République Algérienne Démocratique Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université d'Ibn Khaldoun – Tiaret
Faculté des Mathématiques et de l'Informatique
Département Informatique

Thème

CLASSIFICATION AUTOMATIQUE DE DOCUMENTS NUMERISES PAR LES S V M

Pour l'obtention du diplôme de Master II

Spécialité : Génie Informatique

Option : SITW

Rédigé par : Zaoui Nassira

Dirigé par : Dr Chikhaoui Ahmed

Année universitaire 2014-2015

Dédicaces

Je dédie ce travail :

A mes très chers parents, pour toute l'aide,

A mes très chers frères et sœurs, pour leur encouragement,

A mes amis,

A tous mes professeurs,

A toute ma famille,

A toutes les personnes ayant contribué, matériellement ou moralement, de près ou de loin, à la réussite de ce projet.

A tous un grand merci !

Remerciements

Aucune œuvre humaine ne peut se réaliser sans l'aide de Dieu. Je le remercie en premier lieu de m'avoir donné la santé, le courage ainsi qu'une grande volonté pour aboutir à ce travail.

Je profite l'occasion pour remercier toutes les personnes qui ont contribué de près ou de loin à la réalisation de ce projet de fin d'études.

Je remercie également mon chère mère et mon cher père, et tous mes sœurs et frères pour leurs encouragements, toute la famille, tous les amis et toute la promotion 2014.

A l'occasion de la parution de ce projet de fin d'étude, nous tenons à remercier très chaleureusement notre promoteur Dr Chikhaoui Ahmed pour son aide, ses orientations et pour son soutien moral durant tous les temps de ce travail.

Nous exprimons aussi nos remerciements à l'ensemble du jury qui fera l'honneur de juger notre travail.

Merci à tous !

Résumé

Avec l'avènement de l'informatique et l'accroissement du nombre de documents électroniques stockés sur les divers supports électroniques et sur le Web, particulièrement les données textuelles, le développement d'outils d'analyse et de traitement automatique des textes, notamment la classification automatique de textes, est devenu indispensable, pour assister les utilisateurs, de ces collections de documents, à explorer et à répertorier toutes ces immenses banques de données textuelles.

Ainsi la catégorisation automatique de textes, qui consiste à assigner un document à une ou plusieurs catégories, s'impose de plus en plus comme une technologie clé dans la gestion de l'intelligence, les résultats obtenus sont utiles aussi bien pour la recherche d'information que pour l'extraction de connaissance soit sur internet (moteurs de recherche), qu'au sein des entreprises (classement de documents internes, dépêches d'agences, etc.).

À l'égard des différentes approches de classification automatique de textes, décrites dans l'état de l'art, se reposant sur une architecture classique basée sur un seul point de vue, nous avons introduit une nouvelle utilisation du classifieur « SVM» avec des textes codés en «lemme», basée sur un algorithme d'apprentissage supervisé.

L'objectif principal de nos travaux, est d'améliorer les performances et l'efficacité du modèle de classification.

Mots Clés: Catégorisation, Classification, Texte, Apprentissage, Lemme, SVM.

Abstract

With the advent of information technology and the increase in the number of electronic documents stored on electronic media and on the Web, particulary the textual data, the development of tools for analysis and processing of the texts, including automatic classification of texts, has become indispensable, to assist users, these collection of documents, to explore and identify all these huge banks of textual data.

Thus the automatic categorization of texts, that is to assign a document to one or more Categories, is needed more and more as a key technology useful as well for searching for information for the extraction of knowledge or on the internet (search engines), and with in enterprises (classification of internal documents (dispatches from agencies, etc.).

With respect to the different approaches to automatic classification of texts, described the state of the art, based on classifical architecture based on a single point of view, we have introduced a new use of the classifier «SVM» with texts encoded in "lemma", based on a supervised learning algorithm.

The main objective of our work is to improve the performance and effectiveness of the classification model.

Keywords: Categorization, Classification, Text, Learning, Lemma, SVM.

Sommaire

Liste des Figures

Abréviation

AA : **A**pprentissage **A**utomatique.

ASCII : **A**merican **S**tandard **C**ode for **I**nformation **I**nterchange.

C.D : **C**atégorisation de **D**ocuments.

C.T : **C**atégorisation de **T**extes.

OCR : **O**ptical **C**haracter **R**ecognition.

RI : **R**echerche d'**I**nformation.

SMO: **S**equential **M**inimal **O**ptimization.

SVM: **S**upport **V**ector **M**achine.

TAL : **T**raitement **A**utomatique de la **L**ange.

Introduction Générale

La révolution de l'information bousculée par le développement à grande échelle des accès réseaux Internet/Intranet a fait exploser la quantité d'informations textuelles disponibles en ligne ou hors ligne et la vulgarisation de l'informatique dans le monde des entreprises, des administrations et des particuliers, a permis de créer des volumes importants de documents électroniques rédigés en langue naturelle. Il est très difficile d'estimer les quantités de données textuelles crées chaque mois dans les administrations, les sociétés, les institutions, ou la quantité de publications scientifiques dans les divers domaines de recherche.

L'information textuelle qui prend de plus en plus d'importance dans l'activité quotidienne des chercheurs et des entreprises ainsi que les besoins d'accès intelligents aux immenses bases de données textuelles et leurs manipulations qui ont augmenté très largement, d'une part.

D'autre part les limites d'une approche manuelle qui est coûteuse en temps de travail, peu générique, et relativement peu efficace, ont motivé la recherche dans ce domaine.

Ainsi la recherche des solutions opérationnelles, et la mise en œuvre d'outils efficaces pour automatiser la classification de ces documents devient une nécessité absolue. De nombreux travaux de recherche se focalisent sur cet aspect donnant ainsi un nouvel élan à la recherche dans le domaine qui connait une évolution réelle depuis les deux dernières décennies.

Comment partitionner cette masse d'information en groupes ou classes pour dégager des ressemblances par thèmes, par auteurs, par langue, ou par d'autres critères de classification ou carrément un filtrage de l'ensemble de documents utiles parmi les documents inutiles (Cas des filtres anti-spams). C'est à ce niveau que se positionne notre problématique de classification de textes.

L'objectif de la classification de textes est de rassembler les textes similaires selon un certain critère, au sein d'une même classe.

Deux types d'approches de classification automatique peuvent être distingués :

La classification supervisée et la classification non supervisée. Ces deux méthodes diffèrent sur la façon dont les classes sont générées. En effet dans le cas de la classification non supervisée, les classes sont calculées automatiquement par la machine, par contre, dans l'approche supervisée, la classification de textes consiste à rattacher un texte à une ou plusieurs catégories prédéfinies par un expert, ces catégories pouvant être par exemple le sujet du texte, son thème, l'opinion qui y est exprimée, etc... Nous disposons pour cela d'un ensemble de textes pour lesquels la catégorie est connue (corpus d'apprentissage) et qui nous servent à entraîner nos modèles, modèles qui seront testés et évalués sur d'autres documents pour lesquels la catégorie est connue également (corpus de test), le meilleur de ces modèles sera adopté par la suite pour étiqueter automatiquement des nouveaux documents de catégorie indéterminée.

La problématique de classification nous conduit à nous placer dans l'intersection de plusieurs disciplines variées :

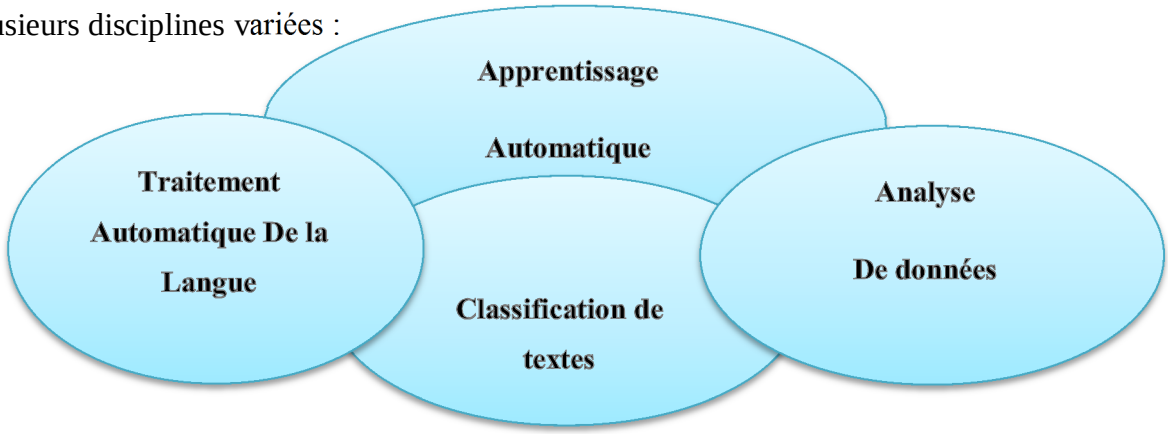

Figure 2: Position de notre problème

Ce mémoire va être organisé de la façon suivante : Un premier chapitre préliminaire pour définir l'ensemble des concepts de base du contexte étudié. Un état de l'art va être étalé au cours de chapitre 2 des techniques employées dans les différentes phases du processus de classification automatique de textes, le troisième chapitre fera l'objet d'une présentation générale de SVM et leur principe de base, alors que le dernier chapitre est L'implémentation pour résoudre notre problème de classification.

Ø Dans ***le premier chapitre***, nous présentons les différents jeux de mots utilisés: classification, catégorisation ou clustering. Les différents objectifs et intérêts et attendus de la classification ainsi que les confluents avec d'autres disciplines comme la Recherche d'Informations sera exposé par la suite puis nous décrivons le processus général de la catégorisation de textes avec toutes ces étapes, pour en finir avec les problèmes spécifiques aux textes lors de l'apprentissage automatique.

Ø Dans ***le deuxième chapitre***, nous allons exposer les différentes opérations de prétraitement nécessaires avant de commencer à vectorisé un texte. La définition et le choix des descripteurs ou termes, qui vont servir à représenter les documents, c'est un choix primordial et important dans la catégorisation de textes. La réduction de dimensionnalité qui va servir à diminuer la taille du vocabulaire avant d'appliquer les techniques de classification les plus complexes et enfin l'attribution des poids à ces termes. Tous ces points vont être étalés dans cette partie.

Ø Dans ***le troisième chapitre*** nous nous contentons d'exposer la méthode de classification le plus utilisée dans la littérature (SVM) en insistant sur les caractéristiques, les avantages et les limites de cette méthode, mais avant ceux-ci nous allons introduire la matière en positionnant notre problème dans un cadre d'apprentissage supervisé pour que le choix de la méthode soit adéquat.

Ø ***Le dernier chapitre*** l'implémentation.

Enfin, nous concluons ce mémoire en résumant les contributions que nous avons pu apporter, et en évoquant les suites de ce travail et les perspectives de recherche dans le domaine.

Chapitre 1: Classification automatique de documents textuels

Introduction

La classification automatique de textes 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. Elle connaît ces derniers temps un fort regain d'intérêt. Cela est dû essentiellement à la forte croissance des documents numériques disponibles et à la nécessité de les organiser de façon rapide.

La classification de textes est une tâche générique qui consiste à assigner une ou plusieurs catégories, parmi une liste prédéfinie, ou non à un document.

Actuellement, la classification de textes est un domaine de recherche très actif et l'automatisation de cette opération est devenue un enjeu pour la communauté scientifique, les travaux évoluent considérablement depuis une vingtaine d'années et plusieurs modèles ont vu le jour comme le filtrage (classification supervisée bi-classe), le routage (classification supervisée multi-classe) ou le classement ordonnée (classement des textes par ordre de pertinence pour chaque catégorie).

Avec ces modèles, des méthodologies de tests et des outils d'évaluation ont été mises en place. Les méthodes de représentation ainsi que les prétraitements correspondants sont maintenant bien connus. Les algorithmes de classification fonctionnent correctement mais déterminer les avantages des uns par rapport aux autres reste souvent délicat ou même améliorer les performances de la même méthode en intégrant d'autres paradigmes comme nous le faisons nous ici dans le présent mémoire reste toujours un domaine de recherche très prometteur. Le domaine du traitement intelligent de données textuelles regroupe tous les outils et méthodes capables d'extraire des informations de textes écrits dans une langue naturelle.

Il existe essentiellement deux domaines de recherche qui traitent cette problématique avec chacune ses propres méthodes :

- Les approches issues de l'analyse de données et de la statistique étudient et chercher surtout à proposer des dispositifs aux statisticiens et aux linguistes pour leur permettre d'analyser les grandes bases de données textuelles en fournissant des informations synthétiques sur les corpus. Les logiciels d'analyse de corpus qui fournissent des listes de fréquence de mots et les représentations graphiques issues de l'analyse factorielle des correspondances font partie de cette catégorie.
- Les approches qui proposent des systèmes de type « boite noire », ces méthodes traitent les documents de façon automatique sans intervention humaine. Elles réalisent souvent des fonctions de bas niveau : analyse lexicale, analyse syntaxique de surface, recherche d'information par mots-clés. Les moteurs de recherche popularisés avec le réseau Internet présentent un exemple typique des applications qui s'appuient sur cette approche.

Dans ce chapitre préliminaire, nous allons entreprendre notre sujet en répondant à la question pourquoi automatisé les documents ? Puis par la présentation des techniques de la classification automatique, ensuite définir la classification et les différents jeux de mots utilisés dans la discipline, ensuite éclaircir la notion de classe et la notion de catégorisation. Les différents objectifs et intérêts et attendus de la discipline ainsi que les confluents avec d'autres disciplines comme la Recherche d'Informations seront exposés par la suite. Nous finirons par développer la démarche classique d'un système de classification automatique de textes de la représentation des documents jusqu'aux évaluations des résultats ainsi que les différentes contraintes qui s'opposent au processus qui sont soit liées à la nature des données traitées (textuelles) soit au corpus lui-même soit aux techniques de représentation ou même le type de classifieur***. [1]***

I. Pour quoi automatiser le texte ?

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

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

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

Nous allons essayer de distinguer les différentes variantes de classification de textes et les vocabulaires utilisés dans la section suivante ***[1]***.

II. Les techniques de classification automatique

1. Catégorisation (Supervisé)

Ainsi, la catégorisation de textes correspond à la procédure d'affectation d'une ou de plusieurs catégories ou classes prédéfinies à un texte. Elle correspond à la classification supervisée pour l'apprentissage automatique et à la discrimination en statistiques alors que la recherche d'informations utilise des termes plus proches de l'application concernée : filtrage ou routage ***[1]***.

Cette problématique a par ailleurs dernièrement trouvé de nouvelles applications dans les domaines du traitement du langage tels que : l'affectation de sujets en recherche d'information, l'aide de l'utilisateur pour l'indexation de documents ***[2]***, la veille technologique, le filtrage personnalisé des documents intéressant un internaute connaissant ses préférences de sujets *(*catégories*)* ***[3]***, le routage de textes (tels que le courrier*)* et l'amélioration de la recherche sur le web ***[4]***, et enfin l'organisation des sources textuelles de plus en plus nombreuses, en particulier des pages web. Aujourd'hui, cette problématique utilise largement des méthodes issues de l'apprentissage automatique et beaucoup d'algorithmes d'apprentissage supervisé lui ont été appliqués (Naïve bayes, K-plus proches voisins, arbres de décision, machines à vecteurs support « SVM », réseaux de neurones, etc…) ***[1]***.

2. Clustering (Non supervisé)

Toutefois quand l'ensemble des catégories n'est pas donné au départ, et qu'il s'agit de le créer en regroupant les textes en classes qui possèdent un certain degré de cohérence interne, on est dans un contexte de classification non supervisée pour l'apprentissage automatique.
La classification non supervisée consiste à trouver de manière automatique une organisation cohérente à un groupe de documents homogènes pour construire des regroupements cohérents (des classes ou clusters), elle correspond en statistiques au clustering, qui est également le terme utilisé en recherche d'informations.

Le clustering consiste donc, à diviser les objets (dans notre cas des textes) en groupes sans connaître à priori leurs classes d'appartenance.

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 ***[1]***.

Ø Dans ce qui suit notre travail va être concentré sur la catégorisation de textes (la classification supervisée) à l'aide de SVM.

III. Définition de la Catégorisation de textes

Dans sa forme la plus simple, la catégorisation de documents consiste à assigner à un texte une ou plusieurs étiquettes permettant d'indexer le document dans un ensemble prédéfini de catégories, Originellement conçue pour assister le classement documentaire d'ouvrages ou d'articles dans des domaines techniques ou scientifiques.

La Catégorisation de Textes (*C.T*) est le processus qui consiste à assigner une ou plusieurs catégories parmi une liste prédéfinie à un document. L'objectif du processus est d'être capable d'effectuer automatiquement les classes d'un ensemble de nouveaux textes.

La catégorisation de documents consiste à apprendre, à partir d'exemples caractérisant des classes thématiques, un ensemble de descripteurs discriminants pour permettre de ranger un document donné dans la (ou les) classe(s) correspondant à son contenu ***[5]***.

Principalement, les algorithmes de catégorisation s'appuient sur des méthodes d'apprentissage qui, à partir d'un corpus d'apprentissage, permettent de catégoriser de nouveaux textes. Ce type de méthodes sont dites inductives car elles induisent de la connaissance à partir des données en entrée *(*les textes*)* et des sorties *(*leurs classes*)*.

Les divers travaux dans le domaine cherchent à trouver un algorithme permettant d'assigner un texte à une classe avec le plus grand taux de réussite possible sans toutefois assigner un texte à trop de classes. Dans un tel contexte, une mesure de similarité textuelle permet d'identifier la ou les catégories les plus proches du document à classer.

Si cette notion de similarité sémantique est un processus souvent intuitif pour l'homme, elle résulte d'un processus complexe et encore mal compris du cerveau.

Le problème de la catégorisation peut se résumer en une formalisation de la notion de similarité textuelle, soit en d'autres termes à trouver un modèle mathématique capable de représenter la fonction de décision d'appartenance des textes aux catégories.

Nous considérons un ensemble de classes *C = {ci}* et un ensemble de documents *D = {dj}* .Un système de classification associe automatiquement à chaque document un ensemble de classes (0,1 ou plusieurs). Le problème de la classification a été formalisé de plusieurs manières, nous vous proposons la formalisation de Sebastiani ***[6]*** reprise par

Yang ***[7].***

Deux fonctions sont définies :

- Une **fonction de décision** qui associe à chaque document un ensemble de classes.
- Une **fonction cible** qui nous renseigne sur l'appartenance exacte d'un document à Un ensemble de classes.

La fonction de décision est une estimation de la fonction cible qu'on ignore. Plus cette estimation est correcte, plus le système de classification est performant.

La fonction de décision et la fonction cible attribuent à chaque couple *(dj, ci)* $\in D \times C$ une valeur booléenne pour indiquer si le document *dj* appartient ou non à la classe *ci.*

La fonction de décision sera définie de la manière suivante :

$$\Phi : D \to D \times C$$

$$\Phi(d,c) = \begin{cases} vrai & \text{si } d \text{ est associé à la classe } c \\ faux & \text{sinon} \end{cases}$$

La fonction cible sera définie de la manière suivante :

$$\Gamma : D \rightarrow D \times C$$

$$\Gamma(d,c) = \begin{cases} vrai & \text{si } d \text{ est associé à la classe } c \\ faux & \text{sinon} \end{cases}$$

Dans les systèmes de classification basés sur des méthodes d'apprentissage, la fonction de décision sera évaluée à l'aide d'un corpus d'entraînement. Cette fonction peut faire intervenir un grand nombre de valeurs numériques qu'un humain ne peut pas saisir. La détermination de cette fonction est appelée ***phase d'apprentissage***, tandis que l'utilisation de cette fonction pour attribuer une catégorie à un document se fera pendant la ***phase de test [1]***.

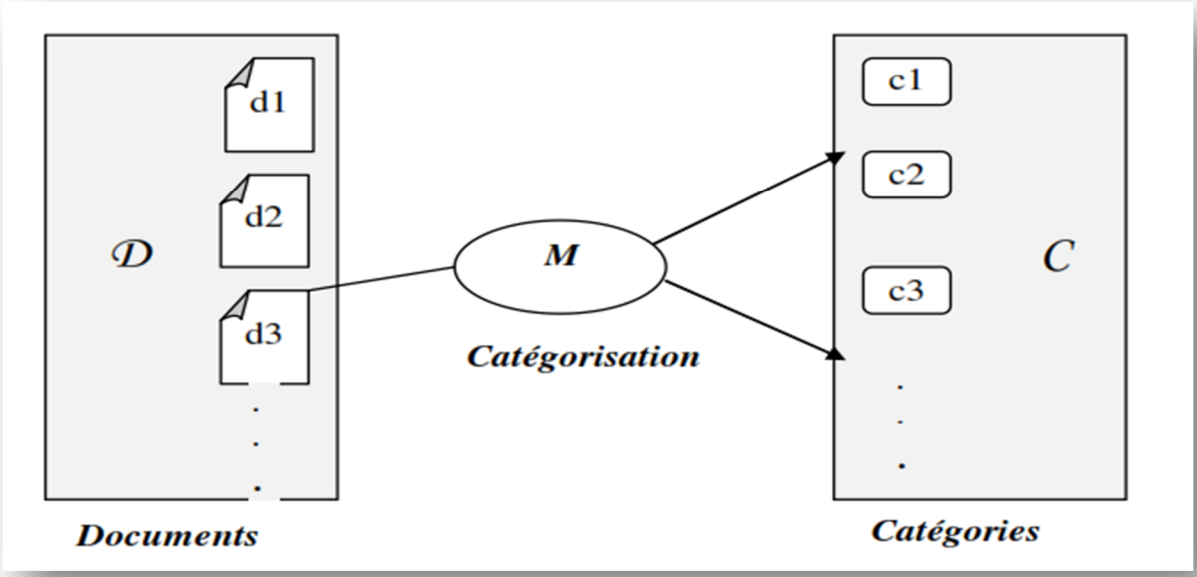

Figure 1.1: Fonctions de catégorisation

IV. La notion de classe pour les systèmes de classification

La notion de classe pour un système de classification a été habituellement synonyme de « thème». Dans ce contexte, classer les documents revient à les organiser par différentes thématiques. Cependant, la problématique de classification a évolué en même temps que les besoins et elle s'intéresse aujourd'hui à différentes tâches pour lesquelles les catégories ne sont pas interprétables comme des thèmes : ainsi, par exemple, les tâches consistant à classer les documents par auteur, par genre, par style, par langue, ou encore selon que le

document exprime un jugement positif ou négatif, etc... Ainsi la classe va correspondre à un besoin d'information d'un utilisateur ou d'une société et n'est donc pas obligatoirement un thème unique. Nous considérerons dans la suite qu'une classe est simplement une étiquette à associer à des documents.

Dans la **figure 1.2,** un système de classification d'emails est représenté où les classes peuvent être de différentes natures (thèmes, messages provenant de certaines personnes, messages d'un certain type, etc…) ***[1]***.

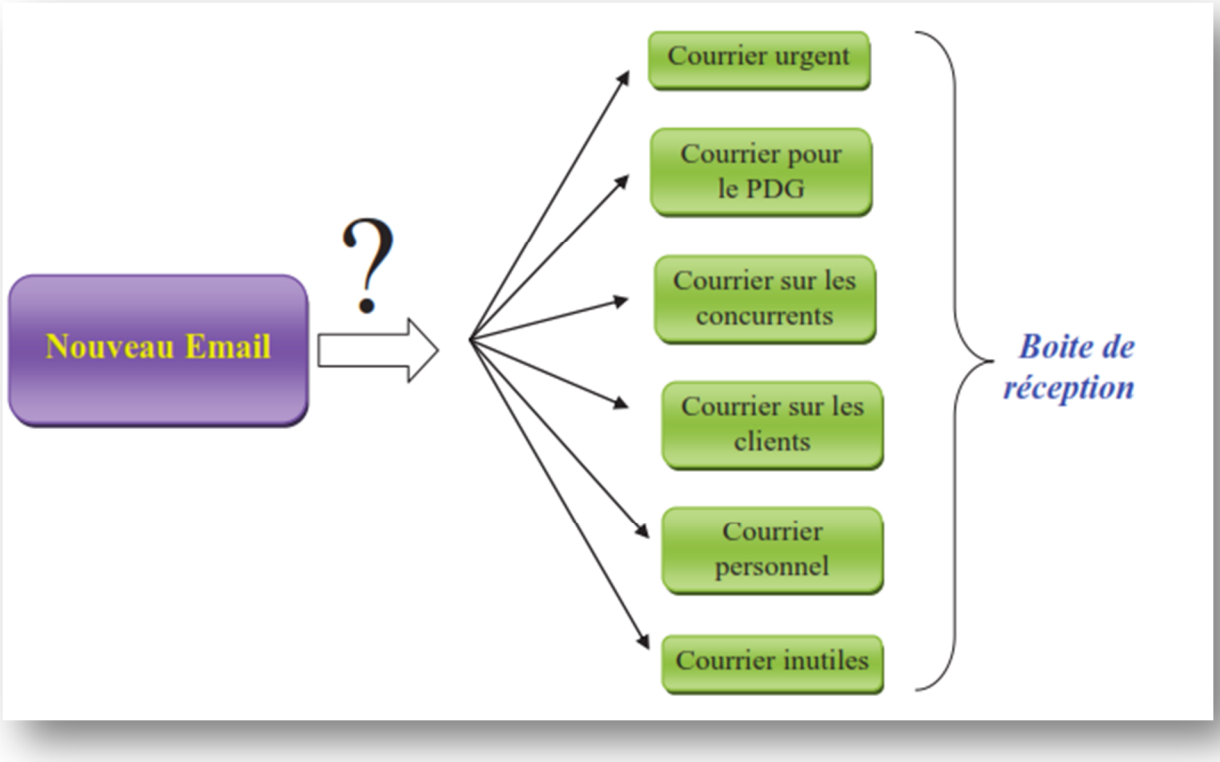

Figure 1.2: Exemple de système de classification

V. Les différents contextes de classification

Plusieurs contextes de classification se distinguent, ils influent directement sur les modèles utilisés. Ludovic DENOYER a bien résumé les différents contextes de classification dans ***[8]***que nous avons reporté dans ce qui suit, les problématiques les plus récentes comme par exemple la classification dans une hiérarchie de classes ***[9]*** ne sont pas abordés ici **.**

1. Classification bi-classe et multi-classe

1.1- La classification bi-classe

La classification bi-classe correspond au filtrage. C'est une problématique pour laquelle le système de classification répond à la question : « Le texte appartient-il à la catégorie C ou non. Cependant quand il s'agit d'effectuer une classification multi-classe qui permet de transmettre le document vers le ou les catégories(s) le(s) plus approprié(s), on parle alors de routage. Cette classification multi-classes, selon le cas, peut être disjointes ou non ***[1].***

1.2- La classification multi-classes disjointes

La classification multi-classes disjointes est le contexte de classification en un nombre de classes supérieur à un et pour lequel un texte est attribué à une et une seule classe. Un système de classification multi-classes disjointes répond à la question « A quelle classe (au singulier) appartient le document ? » ***[1]***.

1.3- La classification multi-classes

Dans un système de classification multi-classes, on peut associer un texte à une ou plusieurs classes voire à aucune classe. Le système répond donc à la question : « A quelles classes (au pluriel) appartient le document ? ». C'est le cas le plus général de la classification. Il correspond par exemple à la problématique de classification du corpus étudié ici dans ce mémoire ***[1]***.

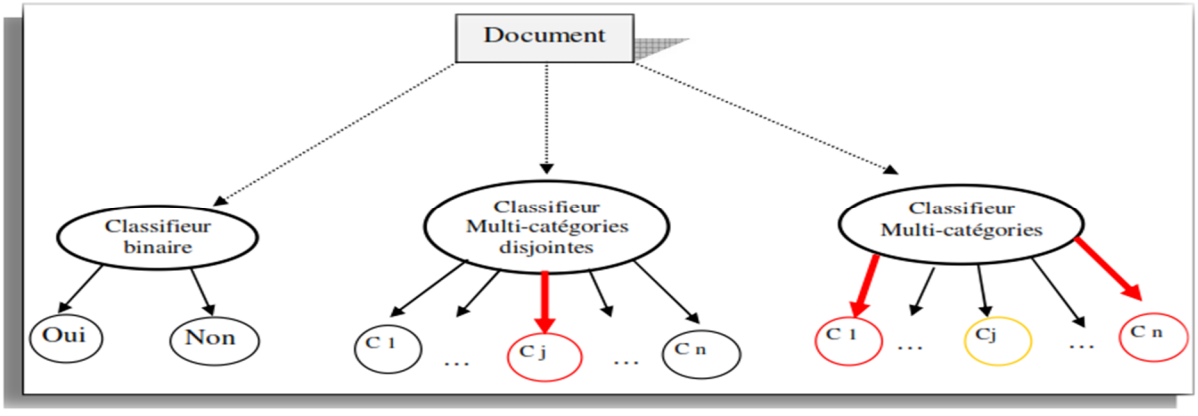

Figure 1.3: les trois paradigmes de la classe

VI. Objectifs et intérêts

Les intérêts des méthodes de classification sont multiples, il peut s'agir d'améliorer les performances des moteurs de recherche documentaire ou aussi classer les documents en fonction de leurs références communes à d'autres documents pour faire apparaître les liens qui les unissent.

Nous pouvons citer six applications typiques qui sont :

- ✓ Le classement automatique de différents communiqués de presse, ou messages sur des forums en différentes matières (« Les actualités de la région », « la bourse » etc.…
- ✓ Indexation automatique sur des catégories d'index de bibliothèques : aide à la classification thématique des différentes rédactions dans une bibliothèque.
  - ❖ La gestion de bases documentaires (mémoire d'entreprise). Ce système peut être utilisé pour présenter l'information à l'utilisateur selon des catégories thématiques, ce qui facilite la navigation.
  - ❖ Sauvegarde automatique de fichiers dans des répertoires.
  - ❖ Les filtres internet en général, et en particulier les filtres anti-spams.
  - ❖ Le classement automatique des emails, et particulièrement la redirection automatique de courriers des clients et fournisseurs en fonction de leur contenu vers les personnes compétentes dans une entreprise (Service commercial, livraison, service après-vente, approvisionnements, etc…) ou vers des répertoires prédéfinis dans un outil de messagerie, ou encore le tri de courriers électroniques dans différentes boîtes aux lettres personnelles et possibilité d'envoi de réponses automatiques ***[1]***.

VII. 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.

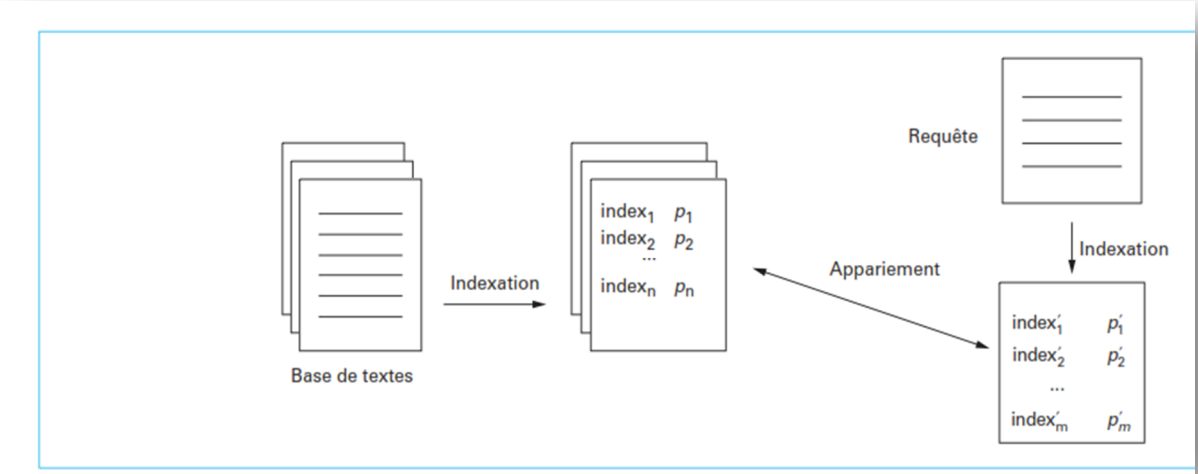

Figure 1.4: schéma de Recherche d'information

- 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 document pertinents. Cet ensemble de documents pertinents définit ce que l'on appelle, un thème ou une catégorie.

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 ***[1]***.

Une autre étude menée par ***[10]*** montre que la classification automatique permet d'améliorer l'efficacité des systèmes de recherche. Un système de recherche documentaire, comme on a vu précédemment, donne, en réponse à une requête, une liste de documents.

La liste des documents trouvés est souvent si longue que les utilisateurs ne peuvent l'examiner entièrement et laissent de côté certains documents pertinents mal classés. L'étude a démontré qu'une classification automatique des seuls documents retrouvés permet d'améliorer la qualité de la recherche documentaire ***[1]***.

VIII. 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) ***[1]***.

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 :

- La phase de préprocessinge consiste à utiliser les techniques de TAL pour transformer les textes en vecteurs de caractéristiques nécessaires pour SVM :
  - 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> ;
- La phase de l'apprentissage consiste à prendre en entrés les vecteurs et produire en sortie un :
  - modèle de classification ;
  - Entrainer le modèle de classification à partir des couples (Document, Classe) ;
  - Évaluer les résultats du classifieur ;
- La phase classification consiste à prendre le vecteur caractéristique du texte et l'utiliser pour la classification selon le modèle appris.

La figure 6 **:** illustre les 3 étapes de classification des flux RSS. ***[11]***

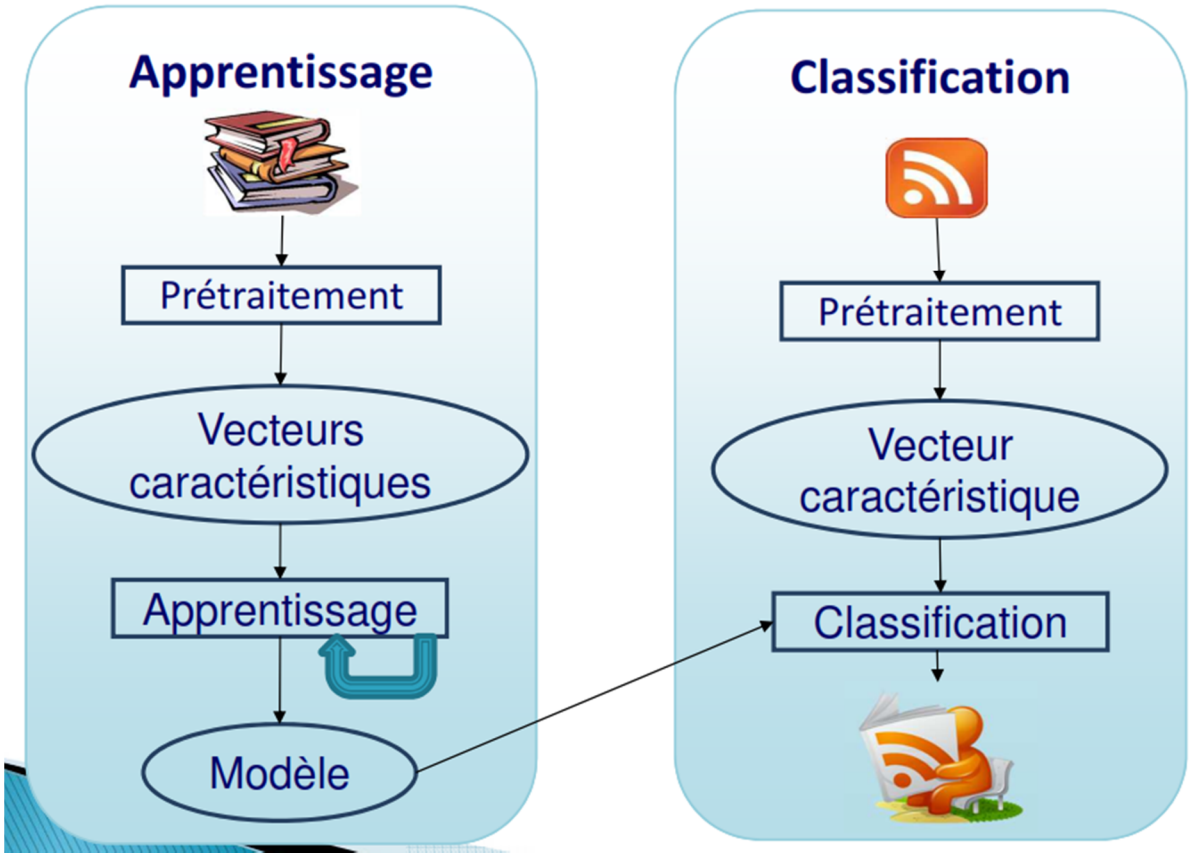

Figure 1.5 : le processus de classification des flux RSS

IX. Problèmes de la catégorisation de textes

Dans ce qui suit nous allons signaler les dix principales difficultés qui s'opposent à la catégorisation de textes :

1. Redandance (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. 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 termes en un groupe sémantique commun.

Pour y remédier, il est alors intéressant de concevoir une ontologie afin de cerner les sens des termes, naturellement, cela engendre des coûts supplémentaires pour sa réalisation et sa maintenance.

2. Polysémie (Ambiguïté)

A la différence des données numériques, les données textuelles sont sémantiquement riches, du fait qui sont conçues et raisonnées par la pensée humaine. Contrairement des langages informatiques, le langage naturel, autorise des violations des règles grammaticales engendrant plusieurs interprétations d'un même propos. 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… Le mot ***livre*** peut désigner une unité monétaire, ou un bouquin. Le mot ***avocat*** peut désigner le fruit, le juriste, ou même au sens figuré, la personne qui défend une cause. Le mot ***table*** de cuisine ce n'est pas le même que dans ***table*** de multiplication. Le mot ***pièce*** peut correspondre à une pièce de monnaie par exemple, ou à une pièce dans une maison, de même pour ***pavillon***, ***bloc, glace, etc…***

3. 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.

(Exemple : avocat en tant que fruit et avocat en tant que juriste). L'homographie et l'ambiguïté génère du bruit qui va causer une dégradation de précision (indicateur nécessaire pour mesurer la performance du classifieur). Il sera alors préférable d'ôter ces ambigüités.

4. 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 ***(Ghelizane, Relizane),*** 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.

Toujours dans ce contexte, ***[12]*** et ***[13]*** affirment que la graphie peut donner une information relative au sens du terme employé. Prenons par exemple le cas pour le mot Histoire dont la majuscule indique qu'il s'agit de la discipline étudiant le passé et non d'un roman ou une blague. La prise en compte de toutes ces variations morphologiques pour la classification automatique de textes n'est pas étudiée dans ce mémoire.

5. Les variations morphologiques

Les conjugaisons, pluriels, influent négativement sur la qualité des résultats puisque les différentes variations morphologiques vont êtes considérées séparément et chacune va être prise comme un élément à part comme par exemple les trois termes : maître, maîtresse, maîtriser sont traités indépendamment quoique en réalité ça pivote sur la même idée. Pour y remédier soit on applique la lemmatisation ou le stemming, à notre texte soit carrément on opte pour une représentation en n-grammes qui peut nous éviter ces prétraitements, tout ceux-ci va être étalé dans le deuxième chapitre.

6. 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 *Arc- en-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.

7. Présence-Absence de termes

La présence d'un mot dans le texte indique un propos que l'auteur a voulu exprimer, on a donc 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, dès 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.

8. Complexité de l'algorithme d'apprentissage

Plus tard, Dans le chapitre 2 : représentation et codage des documents, nous verrons qu'un texte est représenté généralement sous forme de vecteur contenant les nombres d'apparitions des termes dans ce texte. Or, le nombre de textes qu'on va traiter est très important sans oublier le nombre de termes composant le même texte donc on peut bien imaginer la dimension du tableau (textes * termes) à traiter qui va compliquer considérablement la tâche de classification en diminuant la performance du système. De ce fait, une réduction de la taille du tableau, comme nous allons voir par la suite, est primordiale avant d'entamer l'apprentissage.

9. Sur-apprentissage

Le nombre de termes très important et très varié qui ne se répètent dans tous les textes va causer énormément de creux dans le tableau de grande dimension (textes*termes) qui peut provoquer du sur-apprentissage qui s'explique par le fait que le modèle 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 ***[11]***.

10. 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 ***[14]***.

D'après les expériences : Lorsque deux experts humains doivent déterminer les classes d'une collection de textes, il y a souvent désaccord sur plus de 5 % des textes. Il est donc illusoire de rechercher une classification automatique parfaite.

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 chapitre1 de définir la classification ainsi que les notions nécessaires pour l'entame de la suite de ce mémoire.

Chapitre 2 : Vectorisation des textes

Introduction

L'explosion de la quantité d'informations textuelles provoquée par l'évolution à grande échelle des outils de communication essentiellement Internet qui est sorti de l'aspect réservé à un milieu restreint à un aspect de vulgarisation au grand public, a rapidement, fait sentir le besoin de recherche de mécanismes et outils de traitement automatique des quantités d'informations diffusées sur le Web.

Ainsi, avec les bases de données multimédia, les dépêches d'agences de presse, les publications scientifiques, les bibliothèques électroniques, etc... Qui sont consultés habituellement sur le réseau, on dispose de plus en plus de grandes masses de documents non ou faiblement structurées, en particulier les documents textuels qui sont considérés comme étant des documents non structurés, surnommés « documents plats » par quelques auteurs, c'est-à-dire comme une séquence ou un ensemble de mots sans informations complémentaires sur le document.

Le manque de structure au sein de ces collections volumineuses rend difficile l'accès à l'information qu'elles contiennent, d'où la nécessité aujourd'hui, de chercher comment structurer automatiquement ces corpus pour les rendre utilisables d'une façon rapide et optimale pour y faciliter leurs traitements automatiques et notamment la classification.

Pour pouvoir y appliquer les différentes techniques et algorithmes d'apprentissage, une transformation de ces documents non ou peu structurés est indispensable.

La transformation ou le codage de ces documents est une préparation à« l'informatisation» de ces derniers, chaque type de documents comme les images, les vidéos et notamment les textes dispose de ses propres techniques de codage.

Plusieurs approches de représentation des documents textuels ont été proposées dans ce contexte, la plupart étant des méthodes vectorielles.

La représentation vectorielle de texte, même si elle est très différente d'une analyse structurale linguistique, s'avère un modèle performant et rapide. Les méthodes vectorielles ont d'ailleurs, la propriété d'être assez indépendantes de la langue. Dans ce modèle de représentation, les textes sont traités comme des sacs de termes. Les termes peuvent être assimilés à des mots ou à des n-grammes. Cependant, les données résultantes seront très volumineuses, clairsemées et très bruitées. Ceci signifie que l'information pertinente ne constitue qu'une faible partie de l'ensemble total des données disponibles. La vectorisation d'un document crée une matrice terme-segment où chaque case représente la fréquence d'apparition d'un terme dans une phrase (ou segment). Cette matrice peut être très volumineuse. C'est pourquoi, des mécanismes de filtrage et de lemmatisation s'avèrent indispensables afin de réduire la complexité du lexique.

Un état de l'art des différentes approches de représentation de textes est développé dans ce chapitre.

I. Le texte

La numérisation est une opération qui s'est élargie pour atteindre toutes formes de documents et notamment les textes, et ce dans le but de leur exploitation sur les réseaux. Cet élargissement a entraîné derrière lui beaucoup de travaux qui ont un rapport surtout avec le formatage et la normalisation des textes qui ont été développés pour être à la fois rapide et efficace suite au développement de l'Internet.

Depuis les débuts de la numérisation des données textuelles, le texte a été considéré, et c'est encore vrai aujourd'hui dans la plupart des cas, comme tout simplement une séquence de caractères. Ces caractères peuvent être représentés dans différents espaces de codage, le plus courant étant le codage ASCII admettant 256 caractères différents, mais en dépit ce codage ne prenait pas en charge les langues comme l'arabe ou le chinois. Afin de pouvoir représenter ces langues, différentes normes de codages est créés et plus largement utilisées aujourd'hui comme la norme UNICODE qui permet la représentation de 65536 caractères.

Pour la plupart des langues (occidentales et orientales font partie), l'espace de codage au niveau caractère n'est pas un espace très informatif car un caractère seul ne présente pas une information sémantique riche. Un texte est plutôt considéré comme une séquence de mots (un mot lui-même étant une séquence de caractères) et représenté dans un espace de mots dont la dimension est plus grande que celle du caractère (Le nombre de caractères possibles

est limité mais en revanche le nombre de mots qu'on peut avoir est énorme), mais dont chaque dimension est beaucoup plus informative.

Ainsi la représentation informatique de ces textes nécessite un traitement spécifique. Mais Très vite, les méthodes de se sont heurtées au fait qu'un texte n'est pas un sac dans lequel seraient mélangées en vrac ses propres éléments. Le moins qu'on puisse dire sur un texte qu'il est une chaîne linéaire, donc un espace ordonné. (« Voile du bateau » et « Bateau à voile» ont des sens complètement différents).

Evoquer la composition d'un texte fait appel à deux définitions de la composition : il s'agit à la fois de déterminer les unités qui vont constituer le texte, tels les atomes qui composent les molécules, et de constituer un texte c'est-à-dire de distribuer, d'organiser ces unités afin d'atteindre certaines idées, comme une molécule qui possède certaines propriétés en raison de sa structure.

Plusieurs approches de représentation dans cet espace sont proposées dans la littérature. Nous détaillons par la suite ces différentes représentations ***[1]***.

II. Prétraitement de textes

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) ***[1]***.

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 ;**

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-gramme ***[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…

Ou en **Anglais**: Les prépositions (about, after, though…), les déterminants (the, no, one…), les conjonctions (though, and, or.), les adverbes (above, almost, yet…), les pronoms (who, another, few…) et certains verbes (are, can, have, May, will…).

Et en **Arabe** حروف الجر، حروف العطف، أسماء الإشارة، أخوات كان، أخوات إن، الخ...

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 quasitotalité des documents et ont donc un pouvoir discriminant faible en comparaison avec d'autres termes.

D'après la loi de Zipf Leur élimination lors d'un prétraitement 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 ***[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 » ***[1]***.

4. Le traitement morphologique

L'analyse morphologique signifie la manière dont les mots sont construits et quels sont leurs rôles dans la phrase. En pratique, dans le cadre du traitement automatique de la langue, elle consiste à :

- **segmentation et tokenisation ;**
- **Lemmatisation ;**
- **Racinisation(Stemming) ;**

4.1- Segmentation

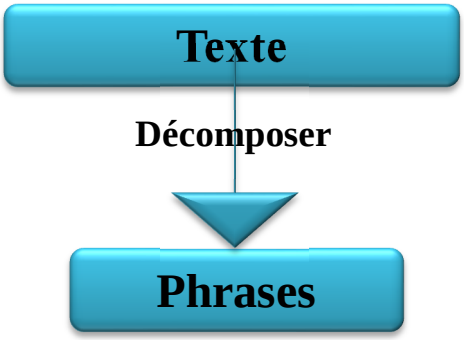

Figure 2.1 : Segmentation de texte

4.2- Tokenisation / Tokens

Figure 2.2 : Tokenisation de texte

4.3- Lemmatisation

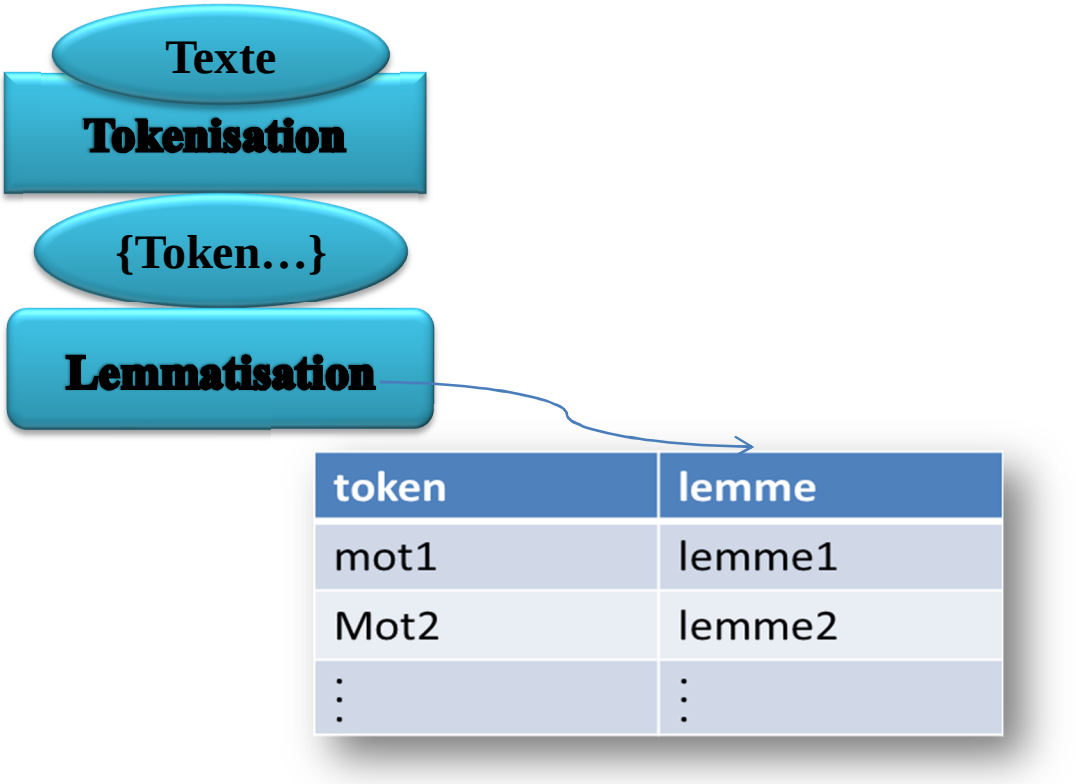

Figure 2.3: Lemmatisation des mots

4.4- Racinisation

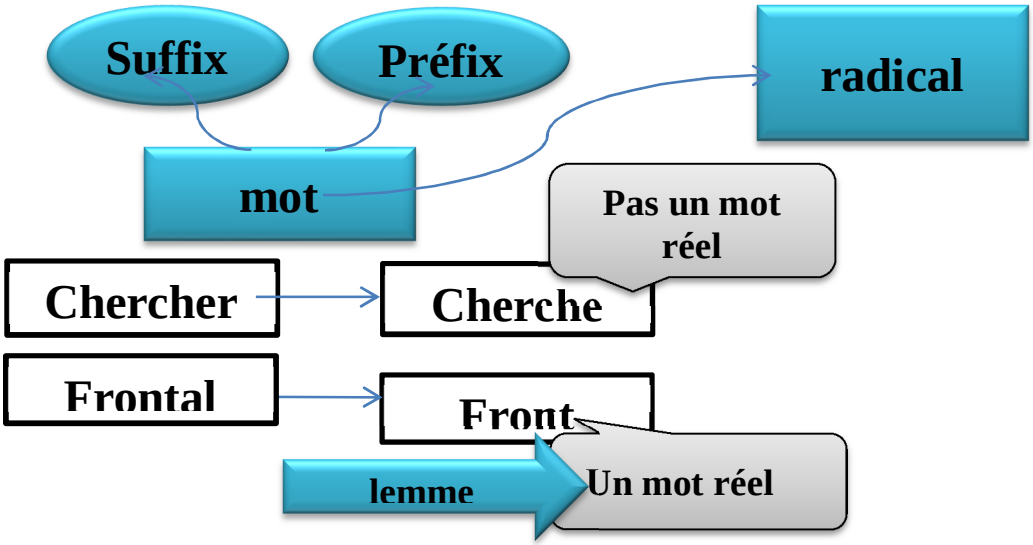

Figure 2.4 : Racinisation des mots

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.

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.

La Figure ci-dessus : présenter les différentes étapes de prétraitement de textes

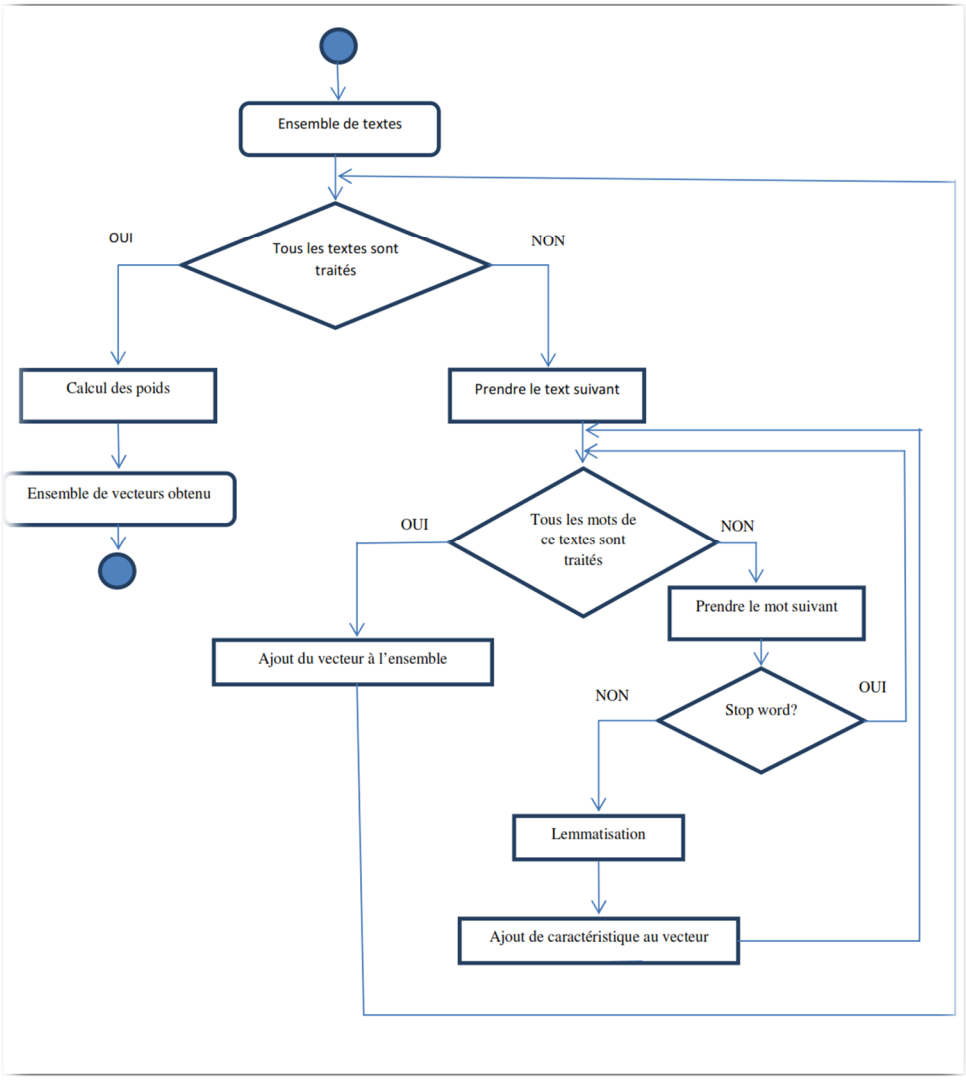

Figure 2.5: Algorithme de prétraitement

III. Définition de descripteurs

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-grammes, etc…

Dans la section suivante, nous allons définir les différentes sortes de termes, utilisés dans la littérature, pour la représentation d'un document texte ***[1]***.

1. Représentation de textes

1.1- **Représentation en « sac de mots** »

Le choix des mots comme descripteurs d'un document c'est le choix le plus intuitive, ainsi un texte sera représenté dans l'espace des mots par un vecteur dont chaque composante correspond au nombre d'apparition d'un mot dans le document, cette représentation est connue par « sac de mots » , « bag of words » .

Pour clarifier la notion de mot, Y. Gilly dans son ouvrage « Texte et fréquence » ***[15]*** l'a considéré comme étant une séquence de caractères appartenant à un dictionnaire, ou formellement, comme étant une suite de caractères séparés par des espaces ou des caractères de ponctuations (Cette définition n'est pas valable pour toutes les langues).

Pour y remédier, un prétraitement linguistique amené par l'application des procédures de lemmatisation et de stemming, avant la représentation des documents, est indispensable.

1.2- Représentation des textes par des collocations

Cette approche proposée ici, consiste à regrouper certains mots (collocations) afin d'obtenir des descripteurs ou expressions plus porteurs de sens au lieu d'utiliser des mots isolés composant le texte.

Identifier des collocations consiste à trouver des mots qui "vont ensemble" et qu'il est naturel de trouver proches dans le langage. Pour former ces groupes de mots, on n'a pas besoin de syntagmes nominaux, juste des paires de mots qui peuvent être séparés par des mots vides, Le but n'est pas ici de chercher à analyser les textes d'un point de vue syntaxique, mais les représenter selon un ensemble d'usages de la langue, qui ont une

influence sur le système classification. (Par exemple : repas-bien-garni, parler-en-connaissance-de-cause, tout-à-fait- normal) ***[1]***.

1.3- Représentation des textes par des phrases

Ces techniques sont venues pour remédier à la déstructuration syntaxique causée par la représentation en **''sacs de mots**".

Les résultats fournis par ce type de représentation **«** sac de mots **»** se basent finalement sur des mots éparpillés composant des textes qui sont très éloignés de ceux qu'ils sont censés représenter. Mais les techniques de traitement statistique **(**Bayes, Markov...**)** s'apprêtent mal des représentations à partir de phrases en raison de leurs caractéristiques irrégulières et exceptionnelles **(**longueur, redondance, bruit, structure compliquée...**).** Beaucoup de chercheurs s'y sont cassé les dents ayant abouti souvent à des solutions dérivées plus simples.

Logiquement, une telle représentation doit obtenir de meilleurs résultats que ceux obtenus via les mots en raison de la richesse sémantique de la phrase, cependant leurs propriétés statistiques ne permettent pas de définir des hypothèses statistiques fiables ***[16]*** car, le grand nombre d'assemblages possibles de mots engendre des faibles fréquences et trop aléatoires, ne permettant pas d'approximer le risque réel de manière correcte grâce au risque empirique.

L'utilisation de **''sac de phrases''** entraîne évidemment un problème de taille **(**pour n mots il y existe potentiellement $\mathbf{n^k}$ combinaisons de longueur **k).**

Pour y remédier, on ne considère pas toutes les séquences possibles mais on tente d'effectuer une sélection des phrases, en privilégiant celles qui sont sémantiquement riche. Dans la phrase

"Le gentil lapin orange mange la carotte bleue" par exemple, on peut dire que des séquences comme "gentil lapin orange", "carotte bleue", "lapin orange", ... sont porteuses de sens. Alors que les séquences "orange mange", "le gentil"…etc. sont insignifiantes.

Une autre approche de ***[17]*** qui proposent d'utiliser des phrases statistiques comme descripteurs au lieu des phrases grammaticales qui ont amélioré considérablement la performance du classifieur. Une phrase statistique est définie par ***[18]***, comme une

collection de mots adjacents (mais pas forcément classés) qui apparaissent ensembles mais qui ne respectent pas forcement les règles grammaticales.

Une autre étude menée par ***[19]***, démontre que ce type de représentation améliore la qualité des résultats par rapport aux méthodes de type «sac de mots» lorsque les documents étudiés sont limités en nombre et taille. De plus, ce type de représentation présente de grandes variations dans la qualité de ses résultats en fonction du type de documents à classifier.

Toutefois, la représentation des textes par des phrases est un domaine dont les recherches restent toujours actives ***[1]***.

1.4- Représentation des textes avec des racines lexicales (stemming)

L'opération de stemming ou **la désuffixation** vise de 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 ***[1]***.

Le stemming est un traitement linguistique moins approfondie que la lemmatisation, ayant deux avantages : Plus 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 ***[20]***.

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.

Plusieurs stemmers ont été développés pour déterminer les racines lexicales, l'algorithme le plus couramment utilisé pour la langue anglaise est celle de **PORTER** ***[21]***.

La représentation des textes par ces stems peut apporter des résultats supérieurs à ceux obtenus par les lemmes (que nous allons voir dans ce qui suit), démontré et approuvé par de Loupy, et bien meilleur que le codage de type « sac de mots » ou chaque variation

d'un mot est considéré comme une nouvelle composante du vecteur. Alors, on peut facilement imaginer combien on va gagner en question de dimensionnalité en optant pour les stems comme descripteurs ***[1]***.

1.5- Représentation des textes avec des lemmes (lemmatisation)

La lemmatisation consiste à utiliser l'analyse grammaticale afin de remplacer les verbes par leur forme infinitive et les noms par leur forme au singulier ***[1]***.

Elle 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 ***[1]***. Un algorithme efficace, nommé *TreeTagger* ***[22]*** 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 ***[1]***.

1.6- Représentation des textes avec la méthode des n-grammes

Une autre approche pour coder les documents émerge : les n-grammes ***[23]*** On définit un n-gramme (n-gram) par est une séquence de n caractères : bi-grammes pour n=2, tri-grammes pour n=3, quadri-grammes pour n=4, etc… On n'a plus besoin de chercher les délimiteurs(les espaces ou les caractères de ponctuations) comme c'était le cas pour les mots. Quelques auteurs admettent les n-grammes comme une chaine non ordonnée de caractères; par exemple un tri - grammes peut être constitué du 2éme, 4éme et 1ér caractère, d'autres auteurs n'autorisent pas ce désordre. Pour notre cas, on va admettre qu'un n-grammes désignera une chaîne de n caractères consécutifs.

Pour un texte quelconque, les n-grammes correspondants sont générés en faisant déplaçer un masque de n caractères sur tout le texte. Ce déplacement s'effectue caractère par caractère, à chaque déplacement la séquence de n caractères est enregistrée, l'ensemble de ces séquences constitue l'ensemble des n-grammes représentant le texte ***[24]***.

Par exemple, pour générer les 3-grammes de la phrase "Tu es libre", on obtient :

"Tu ", "u_e", "_es", "es_", "s_l", "_li", "lib", "ibr", "bre".

Dans cet exemple, l'espace est représenté par le caractère "_", pour faciliter la lecture. Historiquement, les n-grammes étaient conçus pour la reconnaissance de la parole, pour prédire l'apparition de certains caractères en fonction des autres caractères mais par la suite, le concept n-grammes a été bénéfique, pour le domaine de recherche d'information et la classification de textes, avec plusieurs travaux qui ont démontré que cette segmentation ne faisait pas perdre d'information.

Les avantages que présentent les techniques qui s'appuient sur les N-grammes de caractères sont:

- Les N-grammes permettent de capturer automatiquement la racine des mots les plus fréquents. Il n'est pas nécessaire d'appliquer une étape de recherche de racine et/ou de lemmatisation.
- Ces descripteurs sont indépendants de la langue employée dans le corpus. Il n'est pas nécessaire d'utiliser des dictionnaires, ni de segmenter les documents en mots.
- Les N-grammes sont tolérants aux fautes d'orthographe et aux déformations causées lors de la reconnaissance de documents (système OCR). Lorsqu'un document est reconnu à l'aide du système OCR il y a souvent une part non négligeable de bruit.

Par exemple, il est possible que le mot "feuille" soit lu "teuille". Un système fondé sur les N-grammes prendra en compte les autres n-grammes comme "eui", "uil", etc… ***[25]***.

1.7- Représentation des textes par des combinaisons de termes

Au lieu de prendre les termes un par un comme descripteurs, l'idée ici est de combiner linéairement des termes pour améliorer la qualité des résultats. L'intérêt est corriger les anomalies liés aux ambigüités et redondances du vocabulaire en combinant plusieurs termes pour avoir des nouvelles variables artificielles, jouant le rôle de nouveaux «termes» ***[18]***.

2. Sélection de descripteurs

2.1- Besoin de la sélection de descripteurs

Pour une problématique de classification, l'ensemble des descripteurs est constitué de l'ensemble des termes du corpus, un terme pouvant être un mot, un stem ou un n-gramme, etc.., ce qui peut représenter plusieurs centaines de milliers de termes, même après

les prétraitements appliquées dans la première phase qui ont procédé à l'élimination des mots les plus fréquents et les plus rares, soit parce qu'ils n'étaient pas discriminants (Mots vides très faiblement informatifs), soit parce qu'ils n'étaient pas exploitables statistiquement (très faible fréquence), le nombre de termes s'avère encore très élevée (De quelques dizaines de milliers à plusieurs centaines de milliers de termes). Parmi l'ensemble des descripteurs restants, on peut penser que tous ne sont pas nécessairement discriminants voire nuisible pour le système.

Ainsi, Il est nécessaire de diminuer davantage et choisir les descripteurs les plus appropriés (ceux qui assureraient les meilleures performances au classifieur), qui vont être utilisés comme vecteurs d'entrées avant de pouvoir utiliser un modèle d'apprentissage.

La sélection de descripteurs est un des principaux enjeux du processus, puisque du choix des descripteurs et de la connaissance précise de la population va dépendre la mise au point du classifieur. L'information nécessaire à la construction d'un bon modèle de prédiction peut être disponible dans les vecteurs d'entrées mais une sélection inappropriée de descripteurs ou d'exemples d'apprentissage peut faire échouer l'opération.

Evidemment, que quel que soit le modèle statistique utilisé ultérieurement, si la représentation des textes n'inclut pas certains descripteurs ou si les descripteurs retenus sont trop nombreux ou mal choisis, le classifieur aura des performances médiocres.

Les entrées non discriminantes doivent être supprimées pour deux raisons différentes :

- Pour réduire le temps de calcul : Plus le nombre d'entrées est grand, plus le nombre de paramètres à déterminer est élevé, ce nombre intervient dans l'expression de la complexité de l'algorithme qui va exiger un temps de traitement plus important. (Pour les modèles tels que les réseaux de neurones, le nombre de poids du réseau croît linéairement avec le nombre de descripteurs utilisés en entrée du modèle).
- Pour diminuer le sur-apprentissage : Comme les bases d'apprentissage sont limitées, des associations inattendues peuvent apparaître entre des descripteurs non informatifs et des classes ; elles peuvent avoir une influence négative sur la qualité du modèle. Il faut alors disposer d'une base d'exemples plus grande afin de diminuer le sur-apprentissage résultant du nombre trop important de paramètres, dont certains sont de très faible fréquence, par rapport aux textes du corpus

d'apprentissage : on ne peut pas construire des règles stables à partir de quelques apparitions d'un terme dans l'ensemble d'apprentissage.

Le sur-apprentissage dépend aussi beaucoup du modèle d'apprentissage utilisé, en effet certains sont capables de sélectionner les termes informatifs et ne sont pas affectés par un pléthore d'informations inutiles alors que d'autres considèrent que tous les termes sont discriminants, une sélection préalable est donc indispensable.

Les méthodes de sélection de descripteurs ont donc pour but de choisir parmi un ensemble de descripteurs possibles, les descripteurs les plus importants, c'est-à-dire ceux qui vont permettre d'obtenir de bonnes performances sur une base différente de la base d'apprentissage ***[1]***.

2.2- Le nombre de descripteurs conservés

Les méthodes de sélection de descripteurs fournissent, en général, une liste de descripteurs classées par degré d'importance, cette notion d'importance qui dépend de la méthode de classement; l'intérêt des différentes approches de réduction de dimensionnalité est d'avoir un ensemble de descripteurs plus réduit mais informatif. Il reste ensuite à fixer le nombre de descripteurs à garder dans cet ensemble ***[1]***.

La méthode de classification va être forcément, très décisive dans le seuil à fixer pour le nombre de descripteurs à conserver. Comme par exemple, dans un réseau de neurones, réduire la dimension des vecteurs d'entrées est très recommandé alors qu'une approche SVM est capable de traiter des listes plus longues de termes.

Nous cherchons donc, à supprimer des termes de la représentation des textes, tout en sachant que chaque suppression de terme entraîne une perte d'information ; il faut trouver le bon compromis entre, d'une part, la nécessité de réduire l'espace des descripteurs avec moins de redondances possibles et, d'autre part, le nécessité de garder suffisamment d'informations.

Plusieurs auteurs, dans leurs travaux, ont proposé différents nombres de descripteurs pour représenter les textes, de 180 à 100 jusqu' aux 20 premiers descripteurs, sans atteindre les qualités des classifieurs. Une synthèse des différentes expérimentations portantes sur le nombre adéquat de descripteurs retenus, qui n'impliquent pas qu'une grande dimension est nécessaire pour avoir des meilleurs résultats, est évoquée dans ***[18]***.

2.3- Les méthodes de sélection de descripteurs

La majorité des méthodes sont basées sur le calcul d'une statistique pour chaque terme qui représente son importance pour le document où il figure ou pour le corpus complet, puis à sélectionner les termes les plus importants. Il existe plusieurs formules statistiques pour mesurer la quantité d'information apportée à partir du nombre d'apparitions du terme dans la classe et hors de la classe.

Dans cette phase, il s'agit aussi de générer un profil pour chaque catégorie. Le profil d'une catégorie doit contenir tous les termes qui caractérisent cette catégorie par rapport aux autres. Pour mesurer ces statistiques et construire ces profils, il est nécessaire d'utiliser une méthode de sélection de termes. En effet, il existe plusieurs méthodes de sélection de termes.

Dans ce qui suit nous allons présenter la principale formule utilisée pour mesurer la quantité d'informations contenue dans les termes pour les documents ou les classes, dont les performances sont comparées dans plusieurs études.

- **La Fréquence-document (Document Fréquency) ;**

Une méthode de sélection qui peut être considérée comme une méthode de prétraitement approfondie : elle est très simple puisqu'elle correspond simplement au pourcentage de documents dans lesquels le terme apparaît, cette méthode conduit à supprimer les termes très fréquents et très rares afin de conserver les mots les plus importants avec le risque de supprimer des termes très riches et informatifs pour le système. Pour écarter les mots les plus fréquents, nous fixons un seuil maximal de fréquence n'autorisant pas de sélectionner les termes présents dans une très forte proportion de textes (ex : un terme qui apparaisse dans 180 textes d'un corpus de 200 textes n'est pas sélectionné), de même un seuil minimal est fixé pour éliminer les termes très rares (ex : un terme qui a moins de 5 apparitions dans tout le corpus n'est pas sélectionné) ***[1]***.

2.3.1- Sélection des termes par rapport la classe ou tout le corpus

la réduction des dimensions est peut être localement ou globalement :

- Réduction locale : Chaque classe est caractérisée par un profil composé d'un ensemble de termes, et chaque texte sera représenté par une liste de termes dépendante de la catégorie.

- Réduction globale : Contrairement au cas précédent, un texte est représenté par une seule liste de termes dans tous le corpus indépendamment des classes.

3. Traitement numérique : Pondération (ou calcul de poids)

Le tableau (termes x documents) est constitué par le nombre d'apparitions du terme dans le document du corpus. Cette information de base doit être pondérée en fonction de divers paramètres liés au document lui-même (ex : le nombre de termes par document) ou au corpus en intégralité (ex : le nombre de termes du corpus). L'intérêt de cette pondération est mieux exploiter l'information contenue dans le document pour améliorer les performances d'un système de classification de textes ***[1]***.

Plusieurs systèmes de pondération ont été développés dans la littérature, qui se reposent, tous sur les deux hypothèses suivantes :

- Plus le nombre d'apparitions d'un terme dans un texte est important, plus ce terme est discriminant pour la classe associée.
- Plus le nombre d'apparitions d'un terme dans le corpus est important, alors moins ce terme peut discriminer les textes.

Voici, un petit aperçu sur les pondérations les plus habituellement utilisées :

- Commençant par le choix le plus simple, qui ne s'intéresse que sur la présence ou la non- présence d'un terme dans le texte, il consiste à utiliser une pondération binaire : **1** si le terme est présent une ou plusieurs fois dans le document, **0** dans le cas contraire .Cette représentation binaire est historiquement la plus ancienne et la plus simple. Néanmoins, cette fonction est moins utilisée pour les méthodes statistiques, car ce codage supprime de l'information qui peut être utile : l'apparition du même mot plusieurs fois dans un texte peut constituer un élément de décision important.
- Les représentations fréquentielles sont aujourd'hui les plus utilisées et sur lesquelles notre étude va être basée. Plusieurs variantes s'illustrent :
  - Une première approche consiste à utiliser seulement le nombre d'occurrences des termes. Cette pondération ne peut être valable que dans les documents de même taille, sinon elle avantage les termes qui se répètent souvent dans les documents les plus longs.

- Une autre approche simple consiste à utiliser la fréquence d'un terme par rapport au nombre de termes composant le texte.
- La pondération TF x IDF corrige la fréquence du terme (Term Frequency) en fonction de sa fréquence dans le corpus (Inverse Document Frequency). La correction se fait en multipliant du rapport des n textes du corpus sur le nombre de documents contenant le terme. Le logarithme est utilisé pour lisser les résultats.
- Le TFC normalise le TF x IDF en fonction de l'ensemble des termes du document.

D'autres pondérations sont en pleine étude :

- La représentation séquentielle n'a fait l'objet de travaux que récemment car elle nécessite l'utilisation de modèles plus complexes et que son intérêt n'est pas toujours démontré.

Cependant, c'est une représentation naturelle qui permet de conserver l'ordre des mots d'un document.

- D'autres représentations plus riches existent notamment dans le domaine du Traitement de la Langue Naturelle (TALN) comme par exemple les : représentations qui prennent en compte le rôle du mot dans une phrase (Nom, Verbe, Sujet, etc…). Ces représentations s'avèrent efficaces, cependant, les modèles qui les utilisent sont peu performants de ceux qui travaillent dans des espaces plus « simples ». Ces dernières ne sont pas présentées ici.

Un processus de classification automatique de textes employant de méthode essentiellement statistiques, peut-être représenté par un modèle vectoriel.

3.1- Le modèle vectoriel

3.1.1- Représentation binaire

Historiquement, la première représentation d'un texte était sous forme de vecteur binaire, et malgré l'apparition de nouvelles formules pour pondérer les termes, cette façon de représenter un document, est restée toujours largement utilisée en raison du bon compromis fourni entre performance et complexité. Effectivement, en raison de sa simplicité, son temps de traitement est faible et en contrepartie ses résultats ne sont pas mauvais. Elle est appelée représentation « par mots clés ». La méthode consiste à transformer le texte en un vecteur dont les éléments

renseignent sur la présence (valeur égale à 1) ou l'absence (valeur égale à 0) d'un terme dans le texte. Deux exemples de textes sont indiqués dans la figure 2.8, leurs vecteurs binaires correspondants sont représentés dans le tableau 2.1.

« Ce groupe est le groupe de la mort du mondial 2010»

Document 1

« Cette équipe est une équipe Favorite au mondial 2010»

Document 2

Figure 2.6: Deux exemples de documents

Cette façon de représenter un texte, est peu informative car elle ne donne pas les informations nécessaires ni sur les occurrences d'un terme dans le document qui peut être une information importante pour l'opération de classification, ni sur la longueur du texte.

3.1.2- Représentation fréquentielle

Cette représentation consiste à présenter le texte sous forme de vecteur dont les éléments renseignent non seulement sur la présence ou l'absence d'un terme comme dans un vecteur binaire mais aussi informe sur le nombre de présences du terme dans le texte.

Ainsi, un document est transformé en un vecteur dont les composantes vont correspondre au nombre d'occurrences des termes dans le document.

Pour chaque document, un poids est attribué à chacun des termes qu'il contient. Une matrice «**documents /termes**» représente l'ensemble des documents (un vecteur est associé à chaque document, les composantes des vecteurs sont les poids des termes).

Cette méthode conçoit le calcul du poids proprement dit des termes. Aucune analyse linguistique n'est utilisée, ni aucune notion de distances entre les mots.

Ainsi les deux inconvénients majeurs de cette représentation, sont la non prise en charge des interactions des termes entre eux traduit par une indépendance de ces termes d'une part et

d'autre part la déstructuration syntaxique du document causée par le fait que le modèle ne permet pas de conserver l'ordre des mots.

3.1.3- Représentation fréquentielle normalisée

Du point de vue statistique**,** la représentation fréquentielle confronte un problème majeur du fait qu'un texte long sera représenté par un vecteur dont la norme sera supérieure à celle de la représentation d'un document plus court. Il est donc recommandé de normaliser la représentation fréquentielle par rapport à la taille du document. Ainsi le poids du terme sera le nombre d'occurrences de ce dernier dans le texte sur le nombre d'occurrences de tous les termes du texte. Le tableau 1 contient une représentation de ce type.

3.1.4- Représentation TF-IDF

Dans le but d'avoir des représentations plus riches en informations que la représentation fréquentielle basique ou même sa version normalisée, une autre variante des représentations vectorielles s'illustre appelé le codage TF-IDF. Cette représentation se base principalement sur une certaine loi appelée loi de Zipf qui montre la façon dont les mots sont distribués dans un corpus.

3.1.4.1- Loi de Zipf

La répartition des fréquences des termes dans un corpus a été étudiée empiriquement par Zipf. Zipf est parti d'un principe général avant qu'il énonce cette loi mathématiquement. Cette loi réaffirme que la distribution des occurrences des mots dans un corpus donné n'est pas uniforme, certains mots apparaissent très fréquemment, tandis que d'autres apparaissent très rarement.

Le codage TF.IDF a été introduit pour la prise en compte de la loi de Zipf dans le cadre du modèle vectoriel et qui donne parfois son nom à la méthode vectorielle. Le principe de base est que l'élément du vecteur représentant un texte se calcule en multipliant un facteur qui concerne l'importance du terme T dans le texte avec un autre qui concerne l'importance de ce terme dans tout le corpus :

$$L_{Zipf}(T) = \text{Poids dans le document} * \text{Poids dans le corpus.}$$

Donc la formulation de la pondération s'appuie sur deux notions

- La Fréquence du Terme (ou Term Frequency : TF) qui prend en compte le nombre d'occurrences du terme dans le document.

TF(t, D) = fréquence du terme t dans le document D.

- et l'Inverse de la Fréquence en Document (ou Inverse Document Frequency : IDF) qui prend en compte le nombre d'occurrence du terme dans le corpus.

Ces deux notions sont combinées multiplicativement de façon à attribuer un poids d'autant plus fort que le terme apparaît souvent dans le document et rarement dans le corpus complet.

Tf.idf (t, D) = TF(t, D)* IDF(t) ;

Tf.idf = Term Frequency * Inverse Document Frequency;

$$TF\_IDF = TF \times IDF$$

$$\text{Où } IDF = \log\left(\frac{\text{nombre total de documents}}{\text{nombre de documents contenant le mot}}\right)$$

t : terme ; D : Document .

	Doc	Do	Doc	Do	Doc1	Doc	D	Doc	Doc2
Ce	1	0	1	0	0.09	0	1	0.0	0
Cette	0	1	0	1	0	0.1	1	0	0.11
Groupe	1	0	2	0	0.18	0	2	0.0	0
Equipe	0	1	0	2	0	0.2	2	0	0.11
Le	1	0	1	0	0.09	0	1	0.0	0
Est	1	1	1	1	0.09	0.1	2	0.04	0.055
De	1	0	1	0	0.09	0	1	0.0	0
La	1	0	1	0	0.09	0	1	0.0	0
Une	0	1	0	1	0	0.1	1	0	0.11
Mort	1	0	1	0	0.09	0	1	0.0	0
Favorit	0	1	0	1	0	0.1	1	0	0.11
Du	1	0	1	0	0.09	0	1	0.0	0
Au	0	1	0	1	0	0.1	1	0	0.11
Mondia	1	1	1	1	0.09	0.1	2	0.04	0.055
2010	1	1	1	1	0.09	0.1	2	0.04	0.055
Vocabulaire	**Vecteur binaire**		**Vecteur Fréquent**		**Vecteur Fréquentiel**		**DF et Vecteur TF_ IDF**		

Tableau 2.1 : Représentation vectorielles des documents de la figure 2.8

Pour conclure (Salton & Buckley, 1988) et (Joachims, 1999) confirment que la représentation TF-IDF avec toutes ses variantes est la représentation la plus utilisée en recherche d'information aussi bien en recherche documentaire qu'en classification.

Conclusion

Pour pouvoir appliquer les différents algorithmes d'apprentissage sur les documents de type textuels, un ensemble de techniques ont été développé pour montrer comment l'information textuelle est habituellement prise en compte pour la représentation « informatique » de ces documents. Les différentes approches de représentation informatique de textes sont exposées dans ce chapitre.

Ainsi avant la codification des documents, un ensemble d'opérations préliminaires doivent être faites pour épurer le texte de tous les mots inutiles et conserver seulement ceux qui sont porteurs d'informations et utiles pour le processus de classification. Mais malgré tous les prétraitements appliqués sur le document, l'espace des descripteurs, qui peuvent être des n- grammes, des stems, des phrases, des concepts ou tout simplement des mots, reste très grand et très creux, d'où la nécessité d'une diminution préalable de cet espace.

Plusieurs techniques de sélection des descripteurs ou réduction de dimensionnalité sont proposées dans la littérature, une bonne partie de ces approches sont étalées dans ce chapitre. Une fois la liste des descripteurs arrêtés, un degré d'importance ou poids est attribué à tous les termes présents dans la représentation vectorielle puisque chaque terme possède un certain nombre d'occurrences dans le document ou dans le corpus qui est différents aux autres.

Finalement on peut qualifier notre texte, par fichier « informatique » apte à être employé dans les différentes méthodes d'apprentissage automatique.

La matrice TF_IDF est l'entrée de la phase apprentissage pour construire le modèle de la classification, le chapitre suivant illustre cette technique d'apprentissage(SVM).

Chapitre 3 : Classification à l'aide des SVMs

Introduction

La classification des documents est la tâche de regroupement des documents en catégories en fonction de leur contenu ; elle n'a jamais été aussi importante que ce qu'elle est aujourd'hui. La croissance exponentielle du volume de données non structurées associées à une augmentation marquée de règles litiges, de sécurité et de confidentialité ont laissé les organisations tout à fait incapable de faire face aux exigences contradictoires de l'entreprise, les avocats et les régulateurs. L'escalade des coûts et des risques, avec aucune fin en vue. Sans outils efficaces pour faciliter la classification automatisée basée sur le contenu du document, les organisations ont peu d'espoir de rattrapage du retard, encore moins prendre de l'avance. La technologie a créé le problème, et à la technologie incombe la nécessité d'y remédier. ***[26]***

La classification manuelle est hors de question en raison du volume des données, tandis que les approches naïves de recherche ont des résultats médiocres en raison de la complexité du langage humain. De nombreuses approches de pointe ont été proposées pour résoudre ce problème. Les méthodes de classification ont pour but d'identifier les classes auxquelles appartiennent des objets à partir de certains paramètres descriptifs. Elles s'appliquent à un grand nombre d'activités humaines et conviennent en particulier au problème de la prise de décision automatisée. La procédure de classification sera extraite automatiquement à partir d'un ensemble d'exemples. Un exemple consiste en la description d'un cas avec la classe correspondante. Un système d'apprentissage doit alors, à partir de cet ensemble d'exemples, extraire une procédure de classification ; il s'agit en effet d'extraire une règle générale à partir des données observées. La procédure générée devra classifier correctement les exemples de l'échantillon et avoir un bon pouvoir prédictif pour classifier correctement de nouvelles descriptions. Les méthodes utilisées pour la classification sont nombreuses, citons : la méthode des Séparateurs à Vastes Marges (SVM), les Réseaux de Neurones, NB, KPPv, etc…
Pendant ces dernières années, un intérêt particulier a été accordé aux Support Vector Machines (SVMs). Ces algorithmes d'apprentissage ont trouvé des domaines d'application très variés comme la reconnaissance de caractères manuscrits, la détection de visage, la détection de composants audio.

Néanmoins, l'utilisation des SVMs était limitée à un groupe de chercheurs car les algorithmes d'apprentissage standards étaient longs et difficiles à implémenter. Pour pallier à ce problème, John. C.Platt à mis au point un algorithme d'apprentissage pour SVMs appelé SMO (Sequential Minimal Optimization). Cet algorithme est généralement plus rapide, plus simple à implémenter et nécessite un espace mémoire réduit.

Nous présentons dans la suite de ce chapitre une étude détaillée des techniques SVM. Ces méthodes ont montré leurs efficacités dans de nombreux domaines d'applications tels que le traitement d'image, la catégorisation de textes et le diagnostic médical etc…
Le problème de classification peut être limité à l'examen du problème à deux classes sans perte de généralité. Dans ce but le problème est de séparer les deux classes par une fonction qui est induite à partir d'exemples disponibles.

I. Séparateurs à Vaste Marge (SVM)

Les machines à vecteurs de support (Support Vector Machine, SVM) appelés aussi séparateurs à vaste marge sont des techniques d'apprentissage supervisées destinées à résoudre des problèmes de classification. Les machines à vecteurs supports exploitent les concepts relatifs à la théorie de l'apprentissage statistique et à la théorie des bornes de Vapnik et Chervonenkis. La justification intuitive de cette méthode d'apprentissage est la suivante : si l'échantillon d'apprentissage est linéairement séparable, il semble naturel de séparer parfaitement les éléments des deux classes de telle sorte qu'ils soient le plus loin possibles de la frontière choisie. Ces fameuses machines ont été inventées en 1992 par Boser et al. Mais leur dénomination par SVM n'est apparue qu'en 1995 avec Cortes et al. Depuis lors, de nombreux développements ont été réalisés pour proposer des variantes traitant le cas non-linéaire. Le succès de cette méthode est justifié par les solides bases théoriques qui la soutiennent. Elles permettent d'aborder des problèmes très divers dont la classification. SVM est une méthode particulièrement bien adaptée pour traiter des données de très haute dimension.

1. Principe de la technique SVM

Le principe des SVM consiste à projeter les données de l'espace d'entrée (appartenant à deux classes différentes) non linéairement séparables dans un espace de plus grande dimension appelé espace de caractéristiques de façon à ce que les données deviennent linéairement séparables. Dans cet espace, on construit un hyperplan optimal séparant les classes tel que :

• Les vecteurs appartenant aux différentes classes se trouvent de différents côtés de l'hyperplan.

• La plus petite distance entre les vecteurs et l'hyperplan (la marge) soit maximale ***[27]***.

Le principe de la technique SVM est représenté dans la figure ci-dessous :

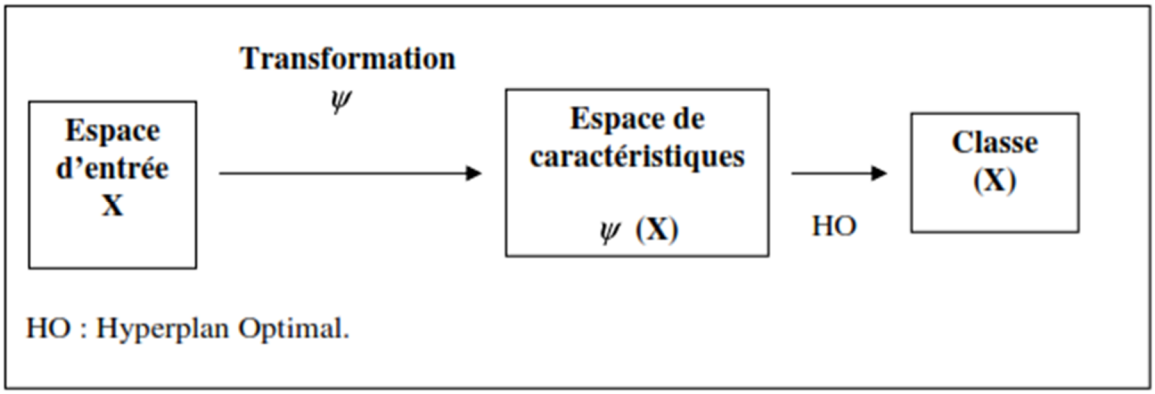

Figure 3.1**:** Principe de la technique SVM

1.1- Classifieur linéaire

Un classifieur est dit linéaire lorsqu'il est possible d'exprimer sa fonction de décision par une fonction linéaire en x. On peut exprimer une telle fonction par:

$$h(x) = \langle w, x \rangle + b = \sum_{i=1}^{n} w_i x_i + b$$

Où : W($\in \mathbf{R^n}$) est le vecteur de poids et b($\in \mathbf{R^0}$) le biais, alors que est la variable du Problème est l'espace d'entrée et qui correspond à R^n, où n est le nombre de composantes des vecteurs contenant les données. Notons que l'opérateur < > désigne le produit scalaire usuel dans R^n. w, b sont les paramètres à estimer de la fonction de décision.

- ✓ On cherche *h* sous forme d'une fonction linéaire : $h(\boldsymbol{x}) = w.\boldsymbol{x} + b$
- ✓ La ***surface de séparation*** est donc l'hyperplan : $w.\boldsymbol{x} + b = 0$
  - Elle est valide si $\forall i$ $u_i h(x_i) >= 0$

- L'hyperplan est dit sous **forme canonique** lorsque $\forall i \ u_i(w.x_i+b) \geq 1$

Ou encore $\min_i |w.x+b| = 1$

Pour décider à quelle catégorie un exemple estimé x' appartient, il suffit de prendre le signe de la fonction de décision **y= sign(h(x'))** , La fonction sign () est appelée classifieur.

Géométriquement (voir figure 1), cela revient à considérer un hyperplan qui est le lieu des points satisfaisant **<w, x> = 0**. En orientant l'hyperplan, la règle de décision correspond à observer de quel côté de l'hyperplan se trouve l'exemple x'.

On voit que le vecteur **w** définit la pente de l'hyperplan (w est perpendiculaire à l'hyperplan).

Le terme **b** quant à lui permet de translater l'hyperplan parallèlement à lui-même.

L'objectif de la discrimination linéaire est de trouver la bonne fonction de décision h(x).

La classe de tous les hyperplans qui en découle sera notée H.

La figure 3.1 représente l'hyperplan séparateur $w.x + b = 0$

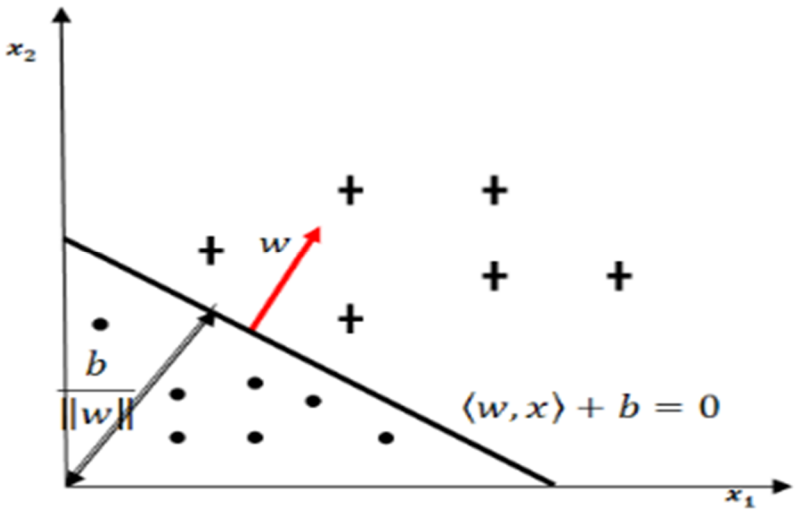

Figure 3.2: hyperplan séparateur

1.2- Marge maximale de l'hyperplan

Les SVM recherchent l'hyperplan dont la distance (appelée marge) minimale aux exemples d'apprentissage est maximale. Les exemples les plus proches de l'hyperplan s'appellent les vecteurs de support et la figure 3.2 illustre ça.

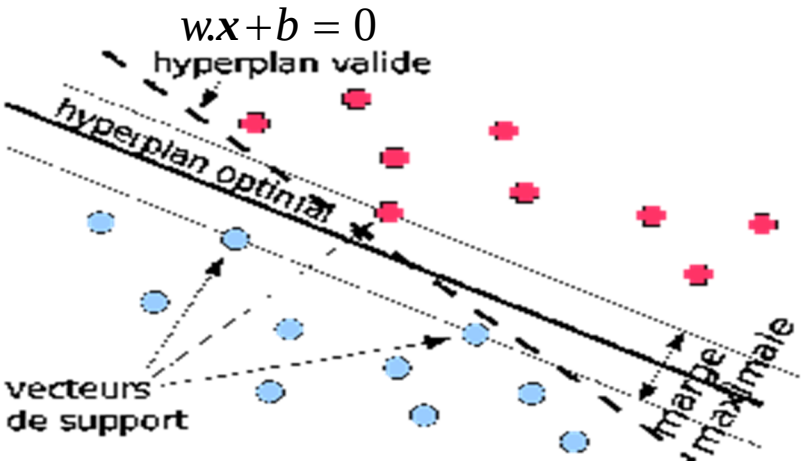

Figure 3.3 : La marge maximale de l'hyperplan

La marge est la distance entre la frontière de séparation et les échantillons

Figure 1 : hyperplan séparateur proches. Dans les SVM, la frontière de séparation est choisie comme celle qui maximise la marge.

- La distance d'un point à l'hyperplan est : $d(\boldsymbol{x}) = \frac{|\boldsymbol{w}.\boldsymbol{x} + w_0|}{\|\boldsymbol{w}\|}$

L'hyperplan optimal est celui pour lequel la distance aux points les plus proches (*marge*) est maximale. Cette distance vaut : $\frac{2}{\|\boldsymbol{w}\|}$

- Maximiser la marge revient donc à minimiser $\|\boldsymbol{w}\|$ sous contr $\begin{cases} \min \frac{1}{2}\|\boldsymbol{w}\|^2 \\ \forall \boldsymbol{i} \;\; \boldsymbol{u_i}(\boldsymbol{w}.\boldsymbol{x_i} + w_0) \geq 1 \end{cases}$

1.3- SVMs: un problème d'optimisation quadratique

- **problème primale**

- Il faut donc déterminer w et w_0 minimisant : $\eta(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w}\|^2$
  (Afin de maximiser le pouvoir de généralisation)

- sous les contraintes (hyperplan séparateur) $u_i[(\boldsymbol{w}.\boldsymbol{x}_i) + w_0] \geq 1, \quad i=1,\ldots,n$

- **Résolution de la forme primale du problème**

  Il faut régler $d + 1$ paramètres

- Possible quand d est assez petit avec des méthodes d'optimisation Quadratique
- Impossible quand d est grand (> qqs 10^3).

➢ **Transformation du problème d'optimisation**

o **Méthode des multiplicateurs de Lagrange**

$$\begin{cases} L(\boldsymbol{w}, w_0, \alpha) = \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^{l} \alpha_i \{(\boldsymbol{x}_i.\boldsymbol{w}+w_0)\boldsymbol{u}_i - 1\} \\ \forall i \ \ \alpha_i \geq 0 \end{cases}$$

➢ **Problème dual**

$$\begin{cases} \max_{\alpha} \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j u_i u_j (\boldsymbol{x}_i.\boldsymbol{x}_j) \\ \forall i \ \ \alpha_i \geq 0 \\ \sum_{i=1}^{l} \alpha_i u_i = 0 \end{cases}$$

- **Propriétés de la forme duale**

La conversion est possible car les fonctions de coût et les contraintes sont strictement convexes (*Th. de Kuhn-Tucker*).

La complexité du problème d'optimisation est

- ∞ *m* (tailles de l'échantillon d'apprentissage)
- et non ∞ *d* (taille de l'espace d'entrée X)
- Possible d'obtenir des solutions pour des problèmes impliquant ≈ 10^5 exemples.

➢ Solution du problème d'optimisation

$$\begin{cases} D(\boldsymbol{x}) = (\boldsymbol{w}^*.\boldsymbol{x} + b^*) \\ \boldsymbol{w}^* = \sum_{i=1}^{m} \alpha_i^* y_i \boldsymbol{x}_i \\ w_0^* = y_s - \sum_{i=1}^{m} \alpha_i^* y_i (\boldsymbol{x}_i.\boldsymbol{x}_s) \end{cases}$$

*** : estimé**

(x_s, y_s) étant n'importe quel point de support

Conclusion

Dans ce chapitre nous avons présenté les concepts de la méthode SVM, dont nous avons expliqué la démarche de construction de l'hyperplan optimal dans les cas des données linéairement séparables et non linéairement séparables ; en se basant sur les fondements mathématiques pour le cas binaire (à deux classes).

L'intérêt principal de la méthode SVM est sa facilité d'emploi, la mise en œuvre des algorithmes des SVM est peu coûteuse, mais le point noir est la durée des phases de tests qui sont assez longues lorsqu'il s'agit de régler les paramètres de la fonction noyau. Parmi d'autres avantages des SVM on cite : qu'ils possèdent des fondements théoriques solides, une optimisation très simple, et un temps de calcul court .
Les SVM peuvent également s'utiliser pour des tâches de prédiction des variables continues en fonction d'autres variables (température, évolution des marchés, etc.). Le champ d'application des SVM est donc large. L'implémentation de la classification automatique de documents textuels par les SVMs est illustrée dans le dernier chapitre.

Chapitre 4: Implémentation et Réalisation

Introduction

La bonne conception représente un facteur déterminant de la réussite d'un projet. C'est pourquoi il est nécessaire de suivre une démarche qui doit être rigoureuse et progressive.

Dans le chapitre précédent de ce mémoire, nous avons présenté une technique d'apprentissage automatique supervisé SVM (Support a Vaste Marge).

Notre projet consiste donc en l'étude, la conception et le développement d'une application de la classification automatique de documents numérisés par les SVMs.

Nous présenterons alors, dans la première partie de ce chapitre, notre environnement de développement avec les technologies utilisées ainsi que les raisons de nos choix, ainsi que les principales interfaces qui le composent à travers des fenêtres de capture.

I. Difficultés rencontrées

Lors de l'élaboration de notre projet (surtout au moment de l'implémentation), nous avons rencontré pas mal de difficultés. La principale difficulté était l'apprentissage de la programmation avec NetBeans. En effet, nous n'avions jamais été confrontés à ce logiciel avant. Aussi la maitrise des techniques comme le **Standford CoreNlp** (pour le prétraitement de nos documents textuels) et l'**opencv** (pour la classification par les SVMs) n'a pas été du tout simple.

II. Environnement de développement

Pour réaliser notre application qui fonctionne selon un environnement (où le niveau d'application est représenté par le NetBeans).

1. NetBeans [8.0.2]

1.1- Présentation de NetBeans

NetBeans est un EDI (Environnement de Développement Intégré, ou IDE en anglais pour Integrated Development Environment) libre pour JAVA. Il est utilisé avec un kit de développement JDK.

Tout comme beaucoup d'EDI (genre Microsoft Visual C++, Borland C++ Builder, Kdevelop, etc), c'est un logiciel qui vous propose un éditeur de texte, de quoi créer et gérer des projets, compiler, débugger votre code source, tester votre programme, gérer les versions, etc.

1.2- Pourquoi utiliser la plate-forme NetBeans ?

NetBeans permet aux programmeurs une grande facilité de développement. Il présente plusieurs avantages parmi lesquels :

- Une large panoplie de fonctionnalités intégrées : xml, documentation, databases, tomcat,…
- Attractif pour travailler avec Java sur des applications de bureau, web et mobile.
- 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.
- Un historique local des dernières modifications.

1.3- Les caractéristiques fondamentales de NetBeans

NetBeans comprend toutes les caractéristiques d'un IDE moderne :

- Editeur en couleur ;
- Projets multi-langage ;
- Editeur graphique d'interfaces et de pages web ;
- Refactoring (remaniement de code) : permettant d'améliorer le logiciel ;
- Fonctionne sur différentes plates-formes : Windows, Linux, Solaris, Mac OS X, …

III. Travail réalisé

Le but du projet est la classification par l'apprentissage supervisé. Pour cela nous nous sommes basés sur SVM et nous avons implémenté un algorithme en deux parties: apprentissage et classification.

1. Implémentation SVM choisie

1.1- Préprocessing

Cette phase est la première phase de l'algorithme implémenté. Elle consiste à préparer les données sous un format que les fonctions d'apprentissage et de classification de Opencv comprennent. (**Opencv** est une librairie open source de traitement et analyse d'images et vidéos avec des interfaces pour les principaux langages de programmation C, C++, Java, C#, Python ...)

Ce format se présente comme suit :

```
<target> <Val>  <Val>   <Val>  <Val> …
```

Où

<target> correspond au label d'une catégorie (1 ou -1)

<Val> : le poids de la dimension correspondante.

Dans notre cas nous avons construit un dictionnaire de mots de tous les textes utilisés à la phase d'apprentissage. Les Val sont les poids correspondants calculés par TF et TF_IDF.

Le préprocessing est très important dans le processus de classification car la construction du modèle est basée sur les données préparées. En effet des données mal préparées risquent de donner un modèle non performant. Ainsi nous avons procédé à un nettoyage des textes pour lemmatiser les mots et supprimer les stop words et les symboles de ponctuations.

1.1.2- Elimination de stop words

Cette phase consiste à supprimer tous les mots standards dans un texte, ce sont des mots très

Communs et utilisés dans pratiquement tous les textes. Leur présence peut dégrader la performance de l'algorithme de classification en termes de coût et en termes de précision de la classification.

Nous pouvons prendre comme exemple pour l'anglais I, am, with, what, you …

Et pour le français il s'agit d'éliminer par exemple :

- Les pronoms pronominaux : il, elle, je, lui, ceux….
- Les auxiliaires : avoir, être et toutes leurs conjugaisons possibles
- Les opérateurs de conjonction : avec, et, ou, aux…
- L', c' …

Pour les deux versions française et anglaise nous avons écrit un petit programme qui permet de déterminer si un mot est un stop word ou pas en référant à un fichier qui contient la liste de tous les stop words.

1.1.3- Tokenization et Lemmatisation

La tokenization est l'extraction des mots d'un texte. La lemmatisation est la réduction d'un mot à sa forme la plus simple, par exemple transformer tous les verbes conjugués à l'infinitif.

Ce traitement permet de normaliser les mots qui dérivent de la même racine et de les traiter comme s'ils étaient le même afin de leurs attribuer un poids unique.

1.1.3.1- Traitement des textes Anglais

Pour les textes anglais, il existe plusieurs outils de TAL qui réalisent parfaitement ce traitement par exemple openNLP et CoreNLP. Dans notre programme nous avons utilisé CoreNLP. C'est un outil qui permet de parser, tokonizer, lemmatiser et d'extraire les entités nommés d'un texte. Il a été créé à l'université de Stanford [12].
Il permet de transformer un texte en un fichier XML de la forme suivante :

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet href="CoreNLP-to-HTML.xsl" type="text/xsl"?>
<root>
<document>
<sentences>
<sentence id="1">
…..
</sentences>
<coreference>
<mention representative="true">
```

<sentence>2</sentence>
<start>3</start>
<end>6</end>
<head>5</head>
</mention>

<mention>
<sentence>2</sentence>
<start>1</start>
<end>2</end>
<head>1</head>
</mention>
</coreference>
</coreference>
</document>
</root>

La partie sentences contient toutes les phrases du texte en entrée.

Chaque phrase est représentée dans une balise sentence comme suit

<tokens>
<token id="1">
<word>Stanford</word>
<lemma>Stanford</lemma>
<CharacterOffsetBegin>0</CharacterOffsetBegin>
<CharacterOffsetEnd>8</CharacterOffsetEnd>
<POS>NNP</POS>
<NER>ORGANIZATION</NER>
</token>
…
<parse>(ROOT (S (NP (NNP Stanford) (NNP University)) (VP (VBZ is) (VP (VBN located)
(PP (IN in)
(NP (NNP California))))) (. .)))
</parse>
<basic-dependencies>
<dep type="nn">

```
<governor idx="2">University</governor>
<dependent idx="1">Stanford</dependent>
</dep>
…
</basic-dépendances>
</tokens>
```

CoreNLP peut prendre plusieurs options pour générer une ou plusieurs parties du fichier XML. Ces options sont :

- **ssplit**: décomposer un texte en un ensemble de phrases.
- **Tokenize**: décomposer une phrase en mots.
- **Lemma**: transformer le token en sa racine.
- **parse**: construit la partie parse du fichier XML.
- **Pos**: construit la partie pos du fichier XML.
- **Ner**: construit la partie ner du fichier XML.
- **etc**…

Dans notre programme nous avons utilisé l'interface java du coreNLP. Cette interface permet de définir les options de coreNLP et fournit des méthodes pour parser le XML généré. Ainsi nous avons pu récupérer toutes les lemmes.

1.1.3.2- Traitement des textes français

Cette tâche était plus difficile car CoreNlp ne permet supporte pas la langue française par défaut.

Nous avons pu lui ajouter un fichier pour permettre la tokenization, mais pour la lemmatisation nous avons été obligés de chercher une solution en dehors de CoreNlp, cette solution consiste à parcourir un fichier contenant tous les mots français et les lemmes qui leur correspondent, ce fichier a été écrit par Anne Vilnat, professeure de l'université de Paris Sud.

1.1.4- Calcul des poids et préparation du vecteur caractéristique

C'est la dernière étape du préprocessing, nous avons implémentés les algorithmes TF et TF-IDF pour réaliser ce calcul, les résultats obtenus par TF-IDF étaient meilleurs donc nous l'avons considéré dans les tests finaux, nous avons choisi le design pattern « Factory » pour permettre de basculer facilement d'un algorithme à un autre et d'en rajouter d'autres. Pour TF-IDF le corpus de documents utilisé est celui des documents d'apprentissage, si dans le

texte à classer un mot n'est jamais apparu dans le corpus d'apprentissage nous lui attribuons un poids négatif et nous l'ignorons dans le processus de classification. Comme indiqué dans le chapitre 2, les vecteurs caractéristiques des textes sont composés des rangs de leurs lemmes dans le dictionnaire (construit également à partir de tous les lemmes du corpus d'apprentissage) accompagnés des poids correspondants. Pour les textes d'apprentissage, la valeur de target est égale à 1 quand le texte fait partie de la catégorie étudiée et -1 sinon, pour les autres textes cette notion n'est pas utilisée.

1.1.5- Exemple de préprocessing

Dans cette partie, nous donnons un exemple de trois documents qui illustrent le travail réalisé dans la partie préprocessing du programme, les trois textes sont en anglais, le premier et le deuxième appartiennent à la catégorie du Data Mining, alors que le troisième traite du domaine de la biologie.

Fenêtre Principale : Bienvenue

Figure 4.1 : Fenêtre Principale

Fenêtre d'Accueil :

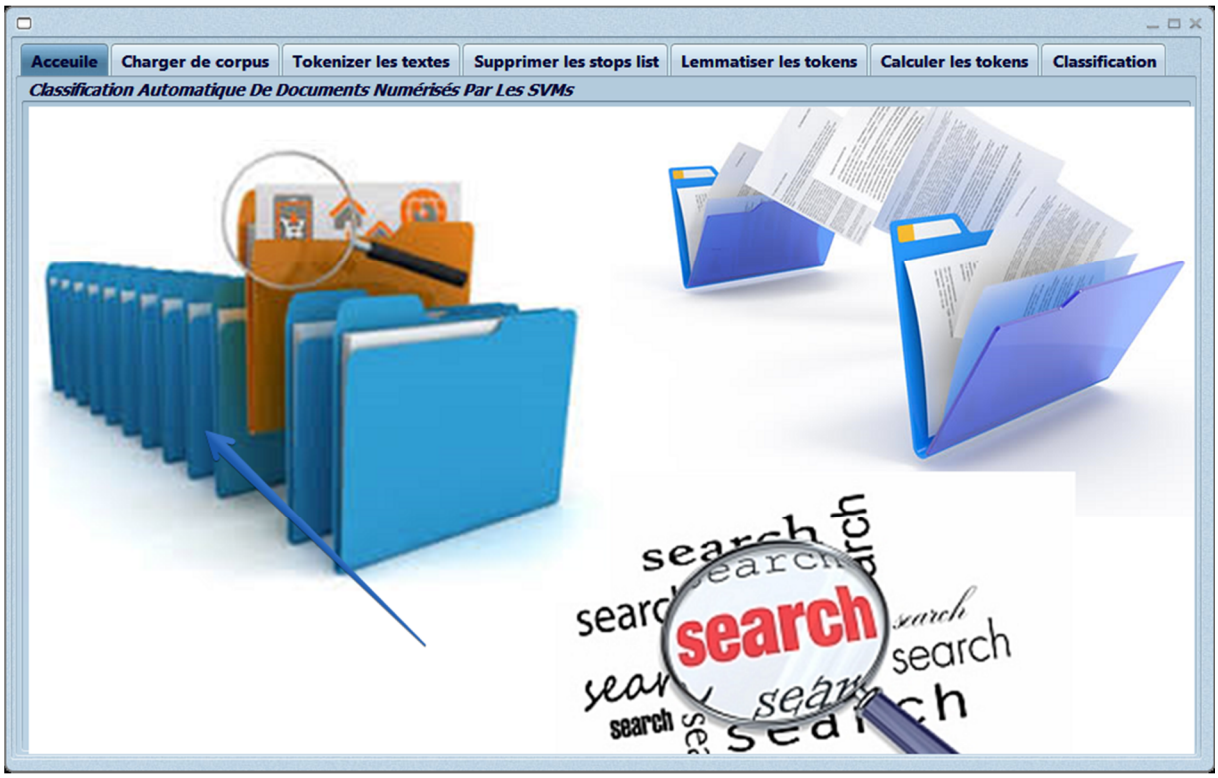

Figure 4.2: Fenêtre d'Accueil

Initialement les trois textes sont :

Document1

Data mining is a powerful new technology with great potential to help companies focus on the most important information in the data they have collected about the behavior of their customers and potential customers.

Document2

It discovers information within the data that queries and reports can't effectively reveal.

Document 3

Biology is a natural science concerned with the study of life and living organisms.

Fenêtre 1 : chargement d'un corpus de documents (.txt)

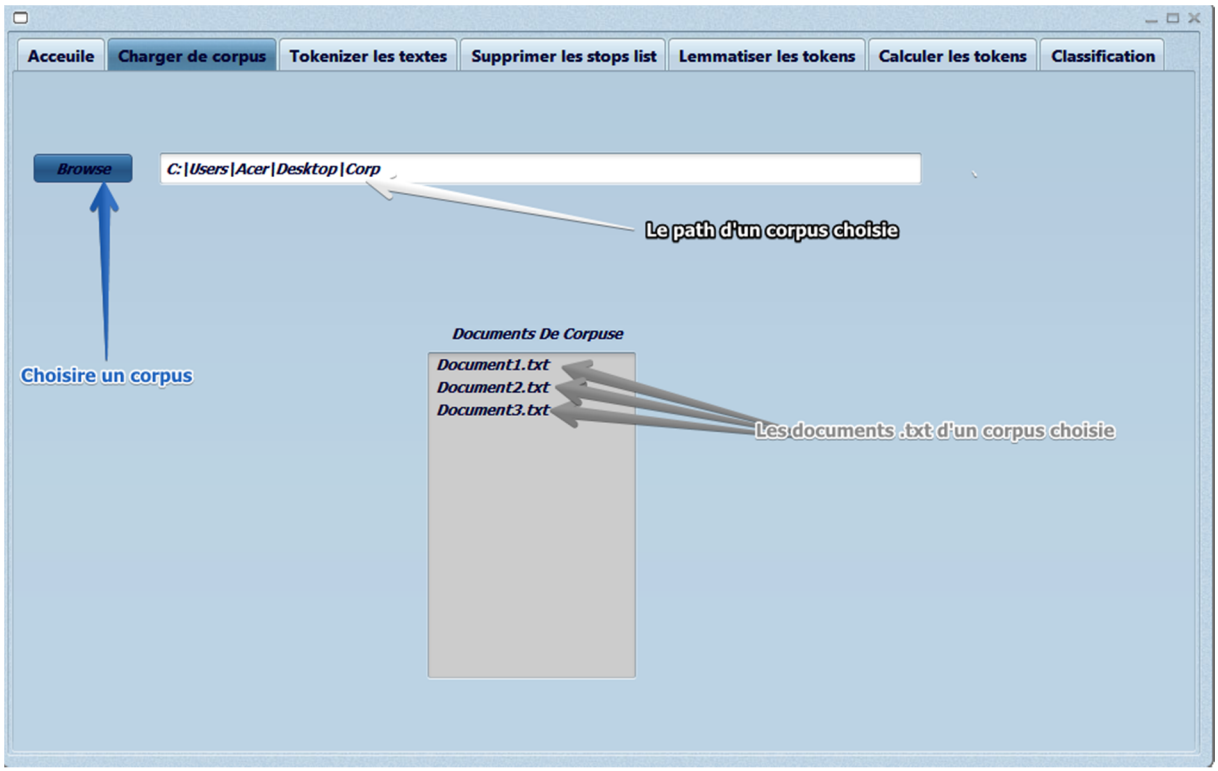

Figure 4.3: chargement d'un corpus de documents (.txt)

Première étape: extraction des mots à partir des textes:

Fenêtre 2: Tokenizer les textes (segmenter les textes par des phrases et les phrases par des mots)

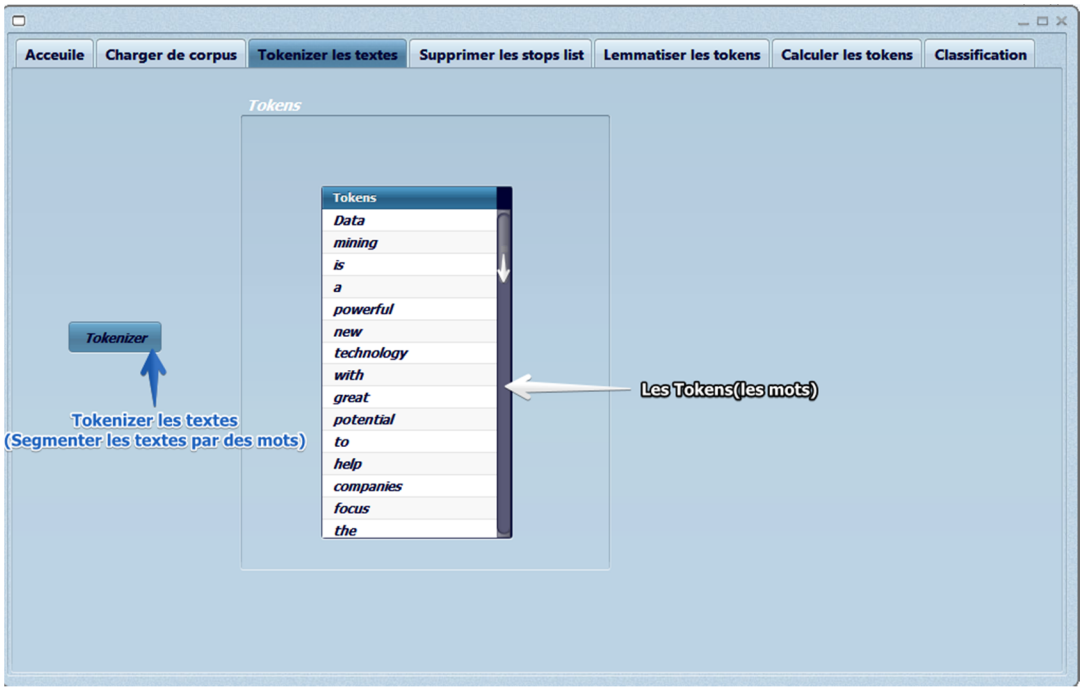

Figure 4.4 : Tokenizer les textes

Deuxième étape : supprimer les stops words

Fenêtre 3 : Supprimer les mots vides(les stops liste)

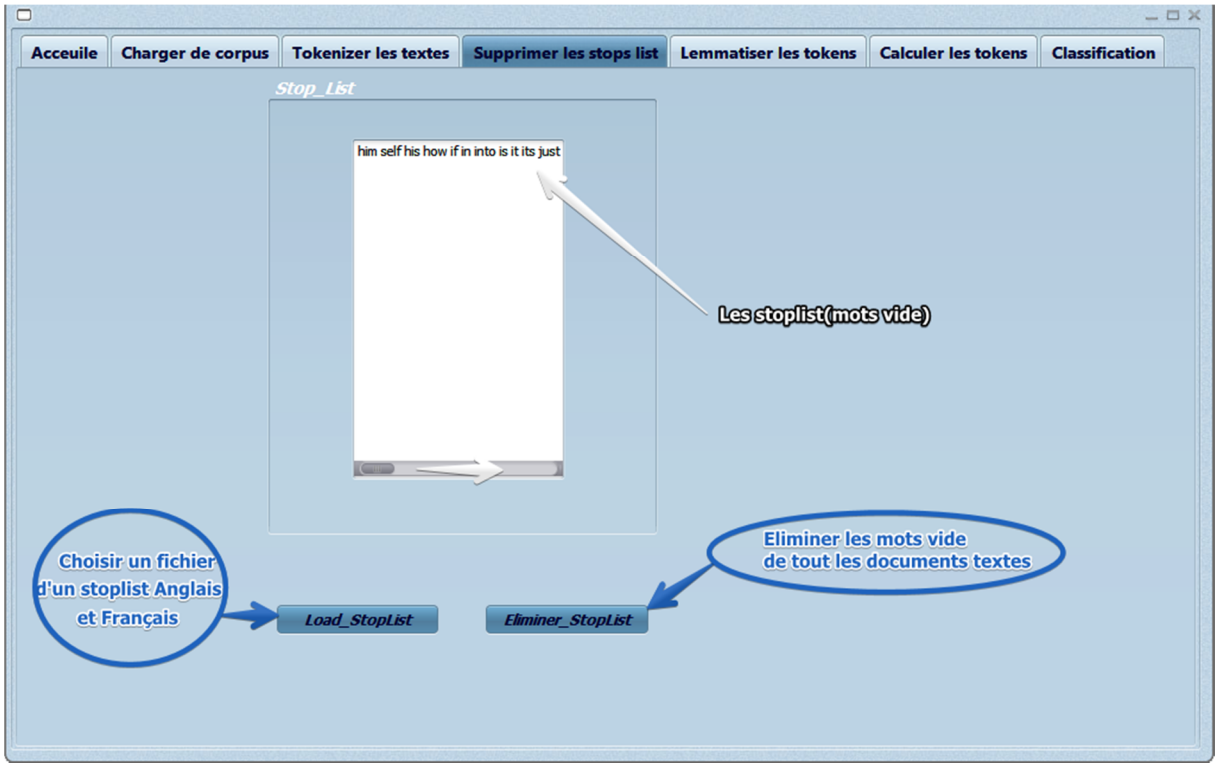

Figure 4.5: Supprimer les mots vides

Le bouton Eliminer_StopList :Supprimer La liste des stops words des 3 documents :

Document1

is/ a/with/to / on/ the/ most / in/ the/ they/ have / about/ the / of/ their/ and /./

Document2

It/ within /the/that/ and/'t/./

Document 3

is /a /with /the / of /and/. /

Troisième étape : Lemmatisation et construction du dictionnaire

Fenêtre 4: **Lemmatiser les tokens (CoreNLP)**

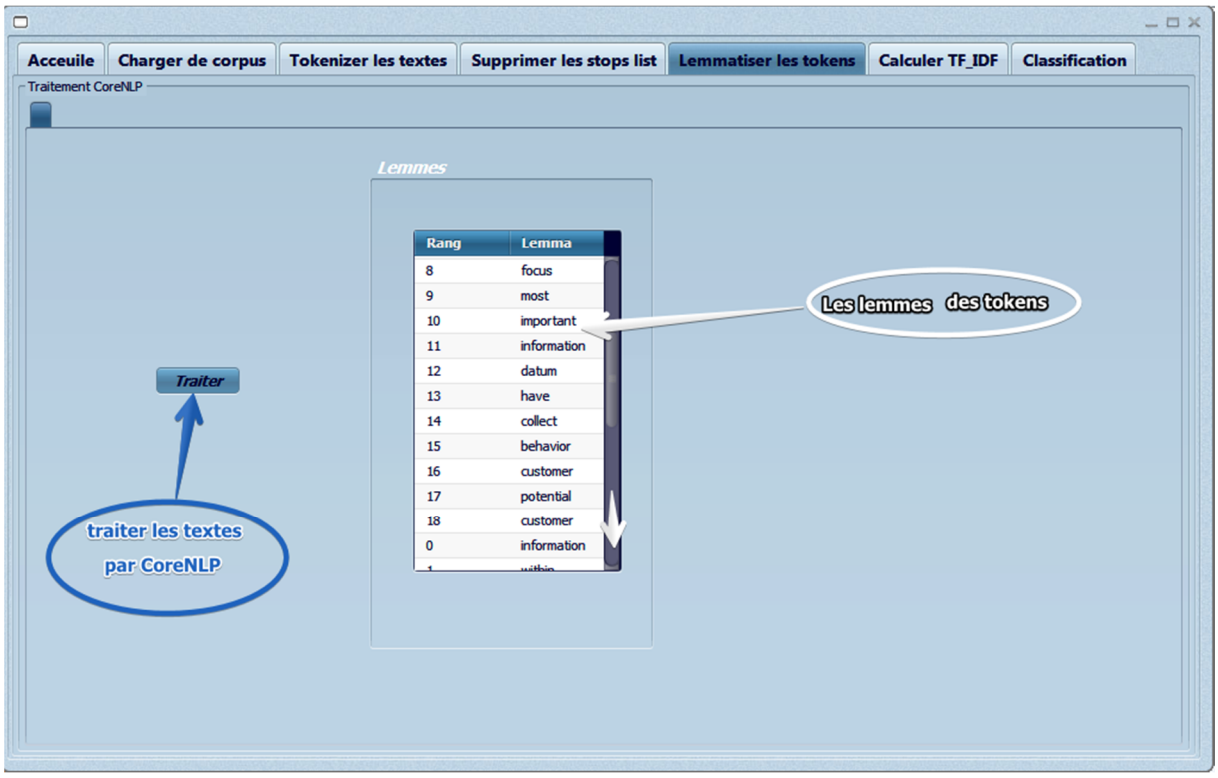

Figure 4.5 **:** Lemmatiser les tokens

Le bouton Traiter : Exécution :

```
Adding annotator tokenize
TokenizerAnnotator: No tokenizer type provided. Defaulting to
PTBTokenizer.
Adding annotator ssplit
Adding annotator pos
Reading POS tagger model from edu/stanford/nlp/models/pos-tagger/english-left3words/english-left3words-distsim.tagger ... done [10,6 sec].
Adding annotator lemma
```

Le résultat est 3 fichiers XML : xmlOutPut1, xmlOutPut2, xmlOutPut3.

Quatrième étape: calcule des poids

Fenêtre 5: Calculer la matrice TF_IDF

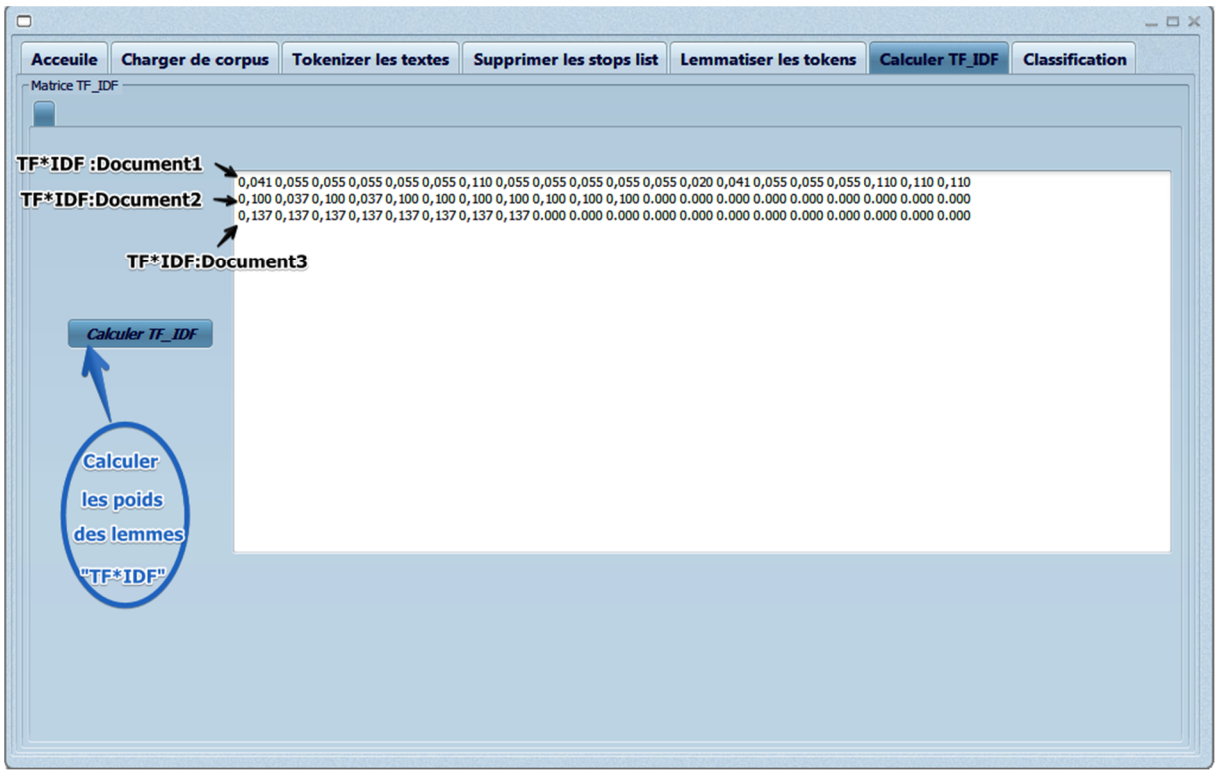

Figure 4.6 **:** Calculer la matrice TF_IDF

Le bouton Calculer TF_IDF :

Nombre de document = 3

- *Nombre de mots dans document* 1 = 18
- *Nombre de mots dans document* 2= 8
- *Nombre de mots dans document* 3= 8
- Nous calculons le poids de chaque mot dans ces textes.

Exemple poids du mot « live » dans le document 3 :

TF26 = 1/8= 0.125

IDF26 = log (3/1) = 0.48

TF_IDF26= 0.125*0.48 = 0.12

Cinquième étape: préparation du vecteur(p :poids)

Document1 :

label 1_p1 2_p2 3_p3 4_p4 5_p5 6_p6 7_p7 . . . . 15_p15

Document2 :

label 16_p16 17_p17 18_p18 19_p19

Document3 :

label 22_p22 23_p23 24_p24 25_p25 26_p26 27_0.06 28_p28

Sixième étape: affectation de label aux vecteurs:

Si nous faisons l'apprentissage sur le classe Data Mining alors l'étiquetage sera comme suit :

Document1 :

1 1_p1 2_p2 3_p3 4_p4 5_p5 6_p6 7_p7 . . . . 15_p15

Document2 :

1 16_p16 17_p17 18_p18 19_p19

Document3 :

-1 22_p22 23_p23 24_p24 25_p25 26_p26 27_0.0.06 28_p28

1.2- Classification

Fenêtre 6:La classification par les SVMs

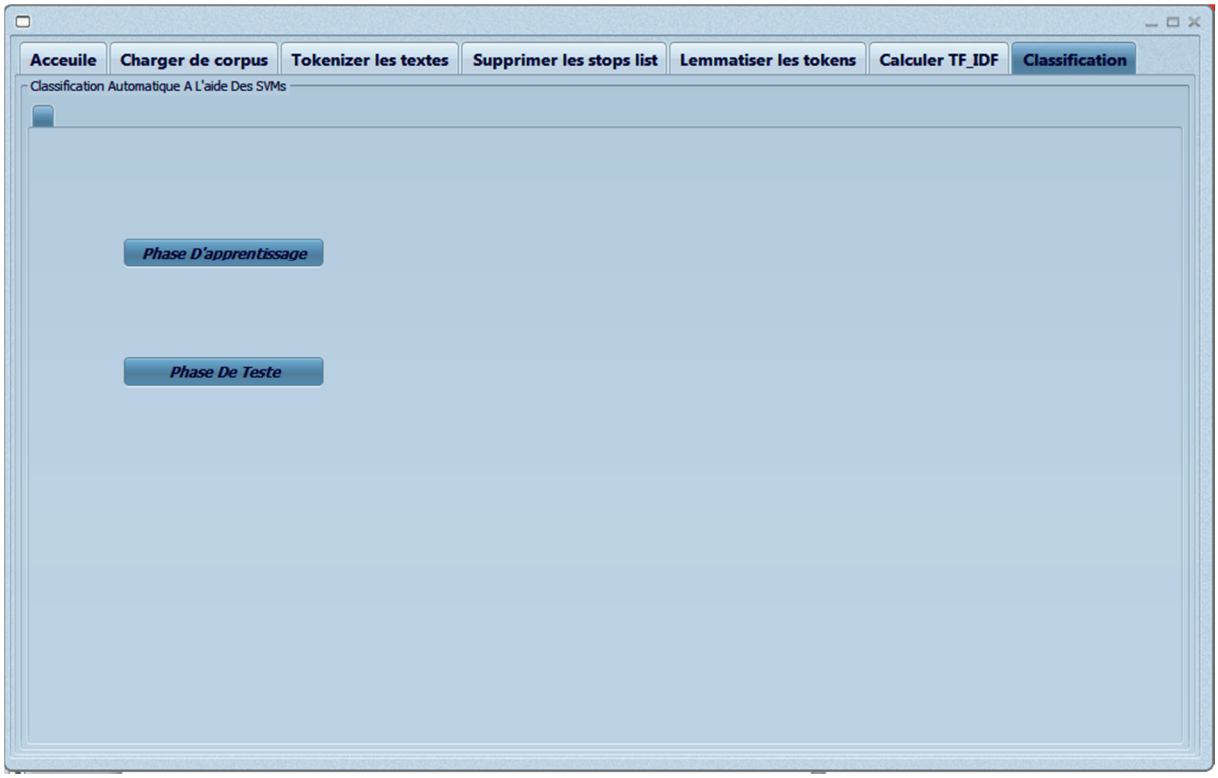

Figure 4.7**:** La classification par les SVMs

Phase d'apprentissage :

Pour déterminer les vecteurs de supports, SVM procède à des calculs de produits scalaires des vecteurs caractéristiques des textes.

Phase de Test :

une phase de test pour vérifier le modèle obtenu (et éventuellement une phase de validation en plus).

Phase de Validation :

une phase de prédiction ou de classement qui consiste à appliquer le modèle à de nouvelles données. C'est la phase déductive.

Les phases d'apprentissage, de test et de validation sont effectuées sur des échantillons distincts de la population.

Conclusion

Dans cette dernier chapitre de notre projet, nous avons présenté des détailles techniques sur notre environnement de travail et les différents outils utilisés pour achever le développement de notre solution. Nous avons aussi fait le tour des différentes interfaces ainsi que les fonctionnalités propre à chacune d'elles.

Conclusion Generale

La classification 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.
Il reste néanmoins difficile de fournir des valeurs chiffrées sur les performances qu'un système de classification peut actuellement atteindre.
Les travaux de recherche dans le domaine se focalisent surtout sur deux aspects **:** l'efficacité et l'amélioration de performances.
Dans ces deux optiques nous avons entamé notre projet de recherche en proposant une approche dans le domaine de classification supervisée intitulée « **Classification automatique de documents numérisés par les SVMs**»,
Cette application propose un couplage original des méthodes issues du Traitement Automatique du Langage Naturel (TALN) et des méthodes d'Apprentissage Automatique (AA). Grâce à cette utilisation associée, il est possible de disposer d'une base d'apprentissage de grande taille et très représentative du problème que l'on cherche à apprendre.
Nous avons décrit dans ce mémoire une nouvelle méthode de classification automatique de textes, dont voici une marque principale :
§ La transformation ou le codage des documents est la préparation à « l'informatisation » de ces derniers, qui va se faire par la technique de lemmatisation par CoreNLP connue pour son indépendance des différentes langues et son non exigence des traitements linguistiques préalables.

Enfin, nous pouvons affirmer que ce modeste travail est présenté dans un axe de recherche qui n'a pas été totalement exploré pendant ces années, malgré tous les efforts fournis dans la discipline, et pour lequel différents problèmes restent à résoudre. Néanmoins des débuts de solutions sont proposés et montrent que cette voie est prometteuse.

Perspectives

Les premiers résultats de cette application semblent prometteurs mais les études dans le domaine doivent être poursuivies, à cet égard et dans la même optique de recherche, on aperçoit de nombreuses pistes qui restent à explorer qui déclarent plusieurs chantiers ouverts :

§ Une première perspective se présente en diminuant les dimensions des profils des documents et catégories soit avec sélection des lemmes soit avec élimination des termes très fréquents (mots outils) et mots très rares sachant que la sélection de descripteurs est un des principaux enjeux du système, puisque du choix des descripteurs et de la connaissance précise de la population va dépendre la mise au point du classifieur. Ces entrées non discriminantes doivent être supprimées pour deux raisons différentes : réduire le temps de calcul et diminuer le sur-apprentissage.

La deuxième perspective s'annonce dans une variation dans la représentation des documents d'entrée en utilisant d'autres techniques de représentation de textes (sac de mots, lemmes, etc..). Chaque agent s'entraînera sur des textes codés par une méthode différemment des autres. Cette diversité dans l'apprentissage ne peut qu'enrichir le processus.

§ Une autre alternative se présente en changeant les pondérations des termes et en s'appuyant sur des représentations plus riches en informations que la représentation fréquentielle basique, à savoir le codage TF-IDF. Le nombre d'occurrences du terme dans la catégorie est la façon la plus simple de calculer cette pondération mais elle n'est pas très satisfaisante au sens où elle ne prend pas en compte les autres catégories, or on désire pouvoir faire une comparaison. Ainsi une autre expérience très intéressante à réaliser, en employant les SVMs avec la pondération la plus largement utilisée à savoir TF-IDF.

Enfin, il faut étendre nos réflexions aux catégorisations des documents manuscrits et ne pas se limiter à la version électronique du corpus (Il existe des versions reconnues du corpus Reuters manuscrit) Sachant que l'information textuelle contenue dans ces documents n'est accessible que grâce à un processus de reconnaissance, qui en toute évidence, induit des erreurs dans le texte résultant.

Bibliographie

[1]: Mattalah.H, ' Classification Automatique de textes Approche Orientée Agent', Soutenu en Février 2011.

[2]: P.Hayes, S.P.Weinstein,'Construe/Tis: A system for content-based indexing of a database of news stories', 1990.

[3]: K. Lang, ' NewsWeeder: Learning to Filter Netnews ', 1995.

[4]: R.Armstrong & all, 'WebWatcher: a Learning apprentice for the World Wide Web', 1995.

[5]: G.Brown, H.A.Chong,' The Guru System in TREC-6 ', 1998.

[6]: F.Sebastiani, 'A tutorial on automated text categorisation', 1999.

[7]: Y.Yang,' An evaluation of statistical approch to text categorization', 1999.

[8]: L.Denoyer, 'Apprentissage et inférence statistique dans les bases de documents structurés : Application aux corpus de documents textuels', 2004.

[9]: McCallum & all, 'Improving Text Classification by Shrinkage in a Hierarchy of Classes', 1998.

[10]: P.Bellot, M.El-Béze, 'Classification locale non supervisée pour la recherche documentaire',2002.

[11]: Ameni.B,' Catégorisation automatique de news à l'aide de techniques d'apprentissage supervisé'.

[12]: C.de Loupy, M.El-Bèze', 'Using few cues can compensate the small amount of resources available for WSD', 2002.

[13]: C.Loupy, 'Évaluation de l'Apport de Connaissances Linguistiques en Désambiguïsation Sémantique et Recherche Documentaire', 2000.

[14]: F.Sebastiani,' Machine learning in automated text categorization', 2002.

[15]: J.Clech, D.A.Zighed,' Une technique de réétiquetage dans un contexte de catégorisation de textes', 2004.

[16] : Y.Gilli, ' Texte et fréquence ', 1988.

[17]: .F.Caropreso, S.Matwin, F.Sebastiani « A learner-independent evaluation of the usefulness of statistical phrases for automated text categorization',2001.

[18] : R. Jalam, 'Apprentissage automatique et catégorisation de textes multilingues', 2003.

[19]: S.Scott, S.Matwin,' Feature Engineering for Text Classification',1999.

[20]: D.Lewis, 'An evaluation of phrasal and clustered representations on a text categorization task ', 1992.

[21]: J.Clech,' Contribution méthodologique à la fouille de données complexes', 2004.

[22]: M.F.Porter, 'An algorithm for suffix stripping', 1980.

[23]: H.Schmid, 'Probabilistic part-of-speech tagging using decision trees', 19994.

[24]: C.Shannon,' The Mathematical Theory of Communication', 1948.

[25]: Sami.L & all' Classification automatique de documents bruités a faible contenu textuel', E-18, juin 2009, PP.25.

[26]: Konstantin.M, Michael.M,' Document Classification with Support Vector'.

[27]: Hanifi.M,' Extraction de caractéristiques de texture pour la classification d'images satellites',2009, PP53.

Webographie

http://dspace.univ-tlemcen.dz/, ***Consulté le 15/12/2014***

http://www.zone-project.org/***, Consulté le 12/2/2015***

http://hal-lirmm.ccsd.cnrs.fr/lirmm-00394668***, Consulté le 4/1/2015***

Annexe

Annexe 1 : AA

L'**A**pprentissage **A**utomatique est l'un des champs d'étude de l'**intelligence artificielle**, est la discipline scientifique concernée par le développement, l'analyse et l'implémentation de méthodes automatisables qui permettent à une machine (au sens large) d'évoluer grâce à un processus d'apprentissage, et ainsi de remplir des tâches qu'il est difficile ou impossible de remplir par des moyens algorithmiques plus classiques.

Les déférentes phases de l'AA :

- **Phase d'apprentissage ;**
- **Phase de teste ;**
- **Phase de la généralisation ;**

Annexe 2 : TALN

(**TAL** ou **TALN**) **T**raitement **A**utomatique de la **l**angue ou **T**raitement **A**utomatique de la **l**angue **N**aturel.

On doit parler de : langage et langue en français alors qu'en arabe ou anglais un seule terme اللغة et language.

- Naturel vers Artificiel ;
- Informel vers Formel ;

Annexe 3 : Loi de Zipf

La **loi de Zipf** est une observation empirique concernant la fréquence des mots dans un texte. Elle a pris le nom de son auteur, ***George Kingsley Zipf (1902-1950).***