REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

N° d'ordre : ……

UNIVERSITE DE M'SILA

FACULTE DES MATHEMATIQUES ET DE L'INFORMATIQUE

Département d'Informatique

MEMOIRE de fin d'étude

Présenté pour l'obtention du diplôme de MASTER

Domaine : Mathématiques et Informatique

Filière : Informatique

Spécialité : Systèmes d'Informations Avancés

Par :OUALI Choayb

SUJET

Classification automatique de textes

Soutenu publiquement le :2014devant le jury composé de :

……	**Université de M'sila Président**
Mr. BRAHIMI Belkacem	**Université de M'sila Rapporteur**
……	**Université de M'sila Examinateur**
……	**Université de M'sila Examinateur**

Promotion : 2013 /2014

Dédicaces

A mes parents

Je vous dois ce que je suis aujourd'hui grâce à votre amour, à votre patience et vos innombrables sacrifices. Que ce modeste travail, soit pour vous une petite compensation et reconnaissance envers ce que vous avez fait d'incroyable pour moi.

Que Dieu, le tout puissant, vous préserve et vous procure santé et longue vie afin que je puisse à mon tour vous combler.

A mes très chers frères

Aucune dédicace ne serait exprimer assez profondément ce que je ressens envers vous, je vous dirais tout simplement, un grand merci, je vous aime.

A mes très chers amis

En témoignage de l'amitié sincère qui nous a liées et des bons moments passés ensemble. Je vous dédie ce travail en vous souhaitant un avenir radieux et plein de bonnes promesses.

OUALI Choayb

Remerciements

En tout premier lieu, je remercie Allah le tout puissant, à la sagesse et au savoir infinis, « Gloire à toi ! Nous n'avons de savoir que ce que Tu nous as appris. Certes c'est Toi l'Omniscient, le sage, le tout miséricordieux le très miséricordieux » (Sourate al-Baqarah, verset 32).

Je tiens à remercier mon encadreur M[r] BRAHIMI Belkacem pour le grand honneur qu'il m'a fait en me proposant le sujet de ce mémoire de fin d'étude. J'ai eu l'honneur et le privilège de travailler sous son assistance et de profiter de ses qualités humaines, professionnelles et de sa grande expérience, il m'a guidé tout au long de ce travail. L'élaboration avec amabilité et dynamisme le caractérisant. Que ce modeste travail puisse satisfaire mes examinateurs, pour qu'ils en témoignent ma gratitude et reconnaissance pour l'aide et les conseils qu'il ma prodigué, ainsi que pour la savoir qu'il ma inculqué.

Je remercie tous mes enseignants de l'université de M'sila.

Mes remerciements vont également aux membres de jury d'avoir accepté de juger mon travail.

Je remercie vivement toute ma famille, en particulier mes parents, pour m'avoir toujours soutenu au cours de mes études. Qu'ils trouvent ici le fruit de leur patience et du soutien permanent qu'ils m'ont prodigué pour affronter tous les moments difficiles.

Je tiens également à remercier mes collègues AMOUR Nasreddine et HADJ HFSI kamal pour son aide et ses conseils précieux.

Merci pour tous ceux qui, m'ont aidé de près ou de loin à réaliser ce travail.

Table des Matières

Chapitre 3 – Classification de texte

Chapitre 4 – Conception

Table des Figures

Liste des abréviations

TAL : *Traitement automatique des langues.*

RI : *Recherche d'information.*

EI : *L'extraction de l'information.*

CAH : *Classification Ascendante Hiérarchique.*

CT : *Classification auTomatique.*

RD : *Recherche documentaire.*

POS Tagger : *Part of Speech Tagger.*

ONU : *Organisation des Nations Unies.*

TF : *Term Frequency.*

IDF : *Inverse Document Frequency.*

TF*IDF : *Term Frequency Inverse Document Frequency.*

SVM : *Machines à support de vecteurs.*

RNA : *Réseaux de neurone artificiel.*

NB : *Naïve Bayes.*

KNN : *K-nearest neighbors.*

WEKA : *Waikato Environment for Knowledge Analysis.*

ARFF : *Attribute-Relation File Format.*

GUI : *Grafical User Interface.*

Csv : *Comma-Separated Values.*

Introduction générale

Introduction générale

De nos jours, les besoins de catégorisation automatique de documents en raison de l'augmentation constante du volume d'informations accessibles électroniquement, la conception et la mise en œuvre d'outils efficaces, permettant notamment à l'utilisateur de n'avoir accès qu'à l'information qu'il juge pertinente, devient une nécessité absolue. Comme la plupart de ces outils sont destinés à être utilisés dans un cadre professionnel, les exigences de fiabilité et de convivialité sont très importantes ; les problèmes à résoudre pour satisfaire ces exigences sont nombreux et difficiles.

Le but de nos travaux est de développer un modèle fondé sur l'apprentissage automatique pour la catégorisation de textes avec le filtrage des noms en utilisant la méthode de naïve bayésienne, donc on peut distinguer deux grandes parties :

- ✓ La catégorisation de textes et textes contiennent les noms.
- ✓ La catégorisation thématique avec le naïve bayésienne.

La structure proposée du mémoire peut être présentée comme suit :

Dans **le premier chapitre** nous introduisons des notions générales sur les domaines de : Data Mining, Text Mining en donnant quelques définitions, les taches principales, les applications de chacun et surtout la relation entre l'apprentissage automatique et le Text Mining.

Le deuxième chapitre vise à présenter le processus de la catégorisation des textes et le prétraitement des textes, ainsi que les difficultés liées à cette catégorisation.

Le troisième chapitre est dédié à la présentation des différents algorithmes d'apprentissage automatique supervisée ainsi que leurs avantages et leurs inconvénients. Nous avons également introduit les différents moyens d'évaluation d'un classificateur.

Le quatrième chapitre en mettant l'accent sur l'algorithme utilisé dans notre travail : le naïve bayésienne.

Et le dernier chapitre permettra d'évaluer les performances des différentes approches implémentées en présentant les résultats obtenus avec interprétation.

Et nous avons terminé par une conclusion qui nous voir ultérieurement.

Chapitre 01 :

DATA MINING ET TEXT MINING

1.1. Data Mining

1.1.1. Introduction

Aujourd'hui, des milliards de données sont collectées chaque jour dans le monde. En effet, les faibles coûts des machines en termes de stockage et de puissance ont encouragé les sociétés à accumuler toujours plus d'informations. Cependant, bien que la quantité de données à traiter ne cesse d'augmenter les spécialistes dans le domaine estiment que la quantité de données collectées dans le monde double tous les 20 mois. Les entreprises étaient jusqu'alors incapables de transformer leurs données en connaissance directement utilisable.

Dans cette optique, un ensemble d'architectures, de démarches et d'outils a été regroupé en une forme homogène sous le terme de fouille de données ou Data Mining.

1.1.2. Définition de Data Mining

La fouille de données ou le Data Mining consiste essentiellement à extraire de l'information d'immenses bases de données de la façon la plus automatique possible.

Plus concrètement, le Data Mining est un processus de traitement informatique d'une très grande quantité de données afin de trouver des informations pertinentes, contrairement à la méthode statistique qui nécessite que l'on établisse une hypothèse de départ qu'il s'agira de vérifier. C'est, des données elles-mêmes, que se dégageront les corrélations intéressantes.

Le Data Mining se situe à la croisée des statistiques, de l'intelligence artificielle et des bases de données **[1]**.

1.1.3. Les taches de Data Mining

Le choix des techniques du Data Mining à appliquer dépend de la tâche particulière à accomplir et des données disponibles pour l'analyse. La première étape consiste à traduire un objectif commercial en une ou plusieurs tâches.

Nous citons ci-dessous les tâches de base:

- **La classification** consiste à examiner des caractéristiques d'un objet afin de l'affecter à une classe d'un ensemble prédéfini;
- **L'estimation** est souvent utilisée pour effectuer une tâche de classification;
- **La prédiction** ressemble à la classification et à l'estimation, mais les enregistrements sont classés selon un certain comportement futur prédit ou à une valeur future estimée;

- **Le regroupement par similitudes** consiste à déterminer les objets qui vont naturellement ensemble;
- **La description** de qualité suffisante suggérera souvent une explication.

Une fois les tâches identifiées, elles sont utilisées pour restreindre la gamme des méthodes prises en compte. En termes généraux, notre but est de sélectionner la technique de Data Mining qui minimise le nombre et la difficulté des transformations de données qui doivent être effectuées pour produire de bons résultats. Les données brutes peuvent demander différentes manières d'être résumées, les valeurs manquantes doivent être traitées, etc. Ces transformations sont nécessairement indépendantes de la technique choisie **[1]**.

1.1.4. Les méthodes de Data Mining

Les outils de Data Mining utilisent les mêmes aspects théoriques que les techniques statistiques traditionnelles. En utilisant la puissance de ces méthodes et en introduisant le concept d'intelligence artificielle et celui de l'apprentissage automatique, le Data Mining constitue un outil très puissant dans le domaine de l'extraction des données au sein de l'entreprise. Cette combinaison technique facilite la résolution, la compréhension, la modélisation et l'anticipation des problèmes.

Le schéma suivant positionne les différentes techniques du Data Mining **[2]**.

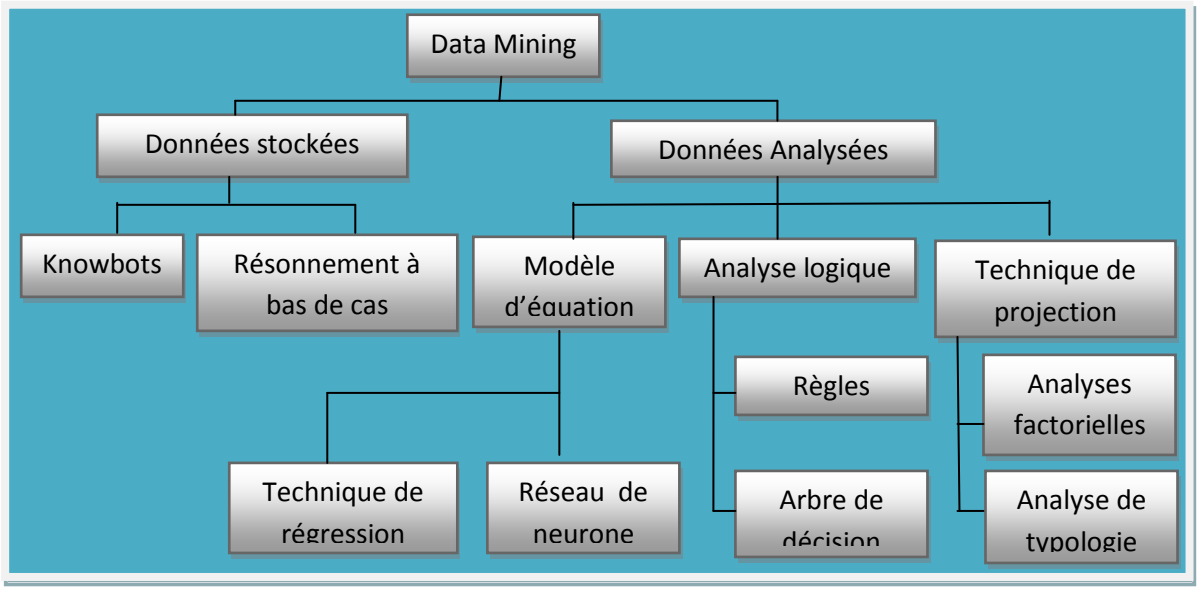

Figure 1.1 Les méthodes de Data Mining

1.1.5. Domaines d'utilisation de Data Mining

Les domaines d'application du Data Mining sont vastes et variés. Cependant un trait commun les lies est le fait qu'ils traitent un volume important de données et tire des informations qui visent à améliorer la qualité du produit ou du service.

Parmi les domaines ou l'utilisation du Data Mining est devenue monnaie courante:

- **Laboratoires pharmaceutiques et médicaux :** La médecine représente un grand chantier pour les outils du Data Mining, notamment dans la détection des tumeurs et des maladies rares à partir des prélèvements d'autres maladies qui présentent des symptômes similaires dans le but d'identifier les meilleures thérapies.
- **Assurances :** Le Data Mining est un outil très puissant pour l'analyse des sinistres et la recherche des critères explicatifs du risque ou de la fraude mais aussi il fournit des modèles de sélection et de tarification.
- **Banques et les grandes administrations :** Le Data Mining fournit des modèles de prédiction de fraudes ainsi que des modèles de pré autorisation de crédit automatique à partir des informations stockées dans leurs bases de données.
- **Automobiles et grandes industries :** Afin d'améliorer la qualité du produit, on utilise le Data Mining pour anticiper les défauts de production. Par ailleurs, les méthodes du Data Mining servent pour la prévention des ventes.
- **Les transports à grandes échelles :** On utilise le Data Mining pour l'optimisation des tournées et dans le Marketing afin de définir des programmes et des promotions selon les classes de clients et leurs caractéristiques.
- **Grande distribution et vente en correspondance :** L'étude dans ce cas se base sur l'analyse des similarités des consommateurs en fonction de critères géographiques et sociodémographiques et l'analyse des comportements des individus **[2]**.

1.2. Text Mining

1.2.1. Introduction

Les textes expriment un grand nombre d'informations de natures diverses mais la manière dont cette information est représentée rend difficile l'analyse automatique. L'information n'est donc pas structurée (texte libre). Cette absence de structure n'autorise pas un accès direct aux informations. Le volume de données est très important rendant impossible toute analyse par un humain.

1.2.2. Définition de text mining :

Le Text Mining, également appelé fouille de textes ou extraction de à partir de textes, est un ensemble de méthodes, de techniques et d'outils pour exploiter les documents non structurés que sont les textes écrits, comme les fichiers bureautiques de type word, les emails, les documents de présentation de type PowerPoint...etc. Pour extraire du sens de documents non structurés, le Text Mining s'appuie sur des techniques d'analyse linguistique. Le Text Mining est utilisé pour classer des documents, réaliser des résumés de synthèse automatique ou encore pour assister la veille stratégique ou technologique selon des pistes de recherches prédéfinies **[3]**.

Schématiquement, on peut énoncer :

TEXT MINING = LINGUISTIQUE + DATA MINING

1.2.3. Les tâches de Text Mining :

Le Text Mining n'est pas un remplacement pour la recherche d'information ou le traitement du langage naturel. Les techniques qui permettent d'organiser un corpus de documents textuels selon leur contenu ont un spectre d'utilisation très large. Le Text Mining cherche des réponses aux questions difficiles ou impossibles à résoudre avec les seuls moteurs de recherche. Des exemples de tels services incluent :

- Résumer des documents qui décrivent une consommation du produit dans certaines régions ;
- Etudier des réclamations des clients, raisons des changements de comportements de consommation, analyse de l'image de l'entreprise, …
- Faire la gestion de la relation client : orienter mes mails clients reçus sur le site vers les services adéquats et les aider à répondre le plus rapidement et correctement possible ;
- Connaître les réseaux relationnels des personnes ou entreprises.

Chacune de ces tâches sera un cas particulier du schéma général de la figure ci-dessous **[4]**:

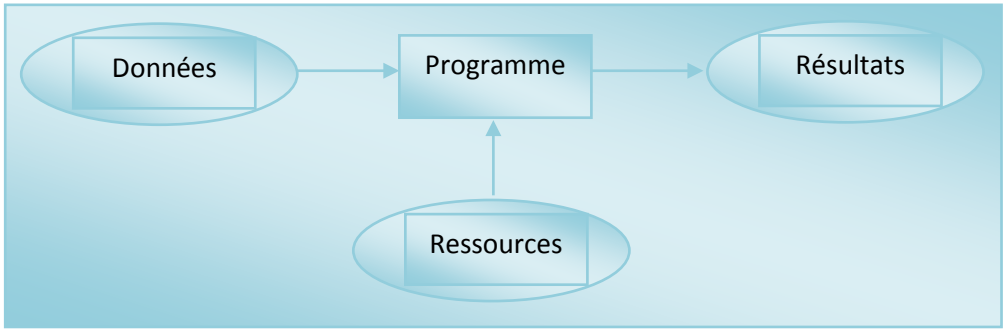

Figure 1.2 Schéma général d'une tache du Text Mining

1.2.4. Processus de Text Mining :

Les étapes nécessaires pour effectuer le processus de text mining sont :

- **L'acquisition** : Source de données telle que : corpus textuels, bibliothèques électroniques, Web…etc ;
- **Le filtrage** : Sélection des mots les plus pertinents (techniques de sélection d'attribut) ;
- **Le nettoyage des données** : Segmentation du texte, élimination des mots vides, lemmatisation;
- **L'identification des mots pertinents** : Analyse statistique (n-gram), analyse sémantique, analyse syntaxique ou structurelle (extraction d'attribut) ;
- **L'extraction des connaissances** : Application de l'un des algorithmes de la fouille de textes **[4]**.

1.2.5. Applications du Text Mining

L'importance de la fouille de textes (Text Mining) ne cesse d'évoluer d'un jour à l'autre. Plusieurs domaines vitaux exploitent les techniques et les outils du Text Mining pour trouver l'information pertinente fouillée dans des quantités énormes de textes de différentes formes, parmi ces domaines on peut citer **[3]**:

- La recherche d'information.
- Les applications biomédicales.
- Le filtrage des communications.
- Les applications de sécurité.
- La gestion des connaissances.
- L'Analyse du sentiment.

1.2.6. Techniques liées à la fouille de textes :

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

1.2.6.1. 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.

1.2.6.2. 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.

1.2.6.3. 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 morpho_syntaxique pour reconnaître les constituants du texte (phrases, mots, verbes, adjectifs), leur nature pour détecter les phrases pertinentes et en extraire les informations voulues **[4]**.

1.2.7. 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 **[5]**.

1.2.8. Méthodes utilisées 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).

1.2.9. Conclusion

La fouille de données est une discipline a pour but de valoriser les bases de données. Elle offre des perspectives nouvelles pour la statistique et répond au défi du traitement des giga bases de données. Les données textuelles en format « libre » disponibles sur supports informatiques représentent environ 70% des données disponibles. La préparation des données est une étape importante, si ce n'est primordial, du processus d'extraction de connaissances à partir de données. En schématisant, il s'agit de définir au mieux les individus et la représentation utilisée pour l'apprentissage.

Chapitre 02 :

Prétraitement et représentation du texte

2.1 Introduction :

Dans ce chapitre nous allons exposer la classification automatique de texte, plus en détail la catégorisation de textes. Nous présentons quelques définitions sur la classification et les différents jeux de mots utilisés : classification, catégorisation ou clustering, ensuite les différents objectifs et intérêts de la classification ainsi que les confluents avec d'autres disciplines comme la Recherche d'Informations, puis nous décrivons le processus général de la catégorisation de textes avec toutes ces étapes, et enfin les problèmes spécifiques aux textes lors de l'apprentissage automatique.

2.2 Définition de la classification

La classification automatique de documents est un problème connu en informatique, il s'agit d'assigner un document a une plusieurs catégories ou classes. Le problème est différent selon la nature des documents en question, en effet la classification de textes diffère de la classification de documents images, vidéo ou encore son. On peut aussi imaginer des classifications selon des paramètres associés aux documents tels que par exemple l'auteur, la date de parution… Dans le cadre de ce projet et dans la suite de rapport nous nous baserons sur la classification de documents de type texte selon leur contenu. Toute référence ultérieure à la classification renvoie donc à cette notion.

La classification de textes est une tâche générique qui consiste à regrouper de manière automatisée des documents qui se ressemblent suivant certains critères à savoir les critères observables tels que le type du document, l'année, la discipline, l'édition, etc... ou le critère du contenu, et à assigner une ou plusieurs catégories, parmi une liste prédéfinie, ou non à un document.

La classification de textes est définie comme une opération qui identifie des classes d'équivalence entre des segments de textes en tenant compte de leur contenu informationnel (mots, n-gram, etc.).

2.3. Définition formelle

Formellement, la catégorisation de texte consiste à associer une valeur booléenne à chaque paire $(d_j, c_i) \in D \times C$, où D est l'ensemble des textes et C'est l'ensemble des catégories selon

que $d_j \in c_i$, ou non. Le but de la catégorisation de texte est de construire une procédure (modèle, classifieur) $\Phi : D \times C \rightarrow \mathbb{B}$ qui associe une ou plusieurs étiquettes (catégories) à un document d_j avec la fonction F: D → C, la vraie fonction qui retourne pour chaque vecteur d_j une valeur c_i **[10]**.

2.4. Automatisation de 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 Méga-octets à plusieurs Giga-octets. [7]

Le nombre de textes à classifier étant énorme, il serait très difficile de pouvoir déterminer de combien de temps a besoin un expert 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.

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'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».

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)

- 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
- 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

2.5. Les méthodes de classification automatique

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.

2.5.2. Apprentissage non supervisé (Clustering)

L'apprentissage non supervisé consiste à apprendre à classer sans supervision. Au début de processus nous ne disposons ni de la définition des classes, ni de leurs nombres. C'est l'algorithme de classification qui va déterminer ces informations. Nous ne disposons pas non plus de données en entrée qui sont déjà classées, c'est aussi à l'algorithme de découvrir par lui-même la structure plus ou moins cachée des données et de former des groupes d'individus dont les caractéristiques sont communes. [11]

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.

Les techniques pour réaliser de tels regroupements constituent un domaine d'étude très riche, qui a donné lieu à de multiples propositions dont le recensement n'est pas l'objet de ce document.

L'apprentissage non supervisé est utilisé dans plusieurs domaines tels que :

- Médecine : Découverte de classes de patients présentant des caractéristiques physiologiques communes.
- Le traitement de la parole : construction de système de reconnaissance de la voie humaine.
- Archéologie : regroupement des objets selon leurs époques.
- Traitement d'images
- Classification de documents.

Dans la littérature il existe plusieurs types d'algorithmes d'apprentissage non supervisé tels que les algorithmes de partitionnements et les algorithmes de classification hiérarchique :

· **Le partitionnement**: consiste au regroupement des données suivant leur degré de similarité. L'algorithme le plus célèbre appartenant à cette classe est K-means : c'est un algorithme qui permet de partitionner un ensemble de données automatiquement en K clusters. Il consiste tout d'abord à choisir k points qui représentent les centres des groupes à créer, puis à affecter les autres points aux centres les plus proches. Cette affectation est faite par le calcul de distance entre les points. Plusieurs distances peuvent être définies telles que la distance euclidienne ou la distance de Manhattan. Par la suite nous procédons à une étape de raffinement des groupes de façon itérative, le raffinement se fait par le recalcule des centres des groupes après chaque itération et par une réaffectation des points aux groupes. L'algorithme s'arrête quand aucun point ne bouge. [11]

· **La classification hiérarchique** : il existe deux types de classification hiérarchique : Ascendante et descendent. La classification ascendante consiste à utiliser une matrice de similarité afin de partir d'une répartition fine vers un groupe unique. Donc, il s'agit de fusionner les groupes jusqu'à ce qu'on obtient un seul groupe englobant tous les autres. Cette classification peut être représentée par un arbre hiérarchique ou dendrogramme. La classification descendante se présente comme l'inverse de la classification ascendante. Donc il s'agit de décomposer un cluster unique en sous-groupes jusqu'à l'obtention des singletons. [9]

2.5.3. Apprentissage supervisé (Catégorisation)

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. [9]

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.

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...).

2.5.4. Avantages et inconvénients

Parmi les avantages et inconvénients liés aux deux approches, on peut citer :

- ☐ Les groupes ou clusters obtenus par la technique supervisée est de meilleure qualité et plus précise que la technique non-supervisée.
- ☐ Dans la technique supervisée, on sait ce qui est attendu favorisant de meilleurs résultats par rapport au non supervisée.
- ☐ Un avantage des techniques non supervisées, est qu'elles accomplissent la tâche de similarité sans avoir besoin des données expertisées.
- ☐ Un inconvénient des approches supervisées, repose sur le fait qu'il peut être difficile de se procurer des données expertisées.
- ☐ L'inconvénient majeur des approches non supervisées qu'elle demande dans l'étape d'évaluation des résultats l'intervention d'un expert.

2.5.5. Classification Supervisée Vs Classification non Supervisée

La classification supervisée consiste à identifier la classe d'appartenance d'un objet à partir de certains traits descriptifs. Cette approche permet l'affectation automatique de documents dans des classes préexistantes.

L'objectif est de trouver une liaison fonctionnelle, que l'on appelle également modèle de prédiction, entre les textes à classer et l'ensemble des catégories. Pour estimer le modèle de prédiction, il faut disposer d'un ensemble de textes préalablement étiquetés, dit ensemble

d'apprentissage, à partir duquel on estime les paramètres du modèle de prédiction le plus performant possible, c'est-à-dire qui produit le moins d'erreurs en prédiction.

A la différence de la classification non supervisée où l'ordinateur doit découvrir lui-même des groupes de documents, la classification supervisée suppose qu'il existe déjà une classification de documents. C'est le cas par exemple d'une bibliothèque ou d'un moteur de recherche. Le but est alors de classer automatiquement un nouveau document. Il s'agit donc d'apprendre d'abord un modèle, ou classifieur, à partir d'un ensemble d'entraînement composé de couples (objet, classe).

Contrairement à la classification non supervisée, la classification supervisée peut mesurer l'importance de chaque mot pour classer de nouveaux documents. Par exemple, une mesure (gain d'information) calcule la typicité d'un terme. Plus un mot est lié à une catégorie et pas aux autres, et plus il est important : si un nouveau document le contient, ce mot sera très discriminant. De nombreuses mesures semblables ont été mises au point.

Enfin, à l'inverse de la classification non supervisée, il est ici simple d'évaluer les résultats d'une classification. Parmi les N exemples de documents classés, on utilise une partie des documents pour l'entraînement, et le reste pour le test. Pendant la phase de test, on soumet chaque document à l'algorithme de classification et on regarde simplement si la machine trouve la bonne classe. Bien sûr, le résultat de ce test n'est en rien garanti lorsque la machine aura à classer de nouveaux documents ! (réussir le test est nécessaire, sans être suffisant)**[9]**.

2.6 Classification de textes et Recherche d'informations

Dans la section suivante, nous allons rappeler les définitions de la recherche d'informations et la catégorisation de textes et essayer de positionner l'un par rapport à l'autre.

La recherche d'informations (RI), aussi appelée recherche documentaire (RD), est la problématique la plus ancienne de ce domaine, elle consiste à trouver, dans une importante base de documents, les documents pertinents correspondant à des requêtes qui peuvent être de différentes natures (liste de mots clefs, langage naturel, langage spécifique comme le SQL par exemple etc.).

La recherche d'informations est généralement effectuée en indexant préalablement tous les documents de la base selon les mots qu'ils contiennent ; la recherche consiste à trouver, le

plus rapidement possible, les documents ayant des mots communs avec la requête de l'utilisateur.

La catégorisation de textes, consiste à trouver dans un flux de documents, ceux qui sont relatifs à un sujet défini par avance. L'une des applications consiste à fournir à un utilisateur, en temps réel, toutes les informations importantes pour l'exercice de son métier. Dans ce cas, l'utilisateur n'exprime pas son intérêt par une requête, mais par un ensemble de documents pertinents. Cet ensemble de documents pertinents définit ce que l'on appelle, un thème ou une catégorie.

La recherche d'information se différencie de la classification ou la catégorisation par le très grand nombre de réponses possibles, qui peut être infini. L'application classique serait la réponse d'un moteur de recherche ou d'intelligence artificielle à une demande. La distinction entre ces deux disciplines peut être simplifiée de la manière suivante : dans le premier cas, la base de documents est fixe et l'interrogation est variable, alors que, dans le deuxième cas, la source de documents est variable et l'interrogation est fixe. Dans la pratique, la catégorisation de textes bénéficie de deux avantages par rapport à la recherche d'information : la stabilité dans le temps de la classe sélectionnée et la quantité réduite de documents à traiter dans le temps. La stabilité de la classe laisse le temps de construire des modèles performants permettant de rechercher la façon dont l'information est codée dans un texte. Le fait de traiter les textes un à un, au lieu de s'attaquer à une base importante de textes, est moins pénalisante pour un système moins performant, et rend possible l'utilisation de modèles plus complexes. [15]

2.7 Démarche à suivre pour la catégorisation de textes

Pour réaliser l'opération de catégorisation automatique de textes comme nous l'avons défini, la démarche commune est la suivante : la première phase consiste donc à formaliser les textes afin qu'ils soient compréhensibles par la machine et utilisables par les algorithmes d'apprentissage. La catégorisation des documents est la deuxième phase, cette étape est bien entendu décisive car c'est elle qui va permettre ou non aux techniques d'apprentissage de produire une bonne généralisation à partir des couples (Document, Classe).

Pour améliorer la performance des modèles, une évaluation de la qualité des classifieurs et la comparaison des résultats fournis par les différents modèles est effectuée en fin de cycle.

La démarche d'une approche standard de classification automatique de textes peut être résumée de la manière suivante :

- Eliminer les caractères de séparation, les signes de ponctuations, les mots vides, etc..;
- Les termes restants sont tous des attributs
- Un document devient un vecteur <terme, fréquence>
- Entrainer le modèle de classification à partir des couples (Document, Classe).
- Évaluer les résultats du classifieur. [17]

2.8. Le processus de la catégorisation des textes

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 d'après **[10]** est habituellement suivies :

- Prétraitement.
- Définition de descripteur.
- Sélection de descripteur.
- Pondération ou calcul de poids.
- La réduction de la taille du vocabulaire.
- Evaluation du modèle.

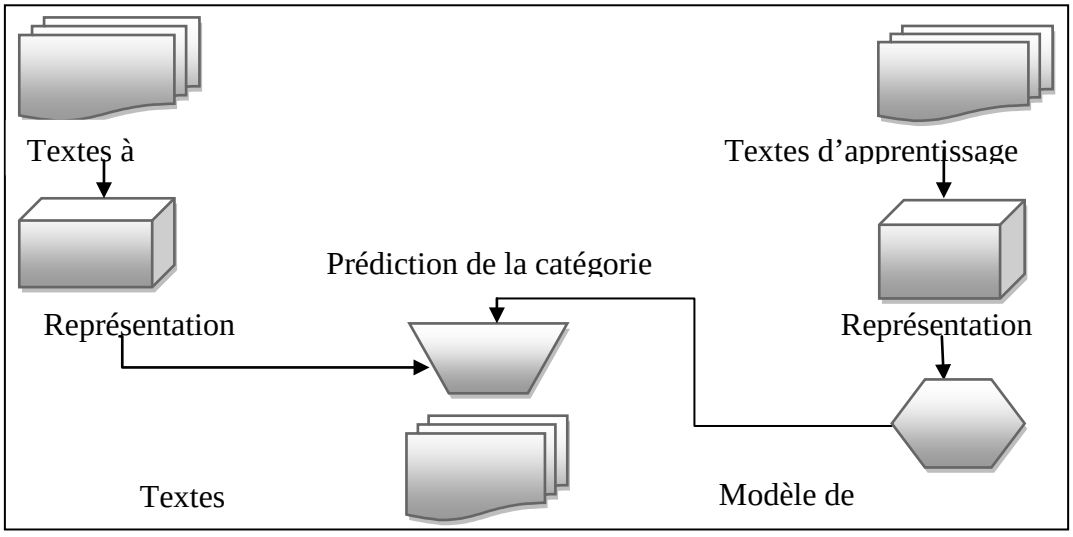

Figure 2.1 Processus de la catégorisation des textes

2.8.1 Prétraitements

Nous allons aborder ultérieurement les différentes méthodes de représentation des documents. Ces représentations sont toutes effectuées à base de mots qui sont eux-mêmes une

séquence de caractères. Il est donc nécessaire d'effectuer, au préalable du codage d'un document dans un espace de mots, une transformation permettant le passage de l'espace du caractère à un espace de mots.

Le prétraitement des textes est une phase capitale du processus de classification, puisque la connaissance imprécise de la population peut faire échouer l'opération. Après la première opération que doit effectuer un système de classification à savoir la reconnaissance des termes utilisés, nous devons expurger le plus possible les informations inutiles des documents afin que les connaissances gardées soient aussi pertinentes qu'il se peut. En effet dans les documents textuels de nombreux mots apportent peu (voir aucune) d'informations sur le document concerné. Les algorithmes dits de "Stop Words" s'occupent de les éliminer. Un autre traitement nommé "Stemming" permet également de simplifier les textes tout en augmentant leurs caractères informatifs comme d'autres méthodes qui proposent de supprimer des mots de faible importance.

Toutes ces transformations et méthodes font partie de ce qu'on appelle le prétraitement. Plusieurs d'entre elles sont spécifiques à la langue des documents (on ne fait pas le même type de prétraitement pour des documents écrits en anglais qu'en français ou encore en arabe). Le prétraitement est généralement effectué en six étapes séquentielles :

- La segmentation
- Suppression des mots fréquents
- Suppression des mots rares
- Le traitement morphologique
- Le traitement syntaxique
- Le traitement sémantique

2.8.1.1 La segmentation

La première opération que doit effectuer un système de classification est la reconnaissance des termes utilisés. La segmentation consiste à découper la séquence des caractères afin de regrouper les caractères formant un même mot. Habituellement, cette étape permet d'isoler les ponctuations (reconnaissance des fins de phrase ou de paragraphe), ensuite découper les séquences de caractères en fonction de la présence ou l'absence de caractères de séparation (de type « espace », « tabulation » ou «retour à la ligne »), puis regrouper les chiffres pour former des nombres (reconnaissance éventuelle des dates), de reconnaitre les mots composés. Eventuellement, nous pouvons unifier les écritures en lettre majuscules ou en lettres minuscules avant ou après les opérations déjà indiquées. C'est un traitement de surface assez

simple dans le principe, mais particulièrement difficile à réaliser de manière exacte sur les documents ayant beaucoup de bruits et des représentations assez variées. Notons que pour des corpus multilingues, une technique de segmentation moins intuitive a été proposée : la segmentation en n-grammes.

2.8.1.2 Suppression des mots fréquents ou élimination des Mots Outils

Les mots qui apparaissent le plus souvent dans un corpus sont généralement les mots grammaticaux, mots vides (empty words) ou mots outils (stop words) : les articles, les prépositions, les mots de liaisons, les déterminants, les adverbes, les adjectifs indéfinis, les conjonctions, les pronoms et les verbes auxiliaires etc.., qui constituent une grande part des mots d'un texte, mais malheureusement sont faiblement informatifs, sur le sens d'un texte puisqu'ils sont présents sur l'ensemble des textes.

A titre d'exemple on peut citer en dans la langue Française, le cas des articles « le », « la », «les » ou de certains mots de liaison « ainsi », « toutefois » etc..

Ces termes très fréquents peuvent être écartés du corpus pour en réduire la dimension. Cette possibilité de réduire la taille des entrées de l'index en éliminant les mots vides s'explique par le fait que ces termes sont présents dans la quasi-totalité des documents et ont donc un pouvoir discriminant faible en comparaison avec d'autres termes.

D'après la loi de Zipf (Voir Section 2.6.1.4). Leur élimination lors d'un pré-traitement du document permet par la suite de gagner beaucoup de temps lors de la modélisation et l'analyse du document.

Ces mots doivent être supprimés de la représentation des textes pour deux raisons :

- D'un point de vue linguistique, ces mots ne comportent que très peu d'informations. La présence ou l'absence de ces mots n'aident pas à deviner le sens d'un texte. Pour cette raison, ils sont communément appelés « mots vides ».
- d'un point de vue statistique, ces mots se retrouvent sur l'ensemble des textes sans aucune discrimination et ne sont d'aucune aide pour la classification. Une répartition des mots outils par rapport les mots utiles dans un corpus est représentée dans la figure 2.1.

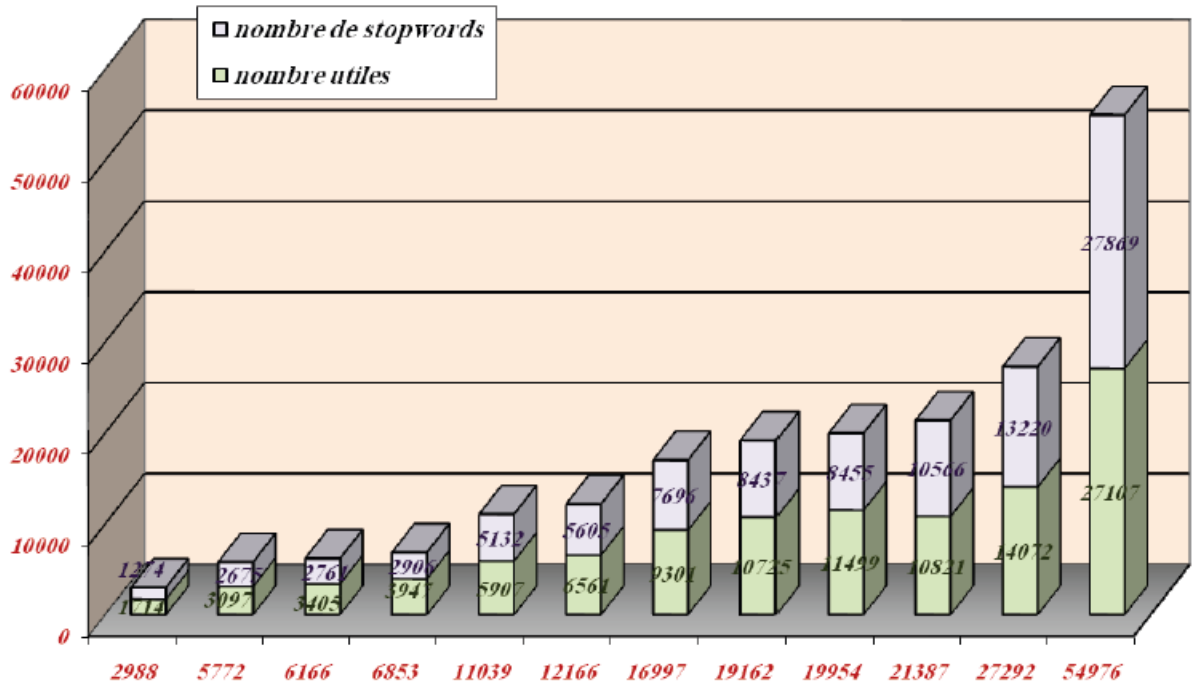

Figure 2.1 : Répartition des mots utiles et des mots vides dans un corpus

L'élimination systématique du corpus des mots vides peut se faire par l'intermédiaire d'une liste prédéfinie de mots pour chacune des langues étudiées.

Cependant, l'établissement d'une telle liste peut poser des problèmes. D'une part, il n'est pas facile de déterminer le nombre de mots exacts qu'il faut inclure dans cette liste. D'autre part, cette liste est intimement liée à la langue utilisée et n'est donc pas transposable directement à une autre langue.

Par exemple Sahami.M dans sa thèse de PHD (Sahami, 1999) définit une liste de 570 mots courant en anglais, plus une liste de 100 mots très fréquents sur le web. Comme on peut les écarter en fixant un seuil maximal de fréquence, pour ne pas sélectionner les mots présents dans une grande partie du corpus.

Une autre manière d'éliminer les mots vides d'un texte passe par l'utilisation d'un étiqueteur syntaxique (Part of Speech Tagger) – Les mots sont écartés en fonction de leur étiquette syntaxique sans avoir besoin de liste prédéfinie.

Enfin, un dernier point concernant les opérateurs de négation (ex : pas, ne, non) qui peuvent être supprimés sans gravité. Dans un contexte de classification de textes, une notion affectée par un opérateur de négation reste inchangée contrairement à une négation dans un contexte de recherche d'information qui peut être déterminante pour les résultats attendus. Dans le

cadre d'une recherche documentaire, le but à atteindre pour l'utilisateur de rechercher l'information en lien avec la requête. En revanche, dans le cade d'une classification de textes en plusieurs catégories, les opérateurs de négation ne vont guère influencer les résultats puisque l'on cherche à distinguer les thèmes les uns des autres. Par exemple les deux phrases suivantes : il est malade et il n'est pas malade traitent toutes les deux le même sujet de santé, et le terme malade, avec ou sans négation, est un terme décrivant cette notion de santé. En évidence, elles ont un sens opposé mais sont toutes liées au sujet de santé.

2.8.1.3 Suppression des mots rares

En général, les auteurs cherchent également à supprimer les mots rares, qui n'apparaissent qu'une ou deux fois sur un corpus, afin de réduire de façon appréciable la dimension des vecteurs utilisés pour représenter les textes, puisque, d'après la loi de Zipf , ces mots rares sont très nombreux.

D'un point de vue linguistique, la suppression de ces mots n'est pas nécessairement justifiée : certains mots peuvent être très rares, mais très informatifs. Néanmoins, ces mots ne peuvent pas être utilisés par des méthodes à bases d'apprentissage du fait de leur très faible fréquence ; il n'est pas possible de construire de statistiques fiables à partir d'une ou deux occurrences ; Une des méthodes communément retenues pour supprimer ces mots consiste à ne considérer que les mots dont la fréquence totale est supérieure à un seuil fixé préalablement.

Notons enfin, que les mots ne contenant qu'une seule lettre sont généralement écartés pour les mêmes raisons précédentes, comme par exemple le mot « D » dans la « Vitamine D » ou le mot « C » dans le « langage C ».

2.8.1.4 Le traitement morphologique

Consiste à effectuer un traitement au niveau de chacun des mots en fonction de leurs variations morphologiques : flexion, dérivation, composition afin de rassembler les mots de sens identiques. Donc, le but est de regrouper par exemple les termes «manger» et «mangent» ou les termes « cheval » et « chevaux » car ils ont la même signification. L'intérêt de cette opération est la réduction des dimensionnalités de l'espace de codage des textes afin d'améliorer davantage la performance du système de classification en matière d'espace mémoire et vitesse de traitement.

Plusieurs traitements morphologiques existent :

➢ ***Le stemming*** ou ***la désuffixation*** regroupe sous un même terme (stem) les mots qui ont la même racine. L'extraction des stems se fait par la technique de racinisation (ou stemming) qui utilise à la place des dictionnaires, des algorithmes simples basées sur des règles de remplacement de chaînes de caractères pour supprimer les suffixes les plus utilisés.[8]

Le stemming est un traitement linguistique moins approfondie que la lemmatisation, ayant deux avantages : Plu rapide que la lemmatisation (algorithmes simples ne faisant pas référence aux dictionnaires et règles de dérivation) et la possibilité de traiter les mots inconnus sans traitement spécifique.[16]

Néanmoins, sa précision et sa qualité sont naturellement inférieures, du fait qu'elle ne gère que les règles principales et ne peut pas prendre en compte les nombreuses exceptions des règles de dérivations. Par exemple, en français l'une des règles préconise de supprimer le « e » final de chaque mot, le mot *« fraise »* est alors transformé en « *frais »* ce qui suppose une relation entre les deux mots qui n'existe pas. Qui fait de cette opération dépendante de la langue, nécessitant une adaptation pour chaque langue utilisée.

➢ ***La lemmatisation*** conserve, non pas les mots eux-mêmes, mais leur racine ou lemme. Ce principe permet de prendre en compte les variations flexionnelles (singulier/pluriel, conjugaisons,…) ou dérivationnelles (substantifs, verbes, adjectifs,…) en regroupant sous le même terme tous les mots de la même famille et donc d'améliorer la classification. La lemmatisation est donc une tâche plus compliquée à mettre en oeuvre que la recherche de racines, puisqu'elle elle s'appuie sur des outils de TALN, ce qui nécessite beaucoup de ressources linguistiques (dictionnaires, règles de dérivation, etc..). De plus les résultats contiennent encore des erreurs à cause des problèmes de polysémie (ambiguïté) et d'incomplétude des dictionnaires.[14]

Un algorithme efficace, nommé TreeTagger (Schmid, 1994) a été développé pour les langues anglaise, française, allemande et italienne. Cet algorithme utilise des arbres de décision pour effectuer l'analyse grammaticale, puis des fichiers de paramètres spécifiques à chaque langue.

Toutes les études montrent que les performances des systèmes de classification, après lemmatisation, sont plus nettement supérieures à celles avant lemmatisation.

2.8.1.5 Le traitement syntaxique

La syntaxe traite les combinaisons et l'ordre des mots dans la phrase.

Le traitement syntaxique identifie et regroupe un ensemble de mots dont la sémantique dépend de leur association. Par exemple, les mots « casque bleu » ne signifient habituellement pas qu'on a affaire à un casque qui est bleu, mais plutôt à une organisation militaire dépendante de l'ONU. L'analyseur syntaxique a pour but d'identifier ce type de cas. La phase d'analyse syntaxique consiste aussi à éliminer des ambiguïtés comme par exemple les problèmes d'homographie.

2.8.1.6 Le traitement sémantique

Le traitement sémantique consiste à extraire la signification des expressions et traiter la polysémie à savoir les différents sens possibles d'un même mot. Par exemple, cette phase permet de différencier le mot « base » qui peut correspondre à une base militaire ou à une base de données. C'est une opération laborieuse, qui fait appelle aux ontologies, et qui n'est pas aujourd'hui bien maîtrisée et dont l'intérêt en terme de meilleures performances, dans les systèmes de classification, n'est pas toujours démontré.

- Notons, en fin de cette section, que les différents traitements appliqués sur un texte avant sa représentation informatique ne sont pas toujours nécessaire pour toutes les méthodes de représentation d'un texte, notamment le codage en n-grammes, qu'on va étaler par la suite, qui s'en passe d'une bonne partie de ces prétraitements en s'attaquant aux documents, pratiquement, dans leurs états bruts.

2.8.2 Présentation de textes

La définition ou l'extraction de caractéristiques au sein d'un texte est une phase décisive puisque la représentation déduite doit conserver au mieux l'information contenue dans le texte.

Ces caractéristiques constituent les éléments informationnels composant le document. Le plus petit élément informationnel étant le caractère, à un niveau supérieur on a le mot, regroupant un ensemble de caractères, puis à un niveau plus global nous pouvons définir les phrases, les paragraphes,…et pour finir le document lui-même.

La difficulté est donc le choix de cet élément de base : descripteur, terme ou caractéristique, puisque le processus de classification de textes en dépend directement.

Différentes méthodes sont proposées pour le choix des termes et les poids attribués à ces termes, des auteurs utilisent les mots comme descripteurs, d'autres utilisent les groupes de mots comme les mots composés, les expressions ou les collocations, comme d'autres qui préfèrent les techniques des n-grams, etc…

Dans la section suivante, nous allons définir les différentes sortes de termes, utilisés dans la littérature, pour la représentation d'un document texte.

2.8.2.1 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 à 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 de distance entre les mots, mais présente comme inconvénient la difficulté de délimiter les mots dans certaines langues telles que l'Arabe ou l'Allemand **[10]**.

2.8.2.2. Représentation par phrases

Un certain nombre de chercheurs proposent d'utiliser les phrases comme unité de représentation au lieu des mots comme le cas dans la représentation « sac de mot», puisque les phrases sont plus informatives que les mots seuls, par exemple « recherche d'information », « world wide web », ont un degré plus petit d'ambigüité que les mots constitutifs, et aussi que les phrases ont l'avantage de conserver l'information relative a la position du mot dans la phrase»**[3]**.

2.8.2.3. Représentation avec les racines lexicales

Cette méthode consiste à remplacer les mots du document par leurs racines lexicales, qui peut être réalisée en utilisant l'algorithme de Porter **[11]** 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. A titre d'exemple : les mots vol, volant, vole ont la même racine vol mais se rendent à trois notions différentes.

2.8.2.4. 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. Par exemple, les mots vol, volant et vole seront remplacés par leurs lemmes : vol, volant et voler selon le contexte. Cette représentation est simple mais elle peut causer une perte d'informations donnée par le contexte nécessaire à la distinction des lemmes polysémiques (possèdent plusieurs sens) et la présence de synonymes, considérés comme des lemmes différents même s'ils font référence au même concept **[12]**.

2.8.2.5. 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 **[10]**.

2.8.2.6. 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). Cette méthode a comme avantage selon **[13]** de réduire l'espace de travail car les mots qui sont synonymes partagent au moins un concept. Cependant, l'inconvénient majeur de cette représentation est qu'il n'existe pas des bases lexicales pour toutes les langues.

2.8.3 Pondération ou calcul de poids

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). L'objectif est de trouver les termes qui représentent le mieux le contenu d'un document. Les méthodes les plus populaires sont :

2.8.3.1. Le codage TFIDF [10]

- **TF (Term Frequency) :** La fréquence d'un terme est simplement le nombre d'occurrences de ce terme dans le document considéré ;
- **IDF (Inverse Document Frequency) :** La fréquence inverse de document est une mesure de l'importance du terme dans l'ensemble du corpus ;
- **TF*IDF(Term Frequency Inverse Document Frequency):**

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

$$TFIDF\ (T_i, D_j) = TF\ (T_i, D_j) * log\ (N/\ DF(T)) \qquad (1)$$

Avec :

- **TF (T_i, D_j)** : la fréquence du terme dans le document ;
- **N** : le nombre total de documents de la base documentaire ;
- **DF(T_i)** : le nombre de documents contenant le terme.

2.8.3.2. Le codage TFC

Le codage TF × IDF ne corrige pas la longueur des documents. Pour ce faire, le codage TFC est similaire à celui de TF×IDF mais il corrige les longueurs des textes par la normalisation en cosinus, pour ne pas favoriser les documents les plus longs **[12']**.

$$TFC\ (t_i, d_j) = \frac{TF*IDF(ti,dj)}{\sqrt{\sum_{s=0}^{|t|}\left(TF*IDF(ts,dj)\right)^2}} \qquad (2)$$

2.8.3.3. Le codage Lnu

Les différents textes qui composent un corpus ont des tailles différentes dont il faut tenir compte dans le codage des termes. Il existe deux phénomènes à considérer dans les textes longs par rapport aux textes courts**[13]** :

- Les mots présents tendent à avoir des fréquences plus élevées,
- Les textes longs sont plus susceptibles de contenir des mots-clefs différents.

$$Lnu = L * u \;;L = \frac{1+log(TF(m,t))}{1+log(\overline{TF}(m))} \;;u = \frac{1}{0.8 + 0.2\frac{U(t)}{\overline{U}}} \quad (3)$$

- TF_m : Fréquence moyenne dans le texte ***t***.
- ***U(t)*** : Nombre de termes uniques dans le texte ***t.***
- ***U***: Nombre moyen de termes sur l'ensemble des textes du corpus.

2.8.3.4. L'entropie

Une dernière approche de pondération significative s'appuie sur l'utilisation de l'entropie. Cette dernière mesure la dispersion d'un descripteur dans un corpus et peut s'avérer une information importante dans le cadre de la sélection de descripteur et/ou de pondération de la représentation fréquentielle d'un corpus.

$$E(t) = \sum_d \frac{P_{td} log_2 P_{td}}{log_2 N} \quad ; \quad P_{td} = \frac{TF_{td}}{GF_t} \quad (4)$$

Où GF_t représente le nombre total de fois où le descripteur i apparaît dans le corpus de ***N*** documents.

Une représentation avec l'approche fréquentielle ***(TF)*** peut alors être la suivante avec pour un terme ***t*** et un document ***d***:

$$w_{td} = (1+E(t)) \; log \; (TF_{td}+1) \quad (5)$$

2.8.4. Réduction de la taille du vocabulaire

Vu la taille impressionnante des bases textuelles, il est difficile de prendre l'ensemble de tous les mots comme étant des attributs, en effet cela engendre une perte de mémoire et de temps de calcul.

Plusieurs techniques de réduction existent pour réduire la dimension de vocabulaire qui se divise en deux grandes familles :

- **Sélection d'attributs**: qui conserve uniquement les mots utiles à la classification selon un critère fixé préalablement tandis que les autres sont rejetés.
- **Extraction d'attributs:** à partir des attributs de départ, elles créent de nouveaux attributs en faisant soit des regroupements ou des transformations **[10]**.

2.9 Applications de la classification

La classification automatique est une technique utilisée dans plusieurs domaines. Sa capacité prédictive la rend rapide et efficace. Parmi les applications où la classification est utilisée, nous trouvons le filtrage de spam, en effet il s'agit de traiter les messages électroniques textuels, identifier leurs caractéristiques et les classer en deux groupes messages désirés ou non désirés.

Une autre application est la détermination automatique du sujet d'un texte pour le classer automatiquement afin de notifier des personnes intéressées par ce sujet de la présence d'un nouveau texte. [11]

2.10 Quelques problèmes rencontrés dans la catégorisation de textes

Beaucoup difficultés peuvent s'opposer au processus de catégorisation de textes. Des problèmes connus dans la discipline liés à l'apprentissage automatique supervisé comme la subjectivité de la décision prise par les experts, le sur-apprentissage, etc.. mais aussi des problèmes particuliers liés à la nature des données traitées à savoir des données textuelles comme la polysémie, la redondance, Les variations morphologiques ou même L'homographie, etc..Nous allons signaler les huit principales Dans ce qui suit :

2.10.1. Sur-apprentissage

Le sur-apprentissage s'explique par le fait que le modèle de prédiction n'arrive pas à bien classer les nouveaux textes, pourtant il l'a bien fait dans la phase d'apprentissage en classant correctement les textes de la base d'apprentissage.

Pour limiter le sur-apprentissage, on doit sélectionner des termes pour réduire la dimensionnalité. D'après les expériences antérieures, le nombre de termes doit être limité par rapport au nombre de textes de la base d'apprentissage.

Quelques auteurs recommandent d'utiliser au moins 50 à100 fois plus de textes que de termes. En général le nombre de textes d'apprentissage est limité, c'est pour cela on cherche à agir sur le nombre des termes utilisés en les diminuant, pour éviter ce sur-apprentissage. Sans bien sûr pénaliser le système en supprimant des termes pertinents.**[16]**

2.10.2. L'homographie

Deux mots sont dits homographes si 'ils s'écrivent de la même façon sans forcément avoir la même prononciation. L'homographie est une sorte d'ambigüité supplémentaire. (Ex : avocat en tant que fruit et avocat en tant que juriste).

2.10.3. Polysémie (Ambiguïté)

Un mot possède, dans différents cas, plus d'un sens et plusieurs définitions lui sont associées. Par conséquent, à cause de la polysémie, les mots seuls sont parfois de mauvais descripteurs ; exemple le mot livre peut désigner une unité monétaire, ou un bouquin.

2.10.4. Les mots composés

Le non prise en charge des mots composés comme : comme Arc-en-ciel, peut-être, sauve-qui-peut, etc. Dont le nombre est très important dans toutes les langues, et traiter le mot Arcen-ciel par exemple en étant 3 termes séparés réduit considérablement la performance d'un système de classification néanmoins l'utilisation de la technique des n-grammes pour le codage des textes atténue considérablement ce problème des mots composés.

2.10.5. La graphie

Un terme peut comporter des fautes d'orthographe ou de frappe comme il peut s'écrire de plusieurs manières ou s'écrire avec une majuscule. Ce qui va peser sur la qualité des résultats. Parce que si un terme est orthographié de deux manières dans le même document (M'sila, m'sila), la simple recherche de ce terme avec une seule forme graphique néglige la présence du même terme sous d'autres graphies, ce qui va influencer les résultats puisque les différentes graphies vont être traitées séparément. Néanmoins du point de vue pratique, le fait qu'un terme inconnu est proche d'un autre terme prouve qu'il a été mal orthographié.

2.10.6. Redondance(Synonymie)

La redondance et la synonymie permettent d'exprimer le même concept par des expressions différentes, plusieurs façons d'exprimer la même chose.
Cette difficulté est liée à la nature des documents traités exprimés en langage naturel contrairement aux données numériques. LE FEVRE illustre cette difficulté dans l'exemple du chat et l'oiseau : mon chat mange un oiseau, mon gros matou croque un piaf et mon félin préféré dévore une petite bête à plumes.[16]

La même idée est représentée de trois manières différentes, différents termes sont utilisés d'une expression à une autre mais en fin compte c'est bien le malheureux oiseau qui est dévoré par ce chat. Lors d'une représentation vectorielle d'un document, ces termes sont représentés séparément, et les occurrences du concept sont dispersées. Il est alors important de rassembler ces terme sen un groupe sémantique commun.

2.10.7. Présence-Absence de termes

La présence d'un mot dans le texte indique un propos que l'auteur a voulu exprimer, on adonc une relation d'implication entre le mot et le concept associé, quoique on sait très bien qu'il y a plusieurs façons d'exprimer les mêmes choses, dés lors l'absence d'un mot n'implique pas obligatoirement que le concept qui lui est associé est absent du document.
Cette réflexion pointue nous amène à être attentifs quant à l'utilisation des techniques d'apprentissage se basant sur l'exclusion d'un mot particulier.

2.10.8. Subjectivité de la décision

Parmi les problèmes classiques usuels dans le domaine de l'apprentissage supervisé c'est la subjectivité de la décision prise par les experts qui décident de la classe à laquelle le texte va être attribué.

Certainement après la lecture du texte à classer, l'expert va trancher à quelle(s) catégorie(s) ce texte appartient en se basant sur le contenu sémantique et le contexte du texte et même en consultant d'autres textes préalablement associés à certaines classes, pour valider la décision prise qui ne peut être que subjective.

Les experts humains ne lisent pas de la même manière ! Ne réfléchissent pas de la même manière ! Donc ne classent pas de la même manière ! Ainsi un même document peut être classé différemment par deux experts, ou encore un même document peut être classé différemment par le même expert, soumis à deux instants différents.[16]

2.11. Conclusion

La catégorisation de textes s'est avérée au cours des dernières années comme un domaine majeur de recherche pour les entreprises comme pour les particuliers. Ce dynamisme est en partie dû à la demande importante des utilisateurs pour cette technologie. Elle devient de plus en plus indispensable dans de nombreuses situations où la quantité de documents textuels électroniques rend impossible tout traitement manuel. La catégorisation de textes a essentiellement progressé ces dix dernières années grâce à l'introduction des techniques héritées de l'apprentissage automatique qui ont amélioré très significativement les taux de bonne classification.

Nous avons tenté dans ce deuxième chapitre de définir la classification, précisément la catégorisation automatique de textes, ainsi que les notions nécessaires pour l'entame de la suite de ce mémoire.

Chapitre 03 :

Classification du texte

3.1 Introduction

La recherche accorde ces dernières années, beaucoup d'importance au traitement des données textuelles. Ceci pour plusieurs raisons : un nombre croissant de collections mises en réseau et distribuées au plan international, le développement de l'infrastructure de communication et de l'Internet. Les traitements manuels de ces données s'avèrent très coûteux en temps et en personnel, ils sont peu flexibles et leur généralisation à d'autres domaines est presque impossible ; c'est pour cela que l'on cherche à mettre au point des méthodes automatiques.

Le domaine de la fouille de textes (text mining) s'est développé pour répondre à volonté à la gestion par contenu des sources volumineuses de textes. A l'heure actuelle, de nombreux logiciels de classification de textes sont disponibles, ils ont fait l'objet de publications et leurs champs d'application s'élargit de jour en jour. En général, ces systèmes sont basés sur des algorithmes d'apprentissage automatique, Nous présentons donc des méthodes d'apprentissage qui, à partir de documents déjà classés, permettent de classer de nouveaux documents.[18]

3.2 Algorithmes d'apprentissage

En apprentissage automatique, différents types de classificateurs ont été mis au point, et cela dont le but d'atteindre un degré maximal de précision et d'efficacité, chacun ayant ses avantages et ses inconvénients. Mais, ils partagent toutefois des caractéristiques communes.

Parmi la panoplie de classificateurs existants, on peut faire des regroupements et distinguer des grandes familles.

Dans les pages qui suivent, nous allons exposer quelques algorithmes en détail, le classificateur bayésienne naïf algorithme qui nous avons utilisé dans notre étude, surpassé par d'autres mais il est souvent utilisé comme point de référence à cause de sa simplicité.

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

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

3.2.1 Algorithme des k-voisins les plus proches KNN

3.2.1.1 Définition

L'algorithme des k-voisins les plus proches («k-nearest neighbors» ou kNN) est une méthode d'apprentissage à base d'instances.

La méthode ne nécessite pas de phase d'apprentissage; c'est l'échantillon d'apprentissage, associé à une fonction de distance et à une fonction de choix de la classe en fonction des classes des voisins les plus proches, qui constitue le modèle.

Lorsqu'un nouveau document à classer arrive, il est comparé aux documents d'entraînement à l'aide d'une mesure de similarité. Ses *k* plus proches voisins sont alors considérés : on observe leur catégorie et celle qui revient le plus parmi les voisins est assignée au document à classer. C'est là une version de base de l'algorithme que l'on peut raffiner. Souvent, on pondère les voisins par la distance qui les sépare du nouveau texte.

3.2.1.2 principe de fonctionnement

L'algorithme de KNN 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.[13]

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 :

$$d\,(x, y)=\sqrt{\sum_{j=1}^{m}(x_j - y_j)^2}$$

La figure suivante illustre la fonctionnement de l'algorithme KNN :

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.

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 :

$$\textbf{\textit{Cosinus (a, b)}} = \frac{\sum(\boldsymbol{a}*\boldsymbol{b})}{\sqrt{\sum \boldsymbol{a^2} * \sum \boldsymbol{b^2}}}$$

- **Mesure de *Distance euclidienne*** La formule de la mesure de Distance est comme suivante :

$$\boldsymbol{D\ (a,\ b)} = \sqrt{\sum |\boldsymbol{a} - \boldsymbol{b}|^2}$$

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

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

3.2.1.3 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**[13]**.

3.2.1.4 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 **[13]**.

3.2.2. Les arbres de décision :

3.2.2.1. Définition :

Les arbres de décision sont plus populaires des méthodes d'apprentissage. Les Algorithmes connus sont ID3 (Quinlan 1986) et C4.5 (Quinlan 1993). Ils sont également populaires pour la classification de document.

Comme toute méthode d'apprentissage supervisée, les arbres de décision utilisent des exemples. Si l'on doit classer des documents dans des catégories, il faut construire un arbre de décision par catégorie. Pour déterminer à quelle(s) catégorie(s) appartient un nouveau document, on utilise l'arbre de décision de chaque catégorie auquel on soumet le document à classer. Chaque arbre répond Oui ou Non (il prend une décision).

Concrètement, chaque nœud d'un arbre de décision contient un test (un IF...THEN) et les feuilles ont les valeurs Oui ou Non. Chaque test regarde la valeur d'un attribut de chaque exemple. En effet, on suppose qu'un exemple est un ensemble d'attributs/valeurs. Pour des documents, chaque attribut peut être un mot et la valeur sera par exemple 0 ou 1 selon que ce mot appartient ou non au document.[2]

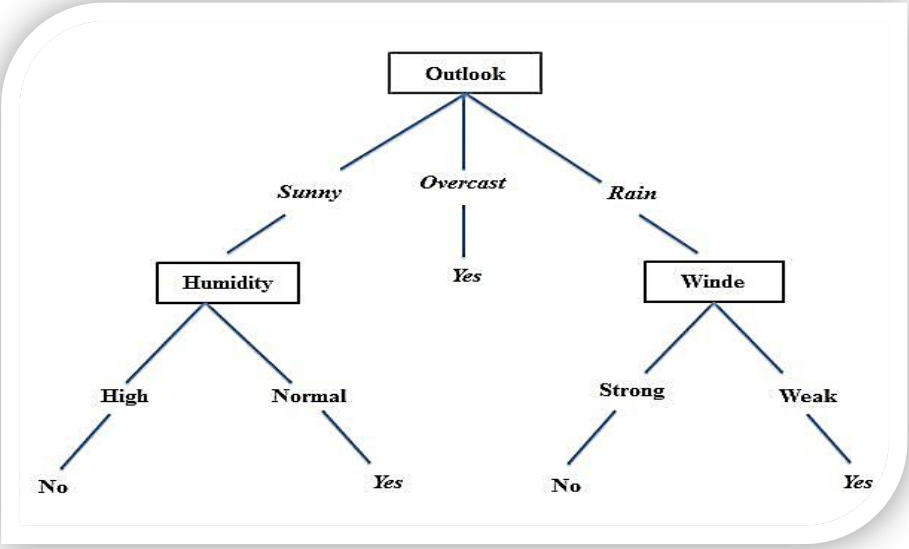

Figure 3.1 : l'arbre de décision

Pour construire l'arbre de décision, il faut trouver quel attribut tester à chaque nœud. C'est un processus récursif. Pour déterminer quel attribut tester à chaque étape, on utilise un calcul statistique qui détermine dans quelle mesure cet attribut sépare bien les exemples Oui/Non.

On crée alors un nœud contenant ce test, et on créer autant de descendants que de valeurs possibles pour ce test.

Exemple : si on teste la présence d'un mot, les valeurs possibles sont Présent/Absent. A chaque fois, on aura donc deux descendants pour chaque nœud..

3.2.2.2. Algorithme :

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

```
Arbre ← arbre vide ; nœud_courantracine
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
```

3.2.2.3 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.

3.2.2.4. 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)**[4].**

3.2.3. 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.

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*.
- ***Vecteurs Support* :** ce sont les points qui déterminent l'hyperplan tels qu'ils soient les plus proches de ce dernier.

Voici un schéma représentatif de ces notions**[4]** :

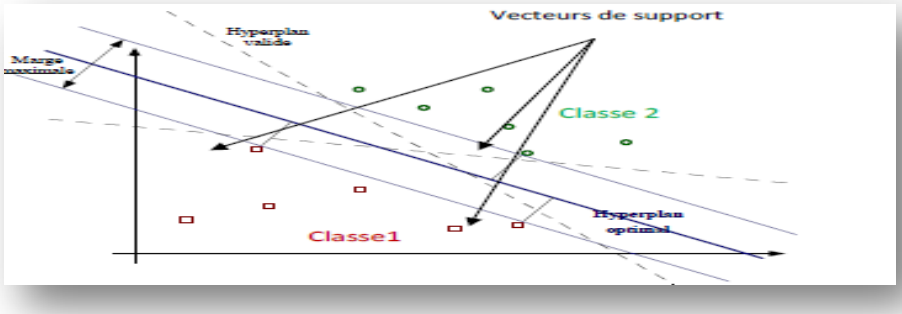

Figure 3.2 Les vecteurs à support

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.

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 **[10]**.

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**[13]**.

3.2.4 Réseaux de neurones

Un réseau de neurones (ou *Artificial Neural Network* en anglais) est un modèle de calcul dont la conception est très schématiquement inspiré du fonctionnement de vrais neurones (humains ou non). Les réseaux de neurones sont généralement optimisés par des méthodes d'apprentissage de type statistique grâce à leur capacité de classification et de généralisation**,** tels que la classification automatique de codes postaux ou la prise de décision concernant un achat boursier en fonction de l'évolution des cours. Ils enrichissent avec un ensemble de paradigmes permettant de générer de vastes espaces fonctionnels, souples et partiellement structurés .Ils appartient d'autre part à la famille des méthodes de l' intelligence artificielle qu'ils enrichissent en permettant de prendre des décisions s'appuyant davantage sur la perception que sur le raisonnement logique forme.

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 associée 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. Mettre l'une derrière l'autre, les différentes couches d'un réseau de neurones reviendrait 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 une non linéarité à chaque étape. 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.

3.2 .5 Classification naïve bayésienne

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 classifieur bayésienne naïf, ou classifieur naïf de Bayes, appartenant à la famille des classifieurs Linéaires.

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

En termes simples, un classifieur 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 classifieur bayésien naïf déterminera que le fruit est une pomme en considérant indépendamment ces caractéristiques de couleur, de forme et de taille.

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

Dans beaucoup d'applications pratiques, l'estimation des paramètres pour les modèles bayésiennes 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.

Malgré leur modèle de conception « naïf » et ses hypothèses de base extrêmement simplistes, les classifieurs 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, [20]. Toutefois, une autre étude de 2006 montre que des approches plus récentes (arbres renforcés, forêts aléatoires) permettent d'obtenir de meilleurs résultats.[21]

L'avantage du classifieur 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.

3.2.5.1 Description du modèle Bayésienne

Le modèle probabiliste pour un classifieur est le modèle conditionne $\boldsymbol{p(C|F1, \ldots \ldots, F\mathfrak{y})}$

où **C** est une variable de classe dépendante dont les instances ou *classes* sont peu nombreuses, conditionnée par plusieurs variables caractéristiques$\boldsymbol{F1, \ldots \ldots, F\mathfrak{y}}$.

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

Par conséquent, nous le dérivons pour qu'il soit plus facilement soluble. À l'aide du théorème de Bayes, nous écrivons

$$p(C|F_1,\dots,F_n) = \frac{p(C)\ p(F_1,\dots,F_n|C)}{p(F_1,\dots,F_n)}.$$

En langage courant, cela signifie :

$$\text{postérieure} = \frac{\text{antérieure} \times \text{vraisemblance}}{\text{évidence}}.$$

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

$$\boldsymbol{p(C, F1, \dots\dots\dots, F\mathfrak{y})}$$

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

$$\begin{aligned}
&p(C, F_1,\dots,F_n)\\
&\quad = p(C)\ p(F_1,\dots,F_n|C)\\
&\quad = p(C)\ p(F_1|C)\ p(F_2,\dots,F_n|C,F_1)\\
&\quad = p(C)\ p(F_1|C)\ p(F_2|C,F_1)\ p(F_3,\dots,F_n|C,F_1,F_2)\\
&\quad = p(C)\ p(F_1|C)\ p(F_2|C,F_1)\ p(F_3|C,F_1,F_2)\ p(F_4,\dots,F_n|C,F_1,F_2,I\\
&\quad = p(C)\ p(F_1|C)\ p(F_2|C,F_1)\ p(F_3|C,F_1,F_2)\ \dots p(F_n|C,F_1,F_2,F_3,..
\end{aligned}$$

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

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

$$p(F_i|C, F_j) = p(F_i|C)$$

$$p(C, F_1, \ldots, F_n) = p(C)\ p(F_1|C)\ p(F_2|C)\ p(F_3|C)\ \cdots$$

$$= p(C) \prod_{i=1}^{n} p(F_i|C).$$

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

$$p(C|F_1, \ldots, F_n) = \frac{1}{Z} p(C) \prod_{i=1}^{n} p(F_i|C)$$

où **Z** (appelé « évidence ») est un facteur d'échelle qui dépend uniquement de $F1, \ldots \ldots \ldots, F\eta$, à savoir une constante dans la mesure où les valeurs des variables caractéristiques sont connues.

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(F*i*|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.

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 2***n***+1 , avec ***n*** le nombre de caractéristiques binaires utilisées pour la classification.

3.2.5.2 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)).

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.

L'espérance, μ, se calcule avec

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$$

Où N est le nombre d'échantillons et x_i est la valeur d'un échantillon donné.
La variance, σ^2, se calcule avec

$$\sigma^2 = \frac{1}{(N-1)}\sum_{i=1}^{N} (x_i - \mu)^2$$

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.

3.2.5.3 Construire un classifieur 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 classifieur 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 classifieur correspondant à cette règle est la fonction **classifieur** suivante :

$$\text{classifieur}(f_1, \ldots, f_n) = \operatorname*{argmax}_c \, p(C = c) \prod_{i=1}^{n} p(F_i = f_i | C = c).$$

3.2.5.4 Analyse

Fait étonnant, malgré les hypothèses d'indépendance relativement simplistes, le classifieur 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 classifieurs 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 façon très précises. Le classifieur 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 classifieurs bayésiens naïfs.

3 .3 Critères d’évaluation des classificateurs

Comme nous l’avons indique auparavant, la décision de classer un document dans une catégorie ou une autre se base sur son contenu et donc elle est subjective. Cette subjectivité entraine une difficulté d’évaluation des décisions prises par le classifieur puisqu’on ne dispose pas d’une définition formelle et précise de ce qui rend un document pertinent a une catégorie ou non. Pour cela, les chercheurs font souvent recours a des méthodes empiriques. [19]

Confirme le précédent en indiquant ≪ comme c’est le cas avec les systèmes de recherche d’information (information retrieval), nous nous basons sur l’expérience pour évaluer un classifieur de textes, plutôt que de procéder analytiquement. La raison en est simple : pour pouvoir évaluer un système analytiquement, on doit posséder une définition formelle du problème à résoudre. Il faudrait pouvoir spécifier à quoi correspondent exactement la rectitude et la complétude. En fait, c’est l’appartenance d’un document à une catégorie, sa pertinence au sein d’une catégorie, qu’il faut définir. Cependant, un caractère très subjectif ressort de ce concept, qui relève aussi de la sémantique d’un texte. Donc, pour l’instant, ce n’est pas formalisable. On se replie alors sur une évaluation empirique des classifieurs ≫. Plusieurs mesures d’évaluation empiriques ont été proposées dans la littérature. Nous allons nous contenter de présenter, dans ce partie, celles souvent utilisées pas les chercheurs du domaine de la classification automatique des documents.

Diverses façons existantes aujourd’hui ont pour objectif de comparer les décisions prises par le classificateur automatique a 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 ci-dessous.

L'ensemble des catégories	Document appartenant à la catégorie	Document n'appartenant pas à la catégorie
Document assignés a la catégorie par le classifieur	***a***	***b***
Document rejetés a la catégorie par le classifieur	***c***	***d***

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

- 1· **Précision** (≪précision≫) : a / (a + b), soit le nombre d'assignations correctes sur le nombre total d'assignations.
- 2· **Rappel** (≪recall≫) : a / (a + c), soit le nombre d'assignations correctes sur le nombre d'assignations qui auraient du être faites.
- 3· **Exactitude** (≪accuracy≫) : (a + d) / (a + b + c + d).
- 4· **Erreur** (≪error≫) : (b + c) / (a + b + c + d).

Comme un document appartient généralement a 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.

Entrainer un classificateur sur la base de l'optimisation d'un de ces deux critères tendrait a 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 [18].

4· **F-Mesure :**

Plusieurs indicateurs ont été crées, mais le plus usuel est la F-mesure qui prenant en compte la valeur relative de la précision et du rappel est calcule comme suit :

$$\text{F\_Mesure} = \frac{(2 * \text{précision} * \text{rappel})}{\text{précision} + \text{rappel}}$$

3.4 Conclusion

La classification supervisée de documents a fait beaucoup de progrès ces dernières années.

Nous avons présente les principales techniques de classification automatique supervisée, utilisées pour classer des unités textuelles en groupes homogènes.

Dans ce chapitre nous avons présente quelques techniques de la catégorisation automatique de texte. Nous basons dans notre travail sur la méthode k-Nearest Neighbor (KNN) et nous avons applique sur elle une panoplie de mesures de similarité. Ainsi Nous basons sur la deuxième méthode Naïve Bayes. Nous avons également introduit les différents moyens d'évaluation d'un classificateur.

Chapitre 04 :

Conception

4.1 Introduction

Le domaine de la segmentation thématique est un domaine qui a donné lieu à de nombreux travaux ces dernières années. Les applications directes de cette tâche sont, certes, peu nombreuses, mais ses applications indirectes sont, elles, bien plus nombreuses.

La segmentation thématique peut par exemple être utilisée pour améliorer les performances de systèmes de question réponse ([V. Prince et A. Labadié. n Text Segmentation Based on Document Understanding for Information Retrieval. z. Proceedings of NLDB'07, pp 295–304, 2007.]) en fournissant des portions de texte thématiquement proches de la question.

Le résumé automatique peut également être amélioré, soit en permettant un résumé thème à thème, soit en proposant un résumé thématique.

4.2 Notre approche

L'idée du notre travaille est de répondre à la question : comment trouver des articles d'une thématique précise issue des articles de presse ? Pour cela, les articles sont évalués comme suivant :

Leur pertinence puis associés à une catégorie thématique (comme culture, politique…, etc.). Cette approche permet de retrouver des informations d'une thématique précise contenues dans les articles.

L'objectif de nos travaux est d'apporter une méthode qui effectue cette classification thématique de manière automatique . Nous proposons dans ce mémoire une approche constituant des nouvelles représentations du corpus original en utilisant des connaissances grammaticales. Afin d'obtenir de telles connaissances, nous utilisons un étiqueteur grammatical.

L'idée derrière notre approche est que les noms sont porteurs de sens concernant les thèmes plus que les autres classes (verbes, adjectifs,……). En effet, les verbes décrivent des événements, des actions, des opérations alors que les adjectifs sont utilisées pour exprimer des opinions et des jugements.

Pour chercher un sujet, l'utilisateur utilise des mots clés pour exprimer ses besoins, pour cela la classe des noms est la classe la plus significative.

Notre solution consiste à filtrer le texte on sélectionne seulement les noms puisque notre classification est thématique ainsi le texte filtré sera réduit a ensemble de noms.

Théoriquement, cette approche présente un avantage de réduction de la taille de corpus, ainsi que le temps d'exécution des classifieurs (temps de prétraitement, temps de classification,…..ect). Il reste a vérifier que les performances de classification (rappel, précision, F_measure,…) sont aussi au meilleurs que dans la classification classique qui utilise le corpus de texte complet (sans filtrage des noms).

4.3 Définition de POS Tagger (part-of-speech tagger)

Une partie du discours Tagger (POS Tagger) est un logiciel qui lit le texte dans une langue et affecte des parties du discours à chaque mot (et autre jeton), tels que nom, verbe, adjectif, etc, bien que généralement de calcul applications utilisent des étiquettes plus fine POS comme «nom-pluriel». Ce logiciel est une implémentation Java des tagueurs log-linéaires partie-de-discours décrites dans ces documents.

Le tagueur est sous licence GNU General Public License (v2 ou plus tard). Source est inclus. Source est inclus. Le paquet inclut les composants de la ligne de commande invocation, fonctionnant comme un serveur, et une API Java. Le code de tagger est double licence (d'une manière similaire à MySQL, etc.) Licences open source est sous l'entière GPL, qui permet de nombreuses utilisations libres. Pour les distributeurs de logiciels propriétaires, les licences commerciales sont disponibles. Si vous n'avez pas besoin d'une licence commerciale, mais souhaitez soutenir l'entretien de ces outils, nous nous félicitons de financement de cadeau.

La description de l'installation sont trouvé dans le site web « tagging text with stanford ».[http://www.galalaly.me/index.php/2011/05/tagging-text-with-stanford-pos-tagger-in-java-applications/]

4.3.1 Les classes de tagger

Le tagger de stanford fonctionne avec les textes, il noté les mots par leur catégorie,

Si le mot est verbe concaténer le mot avec « _V », sont classes est :

- Nom noté(N)
- Adjective noté(A)

- Adverbe noté (ADV)

4.4 Classification thématique

Nous avons décidé de définir le segment thématique comme étant : « La plus petite unité textuelle thématiquement cohérente en son sein et thématiquement distincte des unités textuelles précédentes et suivantes. L'unité atomique du segment thématique est la phrase. ».

La classification de textes consiste à regrouper des contextes (dans notre cas des documents) dans différentes classes qui correspondent à des catégories thématiques (par notre travail, les thèmes politique, sport, la santé, économie, etc.).

4.4.1 Identification d'un thème

Il existe beaucoup de définitions du mot « thème », chacune correspondant à un usage ou un domaine d'usage différent.

Le terme thème vient du grec *thema* qui signifie ce qui est proposé. Si l'on ouvre un dictionnaire,

La définition que l'on lira sera : « Sujet, idée sur lesquels porte une réflexion, un discours, une œuvre, autour desquels s'organise une action ».

En linguistique le thème est : « L'élément d'un énoncé qui est réputé connu par les participants à la communication » ou encore « Terme de la phrase (syntagme nominal) désignant l'être ou la chose dont on dit quelque chose. » (on l'oppose souvent au rhème qui est l'information nouvelle apportée par l'énoncé).

En musique la notion de thème est également présente et signifie : « Fragment mélodique ou rythmique sur lequel est construite une œuvre musicale. ». Nous passerons ici sur les thèmes astraux et militaires, pour nous concentrer sur ce que ces définitions ont en commun. Le thème est l'information centrale sur laquelle s'articule un acte de communication.

Plus simplement, la définition que nous retiendrons de la notion thème dans nos travaux sera : « Ce dont on parle », l'information principale communiquée par l'auteur. En cela nous nous rapprochons de l'étymologie du terme

4.4.2 Définition d'un concept

Un concept est une représentation générale et abstraite de la réalité d'un objet, d'une situation ou d'un phénomène; il n'est pas synonyme de notion car plus abstrait (par exemple, la notion de table, le concept de liberté). Concept vient du participe passé latin conceptus du verbe concipere, qui signifie « contenir entièrement », « former en soi ».

Le concept se distingue donc aussi bien de la chose représentée par ce concept, que du mot, de la notion, ou de l'énoncé verbal, qui est le signifiant de ce concept.

4.5 Approche proposé

Notre traitement consiste a trois étapes principales: prétraitements, tagger, classification

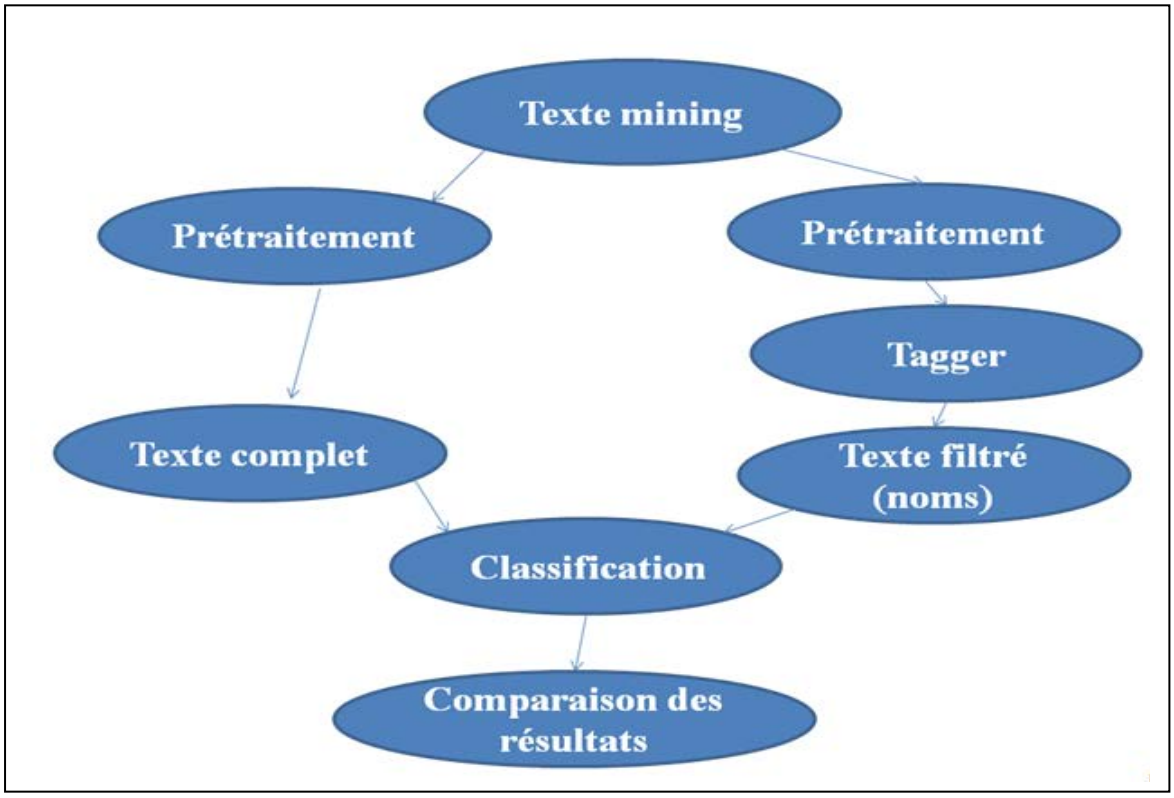

Figure 4.1 Les étapes de notre approche

la figure sous dessus présenter le schéma de notre proposition qui consiste des étages appliquée pour chaque partie (corpus filtré et corpus complet) .

la seule différence que le corpus complet ne passe pas par la phase de tagger (tagger définit dans les points précédents) pour filtré seulement les noms.

4 Conclusion

Au cours de ce chapitre, nous avons présenté et expliquer l'idée de notre travail qui consiste de faire la classification thématique de manière automatique, par proposé un approche de classification du texte par les noms communs après le filtrage (texte contient seulement les noms).

Chapitre 05 :

Implémentation et expérimentation

5.1 Introduction

Au cours de ce chapitre, nous allons présenter notre travail qui consiste à concevoir et à réaliser un système de classification automatique des documents français. Celui-ci est basé sur les naïve bayésienne. Le modèle du naïve bayésienne qu'on va utiliser dans notre système est un modèle perception. Nous commençons par présenter les différents outils utilisés : le Language de programmation (java) et l'environnement de développement (Netbeans IDE 6.8).

Le principe de transcodage seront implémentés et intégrés dans la plateforme de data mining WEKA . WEKA est une plate forme open source qui rassemble une série de méthodes de fouilles de données. Nous présentons aussi la structure de notre système, puis nous montrons les résultats obtenus pour chaque partie : la catégorisation thématique ; et finalement nous présentons quelques interfaces de notre application.

5.2 Outils de développement

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 environnement… etc.

L'outil que nous avons adopté est JAVA sous l'environnement NetBeans, notre choix c'est porté sur cette outil car la plate forme WEKA a été développé en java, ainsi que le nombre phénoménale des composants et classes mise a la porté des utilisateurs.

Pour implémenter notre système nous avons utilisé les outils suivants :

5.2.1 Langage 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.

5.2.2 Environnement de développement

L'environnement de développement utilisé, est NetBeans 6.8 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, …

5.2.3 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 de 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.

Pourquoi l'utiliser

- Il offre la majorité des fonctionnalités listées
- Auto complétion du code et le surlignage de la syntaxe
- Accès plus simple aux différents fichiers du projet
- Commentaires contextuels si le projet est commenté comme il faut (javadoc par ex...)

- ' Aide/doc intégrée à l'ide
- Création et intégration/réutilisation de composants/assemblages
- 1' Indépendance par rapport à l'OS de la machine de développement
- Outils de création d'Interfaces Homme/Machine

5.3 Présentation de la plate forme 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.

5.3.1 Structure de données :

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

5.3.2 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)

5.4 Présentation du corpus d'expérimentation

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 de nos expérimentations un corpus de textes écrits en la langue français.

Notre corpus contient 400 textes répartis en 4 catégories (politique, sport, santé et économie). Et 200 textes pour le test.(ces documents sont prend par reuters, AFP, buteurs,…etc).

Catégorie	Nombre de document
Politique	**150**
Santé	**150**
Economie	**150**
Sport	**150**

Tableau 5.1 Corpus d'apprentissage utilisé dans les expérimentations

Catégorie	Nombre de document
Politique	50
Santé	50
Economie	50
Sport	50

Tableau 5.2 Corpus de test utilisé dans les expérimentations

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

Le prétraitement consiste à :

- Convertir les majuscules en minuscules,
- Enlever les caractères non alphanumériques : les mots sont séparés par des espaces, des signes de ponctuations, des chiffres et les caractères spéciaux…
- Elimination des mots vides, les mots grammaticaux les mots de pays et les noms propres.

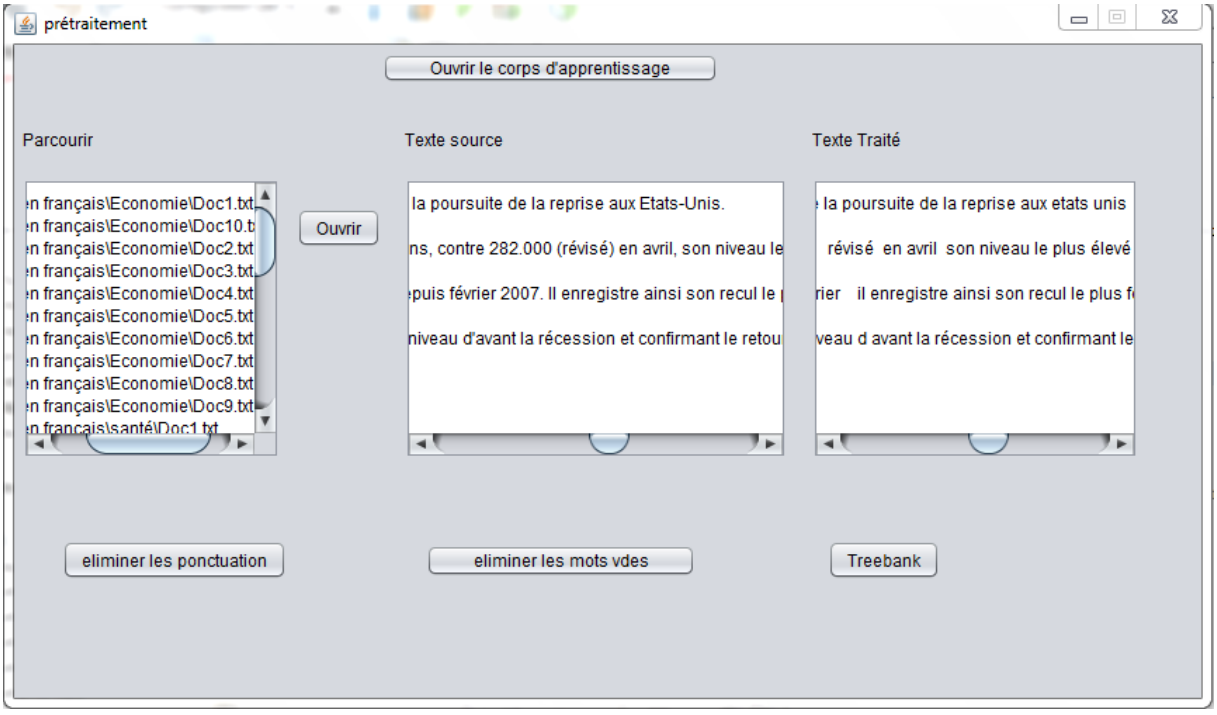

Figure 5.1 Elimination des signes de ponctuation

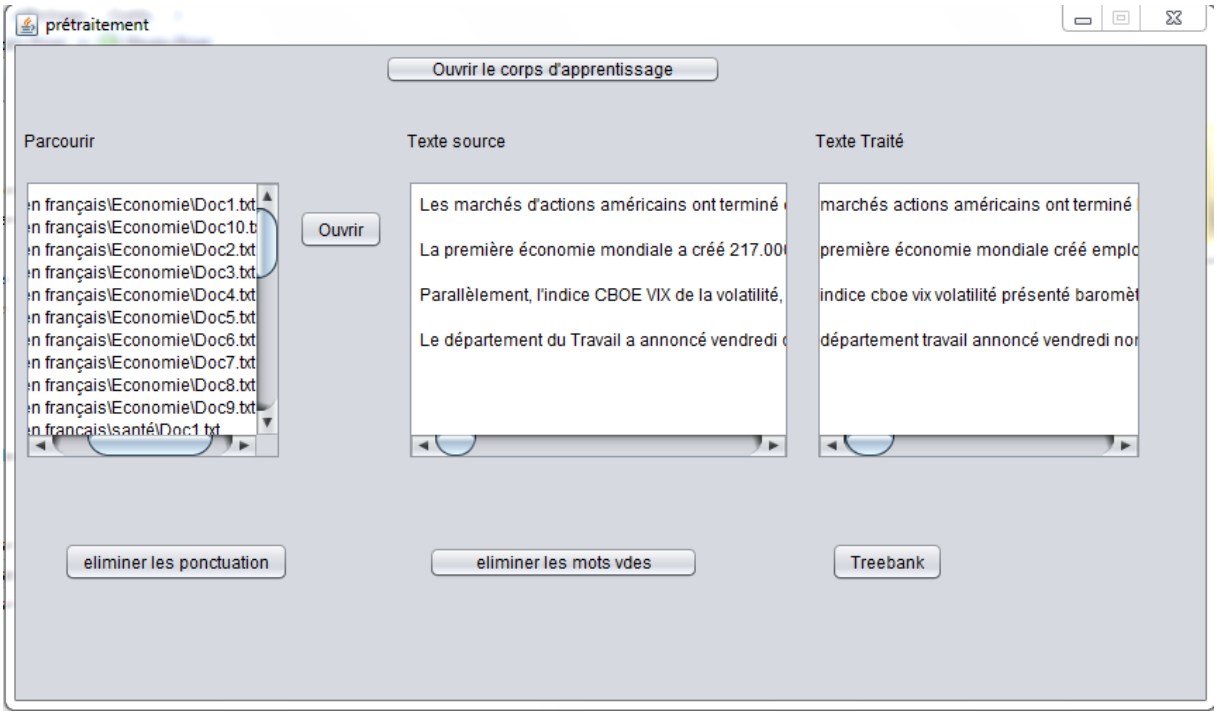

Figure 5.2 Elimination des mots vides (stopwords)

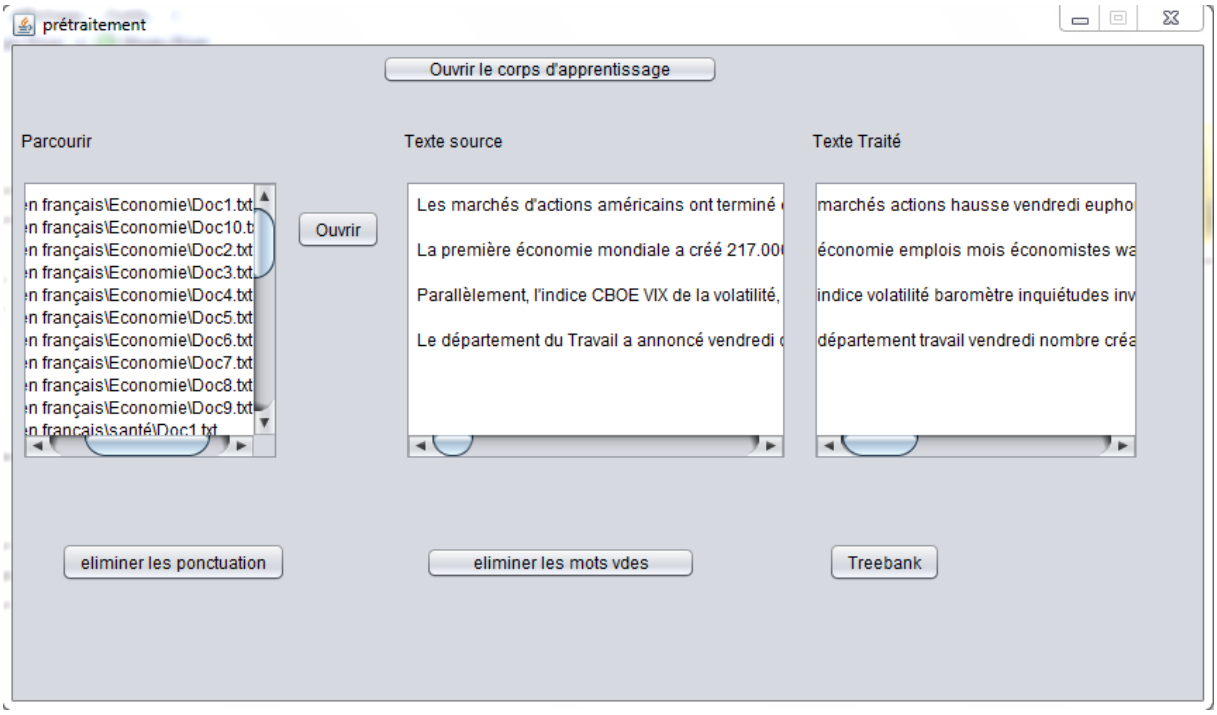

Figure 5.3 Filtrer les noms

5.6 Le processus de classification a travers WEKA

Pour faire la classification par des méthodes (naïves bayes) et mesurer les performances de classifieur (rappel, précision,…..etc.) par WEKA, ce dernier a besoins d'un fichier de format spéciale qui s'appelle ARFF, ARFF définit des attributs par les utilisateurs, ces attributs peut être nominal, numérique, string, date et relationnel.

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.

WEKA traite des données contenues dans des fichiers respectant le format ARFF Attribute_Relation . Il s'agit de fichiers de type texte, décrivant des ensembles des tuples caractérisés par un certain nombre d'attributs communs.

La figure sous dessous représente un exemple de fichier ARFF :

```
@relation nom_text
@attribute  text string
@attribute categorie {Economie santé Sport }
@data

"déclaré ancien entraîneur Carlos Alberto Parreira conduit Brésil.....",Sport
"Actions européennes progressé poursuivre l'augmentation continue..... ",Economie
"actions européennes fermées opérations jour altitude marchés Espagne Italie Portugal",Economie
"Paris Saint-Germain  annoncé engagement joueurs  Marco a eal Madrid  Barcelone respectivement ",Sport
"Cancer groupe maladies cellules hostiles agressif croissance division capacité ",santé
```

Figure 5.4 Fichier arff

Après le lancement du logiciel WEKA, on obtient la fenêtre principale :

Figure 5.5 Fenêtre principale

En choix l'application que l'on désirée (pour notre travail Explorer), et sélectionner le fichier ARFF qui contient les données, Cliquer sur le bouton Explorer disponible dans le menu Application. Apres l'ouverture de la fenêtre explorer le seul onglet visible est l'onglet

preprocess, Cliquez sur « Open file ». Pour notre exemple nous allons choisir le fichier « Resultat_Texte.arff ou Resultat_Noms.arff ».

Notre travail consiste de générer deux fichier ARFF, le premier contient seulement des noms et le deuxième contient le texte (verbe, nom, adjectifs,....etc).

Chaque fichier (Resultat_Texte.arff & Resultat_Noms.arff) représente une entré pour le WEKA pour faire la catégorisation et comparer les résultats de classification pour chaque fichier, et déterminer le meulier entré.

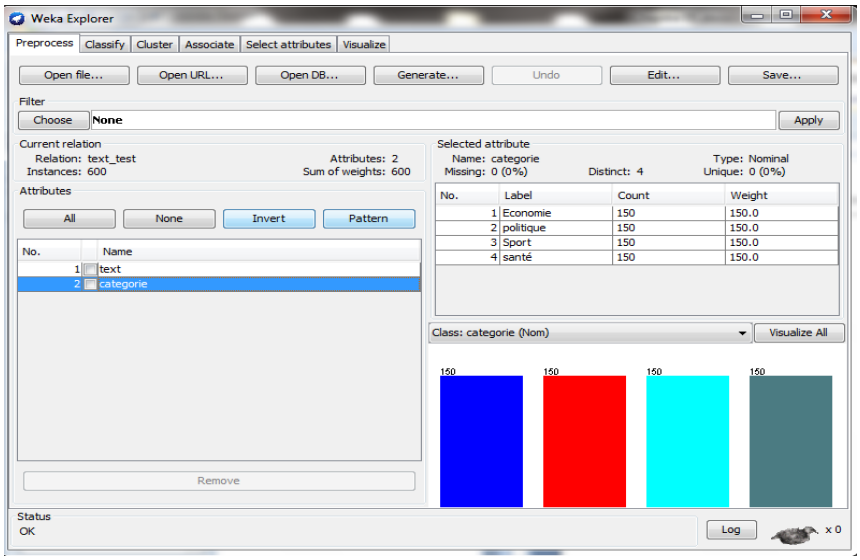

Figure 5.6 Sélection d'un fichier dans WEKA

Apres la sélection du fichier, nous obtenons la fenêtre sous dessus qui affiché les attributs de fichier ARFF (texte, catégorie).

En peut appliquer quelque filtres pour la représentation des documents (StringToWordVector).

StringToWordVector convertis en chaînes de caractères dans un ensemble d'attributs représentant mot occurrence (selon le tokenizer) des informations du texte contenu dans les cordes. L'ensemble des mots (attributs) est déterminée par le premier lot filtré (formation généralement des données).

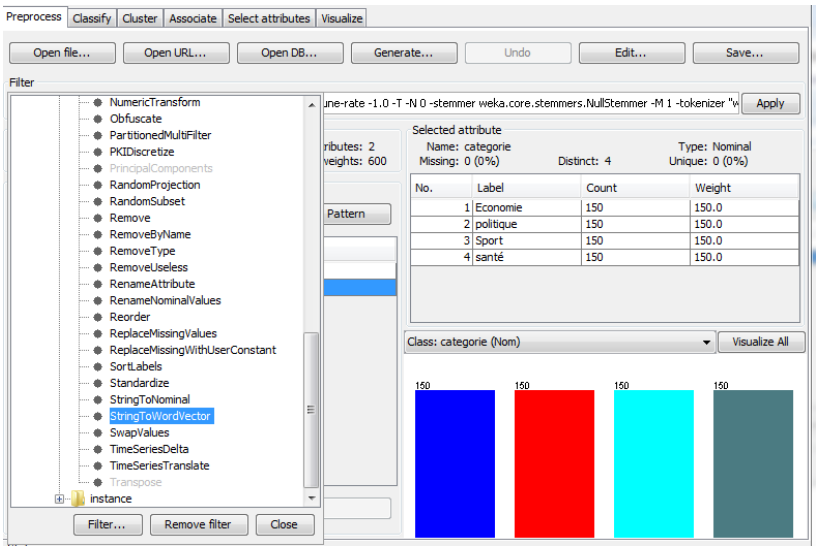

Figure 5.7 choix de filtre

StringToWordVector aussi fournir des méthodes pour choisie quelque paramètres pour la représentation des termes comme TF, IDF, TFIDF, Tokenizer,.....ect.

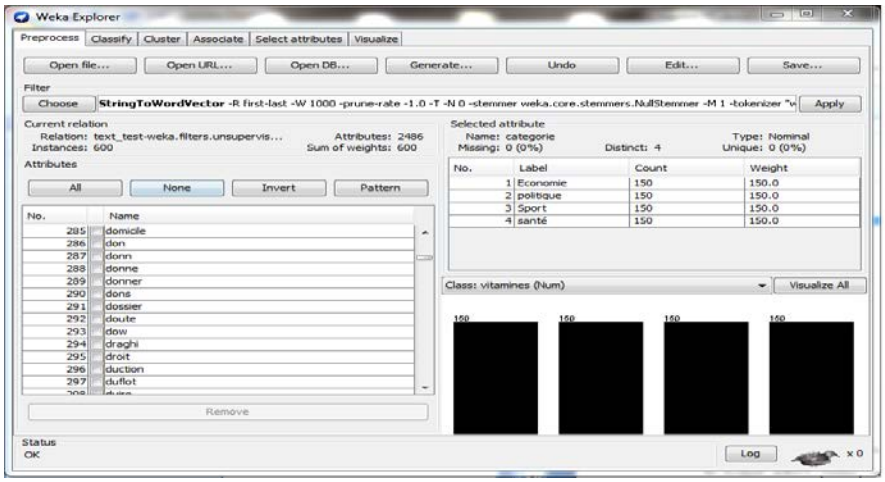

Figure 5.8 représentation des termes

En peut visualisation des mesures de TF, IDF, Tokenizer,....etc, le figure sous dessous représente la valeurs TFIDF pour les termes de documents.

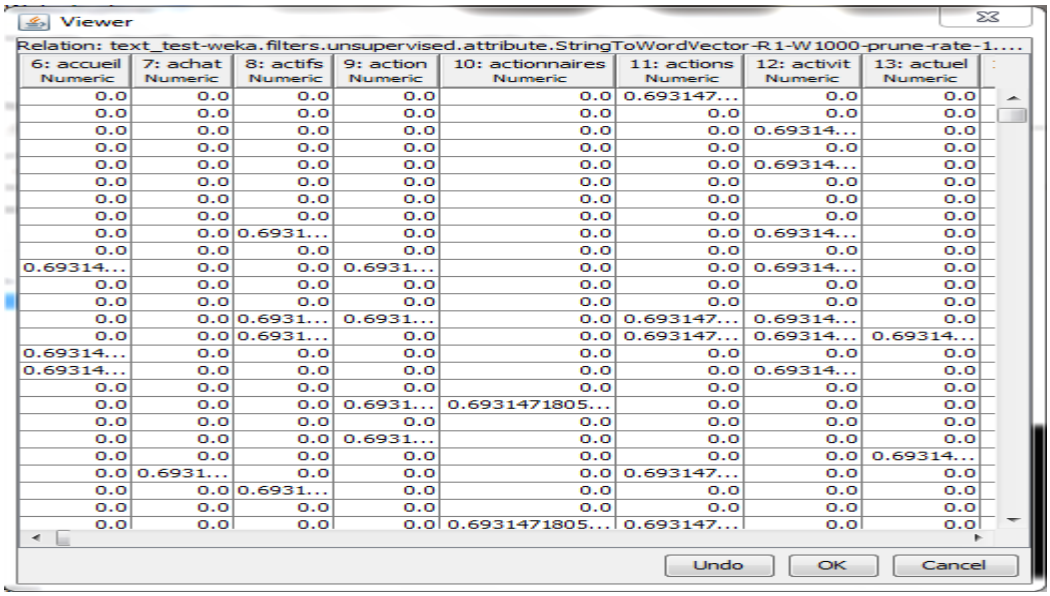

Figure 5.9 Calculer le TF_IDF

pour mesurer les performance d'un classifier on peut diviser le corpus d'apprentissage en deux partie :

- corpus d'entrainement
- corpus de test

Le corpus d' entrainement ayant pourcentage de 66% et le corpus de teste ayant pourcentage de 33% (pourcentage split), le figure suivant illustre comment faire cette étape:

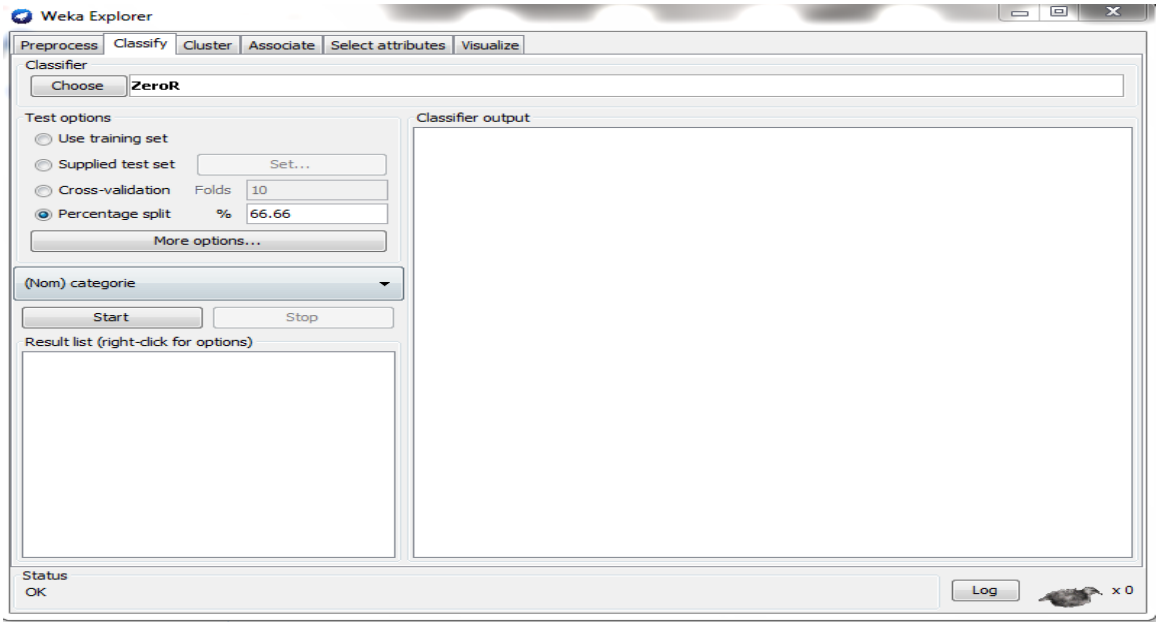

Figure 5.10 Split de corpus d'entrainement

Pour lancer la classification, on clique sur le bouton choose, après on choisi la méthode NaivesBayes parmi les méthodes de classification a base proposés dans WEKA.

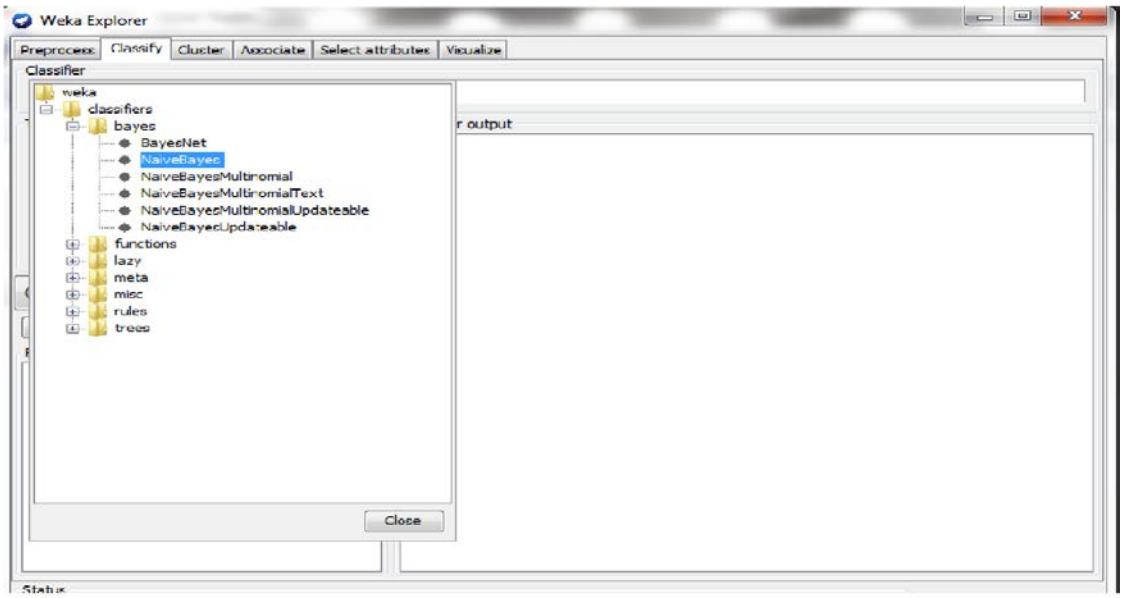

Figure 5.11 Choix de classifieur

Une fois que le méthode de classification est choisie on appuie sur le bouton « start», pour lancer l'exécution et prend les résultats.

```
=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,667    0,021    0,923      0,667    0,774      0,724    0,943     0,851     Economie
                 0,951    0,094    0,722      0,951    0,821      0,779    0,948     0,849     politique
                 0,980    0,013    0,961      0,980    0,970      0,960    0,994     0,985     Sport
                 0,964    0,021    0,946      0,964    0,955      0,938    0,981     0,980     santé
Weighted Avg.    0,885    0,034    0,898      0,885    0,883      0,853    0,967     0,920

=== Confusion Matrix ===

  a  b  c  d   <-- classified as
 36 14  2  2 |  a = Economie
  1 39  0  1 |  b = politique
  0  1 49  0 |  c = Sport
  2  0  0 53 |  d = santé
```

Figure 5.12 résultat des mesures de classification des noms

```
=== Detailed Accuracy By Class ===

                 TP Rate   FP Rate   Precision   Recall   F-Measure   MCC      ROC Area   PRC Area   Class
                 0,704     0,021     0,927       0,704    0,800       0,751    0,955      0,887      Economie
                 0,951     0,082     0,750       0,951    0,839       0,800    0,962      0,858      politique
                 0,980     0,020     0,942       0,980    0,961       0,948    0,998      0,994      Sport
                 0,964     0,014     0,964       0,964    0,964       0,950    0,979      0,963      santé
Weighted Avg.    0,895     0,031     0,905       0,895    0,893       0,865    0,974      0,929

=== Confusion Matrix ===

  a  b  c  d   <-- classified as
 38 12  3  1 |  a = Economie
  1 39  0  1 |  b = politique
  0  1 49  0 |  c = Sport
  2  0  0 53 |  d = santé
```

Figure 5.13 résultat des mesures de classification des textes

Les deux derniers figures représenter les résultats de classifications des noms et textes respectivement .

Comme nous observent dans les résultats précédents que 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.

	Taille de corpus (Ko)	Temps d'exécution (s)	TP rate	FP rate	Precision	Rappel	F_measures
Texte complet	1081	5.36	0.895	0.031	0.905	0.895	0.893
texte filtrée (noms)	431	5.71	0.887	0.034	0.895	0.887	0.885

Table 5.3 comparaison les résultats

5.8 Interprétation des résultats

L'opération de filtrage par noms a permet de réduire la taille de corpus de 60%.

Pour le temps d'exécution, la classification des textes filtrés prend 5.71 s

Par contre, ce temps d'exécution est 5.36 dans le cas de corpus complet.

On remarque une légère dégradation des performances de la méthode de filtrage par noms par rapport a la méthode qui utilise les textes complètes (0.893 texte complet par contre 0.885 pour F_mesure).

Les résultats dépendent énormément de la qualité du corpus utilisé (classes homogènes) et la qualité des classes (classe bien séparés) .

On conclusion et comme attendu la technique de filtrage par noms dépasse la méthode classique qui utilise le texte complet, en deux critères : la taille et le temps d'exécution.

Par contre, concernant le critère de performance de la classification (F_mésure, Taux de bonne classification). une petite dégradation non significative de la méthode de filtrage par noms.

Ces résultat confirment bien notre invitions, le contribution des noms dans la classification thématique est remarquable et elle est presque équivalente a l'utilisation du corpus complet, d'autre coté elle présente l'avantage d'utiliser des noms significatifs, et un corpus réduit.

5.9 Conclusion

Après la génération de deux fichiers ARFF (fichier textes filtrée contient seulement les noms et fichiers textes complet) et utilisation de WEKA pour faire la classification, on conclurons que les résultats obtenir on peut dire que la méthode de naïve bayes donne les bonnes résultats avec textes complet que un texte filtrée

Conclusion générale

Conclusion générale et perspectives

Dans ce mémoire, nous nous sommes intéressés à la catégorisation des documents avec la méthode des NB. Rappelons que le but de la catégorisation est d'apprendre à une machine à classer un texte dans la bonne catégorie en se basant sur son contenu.

Nous avons opté pour une démarche expérimentale qui est la catégorisation thématique qui consiste à classer des documents selon leurs contenus avec le classifieur NB avec texte normal et texte contienne seulement des noms.

Cette phase à pour but de catégoriser un corpus qui contient des documents de thèmes variés et d'associer à chaque document la catégorie adéquate.

Après l'analyse des résultats obtenus, on a pu constater que les taux de classification sont acceptables ; mais ça n'empêche pas l'existence de quelques problèmes parmi les : l'absence de règles absolues permettant de déterminer avec exactitude.

On a rencontré aussi quelques problèmes dans les phases de prétraitement et de représentation des textes en particulier le développement d'un algorithme efficace pour la lemmatisation et le steemming.

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

- Appliquer d'autres approches de représentation des textes, à savoir : l'approche conceptuelle et l'approche des n-grammes.
- Implémenter d'autres classifieurs (KNN, SVM, arbres de décision,RNA) pour avoir l'occasion de les comparer avec notre classifieur RNA.
- Utiliser d'autres modèles de NB pour améliorer les résultats
- Utiliser les ontologies ou les dictionnaires pour enrichir la représentation des textes .

Bibliographie

Bibliographie

[1] B.Agard, A.Kusiak, « Exploration des bases de données industrielles à l'aide du Data Mining – Perspectives », 9ème colloque national AIP PRIMECA, Avril 2005.

[2] R. Lefébure, G.Venturi, « *Le Data Mining* » Edition EYROLLES, deuxième tirage 1998.

[3] A.Taibi, H.LAZREG, «Utilisation des algorithmes d'apprentissage dans la catégorisation automatique thématique de documents Etude de cas : les algorithmes K_PPV, Naïve Bayes», Mémoire de Licence, Université de M'sila, 2011-2012.

[4] 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.

[5] 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.

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

[8] Ameni Bouaziz. « Catégorisation automatique de news à l'aide de techniques d'apprentissage supervisé ». Rapport de Projet de Fin d'Etudes, Université Nice Sophia Antipolis.

[9] S.ABDELOUAHAB, «Processus de classification supervisée de textes arabes par la méthode K PPV Application aux articles de presse», Mémoire de Master, Université de M'sila, 2011-2012 .

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

[11] M. F. Porter, «An algorithm for suffix stripping », Program, pp 130–137, Morgan Kaufmann Publishers Inc, 1980.

[12] C.Ignat, « Représentation de textes a l'aide d'étiquettes sémantiques dans le cadre de la classification automatique », Européen Commission, IPSC, Strasbourg, France,2007.

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

[14] F.Sebastiani.« Machine learning in automated text categorization ». 2002.

[15] P. Lefévre.« La recherche d'information - du texte intégral au thésaurus » 2000.

[16] J.Clech, D.A.Zighed. « Une technique de réétiquetage dans un contexte de catégorisation de textes » 2004.

[17] Matallah Hocine. « Classification Automatique de Textes Approche Orientée Agent ». Mémoire de magister, Département de l'informatique, université d'Aboubekr Belkaid-Tlemcen. Février 2011.

[18] SIMON RÉHEL : « Catégorisation automatique de textes et cooccurrence de mots provenant de documents non étiquetés » Mémoire présenté à la Faculté des études supérieures de l'Université Laval, Québec, Janvier 2005

[19] Sebastián Pena Saldarriaga, « Approches textuelles pour la catégorisation et la recherche de documents manuscrits en-ligne » université de Nantes thèse de doctorat soutenu le 24 mars 2010.

[20] Harry Zhang "The Optimality of Naive Bayes". Conférence FLAIRS 2004

[21] Caruana, R. and Niculescu-Mizil, A.: "An empirical comparison of supervised learning algorithms".Proceedings of the 23rd international conference on Machine learning, 2006.

[22] CLASSIFICATION BAYESIENNE NAÏVE DE TEXTES
Eric Ngouana, Serge Mayaya
Faculté Polytechnique de Mons, 5ième Electricité, Certi_cat Applicatifs Multimédia

Webographie :

[7] http:// www.cuy.be/html/typoweb/chap1.html consulté le l'avril 2014.

[20] (http://www.cs.unb.ca/profs/hzhang/publications/FLAIRS04ZhangH.pdf) consulté le Mai 2014.

[21]
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.122.5901&rep=rep1&type=pdf
Consulté l'avril 2014.

ملخص:

نسعى إلى تقديم مجال تصنيف النص التلقائي.

فمن الممكن لجعل التمثيل النص التقليدي من مجموعة من الكلمات غير ذات صلة.

ويستند نهجنا على تصنيف النصوص لتعيين فئة وفقا لمعيار. التمثيل مع استخدام النص العادي والنص المصفى (أسماء فقط).

في نهجنا وتستخدم خوارزمية بايز وحساب مقاييس الأداء.

كلمات مفتاحية: Classification supervisé de texte, filtrage, naïve bayes

Résumé :

On cherche à présenter le domaine de classification automatique de textes.

Il est possible de faire la représentation classique de texte en ensemble de mots sans relation.

Notre approche est basée sur classifier les textes consiste a lui attribuer une classe selon un critère .avec l'utilisation de représentation de texte brut et texte filtré(seulement les noms).

Dans notre approche on utilise l'algorithme de naïve bayes et calculer les mesures de performances.

Mots clés : Classification supervisé de texte, filtrage, naïve bayes

Abstract :

It seeks to present the field of automatic text classification.

It is possible to make the conventional text representation of set of unrelated words.

Our approach is based on classification of texts is to assign a class according to a criterion.

Representation with the use of plain text and filtered text (only names).

In our approach naïve Bayes algorithm and calculate performance measures are used.

Keywords: Classification supervisé de texte, filtrage, naïve bayes