQUE DE TEXTE :

L'objectif de la CT est de classer de façon automatique les documents dans des catégories qui ont été définies soit préalablement par un expert, il s'agit alors de classification supervisée ou catégorisation, soit de façon automatique, il s'agit alors de classification non supervisée ou encore clustering [9].

2.5.1. Catégorisation (Supervisé)

Contrairement à l'apprentissage non supervisé, nous commençons ici par un ensemble de classes connues et définies à l'avance. Nous disposons aussi d'une sélection initiale de données dont la classification est connue. Ces données sont supposées indépendantes et identiquement distribuées. Elles nous servent pour l'apprentissage de l'algorithme. La classification se fait par l'algorithme selon le modèle qu'il a appris [22].

Ainsi, la catégorisation de textes correspond à la procédure d'affectation d'une ou de plusieurs catégories ou classes prédéfinies à un texte. Elle correspond à la classification supervisée pour l'apprentissage automatique et à la discrimination en statistiques alors que la recherche d'informations utilise des termes plus proches de l'application concernée : filtrage ou routage [9].

Aujourd'hui, cette problématique utilise largement des méthodes issues de l'apprentissage automatique et beaucoup d'algorithmes d'apprentissage supervisé lui ont été appliqués (Naïve Bayes, K-plus proches voisins, arbres de décision, machines à vecteurs support, réseaux de neurones, etc...) [9].

2.5.2. Clustering (Non supervisé)

Toutefois quand l'ensemble des catégories n'est pas donné au départ, et qu'il s'agit de le créer en regroupant les textes en classes qui possèdent un certain degré de cohérence interne, on est dans un contexte de classification non supervisée pour l'apprentissage automatique.

La classification non supervisée consiste à trouver de manière automatique une organisation cohérente à un groupe de documents homogènes pour construire des regroupements cohérents (des classes ou clusters), elle correspond en statistiques au clustering, qui est également le terme utilisé en recherche d'informations.

Le clustering consiste donc, à diviser les objets (dans notre cas des textes) en groupes

sans connaître à priori leurs classes d'appartenance [9].

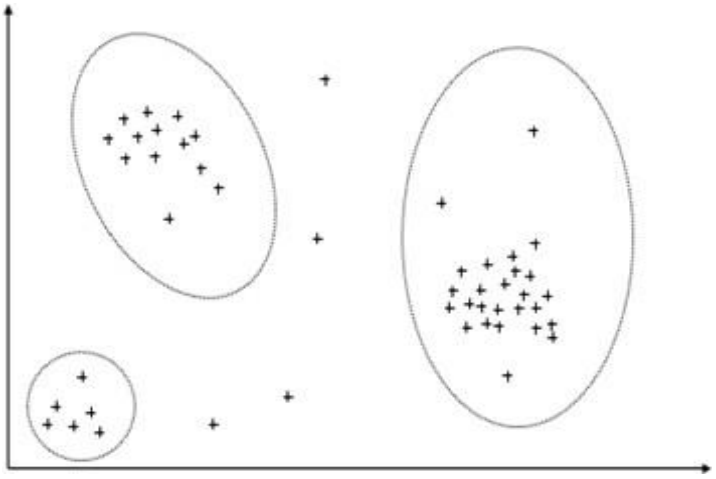


Figure 2.2. Schéma de clustering[26].

2.6. CATEGORISATION DE TEXTE

2.6.1. Définition

La catégorisation automatique de textes est d'apprendre à une machine à classer un texte dans la bonne catégorie en se basant sur son contenu.

Dans une catégorisation de texte : la classification s'apparente au problème de l'extraction de la sémantique d'un texte, puisque l'appartenance d'un document à une catégorie est étroitement liée à la signification de ce texte [21].

La catégorisation de textes comme étant la recherche d'une relation bijective qui consiste à "chercher une liaison fonctionnelle entre un ensemble de textes et un ensemble de catégories (étiquettes, classes)". C'est à dire associer une catégorie à un texte libre, en fonction des informations qu'il contient [16].

Sébastian, définit formellement dans [6] la catégorisation des textes comme le processus qui consiste à associer une valeur booléenne à chaque paire (d_j, c_i) D × C, où D l'ensemble des textes et C l'ensemble des catégories. La valeur V (Vrai) est alors associée au couple (d_j, c_i) si le texte dj appartient à la classe ci tandis que la valeur F (Faux) lui sera associée dans le cas contraire. Le but de la catégorisation de texte est de construire une procédure (modèle, Classificateur) notée : Φ: D × C → {V, F} qui associe une ou plusieurs étiquettes (catégories) à un document dj telle que la décision donnée par cette procédure coïncide le plus possible avec la fonction Φ^ : D × C → {V, F}, la vraie fonction qui retourne pour chaque vecteur dj une valeur c_i. Dans la matrice décisionnelle Une valeur de V pour v_{ij} signifie que le document d_j doit être placé dans la catégorie c_i, bien qu'une valeur de F signifie le contraire.

	C1	C2	…….	….....	……...	……	Cn
d1	V11	V12	……	……	……	……	V1n
d2	V21	V22	…….	……...	……...	……	V2n
……	……..	……	…….	……	……	……	……
……	…..	……	…….	……...	…….	……	…….
Dm	Vn1	Vn2	……...	……	……	……	Vmn

Tableau 2.1 Matrice décisionnelle.

2.6.2. Comment catégoriser un texte ?

Le processus de catégorisation intègre la construction d'un modèle de prédiction qui, en entrée, reçoit un texte et, en sortie, lui associe une ou plusieurs étiquettes. Pour identifier la catégorie ou la classe à laquelle un texte est associé, un ensemble d'étapes est habituellement suivies. Ces étapes concernent principalement la manière dont un texte est représenté, le choix de l'algorithme d'apprentissage à utiliser et comment évaluer les résultats obtenus pour garantir une bonne généralisation du modèle appris. Le processus de catégorisation, intégrant la phase de classement de nouveaux textes, est résumé dans la (figure 2.3). Il comporte deux phases que l'on peut distinguer comme suit [25]:

2.6.2.1. L'apprentissage

Qui comprend plusieurs étapes et aboutit à un modèle de prédiction :

- nous disposons d'un ensemble de textes étiquetés (pour chaque texte nous connaissons sa catégorie) ;
- à partir de ce corpus, nous extrayons les k descripteurs (mots, termes) (t1; ..; tk) les plus pertinents au sens du problème à résoudre ;
- nous disposons alors d'un tableau « descripteurs × individus », et pour chaque texte nous connaissons la valeur de ses descripteurs et son étiquette [25] ;

2.6.2.2. Le classement

Le classement d'un nouveau texte dx, qui comprend deux étapes :

- recherche puis pondération des occurrences (t_1; ...; t_k) des termes dans le texte dx à classer ;

- application d'un algorithme d'apprentissage sur ces occurrences et le tableau précédent afin de prédire l'étiquette de ce texte d_x [25].

2.6.3. Processus de la catégorisation de texte

Le processus reçoit en entrée un document textuel afin de lui trouver sa catégorie, pour cela plusieurs étapes doivent d'être suivies. Ces étapes sont :

- La représentation des textes.
- La Pondération des termes.
- La réduction de la taille du vocabulaire.
- Choix de classificateur.
- Evaluation du modèle [21].

La figure 2.3 résume le processus de catégorisation des textes qui comporte deux phases : l'apprentissage et le classement

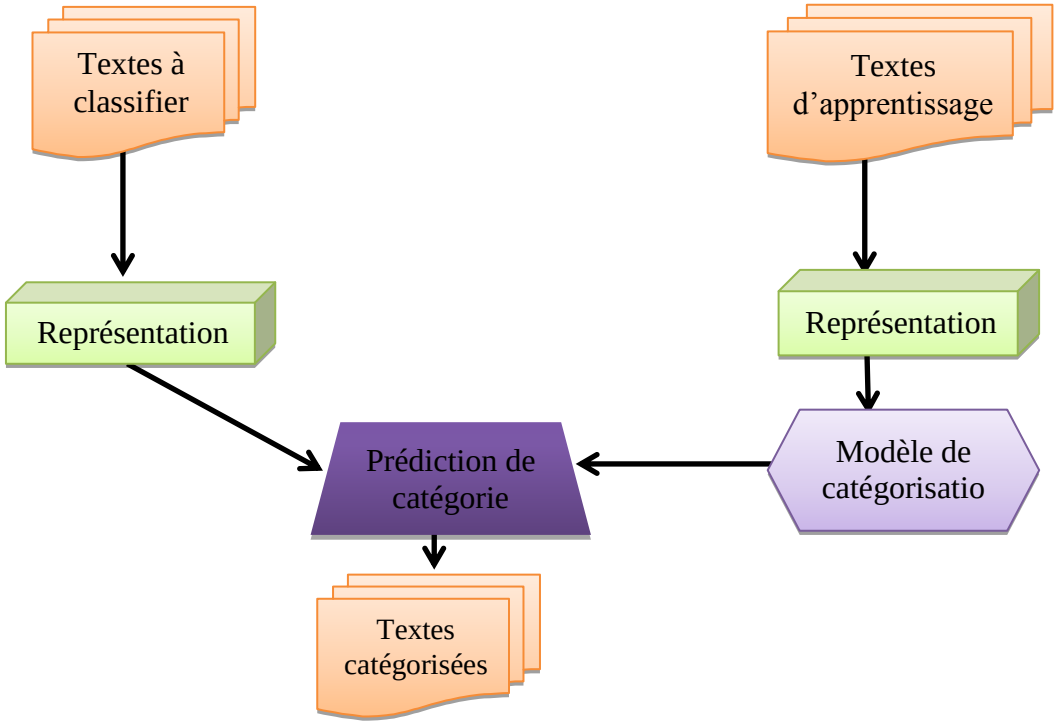


Figure 2.3. Processus de catégorisation de textes.

2.6.4. Représentation de texte (choix des termes)

Dans la catégorisation de textes, comme dans la recherche documentaire, on transforme le document dj en un vecteur $d_j = (w_{1j}, w_{2j}, ..., w|T|j)$, où T est l'ensemble de termes (descripteurs) qui apparaissent au moins une fois dans le corpus (la collection) d'apprentissage. Le poids w_{kj} correspond à la contribution du terme t_k à la sémantique du texte d_j [25].

2.6.4.1. Représentation en sac de mots (bag of words)

Cette méthode consiste à représenter le document sous forme d'un vecteur de mots. Le processus qui permet de convertir le texte d'un document en un ensemble de termes est appelé l'analyse lexicale qui permet de reconnaître les espaces de séparation des mots, les ponctuations, les chiffres,…etc., pour qu'ils seront tous supprimés de la représentation. Cette représentation a comme avantage d'exclure toute analyse grammaticale et toute notion distance entre les mots, mais présente comme inconvénient la difficulté de délimiter les mots dans certaines langues telles que l'Arabe ou l'Allemand [25].

2.6.4.2. Représentation avec les racines lexicales (stemming)

Cette méthode consiste à remplacer les mots du document par leurs racines lexicales, qui peut être réalisée en utilisant un des algorithmes les plus connus pour la langue anglaise qui est l'algorithme de Porter [Porter, 1980] de normalisation de mots qui sert à supprimer les affixes de ces derniers pour obtenir une forme canonique. Cette méthode a comme avantage de regrouper les différentes flexions d'un mot dans une seule composante, et comme inconvénient la perte de sens car la racine extraite peut être commune à des mots se rapportant à des concepts différents [25].

Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. go Text FA to 87121 to receive entry question(std txt rate)T&C\'s apply 08452810075over18\'s',1	
Mot	**Occur**
a	1
apply	1
Cup	1
entry	2
Free	1
FA	2
final	1
go	1
in	1
receive	1
rate	1
Std	1
st	1
S	2
Tkts	1
Text	1
txt	1
wkly	1

Tableau 2.2.Exemple de la représentation en « sac de mots » Les chiffres et dates sont supprimés de la représentation.

2.6.4.3. Représentation avec les lemmes

Cette méthode consiste à remplacer les mots du document par leurs lemmes, elle doit utiliser l'analyse grammaticale afin de remplacer les verbes par leurs formes infinitives et les noms par leurs formes au singulier. En effet, Un mot donné peut avoir différentes formes dans un texte, mais leur sens reste le même [25].

2.6.4.4. Représentation avec les n-grammes

Cette méthode consiste à représenter le document par des n-grammes. Le n-gramme est une séquence de n caractères consécutifs. Cette technique présente plusieurs avantages. Les n-grammes capturent automatiquement les racines des mots les plus fréquents sans passer par l'étape de recherche des racines lexicales, indépendante de la langue, les espaces sont pris en considération parce qu'en effet, la non prise en compte de ces derniers introduit du bruit [25].

2.6.4.5. Représentation conceptuelle

Cette méthode consiste à représenter le document sous forme d'un ensemble de concepts, ces concepts peuvent être capturés en utilisant les réseaux sémantiques ou les sous arbres (un sous arbre représente une hiérarchie de concepts) [25].

2.6.5. Pondération des termes (Codage des termes)

La pondération des termes permet de mesurer l'importance d'un terme dans un document. Cette importance est souvent calculée à partir de considérations et interprétations statistiques (ou parfois linguistiques). Les méthodes les plus populaires sont [21] :

2.6.5.1. Mesure TF (Term Frequency)

Cette mesure est proportionnelle à la fréquence du terme dans le document (pondération locale). Elle peut être utilisée telle quelle ou selon plusieurs déclinaisons (log (TF), présence/absence,. . .) [21].

2.6.5.2. Mesure TFIDF (Term Frequency Inverse Document Frequency)

Le poids d'un terme T dans un document D est calculé comme suit :

$$TFIDF(T, D) = TF(T, D) * \log (N/ DF(T)) \qquad (2.1)$$

Avec : TF(T, D) : la fréquence du terme dans le document, N : le nombre total de documents de la base documentaire. DF(T) : le nombre de documents contenant le terme [21].

Chaque entrée représente un vecteur de termes ou ***pnm*** **est** le poids du terme ***T*****m** dans le document **Dn** et **Ci est** la classe attribuée au document **Di**.

	T1	T2	…	Tm	
D1	P11	P12	…	P1m	Ca
D2	P21	P22	…	P2m	Cb
…	…	…	…	…	...
Dn	Pn1	Pn2		Pnm	CK

Figure 2.4. Matrice Document × Terme.

2.7. ALGORITHMES DE CLASSIFICATION AUTOMATIQUE DE TEXTE

La plupart des algorithmes d'apprentissage supervisée tentent donc de trouver un modèle ou « une fonction mathématique » qui explique le lien entre les documents en entrée et les classes de sortie [13].

Ces jeux d'exemples sont donc utilisés par l'algorithme. Dans le cas de la classification de documents, on fournit donc à la machine des exemples sous la forme (Document, Classe). Cette méthode de raisonnement est appelée inductive car on induit de la connaissance (le modèle) à partir des données d'entrée (les documents) et des sorties (leurs catégories). Grâce à ce modèle, on peut alors déduire les classes de nouvelles données : le modèle est utilisé pour prédire. Le modèle est bon s'il permet de bien prédire [13].

Il existe de nombreux algorithmes d'apprentissage supervisé, notamment :

- L'algorithme des K plus proches voisins (ou K-NN).
- Méthode de Rocchio.
- Les arbres de décision.
- L'algorithme de Naïve Bayes.
- Machines à support de vecteurs (ou SVM).
- Les réseaux de neurones(RNA) [13].

2.7.1. Algorithme des k-voisins les plus proches KNN

2.7.1.1. Définition

L'algorithme des k plus proches voisins est connue en anglais sous le nom K- nearest-neighbor (K-NN) ; est une méthode très connue dans le domaine de la catégorisation des textes pour prédire où classer un nouveau document, il faut le comparer avec ceux déjà classés en cherchant ses K plus proches voisins. Une fois ces derniers déterminés, le nouveau document est classé dans la catégorie qui inclut le maximum de voisins parmi les K trouvés.

Deux paramètres sont utilisés : le nombre **K** et la fonction de similarité pour comparer le

nouveau document à ceux déjà classés telle que la distance euclidienne par exemple qui est donnée par l'équation suivante [13]:

$$\mathbf{D(X, Y)=}\sqrt{\sum_{j=1}^{m}(Xj - Yj)^2} \qquad (2.2)$$

2.7.1.2. principe de fonctionnement

Les K-PPV nécessitent :

- Un entier K.
- Une base d'apprentissage (une base d'objets - des documents par exemple-).
- Une métrique pour la proximité (une mesure de distance) [19].

2.7.1.3. Détails d'algorithme des k-plus proches voisins KNN

L'algorithme des k-plus proches voisins est l'un des algorithmes les plus simples de la classification supervisée, étant donné que l'ensemble des classes est défini à l'avance, cette méthode permet d'obtenir de très bons résultats de classification d'après [16] dont l'algorithme est le suivant [18]:

Paramètre : le nombre K de voisins
Contexte : un échantillon de L textes classés en C = c_1, c_2, ..., c_n classes
Début
  Pour chaque texte T faire
  Transformer le texte T en vecteur T = (x_1, x_2, ..., x_m),
    Déterminer les K plus proches textes du texte T selon une métrique de distance,
  Combiner les classes de ces K exemples en une classe C.
  Fin pour
Fin
Sortie : le texte T associé à la classe C.

Le choix du paramètre K est primordial pour le bon fonctionnement de cette méthode. Une grande base d'apprentissage permet une plus grande valeur de K, et un K petit est nécessaire pour des petites bases d'apprentissage [23].

La distance entre un texte et ses voisins se fait via une métrique de distance. Cette métrique peut être comme suit :

- **Mesure Cosinus** qui consiste à calculer le produit scalaire entre deux vecteurs $\boldsymbol{a}$ et $\boldsymbol{b}$, que nous divisons par le produit de la norme de ces deux vecteurs. La formule de la mesure Cosinus est alors la suivante : $\mathbf{cos}\ (\boldsymbol{a}, \boldsymbol{b}) = \frac{\sum(\boldsymbol{a}*\boldsymbol{b})}{\sqrt{\sum \mathbf{a^2} * \sum \boldsymbol{b^2}}}$ (2.3)

D'autres mesures ont été proposées dans la littérature, parmi lesquelles on peut citer les

mesures de Jaccard et Dice.

- **Mesure de Jaccard** La formule de la mesure de Jaccard est alors la suivante **:**

$$\mathbf{J\ (a,\ b)}=\frac{\sum(\boldsymbol{a}*\boldsymbol{b})}{\sum \mathrm{a}^2+\sum b^2-\sum ab} \quad (2.4)$$

- **Mesure de Dice** La formule de la mesure de Dice est alors la suivante :

$$\mathbf{D\ (a,\ b)\ =\ 2*}\frac{\sum(\boldsymbol{a}*\boldsymbol{b})}{\sum(\boldsymbol{a^2}+\boldsymbol{b^2})} \quad (2.5)$$

2.7.1.4. Critiques de la méthode

L'avantage que présente cette méthode est sa simplicité et son efficacité qui fait d'elle une méthode très utilisée ; toutefois, on peut lui reprocher le fait qu'elle utilise un nombre important d'objets pour calculer la similarité avec un nouvel objet à classer et plus le nombre d'objets est grand plus le temps d'exécution est très important [19].

2.7.1.5. Les domaines d'application

La méthode peut s'appliquer dès qu'il est possible de définir une distance sur les champs. Or, il est possible de définir des distances sur des champs complexes tels que des informations géographiques, des textes, des images, et du son. C'est parfois un critère de choix de la méthode K-PPV car les autres méthodes traitent difficilement les données complexes. On peut noter, également, que la méthode est robuste au bruit [19].

2.7.2. Méthode de Rocchio

Le classificateur de Rocchio parue est l'un des plus simples et plus ancien algorithmes de classification du modèle vectoriel. Ce classificateur a Eté largement utilisé dans la catégorisation textuelle. L'avantage de ce type de classificateur est la simplicité et l'interopérabilité. L'apprentissage de ce type de classificateur est souvent précédé par une sélection et une réduction de termes [16].

2.7.3. Les arbres de décision

2.7.3.1. Définition

Les arbres de décision sont des méthodes puissantes de catégorisation. Leur principe est de construire des modèles de classification à partir d'un ensemble de données d'apprentissage. Il existe plusieurs algorithmes d'apprentissage inductifs basés sur les arbres de décision ; le plus connu est l'algorithme ***C4.5***.

C4.5 a pour rôle de générer un arbre de décision initial à partir des exemples tests. Cet arbre est composé de deux types de nœuds :

- ***La feuille* :** indique la classe.
- ***Le nœud de décision*** : contient les règles de décision à appliquer aux attributs du nouveau vecteur. En effet, les documents sont représentés dans un espace vectoriel.

Lorsqu'un nouveau vecteur doit être affecté à une classe, ***C4.5*** commence par tester les règles de la racine ; ensuite, à chaque nœud de décision, l'algorithme décide, grâce au résultat du test, quelle est la branche à suivre et ainsi de suite, jusqu'à atteindre un nœud feuille où se trouve la bonne classe [19].

2.7.3.2. Détails d'algorithme d'arbre de décision

En général, l'algorithme d'arbre de décision se présente de la façon suivante [18]:

```
Arbre ← arbre vide ; nœud_courant
Répéter
    Décider si le nœud courant est terminal
        Si le nœud terminal alors lui affecter une classe
        Sinon sélectionner un test et créer autant de nœuds fils qu'il y a de réponse au test
        Passer au nœud suivant (s'il existe)
Jusqu'à obtenir un arbre de décision
```

2.7.3.3. Critiques de la méthode

L'arbre de décision est une méthode très utilisée pour des raisons d'efficacité et de simplicité par rapport aux autres méthodes existantes ; en effet, elle est bien compréhensible pour tous les utilisateurs puisque ses règles sont de type « Si...Alors… ». Elle repose sur l'utilisation simultanée de variables qualitatives et quantitatives (discrètes ou continues). Sa classification est rapide : pour classer un nouvel objet, nous parcourons un seul chemin de l'arbre de la racine jusqu'à la feuille qui correspond à sa classe.

Par contre, ses performances sont moins bonnes lorsque les classes sont nombreux, les arbres peuvent être très complexes et ne sont pas nécessairement optimaux. La construction des arbres de décisions nécessite généralement beaucoup de temps car il faut trouver le bon choix des attributs. Si les données évoluent dans le temps, il est nécessaire de relancer la phase d'apprentissage sur un échantillon complet qui contient les nouveaux et les anciens exemples [13].

2.7.3.4. Les domaines d'application

Cette méthode peut être utilisée dans plusieurs domaines tels que :

Les études (pour comprendre les critères prépondérants dans l'achat d'un produit, l'impact

des dépenses publicitaires), les ventes (pour analyser les performances par région, par enseigne, par vendeur), l'analyse de risques (pour détecter les facteurs prédictifs d'un comportement de non-paiement), Le domaine médical (pour étudier les rapports existant entre certaines maladies et des particularités physiologiques ou sociologiques) [19].

2.7.4. Naïve Bayes

L'algorithme Naïve Bayes (NB), est une autre méthode bien connue en apprentissage, elle est également utilisée dans la catégorisation de documents. Elle se base sur un modèle probabiliste, qui vise à estimer la probabilité conditionnelle d'une catégorie sachant un document et affecte au document la catégorie la plus probable [11]. Son utilisation est résumée comme suit : Lors de la phase d'entraînement, le classificateur calcule les probabilités qu'un nouveau document appartienne à telle catégorie à partir de la proportion des documents d'entraînement appartenant à cette catégorie. Il calcule aussi la probabilité qu'un mot donné soit présent dans un texte, sachant que ce texte appartient à telle catégorie. Quand un nouveau document doit être classé, on calcule les probabilités qu'il appartienne à chacune des catégories à l'aide de la règle de Bayes [15].

La probabilité à estimer est donc : $\boldsymbol{P\ (c_j \setminus a_1, a_2, a_3, ..., a_n)}$ où : $\boldsymbol{c_j}$ est une catégorie et $\boldsymbol{a_i}$ est un descripteur. A l'aide du théorème de Bayes, on obtient :

$$\mathbf{P}\ (\boldsymbol{c_j \setminus a_i}) = \frac{\boldsymbol{p(a_1 a_2 a_3 \ldots\ldots a_n \setminus c_j) p(c_j)}}{\boldsymbol{p(a_1 a_2 a_3 \ldots\ldots a_n)}} \qquad (2.6)$$

Ce classificateur a comme avantage: possibilité en ligne et comme inconvénient: lorsque le modèle est mal spécifié, on aura intérêt à utiliser une méthode discriminative [7].

2.7.5. Machines à support de vecteurs (ou SVM)

Les machines à support de vecteurs (SVM) sont à l'origine de nouvelles méthodes de catégorisations, bien que les premières publications sur le sujet datent des années 60 [19].

Avant d'aborder le principe de fonctionnement général des SVM voici quelques notions de base :

- **Hyperplan** : est un séparateur d'objets des classes. De cette notion, nous pouvons dire qu'il est évident de trouver une mainte d'hyperplans mais la propriété délicate des SVM est d'avoir l'hyperplan dont la distance minimale aux exemples d'apprentissage est maximale, cet hyperplan est appelé L'hyperplan optimal, et la distance appelée marge [19].
- **Vecteurs Support :** ce sont les points qui déterminent l'hyperplan tels qu'ils soient

les plus proches de ce dernier [19]. Voici un schéma représentatif de ces notions :

Le principe des SVM consiste en une stratégie de minimisation structurelle du risque mais le problème revient à trouver une frontière de décision qui sépare l'espace en deux régions, à trouver l'hyperplan qui classifie correctement les données et qui se trouve le plus loin possible de tous les exemples. On dit qu'on veut maximiser la marge qui veut dire la distance du point le plus proche de l'hyperplan [19].

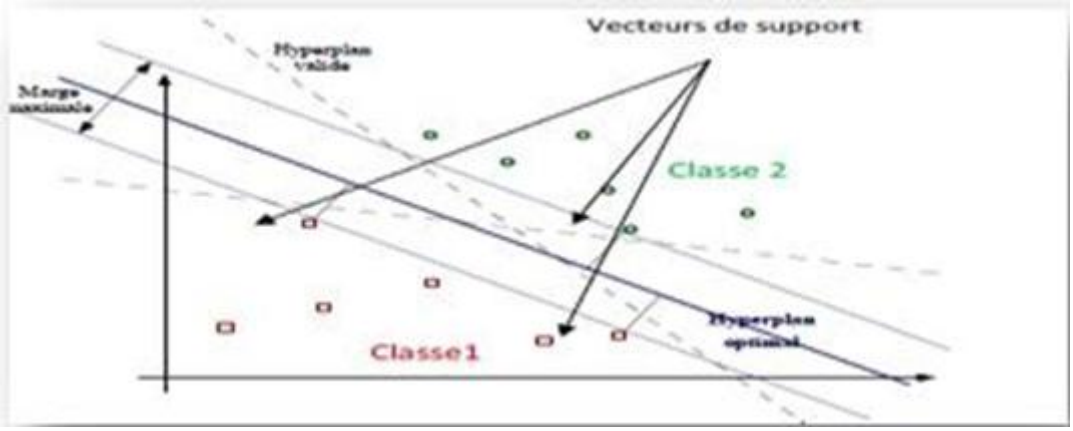


Figure 2.5.Les vecteurs à support [18].

Dans le cas de la catégorisation des textes, les entrées sont des documents et les sorties sont des catégories. En considérant un classificateur binaire, on voudra lui faire apprendre l'hyperplan qui sépare les documents appartenant à la catégorie et ceux qui n'en font pas partie [24].

Les SVM conviennent bien pour la classification de textes parce qu'une dimension élevée ne les affecte pas puisqu'ils se protègent contre le sur apprentissage. Autrement dit, il affirme que peu d'attributs sont totalement inutiles à la tâche de classification et que les SVM permettent d'éviter une sélection agressive qui aurait comme résultat une perte d'information. On peut se permettre de conserver plus d'attributs. Également, une caractéristique des documents textuels est que lorsqu'ils sont représentés par des vecteurs, une majorité des entrées sont nulles. Or, les SVM conviennent bien à des vecteurs dits clairsemés. Un autre aspect positif des SVM est qu'aucun ajustement de paramètres manuel n'est requis, car ils ont l'habileté de trouver automatiquement des paramètres adéquats [15].

2.7.6. Réseaux de neurones

Les réseaux de neurones (Artificial Neural Network) sont généralement optimisés par des méthodes d'apprentissage de type statistique grâce à leur capacité de classification et de généralisation, telles que : la classification automatique de codes postaux ou la prise de

décision concernant un achat boursier [24].

Un réseau de neurone est en général composé d'une succession de couches dont chacune prend ses entrées sur les sorties de la précédente. Chaque couche i est composée de Ni neurones, prenant leurs entrées sur les Ni-1 neurones de la couche précédente. À chaque synapse est associé un poids synaptique, de sorte que les Ni-1 sont multipliés par ce poids, puis additionnés par les neurones de niveau i, ce qui est équivalent à multiplier le vecteur d'entrée par une matrice de transformation [24].

Mettre l'une derrière l'autre, les différentes couches d'un réseau de neurones revient à mettre en cascade plusieurs matrices de transformation et pourrait se ramener à une seule matrice produit des autres, s'il n'y avait à chaque couche, la fonction de sortie qui introduit un non linéarité à chaque étape [24].

Ceci montre l'importance du choix judicieux d'une bonne fonction de sortie : un réseau de neurones dont les sorties seraient linéaires n'aurait aucun intérêt [12].

2.8. QUEL EST LE MEILLEUR CLASSIFIEUR ?

D’après l’étude menée au cours de ce chapitre, On peut constater qu’il existe un nombre important de méthodes pour la classification de documents. Toutes sont issues des recherches sur l’apprentissage et la plupart d’entre elles nécessitent de représenter chaque document sous la forme d’un vecteur. On peut constater également que toutes ces méthodes peuvent s’avérer très lentes étant donné qu’elles traitent une bonne quantité de mots pour chaque document. Néanmoins, l’efficacité de traitement varie d’une méthode à l’autre selon les choix du domaine et des résultats voulus.

Généralement on peut dire que le bon classificateur qui donne la meilleure prévision [13].

2.9. CRITERES D’EVALUATION DU CLASSIFICATEUR

Diverses façons existantes aujourd’hui ont pour objectif de comparer les décisions prises par le classificateur automatique à celles des experts humains et de calculer un score de performance :

Pour mieux illustrer ces différentes mesures on prend pour point de départ la table de contingence illustrée par le tableau 2.3 :

	Documents appartenant à la Catégorie	Documents n'appartenant pas à la catégorie
Documents assignés à la catégorie par le classificateur	A	b
Documents rejetés à la catégorie par le classificateur	C	d

Tableau 2.3. Table de contingence.

On définit à partir des statistiques de cette table les mesures suivantes :

- **Précision:** a / (a + b), soit le nombre d'assignations correctes sur le nombre total d'assignations.
- **Rappel**: a / (a + c), soit le nombre d'assignations correctes sur le nombre d'assignations qui auraient dû être faites.
- **Exactitude**: (a + d) / (a + b + c + d).
- **Erreur**: (b + c) / (a + b + c + d).

Comme un document appartient généralement à un petit nombre de catégories sur l'ensemble, un classificateur qui rejetterait tous les documents présenterait seulement un faible taux d'erreur et une exactitude quand même très élevée. Entraîner un classificateur sur la base de l'optimisation d'un de ces deux critères tendrait à créer un programme qui n'accepte aucun document dans sa catégorie. C'est la raison pour laquelle la précision et le rappel sont les mesures les plus rencontrées dans la littérature [15].

- **F_mesure :** Plusieurs indicateurs ont été créés, mais le plus usuel est la F_mesure qui prenant en compte la valeur relative de la précision et du rappel est calculé comme suit : $$\textbf{F-mesure} = \frac{\left((1+\beta^2)*précision*rappel\right)}{\beta^2*(précision+rappel)} \quad (2.7)$$

Souvent utilisé sous cette forme, (avec β^2 =1) : $$\textbf{F-mesure} = \frac{(2*précision*rappel)}{précision+rappel} \quad (2.8)$$

- **Macro-moyenne et Micro-moyenne :** La précision et le rappel globaux, c'est-à-dire sur toutes les classes, peuvent être calculés à travers une moyenne des résultats obtenus pour chaque catégorie. Deux approches sont distinguées :

- **La micro-moyenne** qui fait d'abord la somme des éléments du calcul – vrais positifs, faux positifs et négatifs – sur l'ensemble des n classes, pour calculer la précision et le

rappel globaux ; **La macro-moyenne** qui calcule d'abord la précision et le rappel sur chaque classe **i**, puis en fait la moyenne sur les n classes

Dans la micro-moyenne chaque classe compte proportionnellement au nombre d'éléments qu'elle comporte : une classe importante comptera davantage qu'une petite classe. Dans la macro- moyenne, chaque classe compte à égalité [3].

Micro-moyenne :

$$\textbf{Précision} = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n} (TP_i + FP_i)} \quad (2.9)$$

$$\textbf{Rappel} = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n} (TP_i + FN_i)} \quad (2.10)$$

Macro-moyenne :

$$\textbf{Précision} = \frac{\sum_{i=1}^{n} \left(\frac{TP_i}{(TP_i + FP_i)} \right)}{n} \quad (2.11)$$

$$\textbf{Rappel} = \frac{\sum_{i=1}^{n} \left(\frac{TP_i}{(TP_i + FN_i)} \right)}{n} \quad (2.12)$$

Avec :

TP_i : nombre de documents correctement attribués à la classe i. FP_i : nombre de documents faussement attribués à la classe i.

FN_i : nombre de documents appartenant à la classe i et non retrouvés par le système.

n : nombre de classes.

2.10. CONCLUSION

Les méthodes d'apprentissage automatique ont montré leur utilité dans de nombreux domaines et ont permis de résoudre un grand nombre de problèmes industriels, c'est pourquoi les chercheurs sont décidé de faire appel à ce type de technologie pour résoudre le problème de catégorisation de documents.

Dans ce chapitre nous avons présenté les algorithmes d'apprentissage supervisés les plus connus ainsi que leurs avantages et leurs inconvénients. Nous avons également introduit les différents moyens d'évaluation d'un classificateur.

Dans le chapitre suivant nous présentons notre méthode d'apprentissage automatique pour la catégorisation de documents : Naïve Bayes (NB).

CHAPITRE 3: EXPERIMENTATION ET IMPLEMENTATION

3.1 INTRODUCTION

La classification bayésienne est utilisée comme méthode d'apprentissage probabiliste. Les classificateurs de Naïve Bayes sont parmi les algorithmes connus les plus réussis pour apprendre à classer des documents de texte. Dans ce chapitre, on vise brièvement les outils et les moyens utilisés pour implémenter la classification textuelle Naïve Bayes. Ainsi que l'environnement de programmation choisi et l'ensemble des interfaces générées par notre application.

3.2.LE CLASSIFICATEUR BAYESIEN NAIF (NAIVE BAYESIAN CLASSIFIER)

La classification naïve bayésienne est un type de classification Bayésienne probabiliste simple basée sur le théorème de Bayes avec une forte indépendance (dite naïve) des hypothèses. Elle met en œuvre un classificateur bayésienne naïf, ou classificateur naïf de Bayes, appartenant à la famille des classificateurs linéaires [13].

Un terme plus approprié pour le modèle probabiliste sous-jacent pourrait être « modèle à Caractéristiques statistiquement indépendantes » [13].

En termes simples, un classificateur bayésien naïf suppose que l'existence d'une caractéristique pour une classe, est indépendante de l'existence d'autres caractéristiques. Un fruit peut être considéré comme une pomme s'il est rouge, arrondi, et fait une dizaine de centimètres. Même si ces caractéristiques sont liées dans la réalité, un classificateur bayésien naïf déterminera que le fruit est une pomme en considérant indépendamment ces caractéristiques de couleur, de forme et de taille [13].

Selon la nature de chaque modèle probabiliste, les classificateurs bayésiens naïfs peuvent être entraînés efficacement dans un contexte d'apprentissage supervisé [13].

Dans beaucoup d'applications pratiques, l'estimation des paramètres pour les modèles bayé siens naïfs repose sur le maximum de vraisemblance. Autrement dit, il est possible de travailler avec le modèle bayésienne naïf sans se préoccuper de probabilité bayésienne ou utiliser les méthodes bayésiennes [13].

Malgré leur modèle de conception « naïf » et ses hypothèses de base extrêmement simplistes, les classificateurs bayésienne naïfs ont fait preuve d'une efficacité plus que suffisante dans beaucoup de situations réelles complexes. En 2004, un article a montré qu'il existe des raisons théoriques derrière cette efficacité inattendue [10]. Toutefois, une autre étude de 2006 montre que des approches plus récentes (arbres renforcés, forêts aléatoires) permettent d'obtenir de meilleurs résultats [4].

L'avantage du classificateur bayésienne naïf est qu'il requiert relativement peu de données d'entraînement pour estimer les paramètres nécessaires à la classification, à savoir moyennes et variances des différentes variables. En effet, l'hypothèse d'indépendance des variables permet de se contenter de la variance de chacune d'entre elle pour chaque classe, sans avoir à calculer de matrice de covariance [13].

3.2.1.Description du modèle Bayésienne

Le modèle probabiliste pour un classificateur est le modèle conditionne $\boldsymbol{p}$ ($\boldsymbol{C\backslash F1}$, $\boldsymbol{Fŋ}$) où C : est une variable de classe dépendante dont les instances ou classes sont peu nombreuses, conditionnée par plusieurs variables caractéristiques $\boldsymbol{F1, Fŋ}$ [13].

Lorsque le nombre de caractéristiques $\boldsymbol{ŋ}$ est grand, ou lorsque ces caractéristiques peuvent prendre un grand nombre de valeurs, baser ce modèle sur des tableaux de probabilités devient impossible [13].

Par conséquent, nous le dérivons pour qu'il soit plus facilement soluble. À l'aide du théorème de Bayes, nous écrivons : $\boldsymbol{p(C\backslash F1, \dots\dots, Fn) = \frac{P(C)P(F1,\dots\dots,Fn\backslash C)}{P(F1,\dots\dots,Fn)}}$ (3.1)

En langage courant, cela signifie : $\boldsymbol{Postérieure = \frac{antérieure \times vaaisemblance}{évidence}}$ (3.2)

En pratique, seul le numérateur nous intéresse, puisque le dénominateur ne dépend pas de C et les valeurs des caractéristiques $\boldsymbol{F}$ sont données. Le dénominateur est donc en réalité constant. Le numérateur est soumis à la loi de probabilité à plusieurs variables [13].

($\boldsymbol{C}$,$\boldsymbol{F1}$,……..,$\boldsymbol{Fŋ}$) (3.3)

Et peut-être factorisé de la façon suivante, en utilisant plusieurs fois la définition de la probabilité conditionnelle :

P(C, F1, …, Fn) = p (C) p(F1,….,Fn\C)
= p(C) p(F1\C)p(F2,……,Fn\C,F1)
= p(C) p(F1\C) p(F2\C,F1) p (F3,…..,Fn\C,F1,F2)
= p(C) p(F1\C) p(F2\C,F1) p(F3\C,F1,F2)p(F4,…,Fn\C,F1,F2,F3)
= p(C) p(F1\C) p(F2\C,F1) p(F3\C,F1,F2) …..p (Fn\C,F1,F2,F3,…) (3.4)

C'est là que nous faisons intervenir l'hypothèse naïve : si chaque $\boldsymbol{F}$ est indépendant des autres caractéristiques$\boldsymbol{y \neq j}$ R alors :

Pour tout $\boldsymbol{\neq j}$ R, par conséquent la probabilité conditionnelle peut s'écrire :

$\mathbf{P(Fi\backslash C, Fj) = p(Fi\backslash C)}$ (3.5)

$\mathbf{P(C,F1,\dots,Fn)\ p(\ C)\ p(\ F1\backslash C)\ P(F2\backslash C)\ p(F3\backslash C) \dots\dots = P(C) \prod_{i=1}^{n} P(\boldsymbol{Fi\backslash C})}$ (3.6)

Par conséquent, en tenant compte de l'hypothèse indépendance ci-dessus, la probabilité conditionnelle de la variable de classe C peut être exprimée par où :

$$\mathbf{P\ (C\backslash F1,\dots\dots,Fn) = \frac{1}{Z} p\ (C) \prod_{i=1}^{n} P(\boldsymbol{Fi\backslash C})} \quad (3.7)$$

où Z (appelé « évidence ») est un facteur d'échelle qui dépend uniquement de $\boldsymbol{F1}$,…, à savoir une constante dans la mesure où les valeurs des variables caractéristiques sont connues [13].

Les modèles probabilistes ainsi décrits sont plus faciles à manipuler, puisqu'ils peuvent être factorisés par l'antérieure P(C) (probabilité a priori de C) et les lois de probabilité indépendantes P(Fi|C). S'il existe K classes pour C et si le modèle pour chaque fonction peut être exprimé selon paramètres, alors le modèle bayésien naïf correspondant dépend de (k − 1) + n r k paramètres [13].

Dans la pratique, on observe souvent des modèles où K=2 (classification binaire) et r=1 (les caractéristiques sont alors des variables de Bernoulli). Dans ce cas, le nombre total de paramètres du modèle bayésien naïf ainsi décrit est de 2n+1, avec n le nombre de caractéristiques binaires utilisées pour la classification [13].

3.2.2.Estimation de la valeur des paramètres

Tous les paramètres du modèle (probabilités a priori des classes et lois de probabilités associées aux différentes caractéristiques) peuvent faire l'objet d'une approximation par rapport aux fréquences relatives des classes et caractéristiques dans l'ensemble des données d'entraînement. Il s'agit d'une estimation du maximum de vraisemblance des probabilités. Les probabilités a priori des classes peuvent par exemple être calculées en se basant sur l'hypothèse que les classes sont équiprobables (i.e chaque antérieure = 1 / (nombre de classes)), ou bien en estimant chaque probabilité de classe sur la base de l'ensemble des données d'entraînement (i.e antérieure de C= (nombre d'échantillons de C) / (nombre d'échantillons total) [13].

Pour estimer les paramètres d'une loi de probabilité relative à une caractéristique précise, il est nécessaire de présupposer le type de la loi en question ; sinon, il faut générer des modèles non-paramétriques pour les caractéristiques appartenant à l'ensemble de données d'entraînement. Lorsque l'on travaille avec des caractéristiques qui sont des variables aléatoires continues, on suppose généralement que les lois de probabilités correspondantes sont des lois normales, dont on estimera l'espérance et la variance [13].

L'espérance, μ, se calcule avec : $$\mu = \frac{1}{N}\sum_{i=1}^{N} xi \quad (3.8)$$

Où N est le nombre d'échantillons et xi est la valeur d'un échantillon donné. La variance, σ^2, se calcule avec : $$\sigma^2 = \frac{1}{(N-1)} \sum_{i=1}^{N} (xi - \mu)^2 \quad (3.9)$$

Si, pour une certaine classe, une certaine caractéristique ne prend jamais une valeur donnée dans l'ensemble de données d'entraînement, alors l'estimation de probabilité basée sur la fréquence aura pour valeur zéro. Cela pose un problème puisque l'on aboutit à l'apparition

d'un facteur nul lorsque les probabilités sont multipliées. Par conséquent, on corrige les estimations de probabilités avec des probabilités fixées à l'avance [13].

3.2.3.Construire un classificateur à partir du modèle de probabilités

Jusqu'à présent nous avons établi le modèle à caractéristiques indépendantes, à savoir le modèle de probabilités bayésien naïf. Le classificateur bayésien naïf couple ce modèle avec une règle de décision.

Une règle couramment employée consiste à choisir l'hypothèse la plus probable. Il s'agit de la règle du maximum a posteriori ou MAP. Le classificateur correspondant à cette règle est la fonction classificatrice suivante [13] :

$$Classificateur\ (F_1, \ldots\ldots, F_n) = argmax\ p(C = c) \quad (3.10)$$

$$\prod_{i=1}^{n} (\boldsymbol{Fi = fi \backslash C = c}) \quad (3.11)$$

3.2.4.Analyse

Fait étonnant, malgré les hypothèses d'indépendance relativement simplistes, le classificateur bayésienne naïf a plusieurs propriétés qui le rendent très pratique dans les cas réels. En particulier, la dissociation des lois de probabilités conditionnelles de classe entre les différentes caractéristiques aboutit au fait que chaque loi de probabilité peut être estimée indépendamment en tant que loi de probabilité à une dimension. Cela permet d'éviter nombre de problèmes venant du fléau de la dimension, par exemple le besoin de disposer d'ensembles de données d'entraînement dont la quantité augmente exponentiellement avec le nombre de caractéristiques.

Comme tous les classificateurs probabilistes utilisant la règle de décision du maximum a posteriori, il classifie correctement du moment que la classe adéquate est plus probable que toutes les autres.

Par conséquent les probabilités de classe n'ont pas à être estimées de façons très précises. Le classificateur dans l'ensemble est suffisamment robuste pour ne pas tenir compte de sérieux défauts dans son modèle de base de probabilités naïves. La documentation citée en fin d'article détaille d'autre raisons pour le succès empirique des classificateurs bayésiens naïfs.

3.2.5.Avantage

Cet algorithme dont le modèle d'apprentissage est très général est utilisé dans de nombreux autres domaines que le texte. Il y'a un ensemble d'avantages du classificateur bayésien naïf, parmi lesquelles :

- Algorithme facile et simple à implémenter.
- Basé sur une théorie mathématique précise.
- Efficacité et rapidité dans l'apprentissage et la classification.
- Facile à mettre à jour avec de nouveaux exemples d'apprentissage.
- Equivalent à un classificateur linéaire, dans sa rapidité d'application.
- L'hypothèse d'indépendance des paramètres assouplit l'algorithme pour qu'il soit :
  - ✓ favorable pour différents types de données
  - ✓ Très efficace avec des petits corpus d'apprentissage
  - ✓ Résiste au bruit existant dans les données d'entrée
  - ✓ Utile pour la classification déterministe comme pour le Ranking puisque il ordonne les classes par degré d'appartenance pour un texte donné
  - ✓ Requiert une petite quantité de données d'apprentissage pour estimer les paramètres

Enfin, le plus important c'est que les méthodes Naïve Bayes donnent de bons résultats [9].

En revanche, l'inconvénient principal à notre avis, c'est bien l'hypothèse d'indépendance entre les descripteurs qui est loin d'être réaliste, mais nous pensons qu'elle n'est pas un handicap majeur dans un contexte de classification.

Tous les avantages cités auparavant et particulièrement la simplicité des calculs, l'efficacité des résultats et la facilité de l'implémentation de cette méthode, au contraire à d'autres techniques plus sophistiquées gourmandes en ressources et en temps d'exécution avec des taux d'amélioration des résultats très minimes, ont stimulé et justifié le choix du modèle d'indépendance conditionnelle (Naïve Bayes classifier) pour nos travaux.

3.3.PROPOSITION

La figure 3.1 présente le schéma de notre proposition qui consiste en des étages appliquée pour chaque partie (Training Data, Test Data).

La seule différence entre le corpus Training Data et Test Data est les classes connues dans Traning data au contraire pour Test Data.

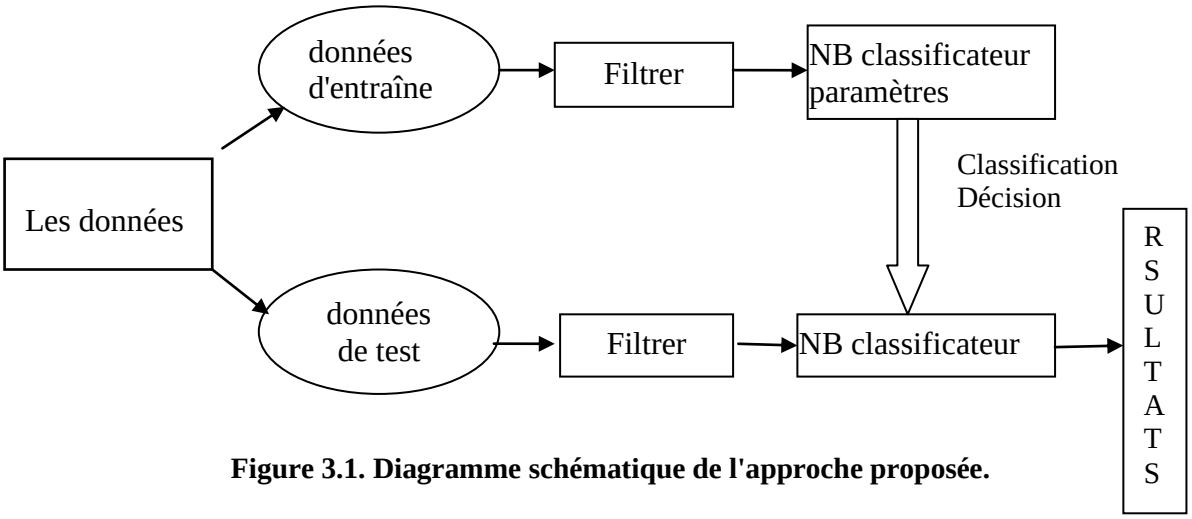


Figure 3.1. Diagramme schématique de l'approche proposée.

3.4.OUTILS DE DEVELOPPEMENT:

Le choix de l'environnement de programmation convenable est très important pour le développement des projets. Cela se fait suivant plusieurs facteurs : la puissance de compilation, la facilité d'utilisation, la disponibilité de plusieurs fonctionnalités, la communication avec d'autre environnements… etc.

L'outil que nous avons adopté est JAVA sous l'environnement NetBeans, notre choix c'est porté sur cet outil car la plateforme WEKA a été développée en java, ainsi que le nombre phénoménal des composants et classes mis- en portée des utilisateurs.

Pour implémenter notre système , nous avons utilisé les outils suivants :

3.4.1.Le langage de programmation (JAVA)

Notre choix pour le langage de programmation s'est porté sur le langage JAVA, et cela parce qu'il est un langage orienté objet simple ce qui réduit les risques d'incohérence et il possède une riche bibliothèque de classes comprenant des fonctions diverses telles que les fonctions standards, le système de gestion de fichiers ainsi que beaucoup de fonctionnalités qui peuvent être utilisées pour développer des applications diverses. Il existe une multitude de bibliothèques développées et fournies pour être utilisées en JAVA. Les API (Application Programming Interface) des autres langages autres que JAVA ne sont pas finalisées et doivent encore être mises à jour.

3.4.2.Environnement de développement

L'environnement de développement utilisé, est NetBeans 8.1 car il possède de nombreux points forts qui sont à l'origine de son énorme succès dont les principaux sont :

- Un environnement de développement intégré (EDI).

- En plus de JAVA, NetBeans3 permet également de supporter différents autres langages, comme Python, C, C++, JavaScript, XML, Ruby, PHP et HTML. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web).
- Les principaux modules de base pour NetBeans concernent le langage de programmation JAVA. Les modules agissent sur des fichiers qui sont inclus dans l'espace de travail (appelé workspace). Ce dernier regroupe les projets qui contiennent une ou plusieurs arborescences de fichiers.
- La construction incrémentale des projets JAVA grâce à son propre compilateur, qui permet en plus de compiler le code même avec des erreurs, de générer des messages d'erreurs personnalisés, de sélectionner la cible, …

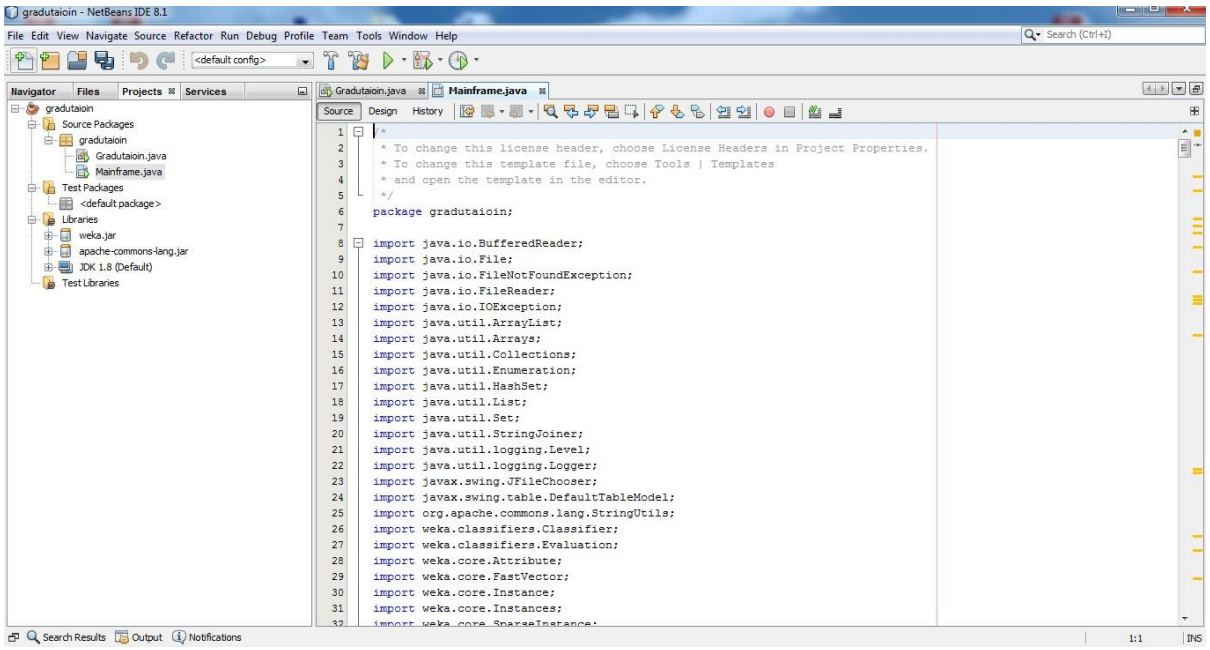


Figure 3.2 : L'interface graphique de NetBeans IDE 8.1

3.4.3.Composants de NetBeans

- Éditeur de Code avec Coloration Syntaxique.
- Support des langages Java, C, C++, XML et HTML.
- Support des technologies JSP, XML, RMI, CORBA, JINI, JDBC et Servlet.
- Support d'Ant, CVS et d'autres Systèmes de Contrôle de Version.
- Support des services de compilation, débogage et d'exécution.

- Outils de conception visuelle qui permet de créer et de manipuler graphiquement des composants visuels. Très pratique pour faire des fenêtres simples et rapidement.
- Assistants et outils de gestion et de génération de code.

3.5.PRESENTATION DE LA PLATEFORME WEKA

WEKA (Waikato Environment for Knowledge Analysis) est un outil de fouille de données (licence GNU) développé en Java. Il a été créé à l'université de Waikato, en Nouvelle- Zélande, par un groupe de chercheurs issus de l'apprentissage automatique, de la reconnaissance de formes et de la fouille de données.

WEKA permet de prétraiter des données (onglet Preprocess dans l'interface graphique), faire de la classification supervisée (Classify) et non-supervisée (Cluster), des régressions (Select Attributes), rechercher des règles d'association (Associate), et de visualiser différentes représentations graphiques des données (Visualize).

C'est un logiciel « open source » gratuit dédié à la fouille de données. Il s'adresse à deux types de publics. D'un côté, il présente une interface graphique, le rendant ainsi accessible à une utilisation de type « chargé d'études » sur des données réelles. De l'autre, du fait que le code source est librement disponible et l'architecture interne très simplifiée, il se prête à une utilisation de chercheurs qui veulent avant tout expérimenter de nouvelles techniques en améliorant celles déjà implémentées ou en introduisant de nouvelles.

3.5.1.Structure de données

WEKA traite des données contenues dans des fichiers respectant le format ARFF Attribute- Relation File Format, le format CSV Comma-Separated Values. Il s'agit de fichiers de type texte, décrivant des ensembles de tuples caractérisés par un certain nombre d'attributs communs.

3.5.2.Caractéristiques principales

- 49 outils de prétraitement de données.
- 76 algorithmes de classification/régression.
- 8 algorithmes de clustering.
- 15 évaluateurs d'attributs et plus de 10 algorithmes de recherche pour la sélection d'attribut.

- 3 algorithmes de recherche de règles d'association.
- 3 interfaces graphiques GUI.
- « Explorer» (explorateur d'analyse de données).
- « Expérimenter » (environnement expérimental).
- «KnowledgeFlow » (le nouveau modèle de processus avec interface).

3.6.EXPERIMENTATION

3.6.1. Présentation du corpus d'expérimentation et l'approches utilisés

Un corpus est un ensemble de documents (textes, images, …) pouvant provenir d'une ou de plusieurs disciplines, regroupés afin d'être soumis à des traitements. Nous avons utilisé pour la première étape d' expérimentations un corpus de textes écrits en langue anglaise.

Notre corpus contient 130 textes répartis en 2 catégories (0, 1). Et 70 textes pour le test.

textes / classes	Classe '0'	Classe '1'
Nombre de textes	160	40

Tableau 3.1. Description des données.

Pour faire la classification par des méthodes (naïves bayes) et mesurer les performances de classificateur (rappel, précision,…..etc.), ce dernier a besoins d'un fichier de format spécial qui s'appelle ARFF, ARFF définit des attributs par les utilisateurs, ces attributs peuvent être nominaux, numériques, strings, dates et relationnels.

La représentation externe d'une instance de classe se compose de :

- Un en-tête: Décrit les types d'attributs.
- Section de données: Liste CSV des données séparé La figure ci_dessous représente un exemple de fichier .ARFF :

Figure3.3 Fichier .arff d'apprentissage et de test utilisé dans les expérimentations.

3.6.2. Prétraitements effectués sur les corpus : d'apprentissage, de test :

Toutes les approches utilisent une représentation « sacs de mots » comme méthode de représentation, issue du modèle vectoriel. Etant donné le grand nombre de descripteurs potentiels, il est, en général, nécessaire d'effectuer une sélection de descripteurs avant de pouvoir utiliser un modèle d'apprentissage.

Cette technique originale de sélection de descripteurs présente plusieurs avantages. Elle est Entièrement automatique et ne nécessite pas de ressources externes (comme une liste de mots les plus fréquents dans une langue donnée) et elle est couplée avec un critère d'arrêt pour trouver le "bon" nombre de descripteurs.

Pour sélectionner le fichier .ARFF qui contient les données Cliquer sur le bouton «Add» :

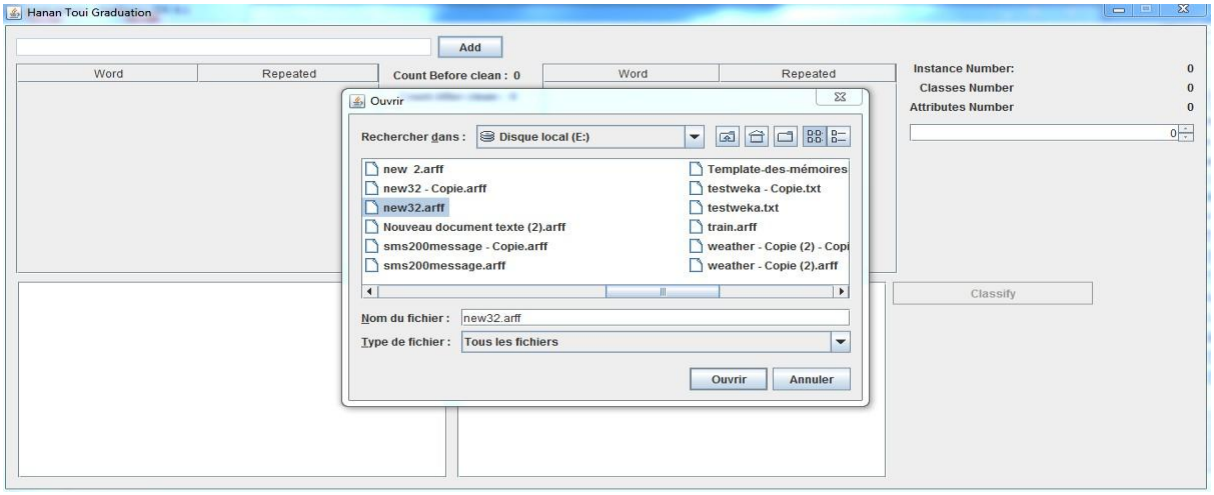


Figure 3.4.sélection le Fichier.arff.

Après la sélection du fichier, nous obtenons la fenêtre sous dessus qui affiché toutes les descripteurs avec leur occurrence qui compose des informations du texte de le fichier .ARFF:

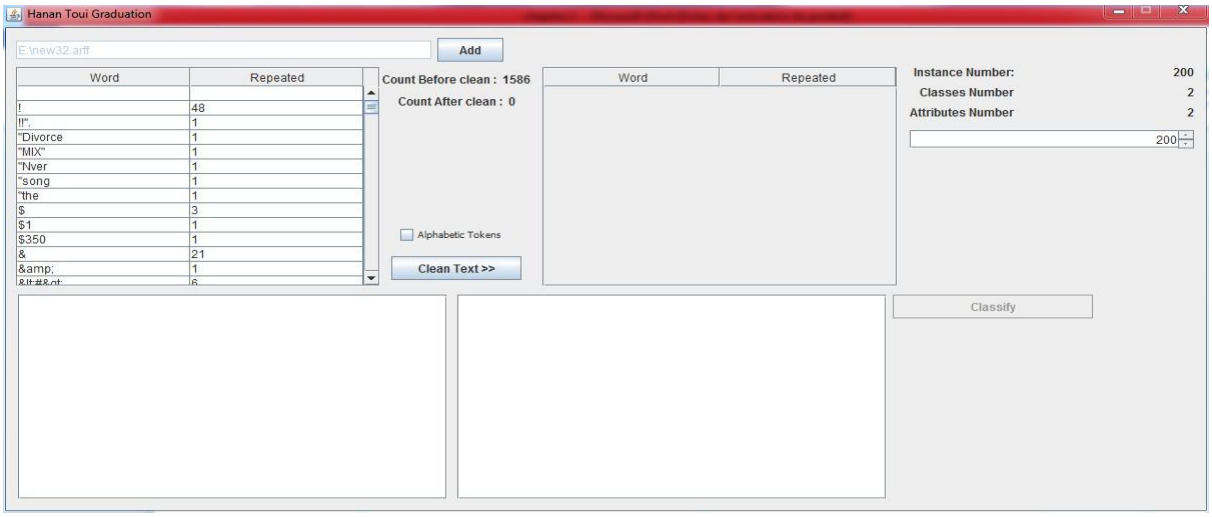


Figure 3.5. Représentation vectoriel de Fichier.arff.

On peut appliquer quelques filtres pour la représentation des documents :

Cliquer sur le bouton « Clean Text »pour Nettoyage « le sac de mot » de mots qui n'utilise pas par l'algorithme bayésien naïf.

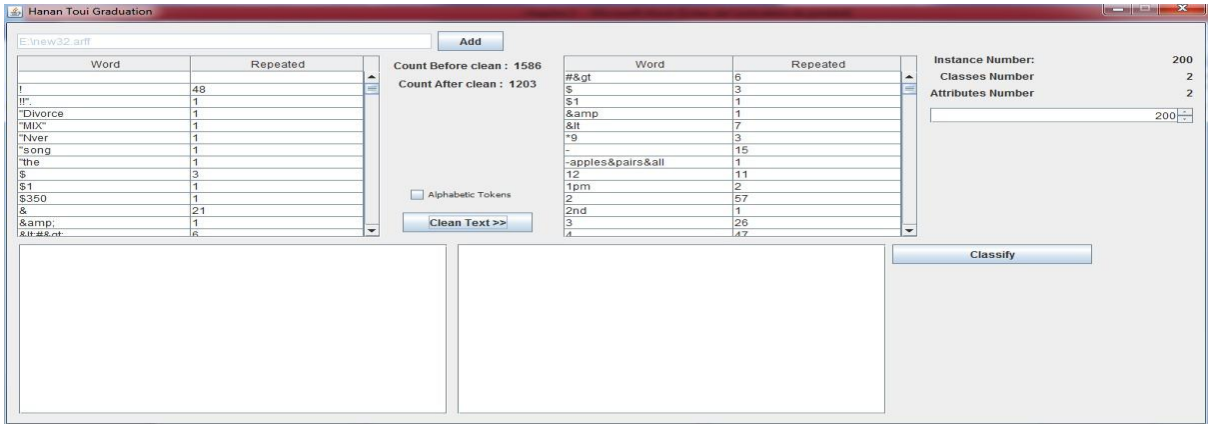


Figure 3.6. Nettoyage le sac de mot.

Cocher la case « Alphabétique ToKens »pour Elimination des signes de ponctuation :

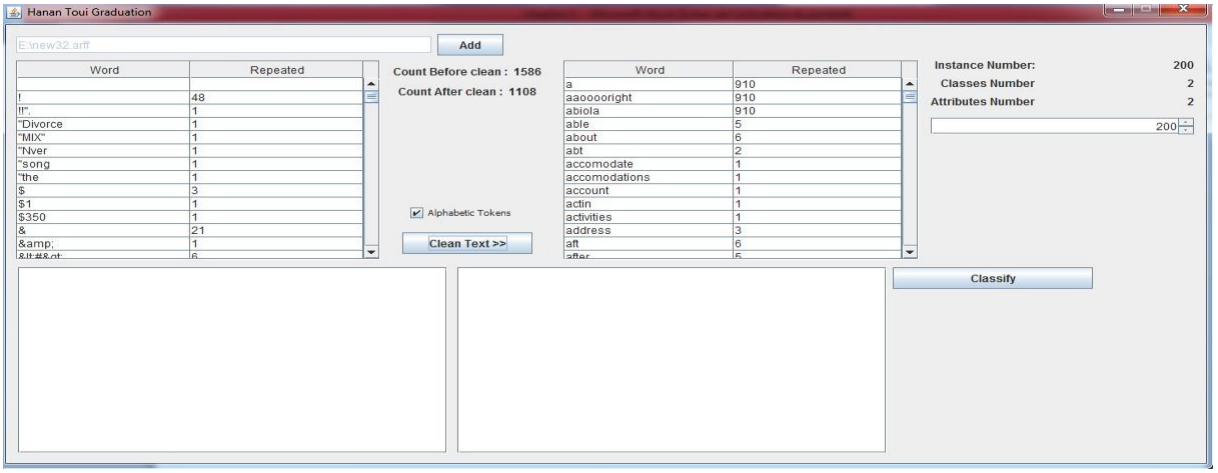


Figure 3.7. Elimination des signes de ponctuation.

REMAREQUE:

Les performances de classification de fichier (Resultat_Texte.arff) par WEKA et mieux sur la classification de fichier filtrée (Resultat_Noms.arff) qui contient seulement les noms.

3.6.3. Application du classificateur bayésien naïf effectué sur les corpus d'apprentissage, de test

Pour mesurer les performances d'un classificateur on peut diviser le corpus d'apprentissage en deux parties :

- Corpus d'entrainement.
- Corpus de test.

Le corpus d'entrainement ayant 130 document et le corpus de test ayant 70 documents, la figure suivante illustre comment faire cette étape:

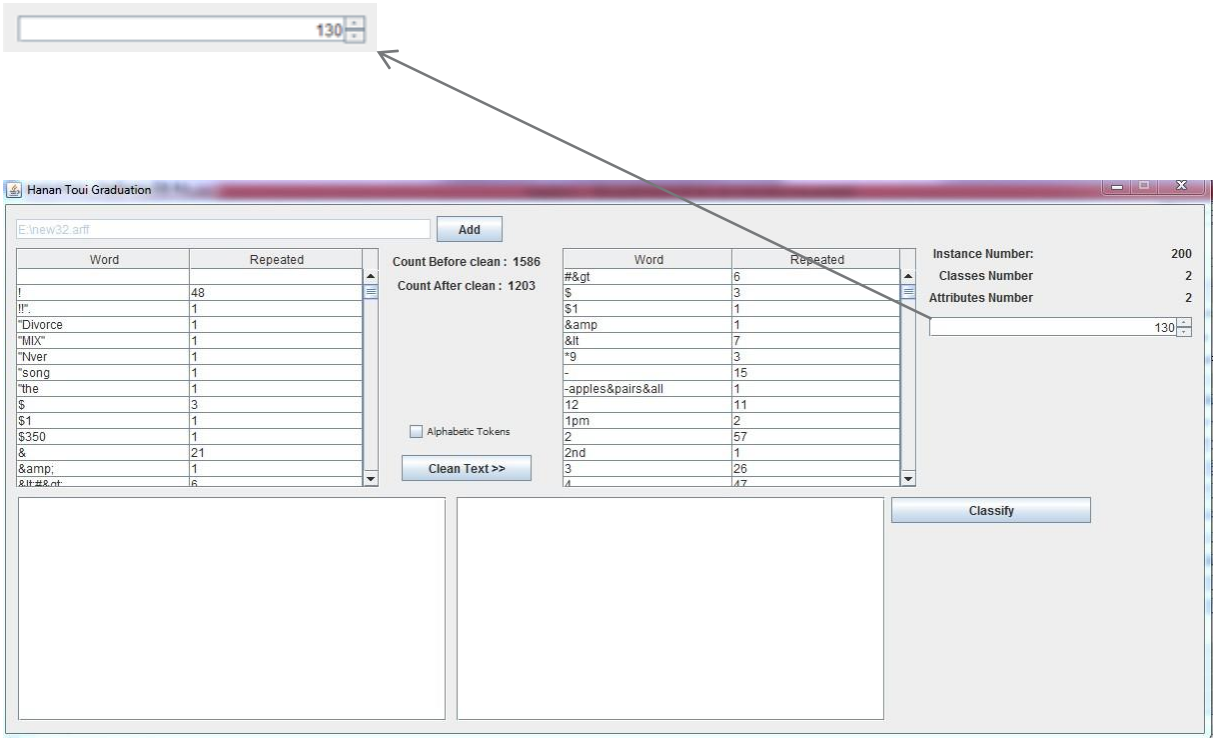


Figure 3.8.Division de corpus d'apprentissage.

Selon la méthode de classification bayésien naïf ,on appuie sur le bouton «classify», pour lancer l'exécution et prend les résultats :

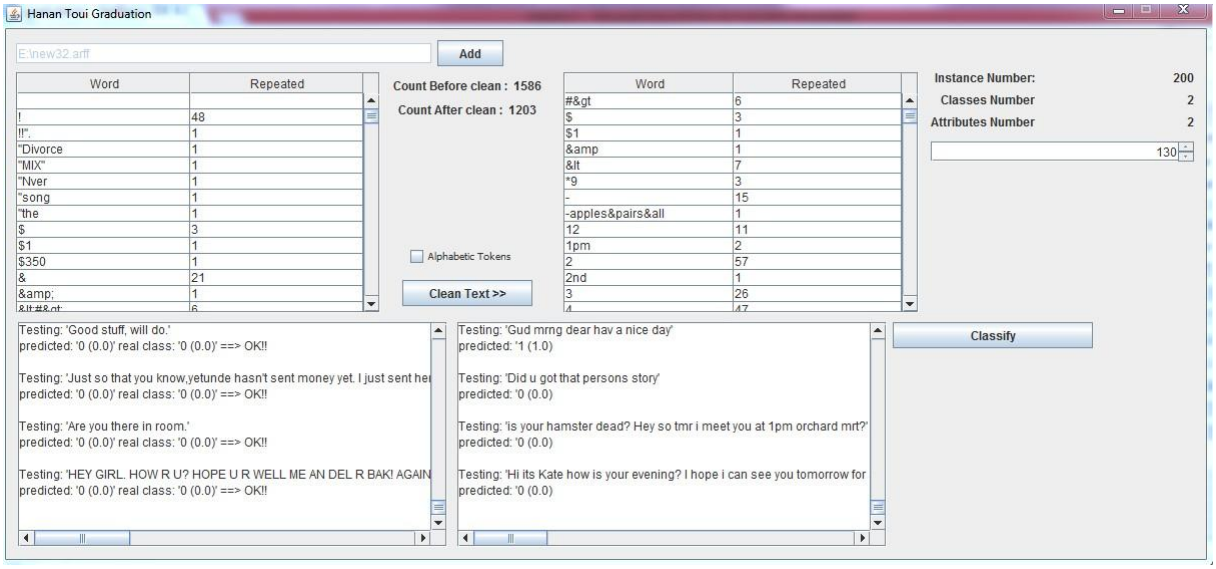
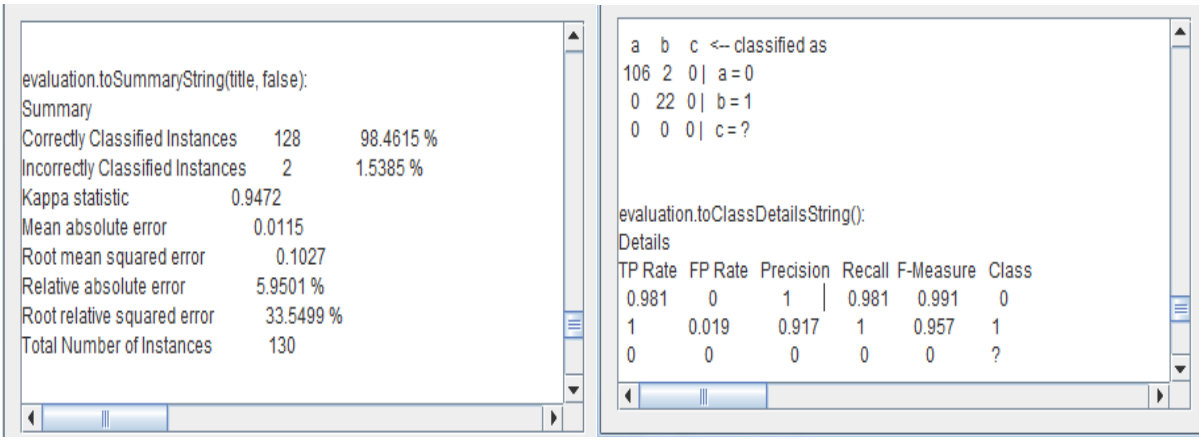


Figure 3.9.Application du classificateur bayésien naïf.



```
evaluation.toSummaryString(title, false):
Summary
Correctly Classified Instances      128          98.4615 %
Incorrectly Classified Instances      2           1.5385 %
Kappa statistic                  0.9472
Mean absolute error              0.0115
Root mean squared error          0.1027
Relative absolute error          5.9501 %
Root relative squared error     33.5499 %
Total Number of Instances          130
```

```
 a   b   c  <-- classified as
106  2   0 |  a = 0
  0  22   0 |  b = 1
  0   0   0 |  c = ?


evaluation.toClassDetailsString():
Details
TP Rate  FP Rate  Precision  Recall  F-Measure  Class
 0.981      0          1      0.981    0.991      0
 1        0.019      0.917    1        0.957      1
 0          0          0      0        0          ?
```

Figure 3.10. Résultat des mesures de corpus d'apprentissage.

```
Time taken to build model: 0.11 seconds

=== Evaluation on test split ===
=== Summary ===

Correctly Classified Instances          62               88.5714 %
Incorrectly Classified Instances         8               11.4286 %
Kappa statistic                          0.6028
Mean absolute error                      0.1077
Root mean squared error                  0.3204
Relative absolute error                 36.331  %
Root relative squared error             76.9734 %
Total Number of Instances               70

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.982     0.467      0.885     0.982     0.931      0.936    0
                 0.533     0.018      0.889     0.533     0.667      0.936    1
Weighted Avg.    0.886     0.371      0.886     0.886     0.874      0.936

=== Confusion Matrix ===

  a  b   <-- classified as
 54  1 |  a = 0
  7  8 |  b = 1
```

Figure 3.11. Résultat des mesures de corpus de test.

Le tableau 3.1 présente les résultats de classification par la méthode de naïve Bayes qui est appliqueé sur un corpus contenu 130 textes d'apprentissage et 70 textes pour le test :

Data set \ Algorithmes		NB	J48
Training Data	Exactitude	98.4615%	85.3846%
	Erreur	1.5385%	4.6154%
Test Data	Exactitude	88.5714%	84.2857%
	Erreur	11.4286%	15.7143%

Tableau 3.2. Comparaison les résultats.

3.7.DISCUSSION

Dans la section précédente, nous avons évalué l'approches implémentées pour les comparer afin d'avoir une meilleure catégorisation des textes. Le tableau 3.1, montre respectivement les résultats de la première approche et les résultats de la deuxième approche. Cette interprétation est confirmée par le taux d'erreur pour l'algorithme Bayes donne

« Taux_Erreur », car sa valeur est moindre, par contre la valeur de Taux_Correct qui est plus élevée dans tous les cas, c.à.d. que l'algorithme NB catégorise les documents avec moins d'erreur.

3.8.CONCLUSION:

Ce chapitre présente la description et la mise en œuvre des étapes implémentées pour notre approche. Après la génération de fichiers ARFF et 'utilisation de WEKA pour faire la classification et présenté les résultats obtenus, à la fin on peut dire que la méthode de Bayes donner de très bons résultats.

Conclusion générale et Perspectives

L'objectif général de ce mémoire est d'implémenter un système de classification automatique de texte pour les documents textes. Ce système comporte deux étapes principales (le prétraitement et la catégorisation).

Malheureusement, le temps attribué à ce travail était très court, d'où il était difficile de fixer certains paramètres pour étudier d'autres algorithmes. Nous proposons comme perspectives :

- Appliquer d'autres approches de représentation des textes, à savoir : l'approche conceptuelle et l'approche des n-grammes.
- Utiliser d'autres modèles de NB pour améliorer les résultats.

Références Bibliographique

Bibliographie

[1] AMEL TERKIA DERDRA, FATIMA ZAHRA BENSFIA, «La Représentation Conceptuelle pour la Catégorisation des Textes Multilingue», Mémoire de mastère, Université Abou Bakr Belkaid– Tlemcen, Algérie, Septembre 2012.

[2] B. LOUNNAS, «Discovery and extraction of motifs and/or profiles in biological sequences, These Doctorat, Université Mohamed.

[3] C.GROUIN, B.ARNULPHY, J.BERTHELIN, S.El AYARI, A.GARCIA-FERMANDEZ, A.GRAPPY, M.HURAULT-PLANTET, P.PAROUBEK, I.ROBBA et P.ZWEIGENBAUMw, « Présentation de l'édition 2009 du Défi Fouille de Textes » (DEFT'09), LIMSI–CNRSbBP 133– F- 91403 Orsay Cedex.

[4] CARUANA, R. and NICULESCU-MIZIL, A.: "An empirical comparison of supervised learning algorithms".Proceedings of the 23rd international conference on Machine learning, 2006.

[5] F. SADOUNE, « Identification de la langue et categorisation thematique de textes d'un corpus multilingue en utilisant des arbres de decision», Mémoire de Master, Université Mohamed BOUDIAF– Msila, Algérie, 2012-2013.

[6] FABRIZIO SEBASTIANI,« Machine learning in automated texte catégorisation », Conseil recherché National, Italie, Mars 2002.

[7] G.OBOZINSKI, « Introduction aux modèles graphiques », cours, Décembre 2010-2011.

[8] HARRAG FOUZI, «Une approche de fouille des textes basée sur la classification et la segmentation thématique : Application au corpus des Traditions Prophétiques "Hadith"», mémoire de doctorat d'informatique, Université Ferhat ABBAS, Sétif, 2011.

[9] H.MATALLAH, «Classificatiion Automatiique de Textes Approche Oriientée Agent», Mémoire de Magister En Informatique, Université Aboubekr Belkaid-Tlemcen, 2011.

[10] HARRY ZHANG,"The Optimality of Naive Bayes". Conférence FLAIRS 2004.

[11] KARIMA ABIDI, « La catégorisation de texte Multilingue », Mémoire de magistère, Ecole supérieur d'Informatique, Algérie, 2010-2011.

[12] M. ZEGGANE, « Algorithmes d'apprentissage pour la classification de documents », Mémoire de licence, Université de Mostaganem- Algérie, 2009.

[13] O.CHOAYB, «Classification automatique de textes », Mémoire de Master, Université Mohamed BOUDIAF– Msila, Algérie, 2013-2014.

[14] R.JALAM, « Apprentissage automatique et catégorisation de textes multilingues », Thèse de doctorat, Université Lumière Lyon 2, France, Juin 2003.

[15] SIMON RÉHEL, « Catégorisation automatique de textes et Cooccurrence de mots provenant de documents non étiquetés », Mémoire, Université Laval Québec, Canada, Janvier 2005.

[16] S.BOUYACOUB, S.KAOUADJI, «Enrichissement de la représentation conceptuelle dans la catégorisation du texte en utilisant les mesures de similarité sémantique», Mémoire de mastère, Université Abou Bakr Belkaid– Tlemcen, Algérie, 2012-2013.

[17] S.BAÂLI , «Conception et mise en place d'un (stemmer) pour la langue arabe dans le cadre de la catégorisation automatique de documents», Mémoire de Master, Université Mohamed BOUDIAF– Msila, Algérie, 2013-2014.

[18] S. SAHRAOUI, «Identification de la langue et catégorisation thématique de textes d'un corpus multilingue en utilisant les réseaux de neurones artificiels RNA », Mémoire de Master, Université Mohamed BOUDIAF– Msila, Algérie, 2012-2013.

[19] S.RAHEEL, « L'Apprentissage Artificiel pour la Fouille de Données Multilingues: Application à la Classification Automatique des Documents Arabes », Thèse de doctorat en Sciences de l'Information et de la Communication, Université Lumière Lyon 2, 2010.

[20] S.BESSOU, « Analyse de Données Textuelles pour la Classification Automatique par les Techniques de Text Mining, application à la Langue Arabe », Mémoire de Magister En Informatique, Université de Sétif, 2007.

[21] S.DELLALI, « Exploitation des Ontologies pour la Classification des Textes Arabes», Mémoire de Master, Université Mohamed BOUDIAF– Msila, Algérie, 2016-2017.

[22] S. ABDELOUHAB, «Processus de classification supervisée de textes arabes par la méthode K PPV Application aux articles de presse», Mémoire de Master, Université Mohamed BOUDIAF– Msila, Algérie, 2011-2012.

[23] S.JAILLET, M.TEISSEIRE, J.CHAUCHE, V.PRINCE, « Classification automatique de documents, Le coefficient des deux écarts », Université Montpellier2-France, 2005.

[24] T.DERDRA AMEL, F.BENSFIA, « La Représentation Conceptuelle pour la Catégorisation des Textes Multilingue », Mémoire de Master, Université Abou Bakr Belkaid– Tlemcen, 2011-2012.

[25] YASMINE HANANE, ZEGGANE MOKHTAR, « Algorithmes d"apprentissage pour la classification de documents», université de mostaganéme- algérie, licence 2009.

Webographie

[26] Aubay, www.aubay.com, consulte le : 12/03/2018.

[27] Journaldunet, www.journaldunet.com, consulte le :12/04/2018.

[28] Piloter , www.piloter.com, consulte le :10/04/2018.

[29] Wikipedia, www.wikipedia.com, consulte le : 11 /04/2018.

[30] Wikipedia, www.wikipedia.com, consulté le : 12/04/2018.

[31] Wixsite,https://kadrisaid28.wixsite.com/sgadri,consulté le:15/04/2018.