Résumé / Abstract

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 « Kppv », basée sur la détection des synonymes. L'objectif principal de nos travaux, est d'améliorer les performances et l'efficacité du modèle de classification. Le corpus de référence Reuters, va servir à mener une étude comparative des résultats obtenus.
Mot clés: Corpus textuels, classification, apprentissage automatique, validation, Reuters.

With the advent of computers and the increasing number of electronic documents stored on various electronic media and web, especially text data, development of analysis tools and automatic processing of texts, including automatic text classification has become essential to assist users of these document collections, to explore and identify all these huge banks of textual data. And automatic categorization of text, which is to assign a document to one or more categories, is becoming increasingly recognized as a key technology in the management of intelligence, the results are useful both for the search information to extract knowledge or on the Internet (search engines), and at the company (ranking of internal documents, news agencies, etc.). In respect of different approaches to automatic text classification, described in the prior art, relying on a conventional architecture based on a single point of view, we introduced a novel use of the classifier " KNN", based on the detection of synonyms. The main objective of our work is to improve the performance and efficiency of the classification model. The reference corpus Reuters will be used to conduct a comparative study of results.
Keywords : Textual Corpus, classification, Automatic learning, validation, Reuters

Dédicace

A ma chère grande mère « Jasmine »
Qui nous a quitter ce 22 Avril
Allah yerhmek très chère.

REMERCIEMENT

Il est difficile d'exprimer en quelques mots ce que le cœur porte incommensurablement. Je n'ai d'habitude, aucunement, de mots pour traduire ma gratitude à ceux qui me soutiennent. Toutefois, je m'oblige à cette occasion car il me semble nécessaire, de jaillir ma plume et de vous remercier infiniment d'avoir était à mes coté durant mon cursus universitaire.

Je suis très heureuse et fière d'avoir préparée et de présenter ce mémoire dont chaque mot, chaque point représente pour moi tous le savoir que vous aviez semé en mon esprit jadis ignare. Ensemble, je crois, nous avions tissé un lien très fort et extrêmement précieux.

Je remercie le département d'informatique pour sa très grande assistance, mon encadrante Mme Maghni Sandid Zoulikha pour son professionnalisme et son humanité.

En dernier, Je remercie ma chaleureuse et grande famille qui m'a toujours porté vers le haut, mes amis qui m'épaulent contre vents et marrées et avec qui je partage les doutes et les épreuves aussi bien que les satisfactions et les récompenses.

Hakima Hamerlain.

Sommaire

Figures

Introduction Générale

1 – Problématique et contexte du mémoire

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 connaît 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

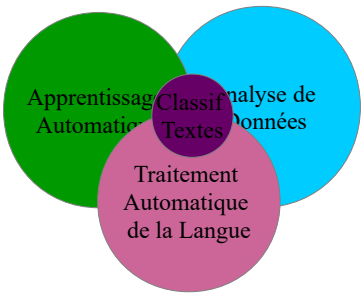

Figure 1. Position de notre problème

2 – Contribution

Nous essayons de réaliser un couplage entre deux grands domaines dans le monde d'informatique qui sont la détections de synonyme et la classification de textes.

L'optique de ce couplage est d'améliorer les performances et l'efficacité des systèmes de classification de textes.

3 – Organisation du mémoire

Ce mémoire va être organisé de la façon suivante :

Un premier chapitre pour définir l'ensemble des concepts de base du contexte étudié. Un état de l'art va être étalé.

Un deuxième chapitre pour traiter les techniques employées dans les différentes phases du processus de classification automatique de textes.

Un troisième chapitre est consacré à motiver toutes les options entreprises ainsi que notre contributions et les résultats expérimentaux.

Chapitre 1
Classification de textes

1 – Introduction

La catégorisation de textes, est un problème qui intéresse les chercheurs depuis relativement longtemps. On retrouve des travaux portant sur ce sujet depuis au moins le début des années 1960. Des avancées importantes ont été observées depuis, et les résultats obtenus aujourd'hui sont encore sujets à amélioration. Pour certaines tâches, les classifieurs sont performants presque aussi bien que les humains, mais pour d'autres, l'écart est encore grand. Sans compter que le besoin d'un traitement efficace de l'information est grandissant, car l'immense bassin de connaissances à notre portée est sans intérêt s'il n'est pas bien géré. Au premier abord, l'essentiel du problème est facile à saisir. D'un côté, on est en présence d'une banque de documents textuels et de l'autre, d'un ensemble prédéfini de catégories. L'objectif est de rendre une application informatique capable de déterminer de façon autonome dans quelle catégorie classer chacun des textes, à partir de leur contenu.

Dans ce chapitre, on va décrire les différents modules qui composent un système de CT.
Pour commencer on va parler de histoire de la classification, ses domaine d'utilisation et les problèmes lié a cette dernière. Ensuite on va décrire le phénomène. Et pour finir on va voir comment représenter nos texte pour les transmettre a la prochaine étape.

2 – Pourquoi on a recours à la classification ?

On assiste aujourd'hui à un accroissement de la quantité d'information textuelle disponible et accessible d'une manière exponentielle. D'après les derniers chiffres, on parle de plus de 200 millions de serveurs hôtes sur Internet et plus de 3 milliards de pages, la taille des corpus tests utilisés est passée de quelques mégaoctets à plusieurs Gigaoctets. [1]

Dans les années 1996-1997, Reuters a produit un peu plus de 800 000 nouvelles en anglais par année. Si l'on ajoute aux articles écrits par les journalistes de l'agence ceux provenant d'autres sources, on arrive à un total de 5.5 millions de textes anglais par année à catégoriser. [1]

À un moment, l'organisation employait 90 personnes dédiées à l'étiquetage de ces documents. Il serait à coup sûr très intéressant de pouvoir déterminer avec précision le coût de classification. De combien de temps a besoin un humain pour associer un texte à une catégorie ? En pratique, il s'agit d'une question difficile à répondre. Assurément, plusieurs variables influencent le phénomène et les lignes qui suivent porteront sur certaines d'entre elles. [1]

Certainement une grande partie du temps consommé pour classer un document est employé dans sa lecture, puis éventuellement à sa relecture. On peut aussi imaginer que la longueur des textes à classer est assez déterminante du temps qui va être requis pour cette opération, et sans doute, d'une personne à une autre, la vitesse de lecture varie. Une fois cette étape achevée, il faut trancher à quelle(s) catégorie(s) ce texte appartient. Au temps de réflexion exigé s'ajoute, certainement, le temps de se référer à la description des classes et éventuellement de consulter d'autres textes préalablement associés à certaines classes, pour valider la décision. D'autres facteurs interviennent également comme, comme par exemple le nombre de classes qui peut faire la différence : plus il y a de classes différentes, autrement dit plus il y a d'étiquettes possibles pour un texte donné, plus il est difficile de faire un choix parmi celles-ci. Aussi, plus la sémantique des catégories est précise, fine, détaillée, plus il faut faire attention avant d'y associer un document. À cet égard, classer des

documents appartenant soit à la catégorie «informatique» soit à la catégorie «mathématiques» est vraisemblablement plus aisée que celle de classer des documents appartenant à l'une ou l'autre des catégories «Intelligence artificielle» , «Génie logiciel» et «Système d'information» . [1]
En conséquence, nous pouvons résumer les contraintes majeures qui s'opposent au traitement manuel de classification des documents textuels dans les trois points suivants :

- La réalisation manuelle de cette tâche par un expert est extrêmement coûteuse en terme de temps et personnel car il s'agit de lire attentivement chaque texte, au vu de la quantité phénoménale de textes aujourd'hui accessibles (par le biais du réseau Internet en particulier). [1]
- Les traitements manuels sont peu flexibles et leur généralisation à d'autres domaines est quasi impossible; c'est pourquoi on cherche à mettre au point des méthodes automatiques
- Cette opération peut être perçue comme subjective puisque basée sur l'interprétation du document, deux experts peuvent classer différemment un même document, ou encore un même expert peut classer différemment un même document soumis à deux instants différents. [1]

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. [1]

3 – Historique de la Catégorisation de textes

C'est une discipline assez ancienne, en 1627, Gabriel Naudé propose un classement selon cinq grands thèmes : théologie, jurisprudence, histoire, sciences et arts, belles lettres. Le désir de maîtriser l'Univers se fait sentir dans la multiplication des encyclopédies. L'encyclopédie de Diderot (parue entre 1751 et 1772) est organisée selon l'ordre alphabétique avec des renvois associatifs alors que celle de Panckoucke (parue de 1776 à 1780) suit une organisation méthodique selon un ordre arborescent. [1]
Le système de classification par thème, apparu dès les débuts de l'écriture et institutionnalisé à Alexandrie conduisit à la création par Dewey, en 1876, d'un système de classification « universel ». Il s'agit d'une classification documentaire de type encyclopédique. [1]
Toutefois l'idée d'effectuer la classification de textes par des machines remonte au début des années 60 et qui a connu des progrès considérables à partir des années 90 avec l'apparition d'algorithmes beaucoup plus performants qu'auparavant. [1]
Jusqu'au début des années 80, pour construire un classifieur, il fallait consacrer d'importantes ressources humaines à cette tâche. Plusieurs experts éditaient des règles manuellement puis les affinaient au fur et à mesure des tests. L'avènement des de l'AA s'est donc traduit par un gain de temps conséquent. Il n'est plus nécessaire par exemple de reconfigurer tout le système en cas de changement d'arborescence. [1]
Ces évolutions technologiques et algorithmes avancées font aujourd'hui de la catégorisation un outil fiable. [1]
Au début des années 90, les travaux proviennent essentiellement de la communauté de Recherche d'Information (RI). En effet, les méthodes de numérisation, les algorithmes de classification et les méthodologies de test ont été adaptés à la CT en particulier au cours des conférences TREC (Text REtrieval Conference.). [1]

La communauté d'Apprentissage Automatique (AA) s'est intéressée elle aussi à ce problème il y a une dizaine d'années en le considérant comme domaine d'application à ces algorithmes de reconnaissance des formes. Actuellement, les méthodes de numérisation de texte restent largement inspirées de la RI alors que les classifieurs les plus performants sont issus de l'AA. [1]
Une autre communauté composée essentiellement de statisticiens et de linguistes, traite également le problème de la CT en s'appuyant sur les méthodes d'analyse de données. Le but ici n'est pas de créer un système qui classe automatiquement des documents sans intervention humaine mais d'extraire des informations synthétiques du corpus. Les problématiques traitées ici sont par exemple l'étude des genres littéraires ou la détermination de l'auteur d'un texte. [1]

4 – 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. [1]

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 », « culture », etc..), (Exemple : Une boite propose un système de tri d'informations dans des flots de dépêches d'agence de presse AFP ou Reuters etc.. ou pages web. Chaque matin les nouvelles importantes sont faxées à différentes entreprises).
- 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]

5 – Description de la catégorisation de texte

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. [2]

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. [2]

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. [2]

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. [2]

- 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) D x C une valeur booléenne pour indiquer si le document dj appartient ou non à la classe ci . [2]

La fonction de décision sera définie de la manière suivante :

D : D x C { vrai, faux } , D (d,c) = Vrai si d est associé à la classe c sinon D (d,c) = Faux

La fonction cible sera définie de la manière suivante :

C : D x C { vrai, faux } , C (d,c) = Vrai si d est associé à la classe c sinon C (d,c) = Faux

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. [2]

6 – Les différents contextes de classification

Il existe Plusieurs contextes de classification.

6.1 – Classification bi-classe et multi-classes

6.1.1 – 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 (i.e. ou à sa catégorie complémentaire ? » (Par exemple, un document est il autorisé aux enfants 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 muti-classes, selon le cas, peut être disjointes ou non. [3]

6.1.2 – 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 ? ». [3]

6.1.3 – 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. [3]

6.2 – Catégorisation déterministe et floue

6.2.1 – Catégorisation déterministe

Le but des classifications précédentes est d'avoir une réponse définitive pour chaque texte (oui ou non, le texte T appartient à la catégorie C) ; qu'on peut qualifier par classification déterministe. Plusieurs fonctions de classement sont utilisées, parmi lesquelles : les règles de décisions, les arbres de décision, SVM. [3]

6.2.2 – Catégorisation floue ou le ranking

Contrairement aux cas précédents, on peut également souhaiter dans certains cas d'avoir simplement une évaluation des classes les plus adéquates -dans l'ordre- pour y classer le texte. Ce qu'on peut appeler par classification floue ou ranking. [3]

Ce type de classification va permettre à l'utilisateur d'être plus indulgent si le texte est "proche" du thème que si le texte n'a absolument rien à voir avec celui-ci dans le cas ou ce dernier est incorrectement attribué à la classe. [3]

Le ranking est une problématique de classification dans laquelle le système, au lieu d'associer un texte à une classe catégoriquement, il ordonne les classes par ordre de pertinence pour un texte donné. [3]

Les méthodes qui évaluent une distance d'un texte à une catégorie permettent facilement ce type de classement de même pour les approches qui estiment des probabilités d'appartenance d'un texte à une classe. [3]

7 – Problèmes de la catégorisation de textes

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

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

7.1 – Redondance (Synonymie)

La redondance et la synonymie permettent d'exprimer le même concept par des expressions différentes, plusieurs façons d'exprimer la même chose. [4]

Cette difficulté est liée à la nature des documents traités exprimés en langage naturel contrairement aux données numériques. LE FEVRE illustre cette difficulté dans l'exemple du chat et l'oiseau : mon chat mange un oiseau, mon gros matou croque un piaf et mon félin préféré dévore une petite bête à plumes. [4]

La même idée est représentée de trois manières différentes, différents termes sont utilisés d'une expression à une autre mais en fin compte c'est bien le malheureux oiseau qui est dévoré par ce chat. [4]

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. [4]

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. [4]

Ce problème est l'objet d'étude de notre mémoire.

7.2 – 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. [4]

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. [4]

7.3 – 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. (Ex : avocat en tant que fruit et avocat en tant que juriste). [4]

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]

7.4 – 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. [4]

7.5 – 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 pré-traitements. [4]

7.6 – Les mots composés

La non prise en charge des mots composés 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. [4]

7.7 – 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 ya plusieurs façons d'exprimer les mêmes choses, 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 technique d'apprentissage se basant sur l'exclusion d'un mot particulier. [4]

7.8 – Complexité de l'algorithme d'apprentissage

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. [4]

7.9 – 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. [5]

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. [5]

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. [5]

Sans bien sûr pénaliser le système en supprimant des termes pertinents.

7.10 – 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. [6]

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 ! [6]

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. [6]

8 – 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 :

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

La figure illustre la démarche de catégorisation de textes avec ses trois étapes qui peuvent être schématisées comme suit : [1]

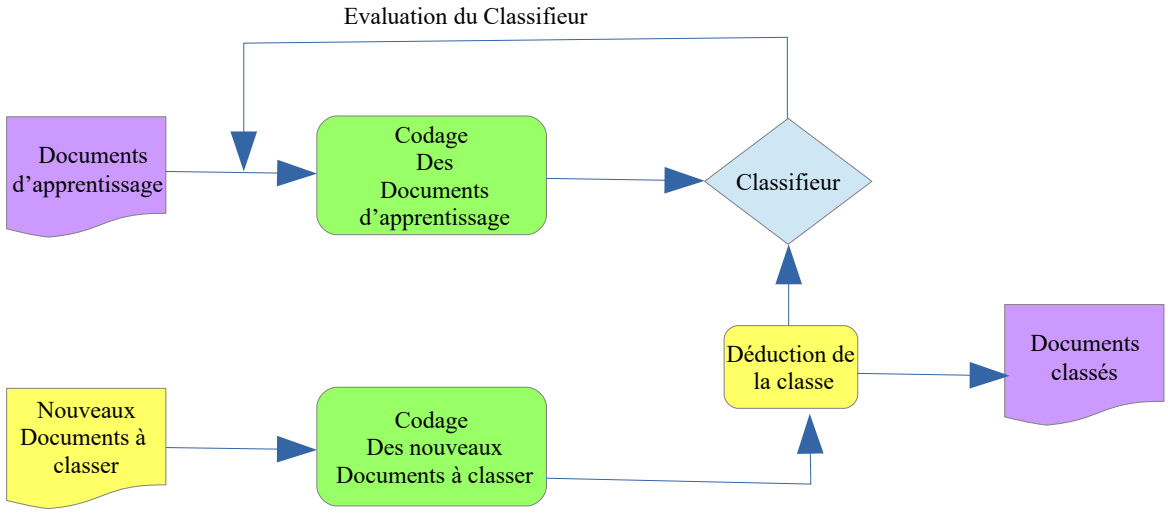

Figure 2. Démarche de la catégorisation de textes.

9 – Prétraitements

Le prétraitement des textes est une phase capitale du processus de classification. 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 d'informations sur le document concerné. Les algorithmes dits de "Stop Words" s'occupent de les éliminer. [1]

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. [1]

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

1. La segmentation. [1]
2. Suppression des mots fréquents. [1]
3. Suppression des mots rares. [1]
4. Le traitement morphologique. [1]
5. Le traitement syntaxique. [1]
6. Le traitement sémantique. [1]

10 – Représentation des documents

La façon dont on va représenter le problème est très importante. Dans le cas de la classification de textes, on doit opter pour une façon efficace de représenter les instances à traiter, soit les textes. Un grand nombre de chercheurs dans le domaine ont choisi d'utiliser une représentation vectorielle dans laquelle chaque texte est représenté par un vecteur de n termes pondérés. À la base, les n termes sont tout simplement les n différents mots apparaissant dans les textes de l'ensemble d'entraînement. Cette approche est aussi appelée sac de mots, en anglais « bag-of-words ». [7]
Il existe aussi plusieurs façons d'associer un poids à un terme. Il peut être tout simplement binaire (1 si le mot est présent dans le texte, 0 sinon). Il peut aussi représenter le nombre d'occurrences du mot dans le texte. [7]
Une façon largement utilisée de calculer le poids d'un terme est la fonction TFIDF (acronyme pour « Term Frequency Inverse Document Frequency »). Issue du monde de la recherche d'information, celle-ci donne plus d'importance aux mots qui apparaissent souvent à l'intérieur d'un même texte, ce qui correspond bien à l'idée intuitive que ces mots sont plus représentatifs du document. Mais sa particularité est qu'elle donne également moins de poids aux mots qui appartiennent à plusieurs documents, pour refléter le fait que ces mots ont un faible pouvoir de discrimination entre les classes. [7]
Le poids d'un terme tk dans un document dj est calculé ainsi :
$TFIDF(tk, dj) = nb(tk, dj) \cdot \log |Tr| / nb(tk)$
où

- nb(tk, dj) est le nombre d'occurrences de tk dans dj.
- |Tr| est le nombre de documents d'entraînements.
- nb(tk) est le nombre de documents d'entraînements dans lesquels tk apparaît au moins une fois.

11 – Réduction de la taille du vocabulaire

S'attaquer au problème de la classification de textes signifie aussi s'attaquer à des difficultés qui sont propres au traitement automatique de la langue naturelle.
La taille impressionnante du vocabulaire peut s'avérer un obstacle à l'utilisation d'algorithmes plus complexes.
Si l'on utilise directement le vocabulaire contenu dans les textes d'entraînement et que l'on crée un attribut pour chaque mot qu'il contient, on se retrouve avec un espace vectoriel ayant une dimension très élevée. Chacun des textes sera représenté par un vecteur ayant autant de termes qu'il y a de mots dans le vocabulaire. [7]
Le traitement d'un tel espace vectoriel demanderait beaucoup de mémoire et de temps de calcul et pourrait nous empêcher d'utiliser des algorithmes de classification plus complexes. Le problème pourrait dans certains cas devenir non soluble. [7]
Utiliser tous ces mots influencerait aussi négativement la précision de la classification. En effet, plusieurs mots, tant en anglais qu'en français, sont vides de sens, donc inutiles pour faire ressortir la sémantique d'un texte. Aussi, si un mot est présent dans un nombre élevé de documents, c'est donc qu'il ne permettra pas de départager l'appartenance d'un texte qui le contient à l'une ou l'autre des catégories. On dit alors que son pouvoir de discrimination est faible. C'est en bonne partie pour ces raisons que certaines techniques ont été mises en place pour réduire la dimension du vocabulaire, des techniques qui se divisent en deux grandes familles. [7]
La sélection d'attributs « feature selection » prend les attributs d'origine et conserve seulement ceux jugés utiles à la classification, selon une certaine fonction d'évaluation. Les autres sont rejetés. D'autres méthodes effectuent plutôt une extraction d'attributs « feature extraction ».
À partir des attributs de départ, elles créent de nouveaux attributs, en faisant soit des regroupements ou des transformations. [7]

12 – Sélection d'attributs

Plusieurs techniques de sélection d'attributs ont été développées en vue de réduire la dimension de l'espace vectoriel. Chacune de ces techniques utilise des critères lui permettant de rejeter les attributs jugés inutiles à la tâche de classification. On obtient alors un vocabulaire réduit, des textes représentés par des vecteurs de moindre dimension, un temps de calcul plus abordable et même dans certains cas une précision de classification accrue. [7]

12.1 – Éliminer les stop-world

Débuter le processus de sélection d'attributs par la suppression de mots vides de sens, appelés en anglais « stop words ». Cette méthode nécessite l'utilisation d'une liste de ces mots qui ne contiennent aucune information sémantique, qui ne modifient pas le sens des mots qui les accompagnent. Par exemple, en anglais, on peut compter des mots comme « a », « and », « the ». Il s'agit souvent de mots fonctionnels comme les prépositions. [7]
Le contenu de cette liste dépend de la langue et possiblement du domaine des textes à classer. Cette étape permet d'éliminer efficacement plusieurs mots inutiles à la tâche de classification. [7]

12.2 – Stemming et Lemmatisation

La recherche de radical « stemming » et la lemmatisation sont d'autres processus pour créer un vocabulaire réduit. Leur but est de regrouper en un seul attribut les multiples formes morphologiques de mots qui ont une sémantique commune. Par exemple, « étude », « étudiant » et « étudier » pourraient être regroupés, et un seul attribut qui sera ajouté à l'espace vectoriel plutôt que trois. Ce faisant, la dimension du vocabulaire s'en trouverait réduite. [7]
Les deux processus précédents sont rarement utilisés seuls. On les considère comme des traitements préalables à des techniques plus élaborées. Parmi ces critères de sélection d'attributs les plus rencontrés, notons :

12.2.1 – La fréquence « document frequency »

Il s'agit d'éliminer les mots dont le nombre de documents dans lesquels ils apparaissent est en dessous d'un certain seuil. L'idée est que ces mots n'apportent pas d'information utile à la prédiction de la catégorie d'un texte ou qu'ils n'influencent pas la performance globale du classificateur. Il y a aussi la possibilité que ces termes soient le résultat d'erreurs, comme un mot mal orthographié. Dans ce cas, leur élimination est donc bénéfique. Opérer de cette façon pour réduire le nombre d'attributs est plutôt simple et rapide. Un bémol à l'utilisation de cette technique est que généralement, on ne l'utilise pas pour une sélection vraiment agressive, parce qu'il est admis que les termes ayant une fréquence faible/moyenne sont relativement informatifs et devraient être conservés. [7]

12.2.2 – Le gain d'information « information gain »

On mesure en quelque sorte le pouvoir de discrimination d'un mot, le nombre de bits d'information obtenue pour la prédiction de la catégorie en sachant la présence ou l'absence d'un mot. Cette méthode est souvent mise en pratique dans les arbres de décisions, pour choisir l'attribut qui va le mieux diviser l'ensemble des instances en deux groupes homogènes.

12.2.3 – L'information mutuelle « mutual information »

Cette façon d'évaluer la qualité d'un mot dans la prédiction de la classe d'un document est basée sur le nombre de fois qu'un mot apparaît dans une certaine catégorie. Plus un mot va apparaître dans une catégorie, plus l'information mutuelle du mot et de la catégorie va être jugée élevée. Plus un mot va apparaître en dehors de la catégorie (et plus une catégorie va apparaître sans le mot), moins l'information mutuelle va être jugée élevée. [7]
Il faut ensuite faire une moyenne des scores du mot jumelé à chacune des catégories. La faiblesse de cette mesure est qu'elle est beaucoup trop influencée par la fréquence des mots. Pour une même probabilité conditionnelle sachant la catégorie, un terme rare va être avantagé, car il risque moins d'apparaître en dehors de la catégorie. [7]

12.2.4 – La statistique du X^2

Mesure statistique bien connue, elle s'adapte bien à la sélection d'attributs, car elle évalue le manque d'indépendance entre un mot et une classe. Elle utilise les mêmes notions de cooccurrence mot/catégorie que l'information mutuelle, mais une différence importante est qu'elle est soumise à une normalisation, qui rend plus comparable les termes entre eux. Elle perd quand même de la pertinence pour les termes peu fréquents. [7]

12.2.5 – La force du terme « term strength »

Il s'agit d'une méthode plutôt différente des autres. Elle se propose d'estimer l'importance d'un terme en fonction de sa propension à apparaître dans des documents semblables. Une première étape consiste à former des paires de documents dont la similarité cosinusoïdale 3 est supérieure à un certain seuil. La force d'un terme est ensuite calculée à l'aide de la probabilité conditionnelle qu'il apparaisse dans le deuxième document d'une paire, sachant qu'il apparaît dans le premier. [7]

13 – Conclusion

Ce chapitre a permis d'avoir un aperçu global sur la classification de textes. Tout d'abord, il s'est attardé sur la nature du problème à résoudre. Par la suite, il a été question sur la représentation des documents traités par un classificateur. Enfin, des techniques de sélection d'attributs ont été exposées, celles-ci visant à réduire la taille du vocabulaire à traiter pour que les algorithmes évoluent dans un espace vectoriel de dimension raisonnable. Maintenant que le problème étudié est mieux cerné, On passe au chapitre suivant avec la présentation d'algorithmes d'apprentissage automatique transposés dans ce domaine.

Chapitre 2

Algorithmes d'apprentissage automatique appliqués à la catégorisation de textes

1 – Introduction

La classification (ou catégorisation) de documents a donné lieu à de nombreux travaux recourant aux méthodes d'apprentissage automatique qui se divise en apprentissage non supervisé et apprentissage supervisé. Les algorithmes les plus utilisées dans ce domaine d'application sont : le classifieur naïf de Bayes, les machines à support vectoriel (SVM) ou encore les arbres de décisions.
Dans ce chapitre Nous allons détailler ces algorithmes. Ensuit on abordera les critères d'évaluation des classificateurs essentiels pour mesurer la performance de ces dernier.

2 – Classification

C'est à cette étape que se fait l'assignation du document à la classe à laquelle il appartient. La détermination de la classe se fait grâce à des algorithmes de classification qui exploitent les données extraites et qui donnent en sortie la classe correspondante. Ces algorithmes se basent sur leur expérience passée où ils ont appris comment classer les textes, c'est de là que vient leur nom « Algorithmes d'apprentissage ». Il existe deux types de ces algorithmes : [8]

- Les algorithmes d'apprentissage supervisé
- Les algorithmes d'apprentissage non supervisé. [8]

3 – Apprentissage automatique

Puisque l'approche manuelle de classification de textes est coûteuse en temps de travail, peu générique, et relativement peu efficace, l'autre solution a été admise, qui consiste à faire apprendre automatiquement à l'ordinateur, sur la base d'un corpus de textes qui servent d'exemples, les paramètres de la fonction de classement.
Ainsi depuis une quinzaine d'années la classification de textes a été considérée comme un problème d'apprentissage automatique et est rapidement devenue un champ d'essai sollicité par les différentes techniques de classification. De toute façon, quelle que soit l'approche retenue, une des particularités de cette tâche est la très grande dimensionnalité de l'espace dans lequel les textes sont représentés, qui comprend généralement plusieurs milliers de termes.
L'apprentissage automatique s'intéresse aux méthodes inductives permettant d'acquérir des connaissances à partir d'observations d'un phénomène. Cette connaissance peut être exploitée pour des tâches de décision ou de prévision : c'est le cadre de l'apprentissage supervisé ; ou à des fins d'analyse exploratoire ou de structuration d'un ensemble de données : c'est le cadre de l'apprentissage non-supervisé. Le contexte de notre étude se situe dans le premier cas. [9]

3.1 – Apprentissage non supervisé

3.1.1 – Principe

L'apprentissage non supervisé consiste à apprendre à classer sans supervision. Au début du processus nous ne disposons ni de la définition des classes, ni de leurs nombres. C'est l'algorithme de classification qui va déterminer ces informations. Nous ne disposons pas non plus de données en entrée qui sont déjà classées, c'est aussi à l'algorithme de découvrir par lui-même la structure plus ou moins cachée des données et de former des groupes d'individus dont les caractéristiques sont communes. L'apprentissage non supervisé est utilisé dans plusieurs domaines tels que : [8]

- Médecine : Découverte de classes de patients présentant des caractéristiques physiologiques communes. [8]
- Le traitement de la parole : construction de système de reconnaissance de la voie humaine.
- Archéologie : regroupement des objets selon leurs époques. [8]
- Traitement d'images. [8]
- Classification de documents. [8]

3.1.2 – Algorithmes d'apprentissage non supervisé

Il existe plusieurs types d'algorithmes d'apprentissage non supervisé tels que les algorithmes de partitionnements et les algorithmes de classification hiérarchique.

3.1.2.1 – Le partitionnement

Consiste au regroupement des données suivant leur degré de similarité. L'algorithme le plus célèbre appartenant à cette classe est K-means : c'est un algorithme qui permet de partitionner un ensemble de données automatiquement en K clusters. Il consiste tout d'abord à choisir k points qui représentent les centres des groupes à créer, puis à affecter les autres points aux centres les plus proches. Cette affectation est faite par le calcul de distance entre les points. Plusieurs distances peuvent être définies telles que la distance euclidienne ou la distance de Manhattan. Par la suite nous procédons à une étape de raffinement des groupes de façon itérative, le raffinement se fait par le recalcul des centres des groupes après chaque itération et par une réaffectation des points aux groupes. L'algorithme s'arrête quand aucun point ne bouge. [10]

3.1.2.2 –La classification hiérarchique

Il existe deux types de classification hiérarchique : Ascendante et descendent.

La classification ascendante consiste à utiliser une matrice de similarité afin de partir d'une répartition fine vers un groupe unique. Donc, il s'agit de fusionner les groupes jusqu'à ce qu'on obtient un seul groupe englobant tous les autres. Cette classification peut être représentée par un arbre hiérarchique ou dendrogramme. La classification descendante se présente comme l'inverse de la classification ascendante. Donc il s'agit de décomposer un cluster unique en sous-groupes jusqu'à l'obtention des singletons. [11]

3.2 – Apprentissage supervisé

3.2.1 – Principe

Contrairement à l'apprentissage non supervisé, nous commençons ici par un ensemble de classes connues et définies à l'avance. Nous disposons aussi d'une sélection initiale de données dont la classification est connue. Ces données sont supposées indépendantes et identiquement distribuées. Elles nous servent pour l'apprentissage de l'algorithme. La classification se fait par l'algorithme selon le modèle qu'il a appris. [8]

3.2.2 – Comment classer ?

Classer les documents revient en réalité à déterminer les paramètres de la fonction de classement. Voici l'idée globale de ce qu'on doit faire :

- Il faut disposer d'un corpus d'apprentissage, qui va servir d'entrée à un algorithme d'apprentissage. On sélectionne un autre corpus qui sert pour l'évaluation (corpus de test).
- Il faut d'abord déterminer les descripteurs (variables de la fonction). [1]
- Il faut fournir à l'ordinateur un type de fonctions de classement lui permettant d'associer une catégorie à un texte. (SVMs, Naïve Bayes, Règles de décision, Arbres de décision, Réseaux de neurones, Autres fonctions …). [1]
- On infère, à partir des données, et par des méthodes mathématiques complexes, les paramètres de la fonction de classement utilisé, qui peuvent être : (Coefficients de l'hyperplan dans les SVMs, Distributions de probabilité dans les classificateurs probabilistes, Règles dans les règles de décision, Conditions et branchements dans les arbres de décision, Poids dans les réseaux de neurones …). [1]
- On se fonde sur la connaissance préalable des bonnes catégories pour les documents du corpus d'apprentissage (apprentissage supervisé). [1]

3.2.3 – Algorithmes d'apprentissage supervisé

Il existe plusieurs algorithmes d'apprentissage supervisé, les plus connus sont K plus proches voisins, LLSF, les réseaux de neurones et Naive Baye.

3.2.3.1 – K plus proches voisins

La méthode des k plus proches voisins ou The k-NN classification (k-Nearest Neighbors) est un classifieur à base d'instances qui fait partie des méthodes géométriques utilisant des mesures de distance. k-NN est un algorithme classique d'apprentissage qui a été longtemps à la base des algorithmes de catégorisation des documents, il a été employée avec succès dans de domaine de classification et a engendré toute une famille de classifieurs connus sous le nom de classifieurs paresseux (lazy learns). [13]

Dans ces systèmes, le seul traitement effectué au cours de la phase d'apprentissage est la mémorisation des exemples sous une forme optimale de façon à pouvoir les extraire ensuite rapidement. Chaque texte est représenté dans un espace vectoriel, dont chacun des axes représente un descripteur. [13]

k-NN est un algorithme de catégorisation dans lequel les classes ne sont pas représentées sous forme de texte "prototype". La partie classification est en contrepartie plus coûteuse en temps : Le classifieur calcule la similarité du nouveau texte à catégoriser avec l'ensemble des autres exemples du corpus d'apprentissage, dont les catégories sont déjà connues, puis il sélectionne les k documents les plus proches du document à classer. Ensuite, pour affecter la catégorie, les relations entre ces k documents et les catégories sont évaluées et un score est calculé par catégorie afin d'évaluer la

pertinence de la catégorie au document. La catégorie (ou les catégories) ayant le score le plus élevé (celle qui contient le plus de textes voisins) est affectée au document. Voici son algorithme général :

Paramètres : le nombre k de voisins

Données : un ensemble d'exemples classés (document, classe)

Entrée : un nouveau document D.

1. déterminer les k plus proches documents de D
2. Sélectionner la classe majoritaire C des classes de ces k exemples

Sortie : la classe de D est C

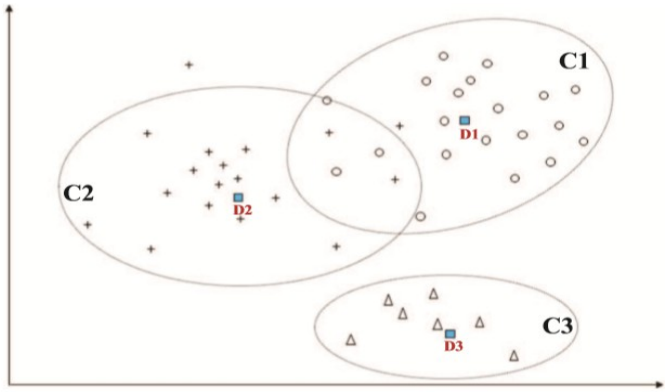

Figure 3. Exemple de la méthode k-PPV [13]

Les trois nouveaux documents D1, D2 et D3 sont associés respectivement aux classes des k plus proches voisins C1, C2 et C3.

3.2.3.2 – Linear least square fit

C'est une approche de mapping développée par Yang, il s'agit d'écrire les données d'apprentissage sous la forme de paires de vecteurs entrée/sortie, le vecteur d'entrée est composé des mots du texte accompagnés de leurs poids respectifs, alors que le vecteur de sortie est composé des différentes catégories avec leur poids binaires (1 si le texte appartient à une catégorie et 0 sinon), la résolution de l'équation suivante permet d'obtenir la matrice des coefficients de régression mot-catégorie. [8]

$F\,LS = \min \| FA - B \|^2$

Où A et B sont deux matrices qui représentent les données d'apprentissage et dont les colonnes sont composées des paires des vecteurs entrée/sortie. La matrice solution permet de donner pour tout texte un vecteur de catégories/poids. Comme pour k-PPV, un seuil est fixé et le document à classer appartient à toute catégorie ayant un poids supérieur au seuil fixé. [8]

3.2.3.3 – Les réseaux de neurones

C'est une structure constituée de suite successive de couches de nœuds et qui permet de définir une fonction de transformation non linéaire des vecteurs d'entrées (composés dans le cas de classification des mots pondérés de leur poids) en vecteur de catégories. La disposition des neurones dans le réseau ainsi que le nombre de couches utilisées ont une influence sur le résultat de

classification. Comparés aux autres méthodes de classification par apprentissage supervisé, les réseaux de neurones ont l'inconvénient que le coût d'apprentissage est assez élevé. [8]

3.2.3.4 – Naïve bayésienne

C'est une méthode de classification probabiliste. Elle consiste à utiliser les probabilités jointes des mots et des catégories pour estimer la probabilité d'une catégorie sachant un texte à classer. Le caractère « naïf » de cette approche est dû au fait que les mots sont considérés indépendants, c'est à dire que la probabilité conditionnelle d'un mot sachant une catégorie est supposée indépendante des probabilités conditionnelles des autres mots sachant la même catégorie, cette assomption rend NB très efficace par rapport aux autres approches bayésiennes. Plusieurs versions de NB sont proposées dans la littérature, le model mixte multinominal par exemple a permis d'avoir de bonnes performances. [8]

3.2.3.5 – SVM (supports vectors machines)

L'algorithme SVM est une méthode d'apprentissage supervisée relativement récente introduite pour résoudre un problème de reconnaissance de formes à deux classes. Le principe de SVM a été proposé par Vapnik à partir de la théorie du risque empirique. [12]

La méthode SVM est un classificateur linéaire utilisant des mesures de distance. En ce qui concerne son application à la problématique de catégorisation de documents, l'algorithme repose sur une interprétation géométrique simple est l'idée générale est de représenter l'espace des exemples (ici des documents) dans un espace vectoriel où chaque document étant un point dans cet espace et de trouver la meilleure séparation possible de cet espace en deux classes. L'espace de séparation est une surface de décision appelée marge, défini par les points « vecteur support ». Ces points se trouvent au minimum de marge. La marge se présente alors comme la plus courte distance entre un vecteur de support et "son" hyperplan. La marge se définit comme la plus petite distance entre les exemples de chaque classe et la surface séparatrice S. [12]

Ainsi la décision s'appuie sur les SVM pour couper l'espace en deux : d'un côté, ce qui est dans la catégorie, de l'autre côté, ce qui n'y est pas. l'approche par SVM permet donc de définir, par apprentissage, un hyperplan dans un espace vectoriel qui sépare au mieux les données de l'ensemble d'apprentissage en deux classes, minimisant le risque d'erreur et maximisant la marge entre deux classes. La qualité de l'hyperplan est déterminée par son écart avec les hyperplans parallèles les plus proches des points de chaque classe. Le meilleur hyperplan est celui qui a la marge la plus importante. [12]

SVM a été étendu pour les points ne pouvant être séparées de manière linéaire (par exemple notre cas des vecteurs de documents), en transformant l'espace initial des vecteurs de données à un espace de dimension supérieure dans lequel les points deviennent séparables linéairement. La figure montre une telle séparation dans le cas d'une séparation linéaire par un hyperplan.

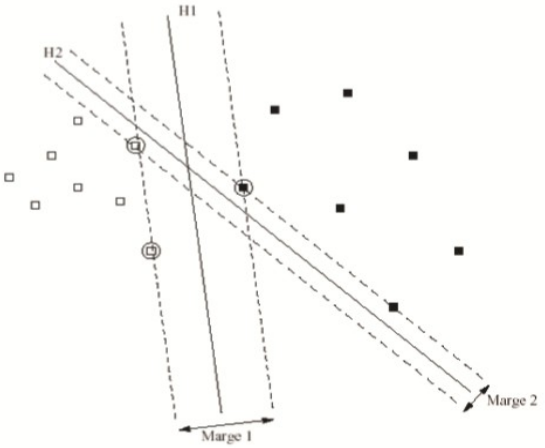

Figure 4. Exemples d'hyperplans séparateurs en dimension deux. [12]

SVM ne peut gérer que des problèmes bi-classes (des extensions commencent à apparaître pour faire du SVM multi-classe). Cet algorithme est particulièrement bien adapté à la catégorisation de textes car il est capable de gérer des vecteurs de grande dimension. Dans la pratique, les catégories sont quasiment toujours linéairement séparables, il n'est donc pas nécessaire d'employer les méthodes avec des noyaux sophistiqués qui alourdissent inutilement les calculs. [12]
SVM a été introduit dans le domaine pour la première fois par Joachims qui a notamment travaillé à rendre SVM compatible avec les données textuelles qui sont caractérisées par de grandes dimensions avec des matrices (documents * termes) très creuses. Depuis, l'approche a été très souvent réutilisé, par exemple pour la détection de courriers électroniques non sollicités ou pour la classification de dépêches. [12]

4 – Mesures de performance de classifieurs

Nous considérons ici un problème simple de classification pour lequel nous nous intéressons à une classe unique C et nous voulons évaluer un système qui nous indique si un document peut être associé ou non à cette classe C. Ce problème est un problème de classification à deux classes

(C et non C noté ¬ C). Si on peut maîtriser ce problème simple, on pourra fusionner par la suite, les mesures de performance de plusieurs systèmes bi-classes afin d'obtenir une mesure de la performance d'un classifieur multi-classes. [1]

4.1 – Matrice de contingence

Pour évaluer un système de classification de ce type, nous utilisons un corpus étiqueté de documents (corpus d'apprentissage) pour lequel on connaît la vraie catégorie de chaque document, et le résultat obtenu par le classifieur. Pour ce corpus, nous pouvons construire la matrice de contingence pour chaque classe, qui fournit 4 informations essentielles : [1]

- Vrai Positif (VP) : Le nombre de documents attribués à une catégorie convenablement. (Documents attribués à leurs vraies catégories)
- Faux Positif (FP) : Le nombre de documents attribués à une catégorie inconvenablement. (Documents attribués à des mauvaises catégories)
- Faux Négatif (FN) : Le nombre de documents inconvenablement non attribués. (Qui auraient dû être attribués à une catégorie mais qui ne l'ont pas été).
- Vrai Négatif (VN) : Le nombre de documents non attribués à une catégorie convenablement (Qui n'ont pas à être attribués à une catégorie, et ne l'ont pas été)

Catégorie Ci		Jugement Expert	
		Oui	Non
Jugement classifieur	Oui	VPi	FPi
	Non	FNi	Vni

Tableau 1. Matrice de contingence de la classe Ci [1]

A partir de ce tableau de contingence, la communauté du TALN calcule divers indicateurs de base, eux-mêmes combinés pour donner d'autres mesures.

4.2 – Précision et Rappel

Certains principes d'évaluation sont utilisés de manière courante dans le domaine de catégorisation de textes. Les performances en termes de classification sont généralement mesurées à partir de deux indicateurs traditionnellement utilisés c'est les mesures de rappel et précision. Initialement elles ont été conçues pour les systèmes de recherche d'information, mais par la suite la communauté de classification de textes les a adoptées. Formellement, pour chaque classe Ci, on calcule deux probabilités qui peuvent être estimées à partir de la matrice de contingence correspondante, ainsi ces deux mesures peuvent être définies de la manière suivante :

1. **Le rappel** étant la proportion de documents correctement classés dans par le système par rapport à tous les documents de la classe Ci.

$$Rappel(Ci) = \frac{Nombre\ de\ document\ bien\ classés\ dans\ Ci}{Nombre\ de\ documents\ de\ la\ classe\ Ci} \quad [1]$$

$$Ri = \frac{VP}{(VPi + FNi)} \quad [1]$$

Le rappel mesure la capacité d'un système de classification à détecter les documents correctement classés. Cependant, un système de classification qui considérerait tous les documents comme pertinents obtiendrait un rappel de 100%. Un rappel fort ou faible n'est

pas suffisant pour évaluer les performances d'un système. Pour cela, on définit la précision. [1]

2. **La précision** est la proportion de documents correctement classés parmi ceux classés par le système dans Ci.

$$Précision(Ci) = \frac{Nombre\ de\ document\ bien\ classés\ dans\ Ci}{Nombre\ de\ documents\ classés\ dans\ Ci} \quad [1]$$

$$Pi = \frac{VPi}{VPI + FPi} \quad [1]$$

La précision mesure la capacité d'un système de classification à ne pas classer un document dans une classe, un document qui ne l'est pas. Comme elle peut aussi être interprétée par la probabilité conditionnelle qu'un document choisi aléatoirement dans la classe soit bien classé par le classifieur.

Ces deux notions sont souvent utilisées dans le domaine de la recherche d'information, car elles reflètent le point de vue de l'utilisateur : si la précision est faible, l'utilisateur sera insatisfait, car il devra perdre du temps à lire des informations qui ne l'intéressent pas. Si le rappel est faible, l'utilisateur n'aura pas accès à une information qu'il souhaitait avoir. Un classifieur parfait doit avoir une précision et un rappel de un (1), mais ces deux exigences sont souvent contradictoires et une très forte précision ne peut être obtenue qu'au prix d'un rappel faible et vice-versa. [1]

4.3 – Bruit et silence

On peut également définir les notions de Bruit (B) et de Silence (S) qui sont respectivement les notions complémentaires de la précision et du rappel. On utilise aussi la notion de bruit qui présente le problème selon le point de vue opposé de la précision. Le bruit est le pourcentage de textes incorrectement associés à une classe par le système :

$$Bruit(B) = 1 - Precision(P) = \frac{FPi}{VPi + FPi} \quad [1]$$

La notion de silence est le point de vue opposé du rappel. Le silence est le pourcentage de textes à associer à une classe incorrectement non classés par le système :

$$Silence(S) = 1 - Precision(P) = \frac{FPi}{VPi + FNi} \quad [1]$$

4.4 – Taux de succès et taux d'erreur

Le taux de succès ou l'exactitude Acc (Accuracy rate) et le taux d'erreur Err (Error rate) sont deux mesures souvent utilisées par la communauté de l'apprentissage automatique. Le taux de succès désigne le pourcentage d'exemples bien classés par le classifieur, tandis que le taux d'erreur désigne le pourcentage d'exemples mal classés. Les deux taux sont estimés comme suit :

$$Acc = \frac{VP + VN}{VP + VN + FP + FN} \quad [1]$$

$$Err = \frac{FP + FN}{VP + VN + FP + FN} = 1 - Acc \quad [1]$$

4.5 – F-measure

Observés conjointement, les indicateurs les plus célèbres à savoir le rappel et la précision, sont une estimation courante de la performance d'un système de classification. Cependant plusieurs mesures ont été développées afin de synthétiser cette double information. Nous ne retiendrons ici la mesure Fβ. La F-mesure est la mesure de synthèse communément adoptée depuis les années 80 pour évaluer les algorithmes de classification de données textuelles à partir de la précision et du rappel. Elle est employée indifféremment pour la classification (Non supervisé) ou la catégorisation (Supervisé), pour la problématique de recherche d'information ou de classification. Elle permet donc, de combiner, selon un paramètre β, rappel et précision. On définit la mesure Fβ comme la moyenne harmonique entre le rappel et la précision :

$$F\beta = \frac{(\beta^2+1)*Precision*Rappel}{\beta^2*Precision+Rappel} \quad [1]$$

Pour utiliser cette mesure, il est donc nécessaire de fixer préalablement un seuil de décision pour le classement, et de calculer la valeur de Fβ pour ce seuil. Le paramètre β permet de choisir l'importance relative que l'on souhaite donner à chaque quantité. On choisit en général de donner la même importance aux deux critères, donc habituellement, la valeur de β est fixée à 1 et la mesure est ainsi notée F1 (noté F) qui s'écrit :

$$F1 = \frac{2*P*R}{P+R} \quad [1]$$

5 – Conclusion

Suite à la présentation de la classification de textes au chapitre 1, il a été question ici des algorithmes d'apprentissage mis à l'œuvre pour traiter ce problème. La variété de techniques d'apprentissage amène une application informatique à classer des textes avec autonomie.
On comprend maintenant mieux le fonctionnement des classificateurs. En plus, la deuxième partie du chapitre a fait la lumière sur le processus d'évaluation des classificateurs.

Chapitre 3

Conception et implémentation

1 – Introduction

Dans ce chapitre, nous allons décrire les environnements matériels et logiciel utilisés pour réaliser notre plate-forme ainsi que l'application développée en termes de conception et d'implémentation.

2 – Environnement de développement

2.1 – L'environnement matériel

Notre plate-forme a été développée sur une machine Intel Core i5, sous le système d'exploitation Microsoft Windows 10.

2.2 – L'environnement logiciel

L'implémentation de l'application a été réalisée avec le langage de programmation Java sous la plate-forme NetBeans IDE8.1.

2.2.1 – Le langage Java :

Nous avons utilisé comme langage de programmation le langage objet « java ». Le choix de java se justifie par les avantages suivants :

- C'est un langage bien connu et largement répandu. Il existe de nombreuses librairies qui facilitent le développement des applications.
- Les applications java s'exécutent en utilisant une machine virtuelle, ce qui les rend indépendantes du système d'exploitation.
- Des machines virtuelles java ont été développées pour la plupart des systèmes actuels, ce qui facilite la portabilité des applications java.
- Les compilateurs java sont gratuits.
- Java permet de définir facilement des interfaces graphiques agréables à utiliser.

Nous avons réalisé notre application java en utilisant JDK 1.6.0.

Ses caractéristiques ainsi que la richesse de sa communauté lui ont permis d'être le choix préféré pour le développement de mon application. [19]

2.2.2 – Environnement NetBeans 8.1

NetBeans est un projet open source ayant un succès et une base d'utilisateur très large, une communauté en croissance constante, et près de 100 partenaires mondiaux et des centaines de milliers d'utilisateur à travers le monde. Sun Microsystems a fondé le projet open source NetBeans en Juin 2000 et continue d'être le sponsor principal du projet.

Aujourd'hui, deux projets existent: L'IDE NetBeans et la Plateforme NetBeans.

L'IDE NetBeans est un environnement de développement, un outil pour les programmeurs pour écrire, compiler, déboguer et déployer des programmes. Il est écrit en Java mais peut supporter n'importe quel langage de programmation. Il y a également un grand nombre de modules pour

étendre l'IDE NetBeans. L'IDE NetBeans est un produit gratuit, sans aucune restriction quant à son usage.
Également disponible, La Plate-forme NetBeans; une fondation modulable et extensible utilisée comme brique logicielle pour la création d'applications bureautiques. Les partenaires privilégiés fournissent des modules à valeurs rajoutées qui s'intègrent facilement à la Plate-forme et peuvent être utilisés pour développer ses propres outils et solutions. [14]

3 – Architecture général de la plate-forme

Dans cette section, nous allons présenter notre plate-forme. Nous décrirons notamment son architecture et son fonctionnement.

3.1 – Architecture structurelle simplifiée du logiciel

La Plate-forme peut être représentée comme un composant logiciel qui reçoit en entrée un Corpus textuels à traiter et fournit en sortie l'affiliation de chaque texte a sa catégorie.

3.2 – Architecture détaillée de l'application

Dans cette section nous allons détailler l'architecture de notre application. Nous présentons les différents composants qui forment notre Plate-forme. Pour chaque composant nous présentons sa structure et son fonctionnement.

3.2.1 – L'indexation automatique

L'indexation automatique est l'opération qui consiste à faire reconnaître par l'ordinateur des termes figurant dans le titre, le résumé, le texte complet (s'il est enregistré avec la notice documentaire) et parfois même l'indexation humaine, et à employer ces termes, soit tels quels soit après conversion en d'autres termes équivalents ou conceptuellement voisins, pour en faire des critères incorporés dans le fichier de recherche et utilisables pour retrouver le document. [15]

En France, l'âge d'or de la recherche en indexation automatique, dans les années 1965 - 70, a été marqué par les travaux du laboratoire d'automatique documentaire et linguistique du CNRS, dirigé par J.CL. GARDIN. C'était une époque marquée par les recherches sur le traitement automatique des langues, les études sur la traduction automatique le démontrent. On a en effet rêvé pouvoir traduire les poèmes de Shakespeare, les romans de Balzac, ou encore pour une activité un peu plus documentaire, produire des résumés automatiquement. Paradoxalement, c'est avec l'avancée des recherches en intelligence artificielle et le développement de l'informatique que les chercheurs sont devenus plus modestes, car on est passé de la traduction automatique à la traduction assistée par ordinateur (TAO), et de l'indexation automatique à l'indexation assistée par ordinateur (IAO).

Il existe deux types d'indexation automatique ou semi-automatique : le premier consiste à enrichir automatiquement l'indexation humaine par autopostage générique ou encore une indexation automatique non sélective (prise en compte de tous les mots non vides du document). Ce type d'indexation est utilisé de façon généralisée. Le deuxième type d'indexation automatique est l'indexation automatique sélective, c'est-à-dire une prise en compte de certains termes seulement jugés par le système comme les plus représentatifs du contenu du document soit en langage naturel soit en langage contrôlé. Une grande majorité des systèmes bâtis sur ce type d'indexation était

encore en expérimentation jusqu'à l'intégration de module linguistique depuis une dizaine d'années environ. [15]

3.2.1.1 – Indexation par Lucene

Lucene est un moteur de recherche textuelle Open Source et passant bien à l'échelle fourni par la fondation Apache. Il est utiliser dans des applications commerciales ou Open Source. Les puissantes APIs de Lucene se concentrent surtout sur l'indexation et la recherche. Il peut être utilisé pour ajouter des capacités d'indexation à des applications comme des clients de courrier, des listes de diffusion, des applications effectuant des recherches sur Internet ou dans une base de données, etc. Des sites Internet tels que Wikipedia, TheServerSide, jGuru, et LinkedIn utilisent Lucene. [16]

Lucene a beaucoup d'atouts :

- Il utilise des algorithmes de recherche puissants, exacts et efficaces. [16]
- Il calcule un score pour chaque document qui correspond à des critères donnés et retourne les documents les plus appropriés classés par score. [16]
- Il propose de nombreux types de requêtes, tels que PhraseQuery, WildcardQuery, RangeQuery, FuzzyQuery, BooleanQuery, et d'autres encore. [16]
- Il permet l'analyse d'expressions de requêtage riches du type de celles qui sont saisies par des êtres humains. [16]
- Il permet aux utilisateurs d'étendre le comportement de la recherche en utilisant des tri personnalisés, des filtres et l'analyse d'expressions de requêtage. [16]
- Il utilise un mécanisme de verrouillage basé sur les fichiers pour empêcher les modifications d'index concurrentes. [16]
- Il permet d'indexer et de rechercher simultanément. [16]

Comme le montre la Figure, construire une application de recherche complète avec Lucene implique principalement l'indexation et la recherche des données, et l'affichage des résultats d'une recherche. [16]

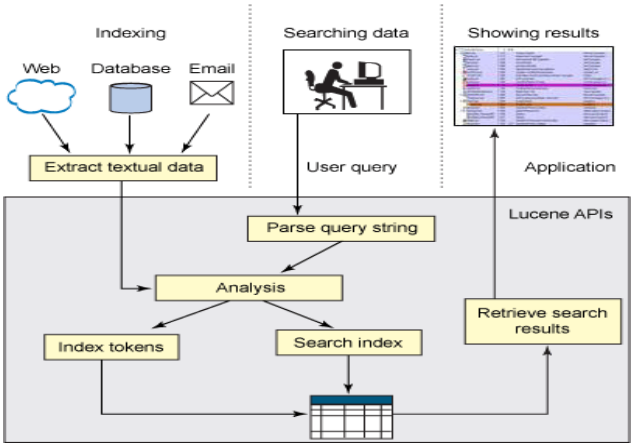

Figure 5. Les étapes de la construction d'applications avec Lucene [16]

3.2.2 – Ontologie

Pour remédier a notre problématique, on va utiliser les ontologies. Une ontologie définit les termes et les relations de base du vocabulaire d'un domaine ainsi que les règles qui indiquent comment combiner les termes et les relations de façon à pouvoir étendre le vocabulaire. [18]

3.2.2.1 – WordNet

WordNet est une ressource lexicale de large couverture, développée depuis plus de 20 ans pour la langue anglaise. Elle est utilisable librement, y compris pour un usage commercial, ce qui en a favorisé une diffusion très large. Plusieurs autres ressources linguistiques ont été constituées (manuellement ou automatiquement) à partir de, en extension à, ou en complément à WordNet. Des programmes issus du monde de l'Intelligence Artificielle ont également établi des passerelles avec WordNet. [17]

L'ensemble constitue un « écosystème » complet couvrant des aspects lexicaux, syntaxiques et sémantiques. Combinées, ces ressources fournissent un point de départ intéressant pour des développements sémantiques en TAL ou dans le cadre du Web sémantique, tels que la recherche d'information, l'inférence pour la compréhension automatique de textes, la désambiguïsation lexicale ou la résolution d'anaphores. [17]

Figure 6. Ressources disposant d'une traçabilité vers WordNet. [17]

3.2.2.2 – Principe de WordNet

WordNet (Miller, 1995) est une base de données lexicale développée depuis 1985 par des linguistes du laboratoire des sciences cognitives de l'université de Princeton. C'est un réseau sémantique de la langue anglaise, qui se fonde sur une théorie psychologique du langage. La première version diffusée remonte à juin 1991. [17]

Son but est de répertorier, classifier et mettre en relation de diverses manières le contenu sémantique et lexical de la langue anglaise. Le système se présente sous la forme d'une base de données électronique qu'on peut télécharger sur un système local. Des interfaces de programmation sont disponibles pour de nombreux langages. [17]

3.2.2.3 – Notion de synset

Le synset (ensemble de synonymes) est la composante atomique sur laquelle repose WordNet. Un synset correspond à un groupe de mots interchangeables, dénotant un sens ou un usage particulier. Un synset est défini d'une façon différentielle par les relations qu'il entretient avec les sens voisins. [17]

Les noms et verbes sont organisés en hiérarchies. Des relations d'hyperonymie « est-un » et d'hyponymie relient les « ancêtres » des noms et des verbes avec leurs « spécialisations ». Au niveau racine, ces hiérarchies sont organisées en types de base. Le réseau des noms est bien plus profond que celui des autres parties du discours. [17]

L'organisation des adjectifs est différente. Un sens « tête » joue un rôle d'attracteur; des adjectifs

« satellites » lui sont reliés par des relations de synonymie. On a donc une partition de l'ensemble des adjectifs en petits groupes. Les adverbes sont le plus souvent définis par les adjectifs dont ils dérivent. Ils héritent donc de la structure des adjectifs. [17]

3.2.2.4 Relations sémantique (entre synsets)

Le tableau suivant présente un comptage des relations sémantiques de WordNet 2.1 par catégorie.

Relation	Entre	Nombre	Exemple
Hypernym/Hyponm	Verbe / verbe	13124	EXHALE / BREATHE
	Nom / nom	75134	CAT / FELINE
Instance Hyponym	Nom / nom	8515	EIFFEL TOWER / TOWER
Part	Nom / nom	8874	FRANCE / EUROPE
Member	Nom / nom	12262	FRANCE / EUROPEAN UNION
Substance	Nom / nom	793	SERUM / BLOOD
Attribute	Adjectif / nom	643	INACCURATE / ACCURACy
Verb Group	Verbe / verbe	1748	GELATINIZE#1 / GELATINIZE#2
Verb Entailment	Verbe / verbe	409	DREAM / SLEEP
Verb Cause	Verbe / verbe	219	ANESTHETIZE / SLEEP
Adjective Similar	Adjectif / adjectif	22622	DYING / MORIBUND

Topic Domain	Nom / adjectif	1108	COMPUTER SCIENCE / ADDRESSABLE
	Nom / nom	4146	COMPUTER SCIENCE / COMPUTER
	Nom / adverbe	37	
	Nom / verbe	1236	COMPUTER SCIENCE / CASCADE
Region Domain	Nom / adjectif	75	
	Nom / nom	1246	FRENCH / FRANCE
Usage Domain	Nom / adjectif	227	
	Nom / nom	563	NEUTRALIZATION / EUPHEMISM
	Nom / adverbe	73	
	Nom / verbe	14	
See Also	Adjectif / adjectif	2683	BLACK/DARK

3.2.2.5 – Mesures de similarité

Une utilisation possible de l'ontologie fournie par WordNet est la définition de métriques heuristiques de « distance sémantique » entre les synsets. Cette métrique est basée sur la distance à parcourir dans le graphe, combinée ou non avec le Contenu Informationnel. Elle permet de quantifier la similarité de deux concepts. Elle peut également servir dans un cadre de désambiguïsation lexicale. [17]

3.2.2 – Apprentissage supervisé

On a utilisé l'algorithme KNN qui permet de calculer le plus proche voisin.

Cette algorithme est proposée dans la section II.3.2.3. Nous avons implémenté dans notre projet. Nous donnons ci-dessous son pseudo-code.

Algorithme de classification par KNN

Paramètre :

Soit D = {(x', c), cϵC)} l'ensemble d'apprentissage

Soit X l'exemple dont on souhaite déterminer sa classe

Début

Pour chaque ((x', c)ϵD) faire

Calculer la distance dist (x, x')

Pour chaque {x'ϵ kppv(x)} faire

Trier dist (x, x') par ordre croissant

Choisir les K premier

Compter le nombre d'occurrence de chaque classe

Attribuer a x la classe la plus fréquente ;

Fin

4 – Illustration du système

Lorsque l'utilisateur lance l'application la fenêtre principale s'affiche

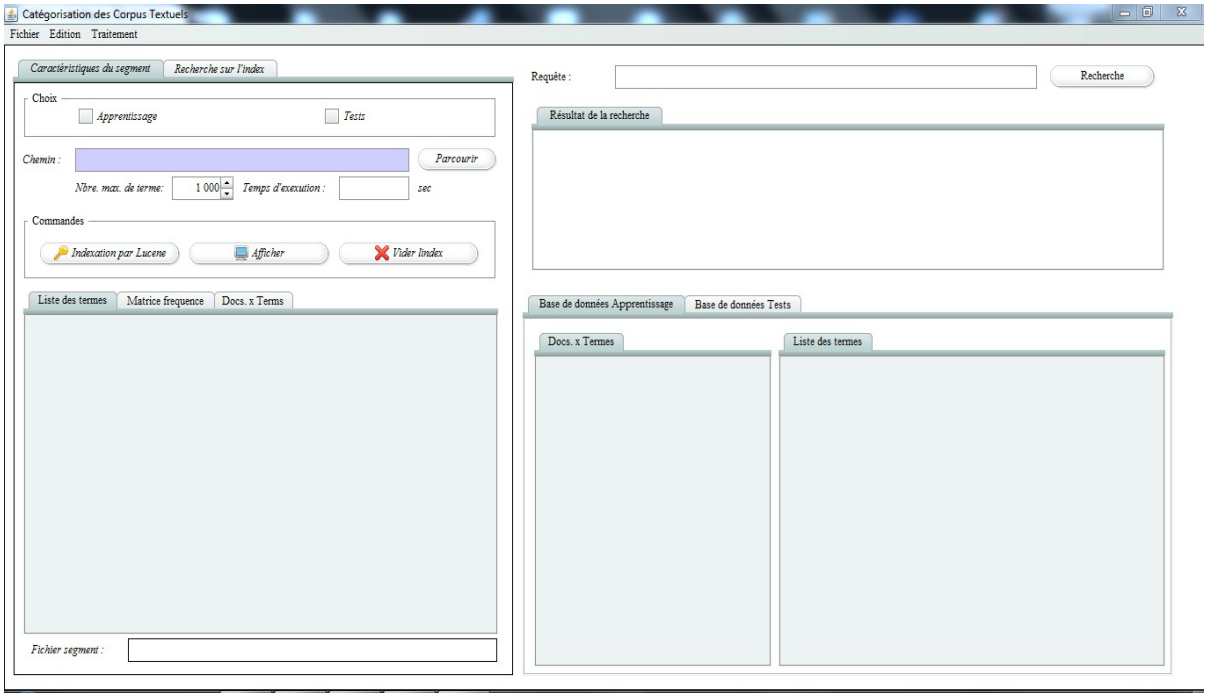

Figure 7. fenêtre principale

Après avoir lancer notre application, on va faire l'indexation de notre corpus (train & test).

On va commencer par le dossier train, on coche la case Apprentissage, on spécifie le nombre de termes, puis en clique sur le bouton Indexation par lucene. Une fois l'indexation finie on affiche le résultat en cliquant sur Afficher. Une fois fini on va la stoker dans une base de donnée afin de pouvoir la charger ultérieurement.

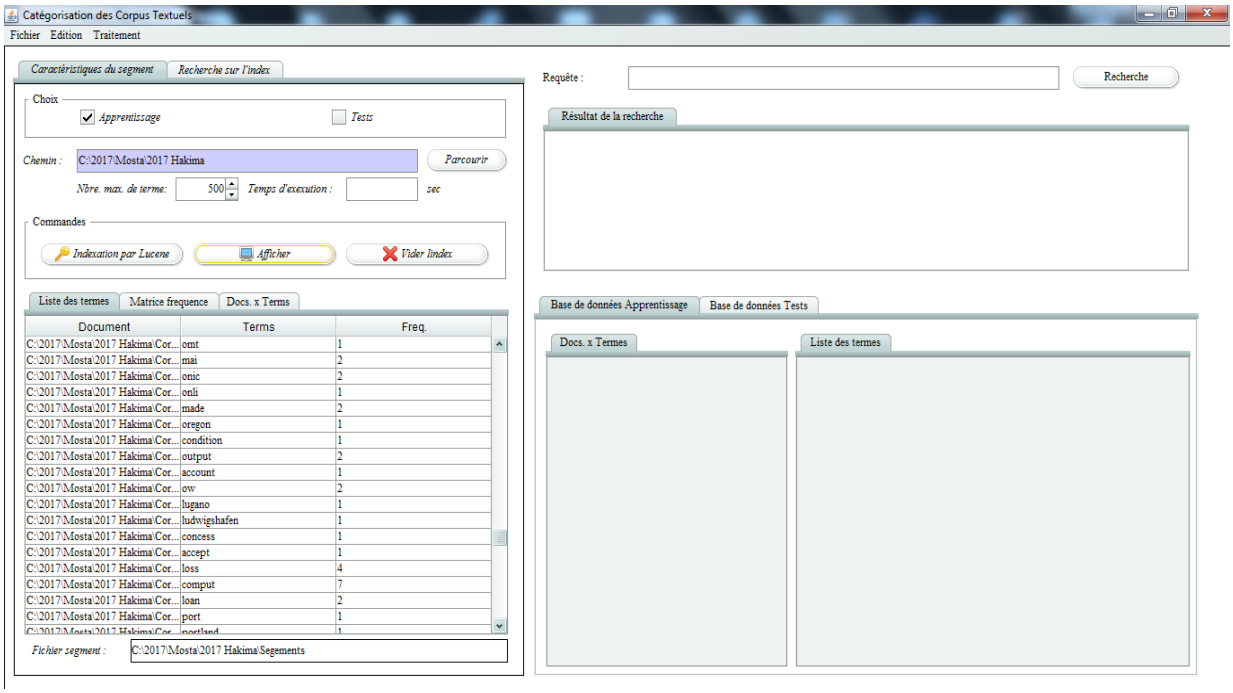

Figure 8. Indexation du dossier train

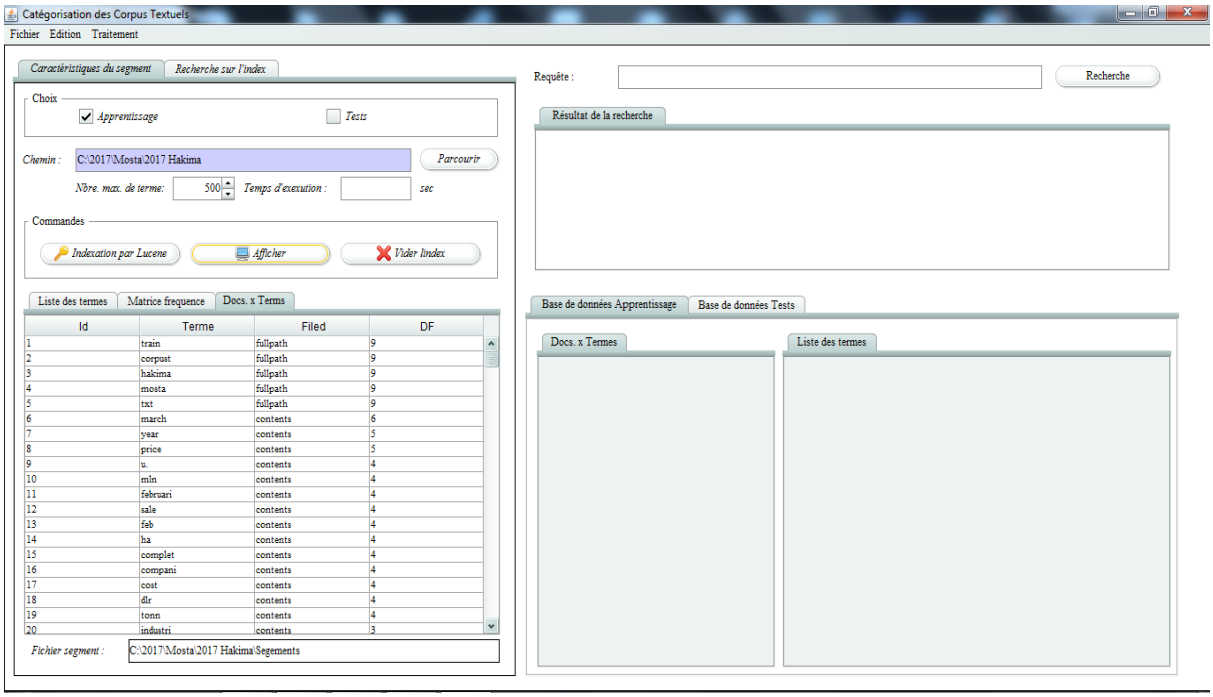

Figure 9. Le poids de chaque terme par rapport au corpus

Lors de l'indexation Lucene calcul le poids de chaque terme en utilisant la mesure TfiDf

On refait la même procédure pour le dossier test, indexation, affichage puis stockage.

On peux charger nos deux base de donnée et les afficher. La base de donnée a été crée pour éviter de faire l'étape de l'indexation plusieurs fois, car cela prend beaucoup de temps.

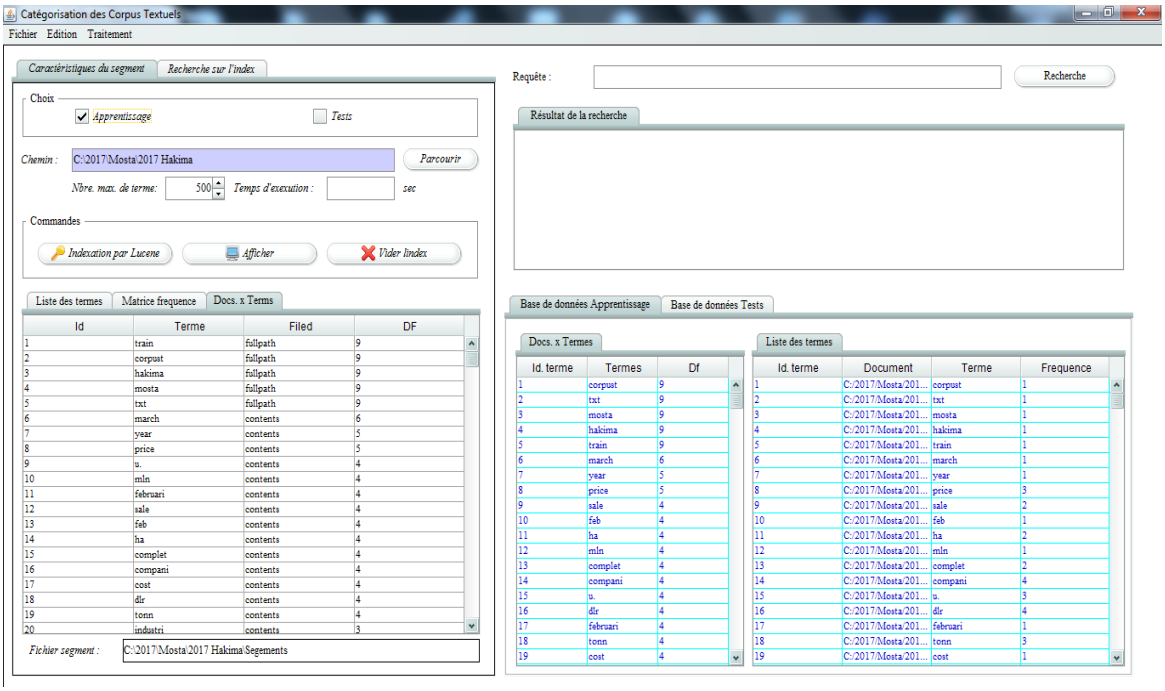

Figure 10. Affichage de la base de donnée (Train & Test)

Une fois l'indexation fini on passe a l'étape traitement du texte par WordNet, nous disposons des termes extrait a partir de l'étape indexation.

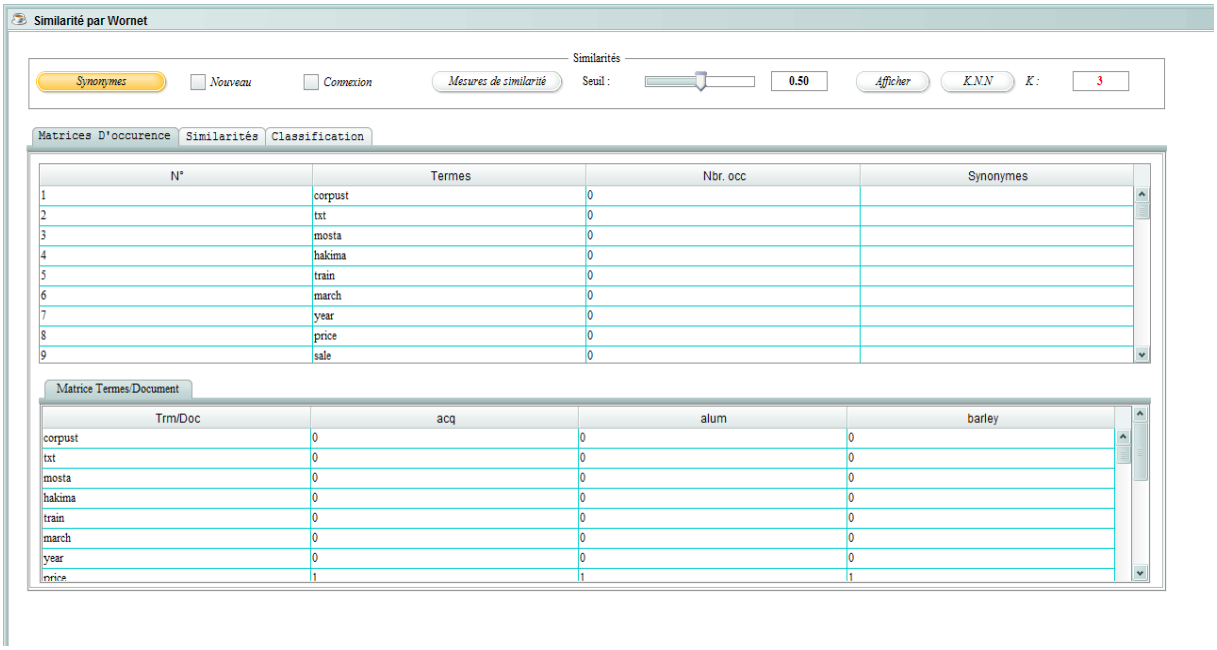

Figure 11. Traitement WordNet

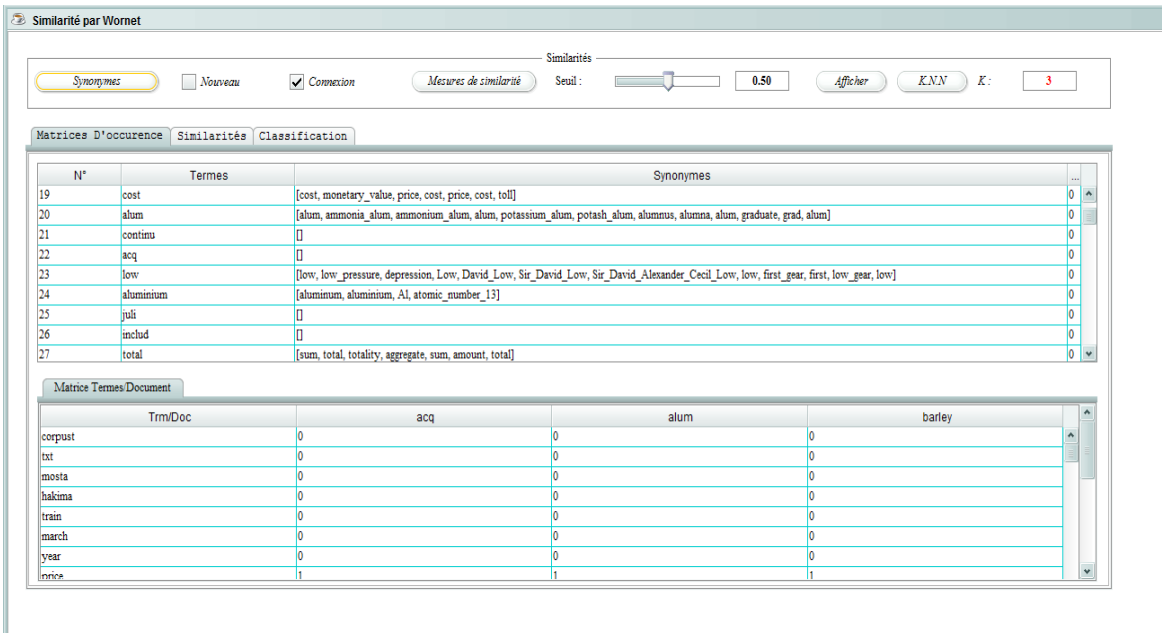

Figure 12. Affichage des synonyme

On clique sur le bouton synonyme pour lancer WordNet et faire la combinaison de chaque termes.

La matrice Termes/Document et une matrice binaire, qui trie les termes par rapport au catégories

(1 si le mot existe dans la catégorie, o sinon)

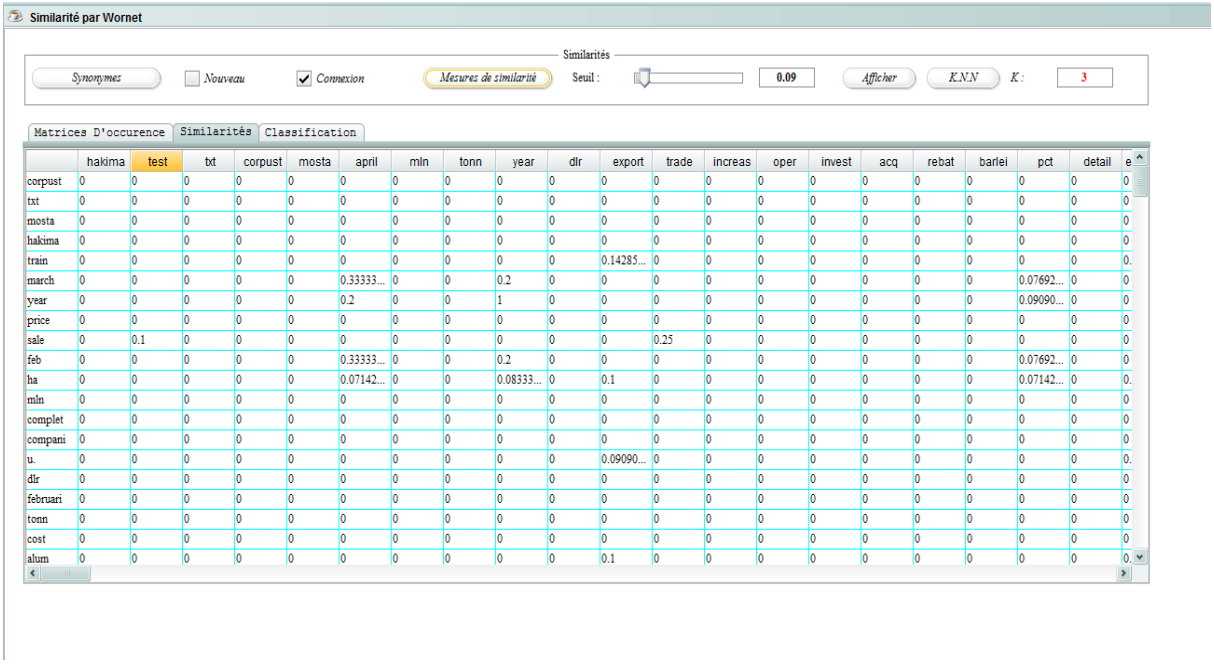

Figure 13. Calcul de Mesure de similarité

Après avoir fait le traitement des synonymes, on passe au calcul de la mesure de similarité.

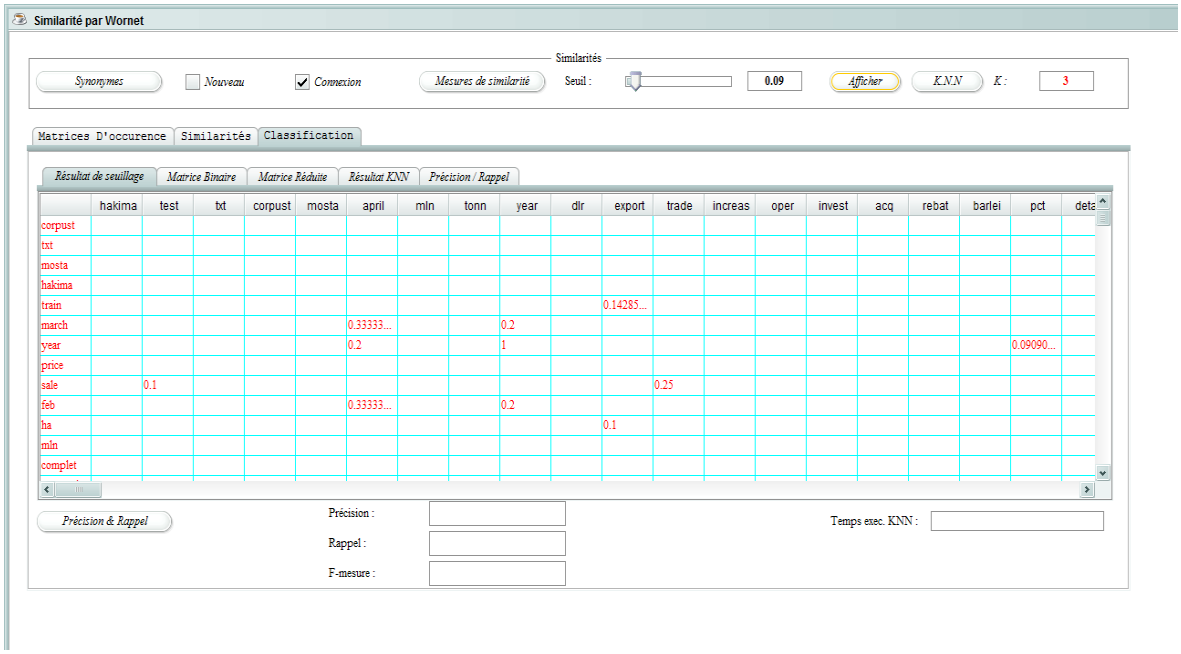

Figure 14. Interface Classification

Le résultat du seuillage a été déduit a partir de la matrice similarité afin de pouvoir calculer les mesures d'évaluation.

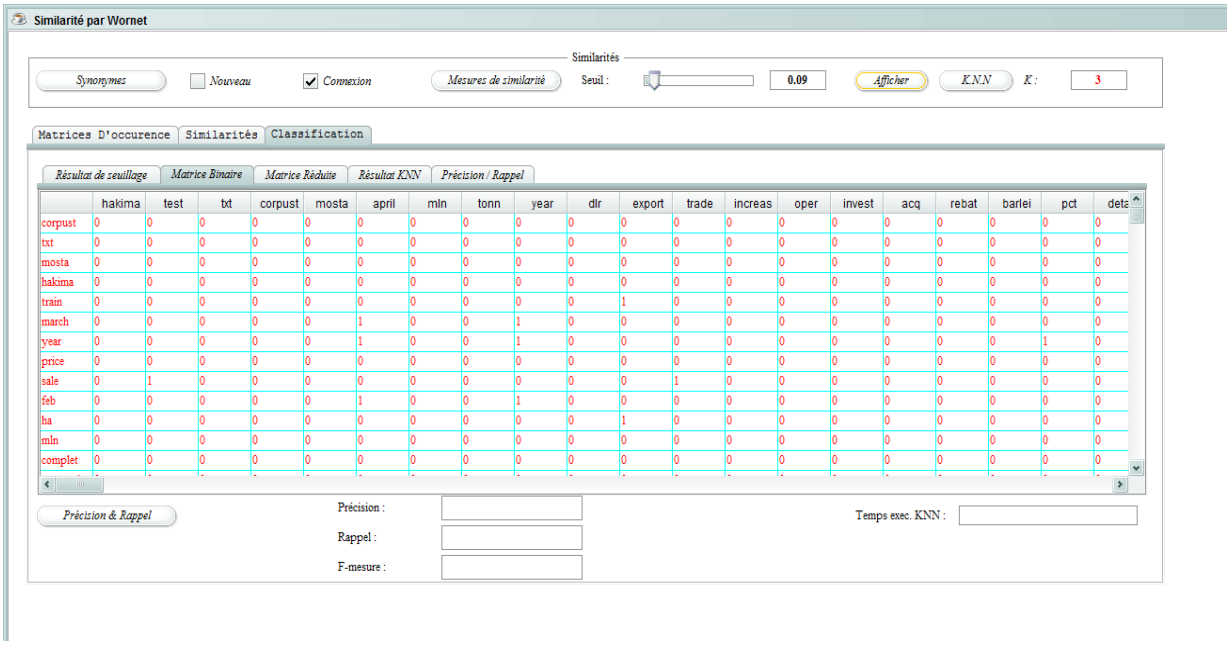

Figure 15. Matrice binaire

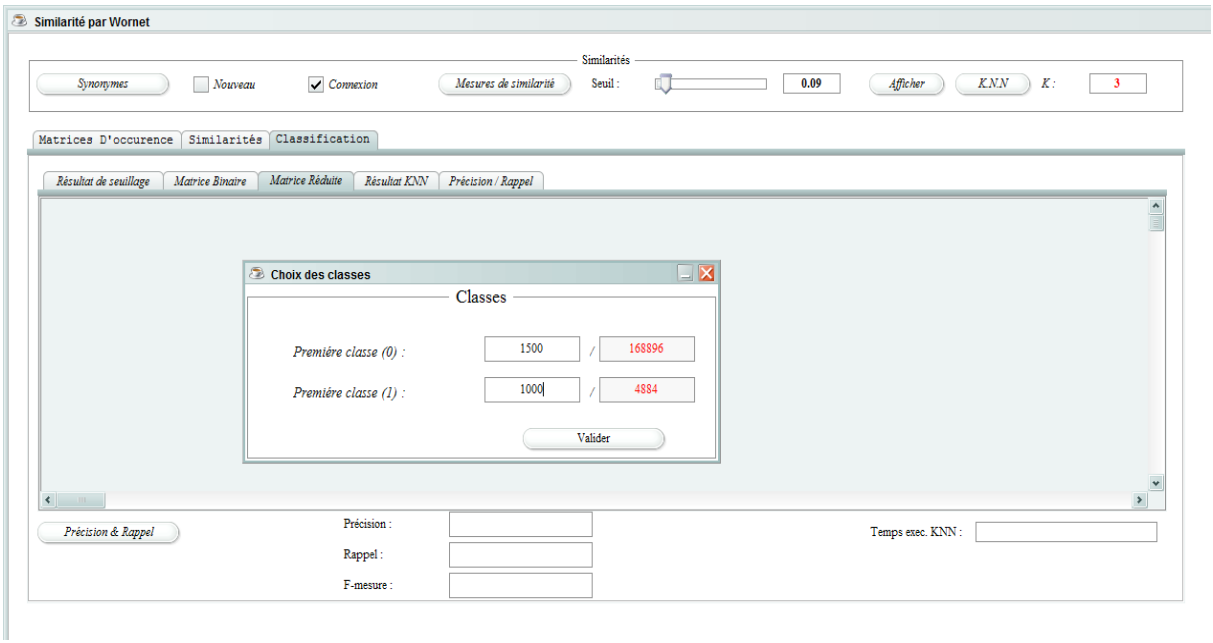

Figure 16. Classe pour définir nb 0 et nb 1

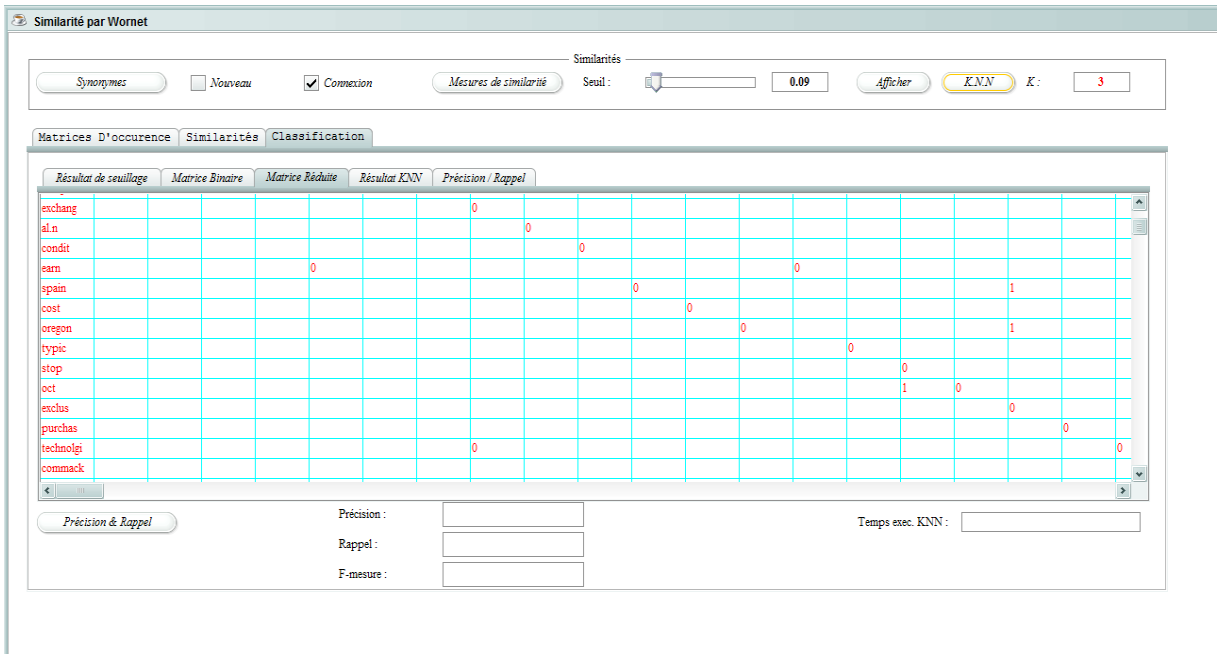

Figure 17. Matrice réduite

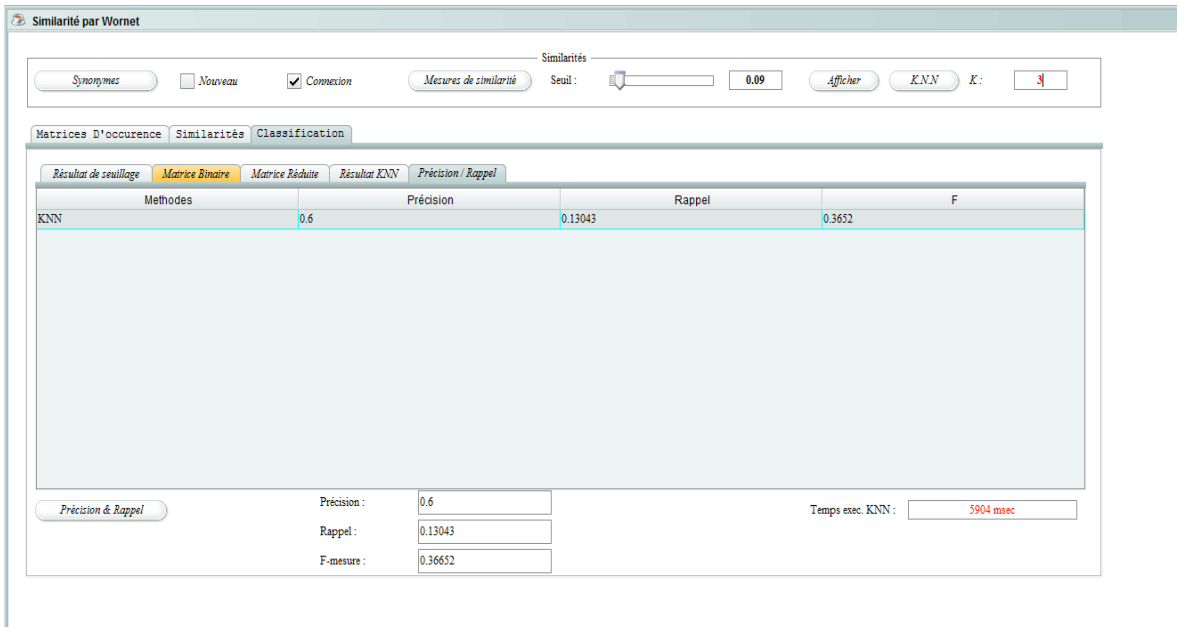

Figure 18. Mesures d'évaluation

5 – Conclusion

Dans ce chapitre, nous avons présenté le volet technique de notre application. Nous avons défini les outils utilisés lors de l'implémentation de notre application. Nous avons illustré par quelques pages écran de la plate-forme.

Conclusion Générale

Les travaux menés dans ce mémoire s'inscrivent dans le cadre de la catégorisation des corpus textuels. Nous avons commencé d'abord par expliquer qu'est-ce que c'est la classification de textes, quels sont ses composants et comment peut on la développer, ensuite nous avons défini les algorithmes d'apprentissage utiliser dans la classification. En fin nous avons présenté notre application avec quelques pages écran.

Dans notre travail nous avons développé une plate-forme informatique qui permet d'indexer deux dossier (Train & Test) et attribuer un poids à chaque terme en utilisant la mesure TfiDF. La plate-forme proposée permet de réaliser notamment les deux fonctionnalités suivantes. La première consiste en la recherche de synonyme et le calcul d'une mesure de similarité de chaque terme en utilisant WordNet. La seconde fonctionnalité permet d'identifier la catégorie d'un fichier modèle à l'aide d'une technique d'apprentissage automatique en l'occurrence l'algorithme des k-plus proches voisins.

En guise de perspectives pour notre travail, nous proposons les orientations suivantes :

Intégrer à la plate-forme de nouvelles techniques de combinaison automatique de valeurs de similarités et offrir un support à leur comparaison.

Intégrer à la plate-forme de nouvelles méthodes d'apprentissage et mettre en œuvre les outils nécessaires à leur comparaison.

Bibliographie

[1] : MATALLAH Hocine. « Classification Automatique de Textes Approche Orientée Agent ». Université Aboubekr Belkaid Tlemcen.

[2] : G.Brown, H.A.Chong. « The Guru System in TREC-6 »

[3] : Ludovic Denoyer. « Apprentissage et inférence statistique dans les bases de documents structuré Application aux corpus de documents textuels ». Paris 6. 2004.

[4] : A.McCallum, R.Rosenfeld, T.Mitchell, A.Y.Nigam. « Improving Text Classification by Shrinkage in a Hierarchy of Classes ».

[5] : F.Sebastiani. « Machine learning in automated text categorization ».

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

[7] : Simon Réhel. « Catégorisation automatique de textes et cooccurrence de mots provenant de documents non étiquetés ». Faculté des sciences et de génie université LAVAL QUÉBEC. Janvier 2005

[8] : Ameni Bouaziz. « Catégorisation automatique de news à l'aide de techniques d'apprentissage supervisé ». Rapport de projet de fin d'études. Université Nice Sophia Antipolis.

[9] : François Yvon. « Des apprentis pour le traitement automatique des langues ».Université Pierre et Marie Curie – Paris.

[10] : Kiri Wagsta, Claire Cardie. « Constrained K-means Clustering with Background Knowledge ».

[11] : Jean-Pièrre Nakache, Josian Confais. « Approche de classification pragmatique »

[12] : V.Vapnik. « The Nature of Statistical Learning »

[13] : Y.Yang. « Problem-based Learning on the World Wide Web in an Undergraduate Kinesiology Class: an Integrative Approach to Education »

[15] : Mabrouka EL HACHANI. « L'INDEXATION AUTOMATIQUE ». DEA SCIENCES DE L'INFORMATION ET COMMUNICATION. Mars 1997

[17] : François-Régis Chaumartin. « WordNet et son écosystème : un ensemble de ressources linguistiques de large couverture ». Université Paris 7. Avril 2007.

[18] : BENAISSA Bedr-Eddine. « Construction semi-automatique d'ontologies à partir de textes arabes ».Université Abou Bakr Belkaid. 2012.

Bibliographie web

[14] : https://netbeans.org/index_fr.html

[16] : http://www.torrefacteurjava.fr/content/utiliser-apache-lucene-pour-effectuer-desrecherches-textuelles

[19] : http://www.futura-sciences.com/tech/definitions/internet-java-485/