REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE MINISTERE DE L'ENSEIGNEMENT
SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE SAÂD DAHLAB BLIDA
FACULTE DES SCIENCES
DEPARTEMENT D'INFORMATIQUE

Mémoire présenté pour l'obtention du diplôme de

Master

en

INFORMATIQUE

Option : Ingénierie des logicielles

Classification des documents médicaux basée sur le Text Mining

Présenté par

Dahmani Houria

Soutenu en 2012 devant la commission du jury composée de :

Président (e) M...Bennamane.....UNIVERSITE DE BLIDA

Examinateur M ...Bakli.....UNIVERSITE DE BLIDA

Examinateur M .....Salhi.....UNIVERSITE DE BLIDA

Promoteur Mme A.ELMAOUHEB, Chef du département réseaux, CERIST

A mon mari Ali

A mes enfants Samy,Maria et Ilyas

A mes parents et beaux parents

A mes sœurs et mes frères

A Guilen

A touts mes collègues du Cerist

A toutes les personnes qui m'aiment

Qu'ils trouvent ici l'expression de ma sincère gratitude

Remerciements

J e remercie ma structure le CERIST, qui m'a permis de rejoindre les bancs de l'université et d'aborder des perspectives nouvelles.

Je remercie Mme ELMAOUHAB Aouaouche d'avoir accepté la lourde charge du promoteur, et d'accorder de son temps pour suivre mon travail.

Je souhaite également adresser mes remerciements à l'ensemble des membres du jury, d'avoir accepté de lire, de juger et de discuter mon travail.

Je remercie également Mr BALA, Mr Ben Nouar et Melle Salhi, pour leur précieuse assistance, pendant le cursus du Master.

Un grand merci pour Insaf, Bdiaa, pour leurs précieuses orientations, qui m'ont permis d'affiner mon travail.

Mes remerciements vont aussi à toute l'administration de la faculté informatique, pour leur gentillesse.

Mes remerciements les plus distingués pour mon mari, Ali, pour son assistance morale et ses encouragements, pour son aide précieuse pendant toute ma formation.

Je remercie également mes chères collègues du Cerist, pour leur soutien moral, spécialement :Amal, Ikram, Hassina, Kahina, Rabab, Nassima et Rym.

Que toutes les personnes qui ont attribué de près ou de loin à l'élaboration de ce travail trouvent ici l'expression de ma haute gratitude.

Résumé/ Abstract/ملخص

Avec l'avènement de l'informatique et l'explosion de nombre de documents stockés sur les supports électroniques et sur le web, qui sont à plus de 80% de type texte, l'utilisation de technologie facilitant leur traitement et leur analyse est devenu indispensable, pour aider les utilisateurs de ces masses de données à les explorer puis à les organiser.

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

A l'égard des différentes approches de classification automatique de textes, décrites dans l'état de l'art, nous avons utilisé l'approche non supervisée (algorithme Kmeans) pour étiqueter nos documents et l'approche supervisée (algorithme Naive Bayes) pour classer les nouveaux documents.

L'objectif principal de notre travail, est d'offrir un modèle fiable de classification de documents médicaux.

Nous utilisons MEDLINE comme corpus de textes, sur lequel nous menons nos expérimentations.

Mots Clés : Catégorisation, clustering, Classification, Texte, Apprentissage, Text Mining, Evaluation, Kmeans, Naïve Bayes, MEDLINE.

With the advent of computers and the explosion of the number of documents stored on electronic media and on the web, which are more than 80% of type text, the use of technology to facilitate their processing and analysis has become essential to help users to explore the masses data and to organize them.

Thus, Text Mining and precisely automatic text classification, which consists in assigning a document to one or more categories, is becoming increasingly recognized as a key technology, the results are also useful for finding information in knowledge extraction or on the Internet (search engines) and within companies (ranking internal documents, news agencies,etc.)..

With regard to the different approaches of automatic text classification, described in the prior art, we used an unsupervised approach (algorithm Kmeans) to label our documents and supervised approach (Naive Bayes algorithm) for classifying new documents.

The main objective of our work is to provide a reliable classification of medical documents.

We use MEDLINE as text corpus, in which we conduct our experiments.

Keywords: categorization, clustering, classification, Text, Learning, Text Mining, Evaluation, Kmeans, Naïve Bayes, MEDLINE.

مع ظهور الحواسيب وانفجار عدد الوثائق المخزنة على وسائط الإعلام الإلكترونية وعلى شبكة الإنترنت، والتي هي أكثر من 80٪ على شكل نص، استخدام التكنولوجيا لتسهيل المعالجة والتحليل أصبح ضروري لمساعدة المستخدمين على استكشاف البيانات وتنظيمها.

وهكذا، تحليل النص و التصنيف الدقيق له، والذي يعود الى اسناده الى صنف واحد ، أصبحت من التكنولوجيات الرئيسية.، كذلك فإن النتائج هي أيضا مفيدة للعثور على معلومات جيدة لاستخراج المعرفة خاصة على شبكة الإنترنت (محركات البحث) وداخل الشركات (ترتيب الوثائق الداخلية ، وكالات الأنباء، وما إلى ذلك) .

وفيما يتعلق بالأساليب التلقائية لتصنيف النص فقد استخدمنا نهج غير خاضع للرقابة (خوارزمية ك مينز) لتسمية وثائقنا الاولية و نهج خاضع للرقابة (خوارزمية بايز الساذج) لتصنيف الوثائق الجديدة.

الهدف الرئيسي من عملنا هو توفير تصنيف موثوق للوثائق الطبية.

استخدمنا نصوص ميدلاين لإجراء تجاربنا.

الكلمات الرئيسية: التصنيف، التجميع، النص، التعلم، تحليل النص ، التقييم، ك مينز، بايز الساذج، ميدلاين

Tableaux et Figures

Liste des tableaux

Liste des figures

*p.77-**Figure 25 :** La silhouette de Kmeans.*

Table des matières

Introduction Générale

Chapitre 1 : Text mining et la classification automatique de textes

Partie 1 : Text Mining

Partie 2 : Classification automatique de textes

Chapitre2 : Codage des textes

Chapitre 3 : Algorithmes d'apprentissage automatique appliqués à la classification de textes

Chapitre 4 : Etude et conception

Chapitre 5: Implémentation et expérimentations

Conclusion générale

Annexe

Bibliographie

Introduction Générale

1- Contexte

MEDLINE est une base de données bibliographique qui couvre tous les domaines médicaux de l'année 1966 à nos jours : plus de 11 millions de références issues de 4 300 périodiques, principalement en langue anglaise, en plus du nombre impressionnant d'articles ajoutés chaque jour à cette base, nous ne pouvons pas imaginer, qu'une intervention humaine pour la classification de ces documents en catégories distinctes soit intéressante.

2- Problématique

La nécessité d'une méthode automatique pour y parvenir s'avère utile. La technologie du Text Mining s'est développée justement, pour faciliter et surtout pour extraire des connaissances utiles enterrées dans des réservoirs de données volumineux. Les techniques de classification du texte ont pour objectif, d'organiser les connaissances, pour pouvoir faire des recherches pertinentes, ou bien extraire des informations utiles.

L'une des motivations principales de ce travail est qu'il n'existe pas, à notre connaissance, des travaux similaires déjà réalisés dans les institutions universitaires algériennes. Ainsi l'intérêt personnel aux systèmes décisionnels.

3- Objectifs

L'objectif global de notre travail, est d'offrir une méthode automatisée pour la classification des documents dans la structure des dépôts de documents. Nous distinguons dans la classification automatique deux types d'approches : Approche supervisée et approche non supervisée.

Après étude de ces deux approches, et après analyse de notre corpus de test, qui n'est pas pré-classé, nous avons trouvé intéressant d'utiliser l'approche non supervisée (l'algorithme Kmeans) pour découvrir de l'information sur ce grand nombre de données non annotées, en les regroupant dans des clusters distincts, et utiliser ensuite une méthode supervisée seulement sur les clusters trouvés, Naïve Bayes sera utilisé pour classer les nouveaux documents.

Pour pouvoir utiliser de tels algorithmes, il est nécessaire de transformer les données, initialement en format XML, en une représentation numérique. Nous avons choisi pour ce faire, la méthode de sélection des termes les plus pertinents. Une fois ce prétraitement terminé, nous pouvons effectuer la classification à l'aide de nos algorithmes ; ensuite nous ferons une évaluation des résultats.

4-Organisation du mémoire

Notre mémoire est divisé en trois parties qui sont l'Etat de l'art, la conception et la réalisation.

La partie **Etat de l'art** est constituée de 3 chapitres:

- **Le chapitre 1**: Résume la technologie du Data Mining et du Text Mining, et Expose la classification automatique de textes, plus en détail, la catégorisation de textes
- **Le chapitre 2** : Présente l'état de l'art des approches de représentation de textes.
- **Le chapitre 3** : Est consacré aux différents algorithmes d'apprentissage automatique appliqués à la classification de textes.

La partie **Conception** est constituée d'un chapitre:

- **Le chapitre 4** : Présente notre étude et conception.

La partie **Réalisation** est constituée elle aussi d'un seul chapitre

- **Le chapitre 5** : Est consacré à l'implémentation et l'expérimentation de notre application (exemple de démonstration).

Chapitre 1

Text Mining et la classification automatique de textes

Sommaire

Partie 1 : Text Mining

Partie 2 : Classification automatique de textes

Partie 1 : Text Mining

1-Introduction

Nous vivons ces dernières années une explosion des données stockées sur des supports numériques. Ces données sont générées par différentes sources telles que, le commerce électronique, les réseaux sociaux, les fichiers logs de tous genres et les bases bibliographiques qui stockent les articles spécialisés.

Le paradoxe, c'est qu'il ya trop de données, mais pas assez d'informations qui permettent leurs exploitation. Pour combler ce besoin, une nouvelle industrie est née : Extraction de connaissances à partir de données (ECD) communément appelée le Data Mining (appellée en français la fouille de données).

Des statistiques montrent que plus de 80% de ces données stockées est de nature textuelle **[Che, 2001]**, ce qui a donné naissance à une technologie data mining pour l'analyse approfondie de ce type de donnée: le **Text Mining**

La première partie du chapitre aborde les définitions et le processus de la fouille de données et la deuxième partie détaille cette technologie appliquée à la donnée textuelle, le Text Mining. Après l'exposition des définitions et les approches du Text Mining, nous expliquons la chaine de traitement pour le processus de fouille de données textuelle. A la fin, nous abordons le lien entre le Text Mining et la classification de textes.

2- Fouille de données (Data mining)

2.1- Définitions

Le Data Mining, textuellement minage de données mais souvent traduit en français par *fouille de données*, se réfère à l'extraction de connaissance à partir de grandes quantités de données **[Han & Kam, 2001].**

D'après **[Pal & Jain, 2005]**, les origines du Data Mining remontent à 1989, lors du premier workshop KDD (*Knowledge Discovery in Database*). L'article le plus ancien contenant le terme *Data Mining* est certainement celui de Jorgenson en 1970.

Cela dit, c'est seulement dans le début des années 1990 que ce terme est adopté avec son sens actuel.

Le Data Mining est un domaine qui consiste à comprendre des données de taille relativement importante, dans le but de découvrir de nouvelles vues, corrélations, tendances, modèles ou relations cachées dans ces données en utilisant un ensemble de moyens matériels et logiciels à l'intersection de l'intelligence artificielle, les statistiques, l'apprentissage automatique et les systèmes de bases de données.

Dans ce qui suit nous abordons brièvement les relations du data mining avec les trois des domaines de recherche abordés : les bases de données, l'apprentissage automatique et les statistiques.

- ✓ *Les bases de données* sont nécessaires afin d'analyser de grandes quantités de données efficacement. L'analyse des données avec des algorithmes de Data Mining peut être soutenue par des bases de données et donc l'utilisation de la technologie de base de données dans le processus du data mining peut être utile.
- ✓ *L'apprentissage automatique* est un domaine de l'intelligence artificielle, qui permet le développement des techniques permettant aux ordinateurs « d'apprendre » par l'analyse de l'ensemble des données.
- ✓ *Les statistiques* sont une branche des mathématiques appliquées. Dans le cadre de la théorie statistique, l'aléatoire et l'incertitude sont modélisés par la théorie des probabilités. Aujourd'hui, de nombreuses méthodes statistiques sont utilisées dans le domaine du data mining.

2.2- Processus du Data Mining

Les outils du Data Mining s'intègrent dans un processus itératif à six phases qui doit être appliqué à un ensemble de données dans le but d'extraire un modèle utile. Ce processus est définit par CRISP-DM (Cross Industry Standard Process for Data Mining) conçu fin 1996 **[Cri, 1999].**

La **Figure 1** suivante montre les phases d'un processus CRISP-DM. La séquence des phases n'est pas rigide, un va et vient entre les différentes phases est toujours nécessaire, il dépend de l'issue de chaque phase.

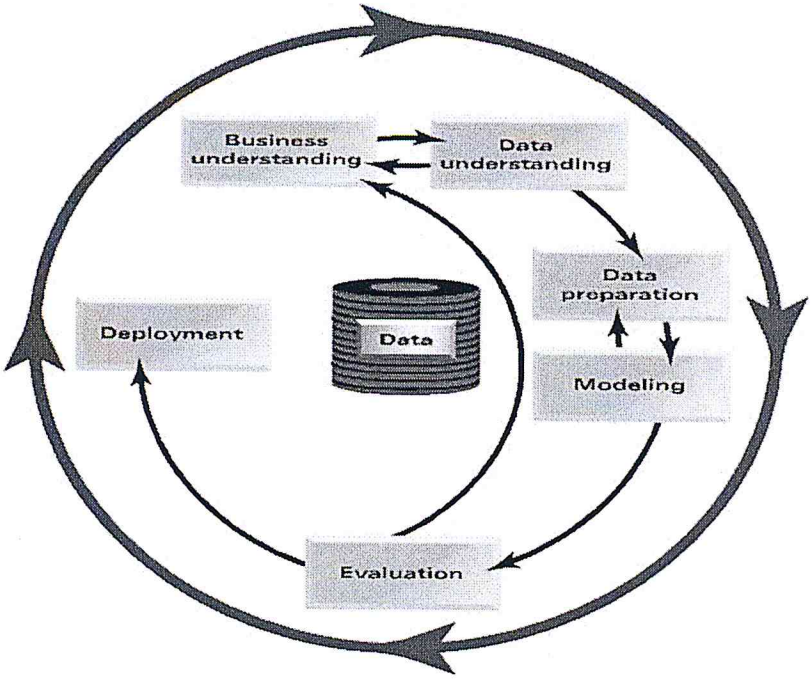


Figure 1 *: Les phases de Crisp DM.*

Dans la table 1, nous décrivons brièvement chaque phase:

- **Les Phases de CRISP-DM**

Phases	Tâches
Compréhension du problème	• Déterminer les objectifs commerciaux. • Evaluer de la situation. • Déterminer les objectives du Data Mining. • Produire d'un plan du projet.
Compréhension des données	• Collecte des données initiales • Description des données • exploration des données • Vérification de la qualité des données
Préparation des données	• Sélection des données • Nettoyage des données • Construction de nouvelles données • Intégration des données • Formatage des données
Modélisation	• Sélection des techniques de modélisation. • Génération d'une conception de test. • Création des modèles. • Evaluation de modèles.
Evaluation	• Evaluation de résultats • Processus de révision • Détermination des étapes suivantes
Déploiement	• Planification du déploiement • Planification de surveillance et maintenance • Production de rapport final • Exécution d'une révision de projet final

***Table 1**: Les phases du CRISP-DM*

3-Fouille de textes

3.1-Text Mining versus Data Mining

Historiquement le Data Mining est à la base du Text Mining au sens où celui-ci est l'extension du même but et du même processus vers des données textuelles.

Néanmoins, les deux technologies se distinguent dans la nature des données à traiter. Le Data Mining s'intéresse aux données numériques et factuelles qui sont bien structurées

dans des bases de données, alors que le Text Mining s'intéresse aux données textuelles non structurées **[Fel & al, 1998]**, généralement exprimées en langage naturel.

3.2-Définition

« La fouille de textes est la découverte à l'aide d'outils informatiques de nouvelles informations en extrayant différentes données provenant de plusieurs documents textuels. Un élément fondamental de ce processus réside dans les relations identifiées entre les informations extraites afin d'identifier de nouveaux faits ou de nouvelles hypothèses à explorer. » **[Hea, 2003].**

Ainsi, à partir d'un document texte, un outil de Text Mining va **générer de l'information sur le contenu du document**. Cette information n'était pas présente, ou explicite, dans le document sous sa forme initiale, elle va être rajoutée, et donc enrichir le document. Dans la pratique, cela revient à mettre en algorithmes un modèle simplifié des théories linguistiques dans des systèmes informatiques d'apprentissage et de statistiques. Les disciplines impliquées sont donc la linguistique calculatoire, l'ingénierie du langage, l'apprentissage artificiel, les statistiques et bien sûr l'informatique.

3.3- Approches du Text Mining

Deux approches, peuvent être envisagées pour faire du Text mining :

3.3.1-Approche statistique

Elle consiste à ne voir le document que via le prisme du nombre et des chiffres. Ainsi l'outil statistique de Text Mining produit des informations sur le nombre d'occurrence d'un terme, la fréquence d'apparition d'un terme dans un document ou un corpus[1].

3.3.2-Approche sémantique

L'analyse sémantique est une technique d'interprétation automatique des textes écrits en langue naturelle, c'est à dire tels qu'on les trouve dans les documents rédigés par et pour les humains. Cela permet à l'ordinateur de « comprendre » ces textes pour y collecter de l'information, pour classer les documents, pour en faciliter la recherche, etc.

[1] Recueil de textes, de documents qui ont trait à une même matière.

La particularité de l'approche sémantique, par rapport aux méthodes à base de mots-clefs, est que le logiciel est doté de réelles compétences linguistiques et ontologiques. Cela lui permet de **raisonner sur le sens** des mots et des phrases (au lieu de compter le nombre d'apparition de tel ou tel mot-clef), et aussi d'exploiter et d'augmenter certaines **connaissances sur le « monde »**. C'est pourquoi cette approche est traditionnellement classée dans le domaine de l'**intelligence artificielle**.

3.4-Chaîne de traitement pour le processus de fouille de données textuelle

Un texte est considéré comme une entité porteuse d'une information qu'il faut préparer, représenter et organiser pour lui permettre d'utiliser des outils de fouille de données et valider les résultats de la fouille, comme illustrer dans la **Figure 2 [Che, 2004]** suivante.

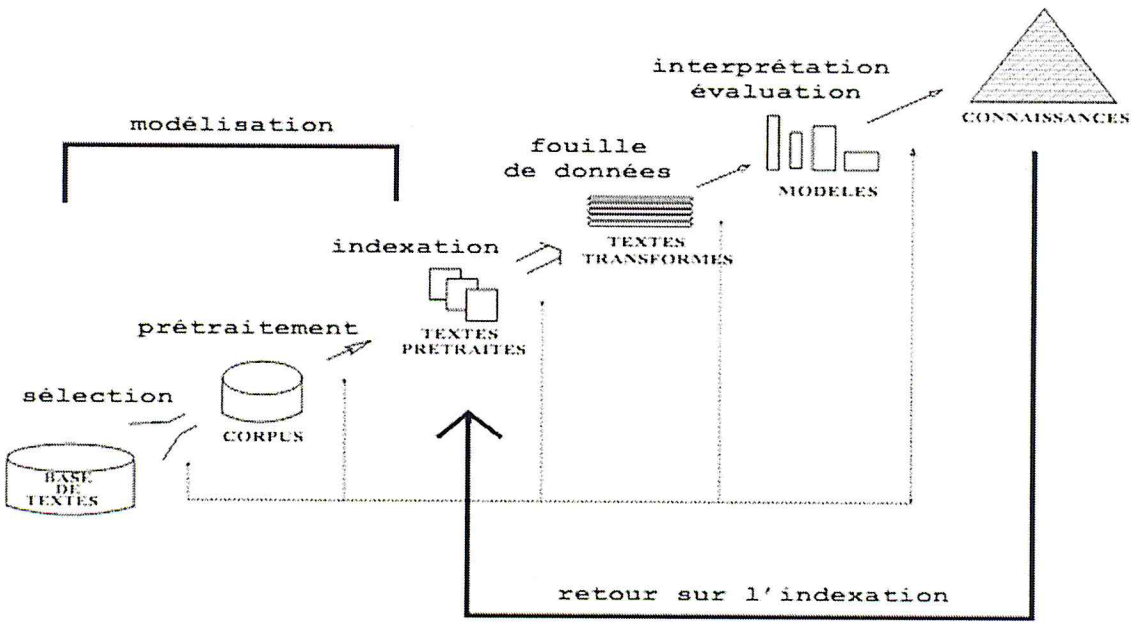


Figure 2** : La chaîne de traitement pour le processus de fouille de textes **[Che , 2004]

- **Le prétraitement:** est une tâche très importante et celle qui consomme le plus de temps, elle englobe les trois premières étapes du modèle CRISP-DM, de la compréhension du problème et des données à la préparation de ces dernières. Cette phase inclut tous les traitements, les processus et les méthodes nécessaires pour la préparation des données pour les opérations de bases de la découverte de connaissance du système Text Mining. L'étape

du prétraitement en général converti les informations de leur source originale en un format intermédiaire.

- **La modélisation :** est le cœur du système Text Mining et inclut les opérations de bases de la fouille qui utilisent les algorithmes de Data Mining pour la découverte de connaissances.
- **L'évaluation :** pour décider de ce qu'il faut faire après, on termine le processus dans le cas où les résultats sont bien adaptés à l'application, sinon si le résultat est signifiant mais non satisfaisant, on réitère et le résultat généré sera utilisé comme une partie de l'entrée d'une ou de plusieurs étapes précoces.

Remarque : Le prétraitement et la modélisation sont les deux plus importantes phases dans n'importe quel système Text Mining, et généralement ils décrivent la série de processus au sein d'une vue généralisée d'une architecture d'un système Text Mining.

3.5 Text Mining et la classification de textes

L'objectif du Text Mining est de faire ressortir, dans une masse très importante de données textuelles, l'information utile afin qu'elle devient exploitable par l'informatique. Il intervient donc dans la :

- Recherche d'information (Information retrieval) : Int errogation de textes par concepts, mots-clés, sujets, phrases visant à obtenir des résultats triés par ordre de pertinence, (ex : Google)

- Construction de résumé (Summarization) : Abstraction et condensation d'un texte pour élaborer une version réduite conservant au maximum la sémantique.

- Extraction d'information (Information extraction) : Identification d'éléments sémantiques dans un texte (entités, propriétés, relations, patterns …)

- Interrogation en langage naturel (Question answering) : Interrogation de bases de données en langage naturel.

Et notamment

- **La classification automatique des documents**

Qui consiste à l'apprentissage de modèles permettant de décrire et de différencier des classes afin de classer les futurs documents.

Remarque : Toutes ces applications sont étroitement liées.

4-Conclusion

Nous avons tenté, tout au long de cette première partie, de présenter les deux technologies : Data Mining et plus en détail le Text Mining, qui en résumé est divisé en deux étapes principales, étape d'analyse qui permet de structurer le texte, et une étape d'interprétation de l'analyse, qui fait appel aux méthodes de fouille de données.

Nous avons précisé le lien entre la fouille de données textuelle et la classification de texte, que nous détaillons dans la partie 2 suivante.

Partie 2 : La classification automatique de textes

1-Introduction

L'intérêt de la classification de textes est de regrouper les textes thématiquement similaires au sein d'un même ensemble, de façon à pouvoir effectuer, par la suite, une recherche ou une extraction d'information utile.

La classification manuelle, pratiquée par des experts humains devient de plus en plus fastidieuse vu le nombre de documents numériques qui s'accroisse, aussi bien sur internet, qu'au sein des entreprises (classement de documents internes, dépêches d'agences, articles, etc.).

Pour remédier à ce problème, des approches visant à créer des classificateurs automatiques de textes sont apparues, elles sont liées directement à l'apprentissage automatique, ce domaine de l'Intelligence Artificielle qui s'intéresse à donner aux machines la capacité de s'améliorer à l'accomplissement d'une tâche, en interagissant avec leur environnement.

On distingue dans L'apprentissage automatique deux types d'approches : **supervisée** et **non supervisée**. Ces deux méthodes diffèrent sur la façon dont les classes sont générées. En effet dans le cas de l'apprentissage non supervisée, les groupes de documents (classes) sont calculés automatiquement par la machine **[Sal, 1983], [Iwa, 1995]**, tandis qu'ils sont, dans l'approche supervisée **[Joa 1998b, Seb 2002, Yan 1999a]**, définis par un expert.

Ce chapitre traitant la classification automatique de textes est organisé comme suit : Après une introduction, nous expliquons le besoin de la classification automatique ainsi que le vocabulaire utilisé. Nous donnons par la suite une définition précise du problème de la catégorisation automatique de textes.

Nous exposons après, la démarche classique d'un système de catégorisation automatique de textes, de la représentation des documents jusqu'aux évaluations des résultats. Nous finissons par citer quelques lacunes rencontrés lors de la catégorisation.

2-Pourquoi automatiser la classification?

On assiste aujourd'hui à un accroissement de la quantité d'information textuelle disponible et accessible d'une manière exponentielle. D'après les derniers chiffres, **1 trillion, c'est-à-dire mille milliards (1.000.000.000.000)** est le nombre de

pages web distinctes que Google a répertorié et plus de 200 millions de serveurs hôtes sur Internet[2].

En début 2012, Notre corpus de test MEDLINE (une présentation détaillée de ce corpus est faite dans le dernier chapitre), conten ait plus de 19 millions d'articles référencés, provenant de plus de 5 000 sources différentes (revues en biologie et en médecine) dont les plus anciennes remontent à 1902[3].

Le nombre de textes à classifier étant énorme, il serait très difficile de pouvoir déterminer de combien de temps a besoin un expert pour associer un texte à une catégorie ? En pratique, il s'agit d'une question difficile à répondre. Assurément, plusieurs variables influencent le phénomène et les lignes qui suivent porteront sur certaines d'entre elles.

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

---

[2] **D'après le site http:// www.cuy.be/html/typoweb/chap1.htm**
[3] D'après le site http://fr.wikipedia.org/wiki/MEDLINE

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

1. La réalisation manuelle de cette tâche par un expert est extrêmement coûteuse en terme de temps.
2. 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 **[Mou, 1996], [Seb, 2002]**
3. La classification faite par les humains est subjective, deux experts peuvent classer différemment un même document, ou encore un même expert peut classer différemment un même document soumis à deux instants différents **[Cle & Zig, 2004].**

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 notre travail se concentre.

3-Vocabulaire utilisé

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

Classification, catégorisation ou encore clustering ? C'est des termes qu'on peut rencontrer dans la littérature puisque la CT provient de plusieurs domaines scientifiques différents qui n'utilisent pas toujours le même vocabulaire pour la dénomination des différentes tâches.

Nous allons essayer de distinguer entre les différentes variantes de classification de textes et le vocabulaire utilisé dans la section suivante.

3.1-Catégorisation (classification supervisée)

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.

Aujourd'hui, cette problématique utilise largement des méthodes issues de l'apprentissage automatique et beaucoup d'algorithmes d'apprentissage supervisé lui ont été appliqués (Naïve bayes, K-plus proches voisins, arbres de décision, machines à vecteurs support, réseaux de neurones, etc...), nous détaillerons quelques algorithmes dans le chapitre 3.

3.2-Clustering (Regroupement, classification automatique)

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. Le processus d'attribution de nom à ces classes est appelé **labelling**.

Exemple : points dans le plan

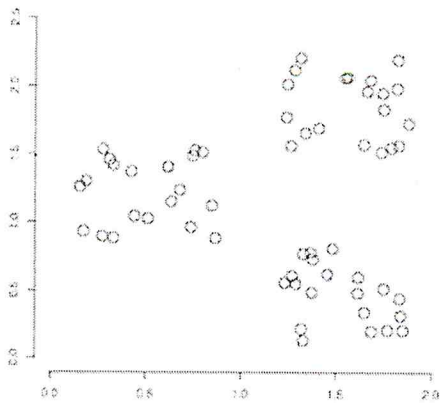


Figure 3 : *Regroupement d'objets similaires*

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 seront aussi détaillées dans le chapitre 3.

4-Avantages et inconvénients

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

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

Remarque : Nous nous intéressons particulièrement à l'apprentissage supervisé, sans exclure la possibilité de recours aux techniques de regroupement.

5-Définition de la catégorisation automatique de textes

Le but de la catégorisation automatique de textes est d'apprendre à une machine à classer un texte dans la bonne catégorie en se basant sur son contenu. Habituellement, les catégories font référence aux sujets des textes.

La recherche dans ce domaine est toujours très pertinente, car les résultats obtenus aujourd'hui sont encore sujets à amélioration. Pour certaines tâches, les classifieurs[4] automatiques performent presque aussi bien que les humains, mais pour d'autres, l'écart est

[4] Un classifieur est un modèle de classification

encore grand. 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, tel qu'illustré à la **Figure 4 [Sim, 2005].**

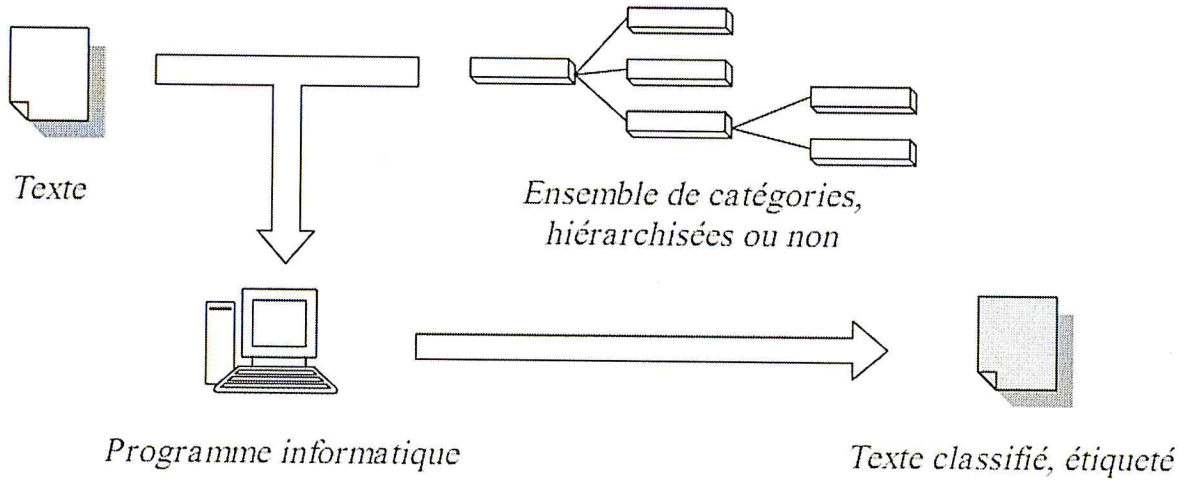


Figure 4** : La tâche de classification **[Sim, 2005]

6-Démarche de la catégorisation automatique de textes

Pour réaliser l'opération de catégorisation automatique de textes suivant la **Figure 5**,

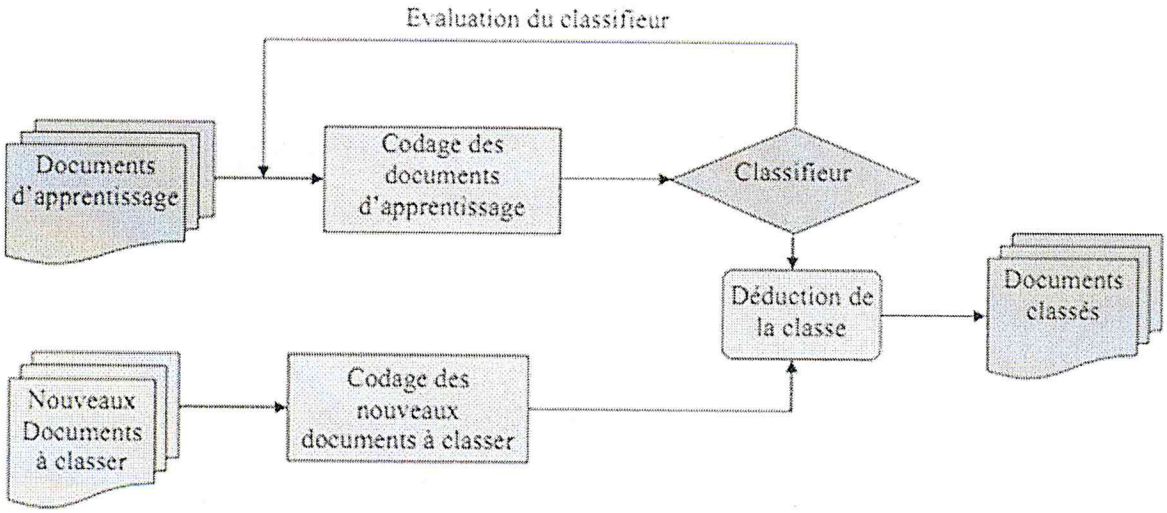


***Figure 5** : Démarche de la catégorisation de textes*

La démarche commune est:

- ✓ la première phase consiste donc à formaliser les textes afin qu'ils soient compréhensibles par la machine et utilisables par les algorithmes d'apprentissage.
- ✓ La catégorisation des documents est la deuxième phase, cette étape est bien entendu décisive car c'est elle qui va permettre ou non aux techniques d'apprentissage de produire une bonne généralisation à partir des couples (Document, Classe).
- ✓ Pour améliorer la performance des modèles, une évaluation de la qualité des classifieurs et la comparaison des résultats fournis par les différents modèles est effectuée en fin de cycle.

Le processus de catégorisation intègre la construction d'un modèle de prédiction qui, en entrée, reçoit un texte et, en sortie, lui associe une ou plusieurs étiquettes.

Pour identifier la catégorie ou la classe à laquelle un texte est associé, un ensemble d'étapes est habituellement suivies. Ces étapes concernent principalement la manière dont un texte est représenté, le choix de l'algorithme d'apprentissage à utiliser et comment évaluer les résultats obtenus pour garantir une bonne généralisation du modèle appris (le **classifieur**).

Le processus de catégorisation, intégrant la phase de classement de nouveaux textes, est résumé dans la **Figure 6**. Il comporte deux phases que l'on peut distinguer comme suit :

1. l'apprentissage, qui comprend plusieurs étapes et aboutit à un modèle de prédiction:

a) nous disposons d'un ensemble de textes étiquetés (pour chaque texte nous connaissons sa catégorie) ;

b) à partir de ce corpus, nous extrayons les k descripteurs (mots, termes) (t_1; ..; t_k) les plus pertinents au sens du problème à résoudre ;

c) nous disposons alors d'un tableau « descripteurs × individus », et pour chaque texte nous connaissons la valeur de ses descripteurs et son étiquette;

2. le classement d'un nouveau texte d_x, **Figure 7 [Sim, 2005]**, comprend deux étapes :

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

b) application d'un algorithme d'apprentissage sur ces occurrences et le tableau précédent afin de prédire l'étiquette de ce texte d_x. **[Rad, 2003].**

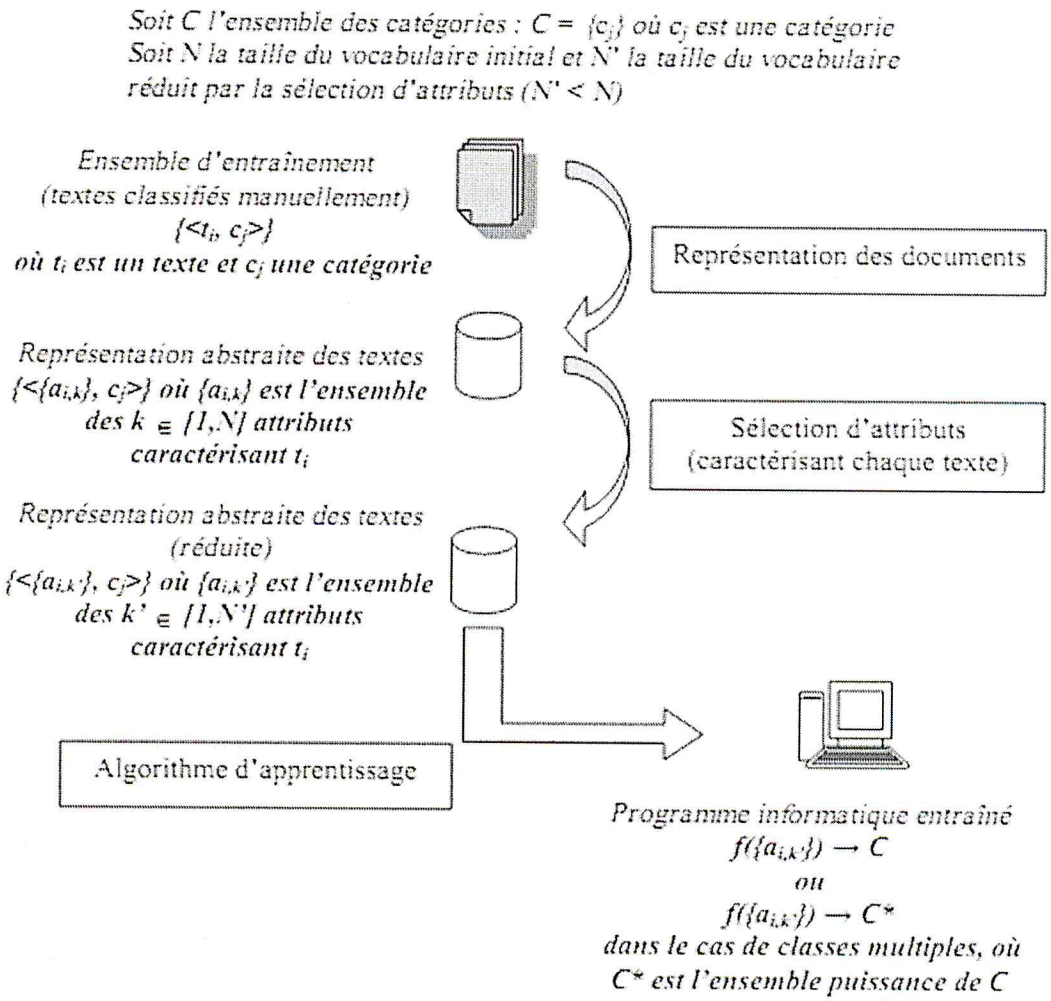


Figure 6** : Entraînement d'un système de classification automatique de textes **[Sim, 2005].

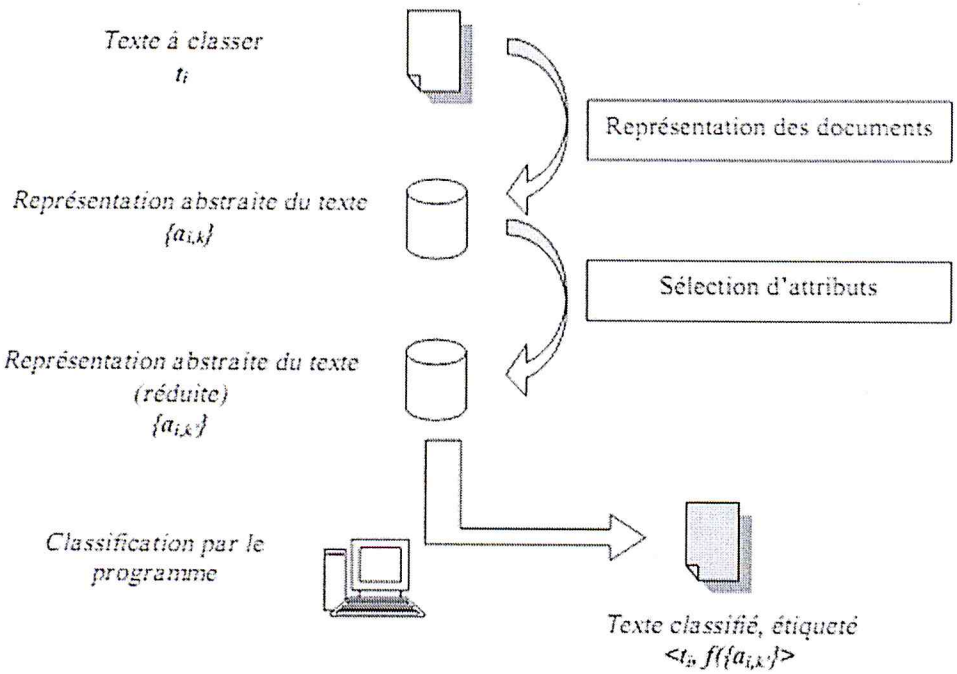


Figure 7 : *Classification d'un nouveau document* ***[Sim, 2005].***

7-Quelques problèmes rencontrés dans 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..

7.1-Sur-apprentissage

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

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

Quelques auteurs recommandent d'utiliser au moins 50 à100 fois plus de textes que de termes. En général le nombre de textes d'apprentissage est limité, c'est pour cela on cherche à agir sur le nombre des termes utilisés en les diminuant, pour éviter ce sur-apprentissage. Sans bien sûr pénaliser le système en supprimant des termes pertinents **[Seb, 2002].**

7.2-L'homographie

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

7.3- Polysémie (Ambiguïté)

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

8-Conclusion

La classification de textes est devenue de plus en plus indispensable, devant la quantité énorme de textes électroniques.

Ces dix dernières années la classification a connu beaucoup de progrès, grâce à l'introduction des techniques d'apprentissage automatique. Dans cette deuxième partie, du premier chapitre, nous avons essayé de définir, la classification, précisément la catégorisation automatique de textes, ainsi que quelques notions nécessaires pour la suite de ce mémoire.

1-Introduction

L'information textuelle, est actuellement stockée sous différents formats de fichiers, tels que HTML, XML, CSV, etc. Ces collections sont peu structurées, ce qui rend difficile l'accès à l'information qu'elles contiennent. D'ou la nécessité de chercher comment structurer ces corpus pour qu'ils deviennent exploitables, notamment par la classification.

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

Dans ce qui suit nous allons exposer les différentes approches de représentation de documents textuels.

2-Caractéristiques de la donnée textuelle

Un texte peut être vu comme une suite de mots séparés par des espaces et par un ensemble de caractères de ponctuation.

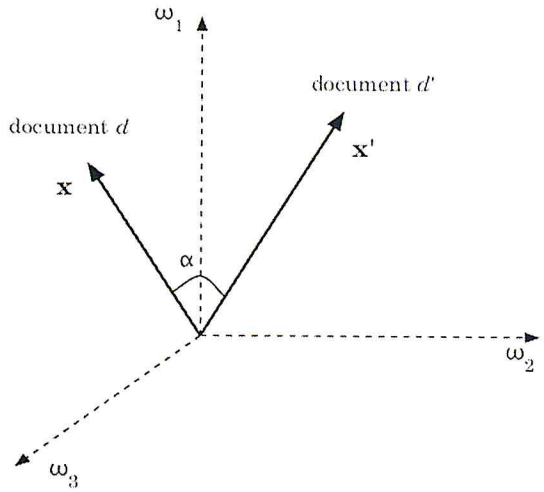
La composition d'un texte fait appel à deux définitions de composition :

- Déterminer les unités constituantes le texte.

- Constituer un texte : dans le sens de distribuer et d'organiser ces unités afin d'atteindre certaines idées.

Un document (texte) di est représenté par un vecteur numérique de la façon suivante : di = (V1i, V2i, ..., V | T | i)
Où T est l'ensemble des termes (ou descripteurs) qui apparaissent au moins une fois dans le corpus.
(|T| est la taille du vocabulaire), et Vki représente le poids (ou la fréquence).

- Chaque document est vu comme une séquence de mots
- Le nombre de mots du lexique présents dans les documents du corpus détermine la dimension de l'espace



Plusieurs approches sont proposées pour la représentation de la donnée textuelle.

La dimension linguistique exprimée dans un texte est considérée comme un moyen d'accès à l'information que représente un texte. **L'étude exclusivement linguistique d'un texte n'est d'ailleurs pas l'objet de notre travail.**

3-Prétraitement

Le prétraitement des textes est une phase capitale du processus de classification, puisque la connaissance imprécise de la population[1] peut faire échouer l'opération.

Après la première opération que doit effectuer un système de classification à savoir la reconnaissance des termes utilisés (La segmentation), il faut extraire 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 *désuffixation (*ou *stemming en anglais*), qui consiste à rechercher les racines lexicales, 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.

[1] On appelle population l'ensemble soumis à une étude statistique

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).

4-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 ou les expressions.

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

4.1-Représentation en « sac de mots »

La représentation de textes la plus simple a été introduite dans le cadre du **modèle vectoriel** (qui est détaillé dans la section **6.2.1**) elle porte le nom de « sac de mots ». L'idée est de transformer les textes en vecteurs dont chaque composante représente un mot. Les mots ont l'avantage de posséder un sens explicite. Cependant, plusieurs problèmes se posent.

Il faut tout d'abord définir ce qu'est « un mot » pour pouvoir le traiter automatiquement.

On peut le considérer comme étant une suite de caractères appartenant à un dictionnaire, ou bien, de façon plus pratique, comme étant une séquence de caractères non délimiteurs encadrés par des caractères délimiteurs.

Les composantes du vecteur sont une fonction de l'occurrence des mots dans le texte. Cette représentation des textes exclut toute analyse grammaticale et toute notion de distance entre les mots : c'est pourquoi cette représentation est appelée « sac de mots » ; d'autres auteurs parlent d'« ensemble de mots » lorsque les poids associés sont binaires. **[Rad, 2003].**

4.2-Représentation des textes par des phrases

Malgré la simplicité de l'utilisation de mots comme unité de représentation, certains auteurs proposent plutôt d'utiliser les phrases comme unité. Les phrases sont plus informatives que les mots seuls, car les phrases ont l'avantage de conserver l'information relative à la position du mot dans la phrase.

Logiquement, une telle représentation doit obtenir de meilleurs résultats que ceux obtenus via les mots. Mais les expériences présentées ne sont pas concluantes car, si les qualités sémantiques sont conservées, les qualités statistiques sont largement dégradées. **[Rad, 2003].**

4.3 Représentation des textes avec des racines lexicales et des lemmes

Dans le modèle précédent (représentation en «sac de mots»), chaque mot est considéré comme un descripteur différent et donc une dimension de plus; ainsi, les différentes formes d'un verbe constituent autant de mots. Par exemple: les mots «déménageur, déménageurs, déménagement, déménagements, déménager, déménage, déménagera, etc.» sont considérés comme des descripteurs différents alors qu'il s'agit de la même racine « déménage ». Les techniques de *désuffixation*, (qui consistent à rechercher les racines lexicales), et de *lemmatisation (*qui consiste à remplacer les verbes par leur forme infinitive, et les noms par leur forme au singulier), cherchent à résoudre cette difficulté. Pour la recherche des racines lexicales, plusieurs algorithmes ont été proposés; l'un des plus connus pour la langue anglaise est l'algorithme de Porter. Pour la

lemmatisation l'algorithme le plus efficace est TreeTagger pour l'anglais, le français, l'allemand et l'italien.

L'extraction des *stemmes* repose sur des contraintes linguistiques bien moins fortes. De ce fait, les algorithmes sont beaucoup plus simplistes et mécaniques que ceux permettant l'extraction des lemmes; ils sont donc plus rapides; mais leur précision et leur qualité sont naturellement inférieures. **[Rad, 2003].**

5-Sélection de descripteurs (Réduction)

5.1-Pourquoi réduire

S'attaquer au problème de la classification automatique de textes signifie aussi s'attaquer à des difficultés du traitement automatique de la langue naturelle. La taille impressionnante du vocabulaire peut s'avérer un obstacle à l'utilisation d'algorithmes plus complexes, pourquoi ? Si l'on utilise directement le vocabulaire contenu dans les textes et que l'on crée un attribut pour chaque mot qu'il contient, on se retrouve avec un espace vectoriel de dimension très élevée. Chacun des textes sera représenté par un vecteur ayant autant de termes qu'il ya de mots dans le vocabulaire. Le traitement d'un tel espace vectoriel demanderait beaucoup de mémoire et de temps de calcul et même il pourrait nous empêcher d'utiliser des algorithmes de classification plus complexes. Utiliser tous ces mots influencerait aussi négativement sur la précision de la classification. En effet plusieurs mots sont vides de sens. Aussi si un mot est présent dans plusieurs documents, c'est donc il ne permettra pas de départager l'appartenance d'un texte qui le contient à l'une ou l'autre catégorie.

Ainsi, Il est nécessaire de diminuer d'avantage 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.

5.2-Le nombre de descripteurs conservé

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 chercheurs dans le domaine ont essayé de réaliser ce bon compromis, comme par exemple **[Dum & al., 1998]** construit son modèle à base des SVM[2] en prenant en considération seulement 300 termes sur le corpus Reuters[3].

5.3-Méthodes de sélection de descripteurs

Des techniques sont développées pour réduire la dimension du vocabulaire. Principalement ces techniques sont divisées en deux grandes familles ; **la sélection d'attributs** et **l'extraction d'attributs.**

a- Sélection d'attributs :

Cette technique prend les attributs (mots dans notre cas) d'origine et conserve seulement ceux jugés utiles à la classification, selon bien sur une certaine fonction d'évaluation.

b- Extraction d'attributs :

Contrairement à la sélection, cette méthode crée de nouveaux attributs à partir des attributs de départ, en faisant soit des regroupements ou des transformations.

6-La pondération

La pondération des termes est une mesure statistique, le principe de pondération s'appuie sur l'observation suivante **[Rij, 1979] [Sal & McG, 1983]** : « la fréquence d'apparition des mots dans les textes en langage naturel est significative de l'importance de ces mots dans le seul but de représenter le contenu de ces textes ».

Il existe différentes méthodes pour calculer le poids sachant que, pour chaque terme, il est possible de calculer non seulement sa fréquence dans le corpus, mais aussi le nombre de documents contenant ce terme.

[2] Machine à vecteur support : un algorithme d'apprentissage supervisé.
[3] Base de données de dépêches d'information en langue anglaise.

6.1-Formules de pondération

6.1.1-Term frequency (TF)

- Un terme qui apparait plusieurs fois dans un document est plus important qu'un terme qui apparaît une seule fois
- w_{ij} = Nombre d'occurrences du terme t_i dans le document d_j
  TF_{ij} = Fréquence du terme t_i dans le document d_j

$$TF_{ij} = \frac{w_{ij}}{|d_j|}$$

6.1.2-Inverse document frequency (IDF)

- Un terme qui apparaît dans peu de documents est un meilleur discriminant qu'un terme qui apparaît dans tous les documents
  - df_i = nombre de documents contenant le terme t_i
  - d = nombre de documents du corpus

$$IDF_i = \log \frac{d}{df_i}$$

6.1.3-TF-IDF

- TF-IDF signifie Term Frequency x Inverse Document Frequency :
  - Proposée par **[Sal, 1989]**, mesure l'importance d'un terme dans un document relativement à l'ensemble des documents.

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

  - $tf_{i,j}$ = fréquence du terme i dans le document j
    df_i = nombre de documents contenant le terme i
    N = d =nombre de documents du corpus

6.2-Modèles de représentation de document

La représentation de document est une étape préliminaire qui consiste en la représentation de chaque document par un **vecteur**, dont les composantes sont par exemple les mots contenus dans le texte, afin de le rendre exploitable par les algorithmes d'apprentissage. Une collection de textes peut être ainsi représentée par **une matrice** dont les lignes sont les termes qui apparaissent au moins une fois et les colonnes sont les documents de cette collection. **[Rad, 2003].**

Plusieurs modèles ont étés proposés pour la représentation de texte, tel que le modèle probabiliste, le modèle séquentiel et le modèle vectoriel.

Nous détaillerons dans ce qui suit, le modèle le plus utilisé, qui est le modèle vectoriel.

6.2.1-Le modèle vectoriel

En dépit de sa structure de données simple, sans utiliser aucune information sémantique explicite, le modèle vectoriel permet une analyse très efficace de grandes collections de documents.

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.

6.2.1.1-Représentation binaire

Cette représentation est la plus simple et la plus ancienne, elle 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 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.

6.2.1.2-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 (fréquence d'apparition des termes).

6.2.1.3-Vecteur TF-IDF

L'idée de base est de représenter les documents par des vecteurs et de mesurer la proximité entre documents par l'angle entre les vecteurs, cet angle étant donc supposé représenter une distance sémantique.

Le principe est de coder chaque élément du sac de mot par un scalaire (nombre) appelé TF-IDF (présenté précédemment) pour donner un aspect mathématique aux documents textes.

Cette représentation donne plus de poids aux **termes qui apparaissent avec une haute fréquence dans peu de documents**. L'idée est que de tels mots aident à discriminer entre textes ayant différent sujet.

Le TF-IDF a deux limites fondamentales :

- ✓ La première est que les documents plus longs ont typiquement des poids plus forts parce qu'ils contiennent plus de mots, donc « la fréquence des termes » tendent à être plus élevées.
- ✓ La deuxième est que la dépendance de « la fréquence des termes » est trop importante. Si un mot apparaît deux fois dans un document dj, cela ne veut pas nécessairement dire qu'il a deux fois plus d'importance que dans un document dk où il n'apparaît qu'une seule fois.

Remarque : L a fonction TFIDF a démontré une bonne efficacité dans des tâches de catégorisation de textes, et, en plus, son calcul est simple **[Seb, 1999].**

7-Conclusion

Pour pouvoir appliquer les différents algorithmes d'apprentissage sur les documents de type textuels, des techniques ont été développées 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 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, deux approches sont étalées dans ce chapitre.

Une fois la liste des descripteurs est arrêtée, 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. La pondération est les modèles de représentation sont détaillés dans la dernière partie de ce chapitre.

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

Maintenant que le problème étudié est mieux cerné, poursuivons au chapitre suivant avec la présentation d'algorithmes d'apprentissage automatique utilisés dans ce domaine.

Chapitre 3

Algorithmes d'apprentissage automatique appliqués à la classification de textes

Sommaire

1-Introduction

« L'apprentissage automatique fait référence au développement, à l'analyse et à l'implémentation de méthodes 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 »[1].

L'objectif de l'apprentissage artificiel est de concevoir des programmes pouvant s'améliorer automatiquement avec l'expérience.

Nous allons exposer dans ce chapitre quelques algorithmes d'apprentissage automatique qui se divisent en deux grandes catégories : **Algorithmes d'apprentissage supervisé, Algorithmes d'apprentissage non supervisé.**

Nous aborderons aussi la notion distance, utilisée par ces algorithmes et les différentes formules de son calcul.

Notons que nous avons repris, dans cette section, quelques définitions proposées dans (www.fr.wikipedia.org).

2- Algorithmes d'apprentissage supervisé

La majorité des algorithmes d'apprentissage supervisés tentent de trouver un modèle (une fonction mathématique) qui explique le lien entre des données d'entrée et les classes de sortie. Ces jeux d'exemples sont donc utilisés par l'algorithme.

Il existe de nombreuses méthodes d'apprentissage supervisé :

- Machine à vecteur support : SVM
- K plus proche voisins :KPPV
- Les arbres de décision
- Rocchio
- Naive Bayes

Dans ce qui suit nous allons détailler deux algorithmes, SVM et Naive Bayes qui sont très utilisés dans la classification des textes.

[1] D'après le site http:// fr.wikipedia.org/wiki/Discussion:Apprentissage_automatique

2.1-Machine à vecteur support : SVM

L'algorithme de machines à support vectorielles est très rapide et efficace pour les problèmes de classification de texte **[Joa, 1998]**.

Un document d est représenté par un vecteur (t_{d1}, t_{d2},…,t_{dn}) des mots qui le compose. Un SVM simple peut seulement séparer deux classes : une classe positive L_1 (indiqué par y = +1) et une classe négative L_2 (indiqué par y = -1)

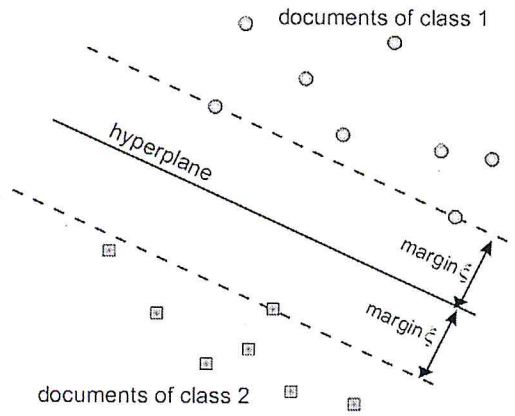


***Figure 8**: hyperplan avec distance maximal (marge) aux exemples de classes Positives et négatives.*

Dans l'espace de vecteurs introduis, un hyperplan peut être définie en donnant à y la valeur nulle (y = 0) dans l'équation linéaire suivante :

$$y = f(\vec{t_d}) = b_0 + \sum_{j=1}^{N} b_j t_{dj}$$

L'algorithme SVM détermine un hyperplan qui est localisé entre les exemples positifs et négatifs de l'ensemble d'entrainement. Les paramètres *bj* sont adaptés de telle sorte que la distance $\mathcal{E}$ (appelé marge) entre l'hyperplan et l'exemple positif et négatif le plus proche soit maximisé.

Les documents ayant une distance $\mathcal{E}$ de l'hyperplan sont appelés **supports de vecteur** et détermine l'endroit actuel de l'hyperplan. En général une petite fraction de documents est supports de vecteur. Un nouveau document avec un terme vecteur $\vec{t_d}$ est classifié en L_1 si la valeur $f(t_d) > 0$ et en L_2 autrement.

La propriété la plus importante du SVM est que l'apprentissage est quasiment indépendant des dimensions de l'espace.

- **Critiques de l'approche :**

Actuellement, l'algorithme SVM est considéré parmi les plus performants en raison de sa modélisation simple et rapide à calculer par une machine.

Seulement il introduit des concepts complexes peu adaptés aux corpus de grandes tailles non fixes.

2.2- Naïve bayes[2]

La classification bayesienne naïve de textes est une approche probabiliste de classification simple. Cette approche est basée sur un modèle probabiliste dérivant du théorème de Bayes qui fait l'hypothèse que les mots qui apparaissent dans un document sont indépendants les un des autres. Ce qui n'est pas tout à fait le cas dans la pratique. Il est supposé que **la classe du document a un rapport avec les mots** qui apparaissent dans le document.

- **Description du modèle Bayésien appliqué à la classification de textes**

Supposons que nous disposons de n catégories de documents, déterminer à quelle catégorie Ci sera associé un document D revient à calculer la probabilité d'appartenance du document D à la catégorie Ci. En se basant sur le théorème Bayes, on peut calculer cette probabilité de la façon suivante :

$$P(C_i|D) = \frac{P(D|C_i) * P(C_i)}{P(D)}$$

Dans cette formule :

- P (Ci |D) représente la probabilité d'appartenance du document D à la catégorie Ci qui peut être également déterminée en évaluant la fréquence d'apparition des mots du document D qui sont associés à la catégorie Ci.
- P (D |Ci) est la probabilité selon laquelle, pour une catégorie donnée, les mots du document D sont associés à la catégorie Ci .

[2] Ce paragraphe est inspiré de **[Ngo& May, 2005]**

- P (Ci) est la probabilité qui associe le document D à la catégorie Ci indépendamment du contenu du document.
- P (D) est la probabilité propre du document D.

Pour réellement déterminer à quelle catégorie un document appartient, il faut calculer

P (Ci |D) pour chacune des catégories. Étant donné que P (D) reste constant pour toutes les catégories, déterminer P (Ci |D) se résume juste au calcul de P (D |Ci) * P (Ci).

Comment calcule t-en ces probabilités ?

En considérant que le document D est composé d'un ensemble de mots que nous noterons W 1 , ..., W m , calculer P (D |Ci) reviendrait à calculer le produit des probabilités d'apparition de chaque mot W i dans la catégorie Ci . Ce calcul se justifie par l'hypothèse selon laquelle tous les mots apparaissent indépendamment les un des autres dans un document. Ce qui permet finalement d'écrire :

$$P(D|C_i) = P(W_1|C_i) * P(W_2|C_i) * ... * P(W_m|C_i)$$

Pour chacune des catégories, P (W i |Ci) est le rapport entre le nombre de fois que le mot W i apparaît dans la catégorie Ci et le nombre total de mots que comprend la catégorie Ci .

P (Ci) est calculé en divisant le nombre total de mots pour la catégorie Ci par la somme du nombre total de mots dans toute les catégories. D' où la formulation suivante :

$$P(C_i|D) = P(W_1|C_i) * ... * P(W_m|C_i) * P(C_i)$$

Ce calcul est effectué pour chaque catégorie et on considère la probabilité la plus élevée pour choisir à quelle catégorie sera associée le document qu'on souhaite classer.

Critiques de l'approche :

L'algorithme NB est connu par son efficacité et sa simplicité qui revient à l'effet admis, d'indépendance entre les différents descripteurs et à cause de cette hypothèse d'indépendance des mots dans ce modèle, on le qualifie souvent de "Naïve", "Idiot", "Simple". En général, ce type d'algorithmes permet de faire le même

travail de classification que les autres algorithmes qui ont déjà prouvés dans le domaine. Ce classifieur est très favorable pour les documents courts qui donne des résultats très intéressants, néanmoins ces performances sont réduites quand il s'agit d'un vocabulaire important à traiter, ainsi le manque d'une meilleure prise en compte de la taille des documents, fait que ses performances en qualité de classement se dégradent avec l'augmentation du nombre de caractéristiques. En effet, si le nombre de termes augmente, alors le nombre des dépendances entre l'ensemble des termes augmentent aussi, et donc, la vérification de l'hypothèse de Naïve Bayes diminue. **[Hil, 2009]**

2.3-Evaluation

Il existe plusieurs mesures statistiques qui servent à évaluer les résultats des classifieurs, nous citons :

2.3.1-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 (Voir ***Tableau 2***), qui fournit 4 informations essentielles :

- 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	**VP_i**	FP_i
	Non	FN_i	**VN_i**

Tableau 2 : *Matrice de contingence de la classe Ci*

2.3.2-Précision et Rappel

On définit à partir de la matrice de contingence les mesures suivantes :

- **Rappel** étant la proportion de documents correctement classés par le système par rapport à tous les documents de la classe Ci.

$$Rappel\ (C_i) = \frac{Nombre\ de\ documents\ bien\ classés\ dans\ C_i}{Nombre\ de\ documents\ de\ la\ classe\ C_i}$$

$$R_i = \frac{VP_i}{VP_i + FN_i}$$

- **Précision** est la proportion de documents correctement classés parmi ceux classés par le système dans Ci. 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.

$$Précision\ (C_i) = \frac{Nombre\ de\ documents\ bien\ classés\ dans\ C_i}{Nombre\ de\ documents\ classés\ dans\ C_i}$$

$$P_i = \frac{VP_i}{VP_i + FP_i}$$

- **F-mesure** : F1 permet de combiner, les deux mesures classiques le Rappel (R) et la Précision (P) pour obtenir une moyenne harmonique entre ces deux indicateurs, définit par :

$$F_i = \frac{2 * P * R}{P + R}$$

3-Algorithmes d'apprentissage non supervisés

Il existe deux types de structures produites par les algorithmes d'apprentissage non supervisés. Celles qui sont **hiérarchiques** et celles qui sont non hiérarchiques (**linéaires**). Les méthodes hiérarchiques sont souvent préférables lorsqu'on désire effectuer une analyse détaillée des données tandis que les méthodes non hiérarchiques sont plus rapides et choisies lorsque le nombre de données est grand.

3.1-Hiérarchique

Le premier type d'algorithme essaie de créer une hiérarchie des clusters, les documents les plus similaires sont regroupés dans des clusters aux plus bas niveaux, tandis que les documents moins similaires sont regroupés dans des clusters aux plus hauts niveaux. Selon comment la hiérarchie est créée, ce type d'algorithmes peut encore se diviser en deux: **divisif** ou **agglomératif**. En partition, on tente de diviser un grand cluster en 2 plus petits (approche descendante). En regroupement, on tente de regrouper 2 clusters en un plus grand (approche ascendante, Classification Ascendante Hiérarchique **CAH**).

Principe de l'agglomération (voir **Figure 9**) :

- Chaque individu représente un groupe.
- Trouver les deux groupes les plus proches.
- Grouper ces deux groupes en un nouveau groupe.
- Itérer jusqu'à N groupes.

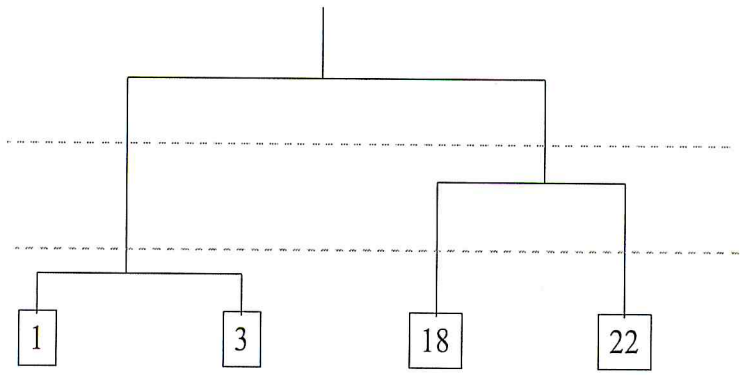


Figure 9 : *Agglomération*

Critique:

- Tourne lentement
- La CAH est peu robuste: il suffit de modifier une distance pour que le saut change (sensibilité aux valeurs extrêmes).

3.2-Non-hiérarchique : Linaire

Le deuxième type d'algorithmes ne crée pas une hiérarchie. Les clusters sont au même niveau, plusieurs algorithmes on étés proposés, Nous détaillerons dans ce qui suit l'algorithme le plus utilisé, **Kmeans**.

3.2.1-Kmeans

Connue aussi avec le nom des centres mobiles, est l'un des algorithmes de clustering les plus fréquemment utilisés en pratique dans le domaine de l'exploration de données et des statistiques **[Har, 1975].** La procédure, qui vient du domaine des statistiques, est simple à mettre en œuvre et peut également être appliquée à de grands ensembles de données. Il s'est avéré que, notamment dans le domaine des données textuelles, le kmeans obtient de bons résultats. En partant d'une solution de départ dans laquelle l'ensemble des documents sont distribués sur un nombre donné de groupes, on tente d'améliorer la solution par des modifications spécifiques de la répartition de documents aux clusters. un ensemble de variantes existe, et le principe de base remonte à Forgy 1965 **[For, 1965]** ou MacQueen 1967 **[Mac, 1967].**
Le principe de base est indiqué dans l'algorithme suivant:

Entrée : un ensemble D, mesure de distance *dist,* un nombre *k* de clusters.

Sortie : un partitionnement P de l'ensemble D (en ensemble de k clusters disjoints de D avec $\bigcup_{P \in P} P = D$

1. choisir k points de données arbitraires de D comme des centroides de départ.
2. Répéter.
3. Assigner à chaque point de P le centroide le plus proche en respectant la mesure dist.
4. Recalculer les centroides.
5. Jusqu'à ce que les centroides des clusters sont stables.
6. Retourner l'ensemble **P** de *k* clusters.

Critique de l'approche :

L'algorithme Kmeans doit sa popularité à sa capacité de traiter de large ensemble de données.

Le principal inconvénient de l'algorithme est que la partition finale dépend du choix de la partition initiale. L'algorithme est sensible à la sélection initiale des centroides, et nécessite que l'utilisateur lui fournisse le nombre K de classes.

Remarque : Plusieurs méthodes ont étés proposées, pour estimer ce nombre tels que : ACP, Scree test (test de coude), utiliser une CAH avant Kmeans, peut aussi nous renseigner sur le nombre de clusters. Pour plus de détails sur toutes ces méthodes, Voir **Annexe.**

3.3-Evaluation (Validation de classes)

La validation des structures de regroupement est la partie la plus difficile dans le processus de clustering.

Nous avons vu précédemment que pour la classification supervisée, nous avons une variété de mesures pour évaluer la qualité de notre modèle, qu'on est-il pour la classification automatique ?

Plusieurs méthodes ont été proposées pour l'évaluation des résultats des algorithmes du clustering, la plus utilisé est :

Le coefficient de silhouette :

Le coefficient de silhouette permet de mesurer la qualité d'appartenance d'un document à un cluster. Cette mesure génère des valeurs entre -1 et 1 et plus la valeur est haute, plus le document est bien placée. En faisant la somme des coefficients de silhouette, on retrouve une mesure de qualité du clustering, plus la somme est haute, plus les clusters sont bien séparés.

Calcul du coefficient de silhouette :

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

Où

a(i) représente la distance moyenne entre le document i et les documents de son cluster, et b(i) représente la plus petite distance moyenne entre le document i et les documents d'un autre cluster.

Remarque : Notons que l'avis d'un expert est le meilleure moyen pour valider les résultats des deux types d'algorithmes, mais malheureusement cette solution est très couteuse est subjective.

4-Formules de distance

Plusieurs algorithmes de classification, s'appuient sur le principe des mesures de distance.

Il existe plusieurs variantes de distance entre documents et classes qui peuvent être utilisées, dont l'influence sur les performances d'un système de classification est démontrée.

4.1-Calcul de distance

4.1.1- Définition de la distance

Une distance est une fonction de $E \times E$, où E est un espace vectoriel.

Cette fonction est caractérisée par les propriétés suivantes :

$D(x, y) \geq 0$ $\quad D(x, y) = 0 \Longleftrightarrow x = y$

$D(x, y) = D(x, y)$ $\quad D(x, y) \leq D(x, z) + D(z, y)$

x, y, z sont des éléments de l'espace E.

Dans notre contexte, ces éléments sont soit des textes soit des classes.

4.1.2- Variantes de distance

Il existe plusieurs fonctions de distance, nous citons :

4.1.2.1-La distance Euclidienne

$$D_e(x, y) = \sqrt{\sum_i (x_i - y_i)^2}$$

4.1.2.2-La distance Manhattan

$$D_m(x, y) = |x_1 - y_1| + |x_2 - y_2| + \ldots + |x_n - y_n|$$

4.1.2.3- La distance Cosinus

$$D_c(x, y) = \max\{ |x_1 - y_1|, |x_2 - y_2|, \dots, |x_n - y_n| \}$$

Remarque : La distance euclidienne est fréquemment utilisée en tant que mesure de distance entre deux documents.

5-Conclusion

Nous avons tenté tout au long de ce chapitre de présenter les algorithmes d'apprentissage les plus utilisés. Les algorithmes supervisés proposent justement un modèle grâce auquel on peut déduire les classes des nouvelles données. De façon simple, le but de ces techniques est de découvrir pourquoi chaque document d'exemple a été rangé dans telle ou telle classe, afin de **prédire** la classe de **nouveaux documents** à ranger dans le futur. A la différence des algorithmes non supervisés qui segmentent des documents hétérogènes en un certain nombre de classes plus au moins hétérogènes, en fonction de leur similitude. Nous avons abordé pour les deux types d'algorithmes comment évalué leurs résultats.

Nous avons exposé aussi les mesures de distance entre documents, qui sert à regrouper les documents les plus proches dans la même catégorie, et les plus distants dans des catégories distinctes.

Pour atteindre nos objectifs, nous avons choisi d'utiliser l'apprentissage non supervisé, pour la classification de notre corpus et l'apprentissage supervisé pour le traitement des nouveaux documents.

Chapitre 4

Etude et conception

Sommaire

1- Introduction

Rappelons que le but de notre travail est d'utiliser et d'évaluer une ou plusieurs méthodes du Text Mining dans le but d'obtenir un système qui soit capable de classifier automatiquement les notices (une présentation détaillée de la structure des notices est faite dans le dernier chapitre) de la base de données médicale MEDLINE, et permettre aux administrateurs une meilleure organisation de cette base.

Dans ce chapitre, nous présentons la conception de notre système. Pour cela, nous détaillons les différentes phases de notre modélisation, aboutissant au système fini et opérationnel.

2- Présentation de la méthode de conception

Pour la conception de notre système, nous avons opté pour l'approche orientée objet, en prenant soin de le modéliser en utilisant le langage **UML** (Unified Modeling Language).

Cependant, UML n'étant qu'un langage et pas une méthode (car il ne donne pas une démarche particulière à suivre pour effectuer le travail), nous avons choisi de suivre la méthode **OMT** (Object Modeling Technique) **[Man, 1999]**.

2.1-La méthode OMT

C'est une méthode d'analyse et de conception qui a pour but de formaliser les étapes de développement d'un système afin de le rendre plus fidèle aux besoins de l'utilisateur du système.

Cette méthode est basée sur une modélisation entités/relations étendue afin d'intégrer les concepts objets.

Notre système est décrit selon les trois axes présentés dans la **Figure 10.**

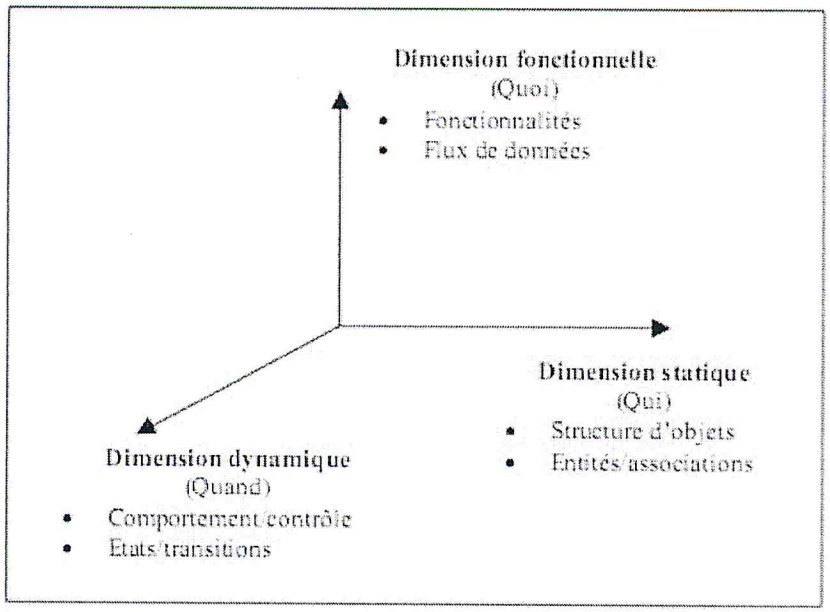


Figure 10: *Représentation des trois axes de description d'un système.*

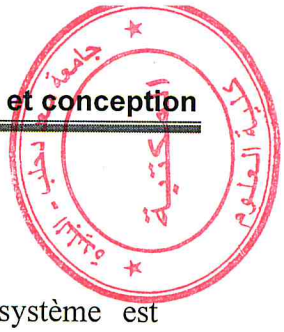
✓ ***Le modèle fonctionnel :*** il permet une meilleure visibilité de ce qui doit être fait par le système et décrit en particulier les transformations des données par rapport aux fonctions du système, la nature et le circuit des échanges au moyen de représentations de flux de données, en dehors de toute considération temporelle.

✓ ***Le modèle dynamique :*** représente les différentes séquences d'exécutions des opérations, à travers le diagramme d'activité du système.

✓ ***Le modèle statique :*** représente la structure statique par l'ensemble des classes et des relations (associations, agrégations, généralisations et spécialisations) qui les relient, ainsi que les attributs, contraintes et opérations.

Dans ce qui suit, nous décrivons notre système selon les trois aspects :

- Fonctionnel.
- Dynamique.
- Statique.



3-Aspect fonctionnel

La définition des besoins doit traduire ce que le nouveau système est susceptible d'apporter aux utilisateurs, en faisant abstraction de la manière dont il sera construit. Cette étape décrit les différentes fonctionnalités du système et surtout la façon de les utiliser.

L'emploi du modèle de cas d'utilisation est une bonne approche pour collecter les besoins des futurs utilisateurs du système. Pour cela nous procédons comme suit :

- ✓ L'identification des acteurs.
- ✓ L'identification des cas d'utilisation du nouveau système (par rapport aux objectifs).
- ✓ La description textuelle de chaque cas d'utilisation.
- ✓ Et enfin, élaborer le diagramme de cas d'utilisation.

La phase de définition des besoins permet de savoir :

- ✓ Ce que le nouveau système apportera aux futurs administrateurs de la base MEDLINE.
- ✓ Comment le système se comporte face à un administrateur.

3.1- Identification des acteurs

Les acteurs sont les différents utilisateurs du système. Dans notre cas nous avons un seul utilisateur (administrateur qui es noté **admin**).

admin: est le superviseur qui utilise notre système pour classifier la base MEDLINE.

3.2- Identification des cas d'utilisation

Un cas d'utilisation est une suite d'événements, souvent initiés par un acteur. Il correspond à l'une des possibilités d'utilisation du système. Nous regroupons dans le **tableau 3** la liste des cas d'utilisation du système.

#	Cas d'utilisation
1	Préparer les données
2	Regrouper les notices
3	Créer le modèle de classification (Naïve Bayes)
4	Classer une nouvelle notice

Tableau 3 : *Liste des cas d'utilisation.*

3.3- Description textuelle des cas d'utilisation

Dans cette phase, nous décrivons textuellement les différents cas d'utilisation.

✓ ***Préparer les données***

Ce cas d'utilisation

- ***But :*** permet de préparer les données d'entrée de notre système.
- **Acteur :** *admin*

✓ ***Regrouper les notices***

Ce cas d'utilisation

- ***But :*** permet de créer un catalogue global de notre base de textes, ceci revient à attribuer une classe pour chaque document du corpus.
- **Acteur :** admin

✓ ***Créer le modèle de classification Naive Bayes***

Ce cas d'utilisation

- ***But :*** permet de créer le modèle de classification Naive Bayes, qui est entrainé avec les documents pré-classés, par regroupement.
- **Acteur :** admin

✓ ***Classer une nouvelle notice***

Ce cas d'utilisation

- ***But :*** permet de prédire la ou les classes des nouveaux documents, à base du modèle appris (classifieur) Naive Bayes.
- **Acteur :** admin

3.4- Diagramme des cas d'utilisation

Les diagrammes de cas d'utilisation représentent :

- Les acteurs.
- Les cas d'utilisation inclus dans le système.
- Ainsi que l'interaction des deux, représentée par une flèche.

Système de classification de documents médicaux :

Notre système permet à l'administrateur de classifier le corpus MEDLINE avec Kmeans, de construire un modèle de classification Naive Bayes, capable de classer les nouveaux documents.

La (**Figure 11**) structure les besoins et les objectifs du système.

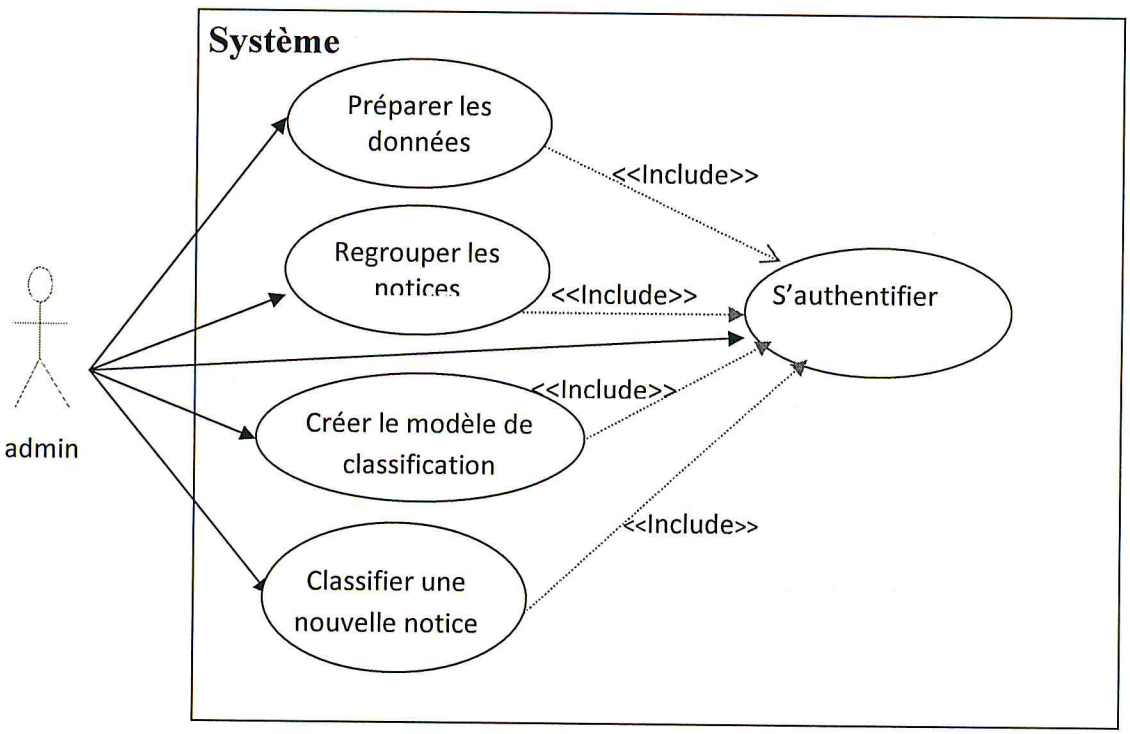


Figure 11 : *Diagramme des cas d'utilisation du système.*

4- Aspect dynamique

Pour effectuer l'analyse dynamique du système, nous établissons des diagrammes d'activités. Cela, afin d'illustrer le dynamisme de notre système.

4.1-Elaboration des diagrammes d'activités

Le diagramme d'activité représente la dynamique du système. Il montre l'enchaînement des activités d'un système ou même d'une opération. Le diagramme d'activité représente le flot de contrôle qui retrace le fil d'exécution et qui transite d'une activité à l'autre dans le système.

Dans notre système deux processus peuvent être distingués :

- **Processus N°1** : résume la création du catalogue du corpus utilisé et la création du modèle d'apprentissage. **La Figure 12**, représente les activités globales du premier processus.

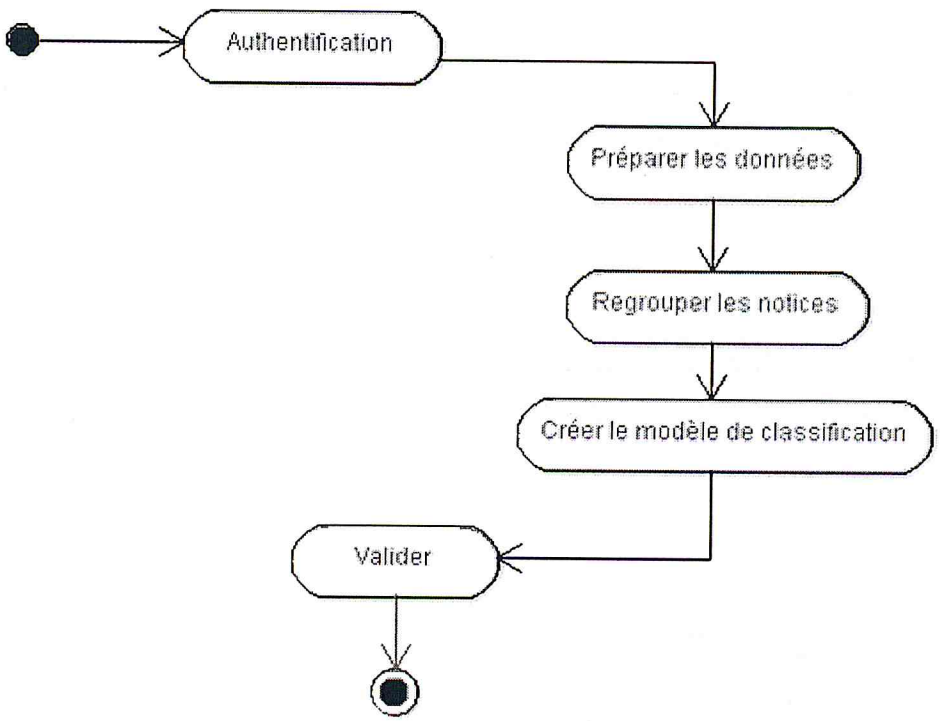


Figure 12 : *Processus de classification initial.*

- **Processus N°2** : qui sert à classer les nouveaux documents. **La Figure 13**, représente les activités globales du deuxième processus.

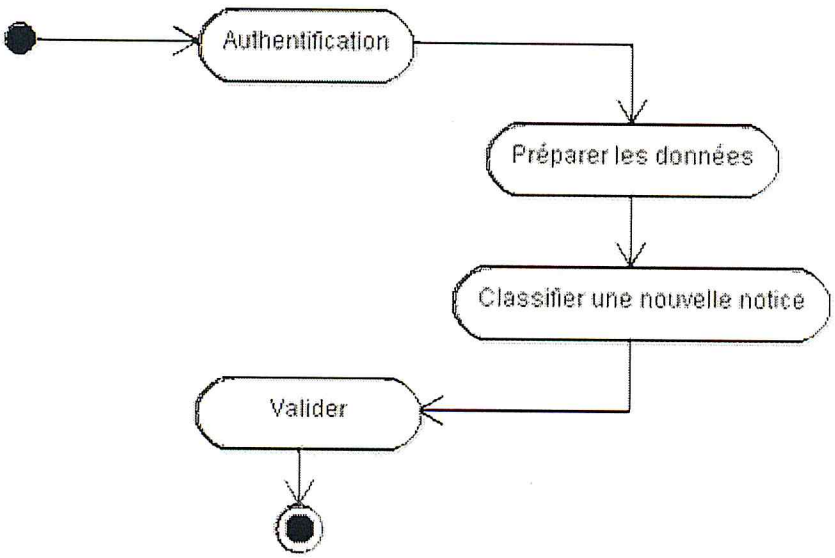


Figure 13 : *Processus de classification d'une nouvelle notice.*

✓ ***Cas d'utilisation Préparer les données***

La Figure 14 représente les activités qui s'exécutent pour rendre notre fichier de données des notices MEDLINE compréhensible par les algorithmes d'apprentissage.

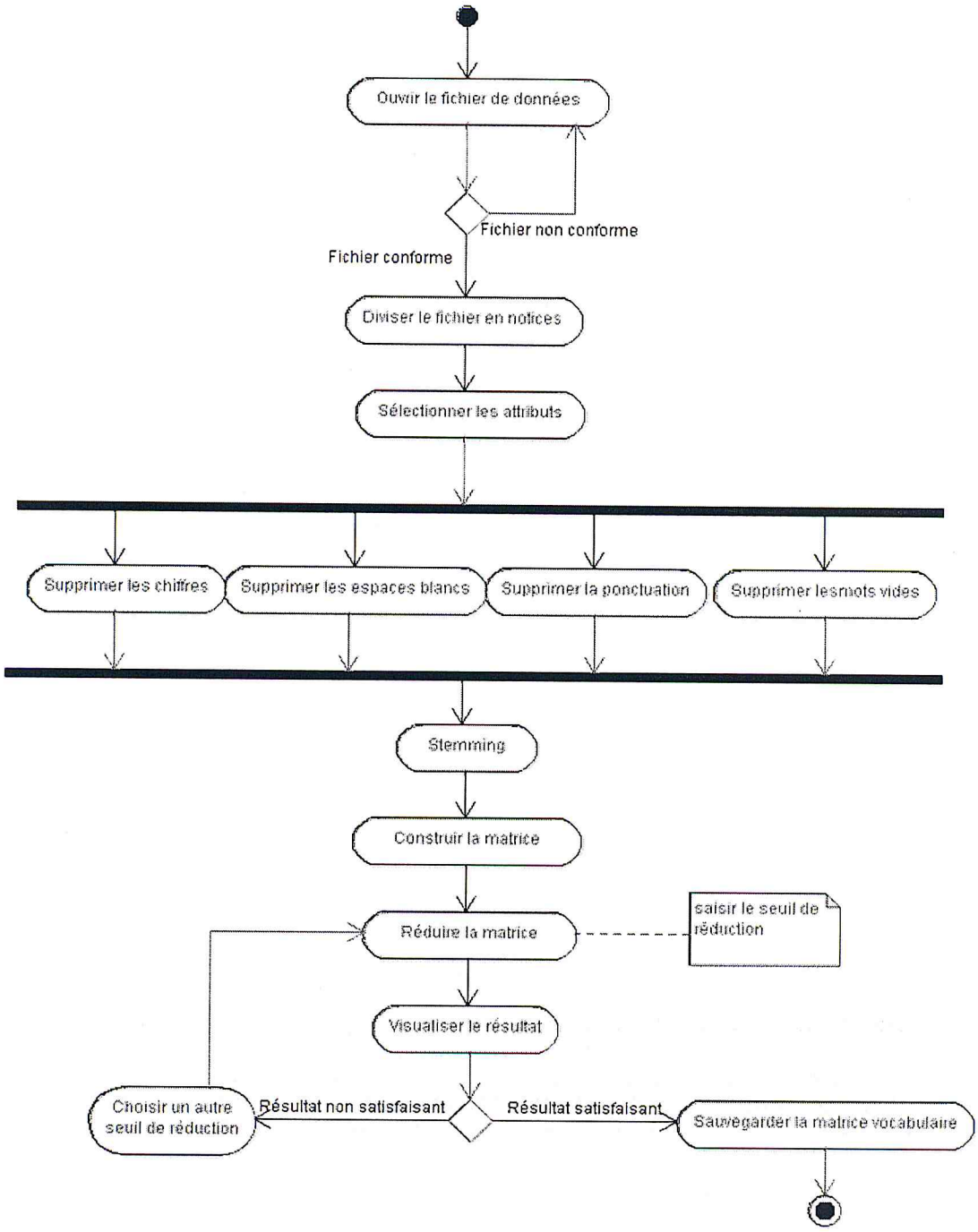


Figure 14 : *Diagramme d'activités pour le cas d'utilisation Préparer les données.*

Détails :

- **Ouvrir le fichier de données :** L'administrateur ouvre le fichier de données de la base bibliographique disponible sous format XML (préalablement téléchargé et stocké sur son disque).
- **Diviser le fichier en notices :** nous divisons le fichier de données d'entrée en notice stockée en format XML.
- **Sélectionner les attributs :** notre système s'appuie exclusivement sur les attributs **tire de l'article** (TI), **le résumé** (AB) et l'ensemble des **descripteurs (MH ou MeSH).**
- **Supprimer les chiffres :** consiste à supprimer les chiffres, que nous jugeons vide d'information, utile pour la classification.
- **Supprimer les espaces blancs :** Supprimer tout les espaces blancs.
- **Supprimer la ponctuation :** éliminer tout symbole qui ne correspond pas à une lettre de l'alphabet (points, virgules, traits d'union, chiffres etc.). Cette opération est motivée par le fait que ces caractères ne sont pas liés au contenu des documents et ne change rien au sens s'ils sont omis et par conséquent ils peuvent être négligés.
- **Supprimer les mots vides :** :(stoping, en anglais): qui correspond à la suppression de tous les mots qui sont trop fréquents (ils n'aident donc pas à distinguer entre les documents) ou jouent un rôle purement fonctionnel dans la construction des phrases (articles, prépositions, etc.). Les mots à éliminer, connus comme stopwords, sont récoltés dans la stoplist qui contient en général entre 300 et 400 éléments.
- **Stemming :** qui consiste à remplacer chaque mot du document par sa racine.
- **Construire la matrice** : Les notices sont représentées selon **l'approche vectorielle**, chaque attribut représentant un mot du vocabulaire. Nous avons retenu la technique **TF-IDF** pour la pondération des attributs.
- **Réduire la matrice :** Pour réduire la taille des vecteurs tout en éliminant les mots inutiles, une **sélection d'attributs** est appliquée.
- **Visualiser le résultat :** Un nuage de mot est affiché à l'administrateur, qui résume tout les mots utiles extraits des notices.
- **Sauvegarder la matrice vocabulaire :** si l'administrateur est satisfait des résultats de la transformation de son fichier de données en une matrice de mots, la matrice sera sauvegardée.

- **Choisir un autre seuil de réduction :** si l'administrateur juge que le nuage de mots présente beaucoup / ou peu de mots clès, alors il n'a qu'a augmenter ou bien diminuer le seuil de réduction.

✓ ***Cas d'utilisation Regrouper les notices***

La Figure 15 représente les activités qui ce déclenche lors du regroupement des notices MEDLINE.

Pour grouper notre corpus en classes, nous avons choisi l'algorithme Kmeans car c'est l'algorithme le plus simple et le plus efficace qui puisse être aisément modifié et contrôlé.

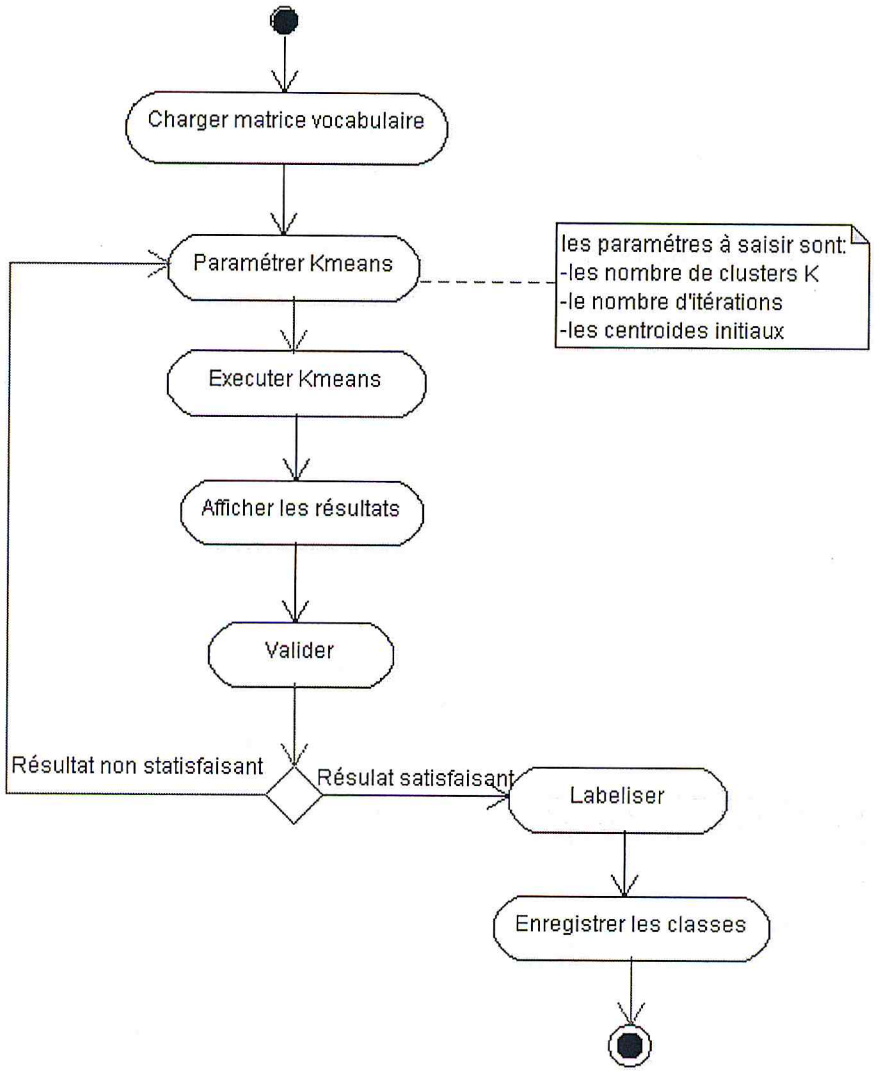


Figure 15 : *Diagramme d'activités pour le cas d'utilisation Regrouper les notices.*

Détail :

- **Charger la matrice vocabulaire :** la matrice créée précédemment, est en format adéquat pour l'exécution de notre algorithme d'apprentissage.
- **Paramétrer Kmeans :**

L'algorithme Kmeans consiste à grouper les notices selon un critère bien déterminé. L'entrée de l'algorithme est le nombre k de groupes (cluster), nombre maximum d'itérations ainsi que les centres initiaux, si l'administrateur ne lui donne pas les deux derniers paramètres l'algorithme les choisira aléatoirement, et les résultats seront différents jusqu' a stabilisation de l'algorithme.

- **Exécuter l'algorithme Kmeans :** Lancer l'algorithme Kmeans avec les paramètres d'entrées cités auparavant.
- **Afficher les résultats :** les résultats sont affichés à l'aide d'un graphe silhouette[1], qui donne une idée générale sur les groupes de documents (clusters).La liste des documents attribués à des classes est aussi affichée.
- **Valider :** Nous utilisons le coefficient de silhouette pour valider les résultats de Kmeans, l'administrateur doit ré-exécuter l'algorithme jusqu'à satisfaction des résultats en fonction de ce coefficient, qu'il doit maximiser, il doit être le plus proche de 1. Nous estimons qu'entre 0.25 et 0.7 le regroupement est satisfaisant.
- **Labeliser :** consiste à donner des noms au classes trouvées, les 10 mots clés de la classe sont affichés, et l'administrateur les étudie pour décider de l'appellation finale de la classe.
- **Enregistrer les classes :** Si l'administrateur est satisfait des résultats, les clusters sont enregistrés.

✓ ***Cas d'utilisation Créer le modèle de classification***

La Figure 16 représente les activités qui ce déclenchent lors de la création du modèle d'apprentissage.

Nous avons choisi le modèle Naive de Bayes, qui est d'après David.D Lewis dans **[Lew, 2004]** et Hassane Hilali dans **[Hil, 2009]**, un algorithme facile et simple à implémenter, Requiert une petite quantité de données d'apprentissage pour estimer les

[1] Est un graphe qui permet d'afficher les clusters.

paramètres. Et le plus important c'est que les méthodes Naïve Bayes donnent de bons résultats.

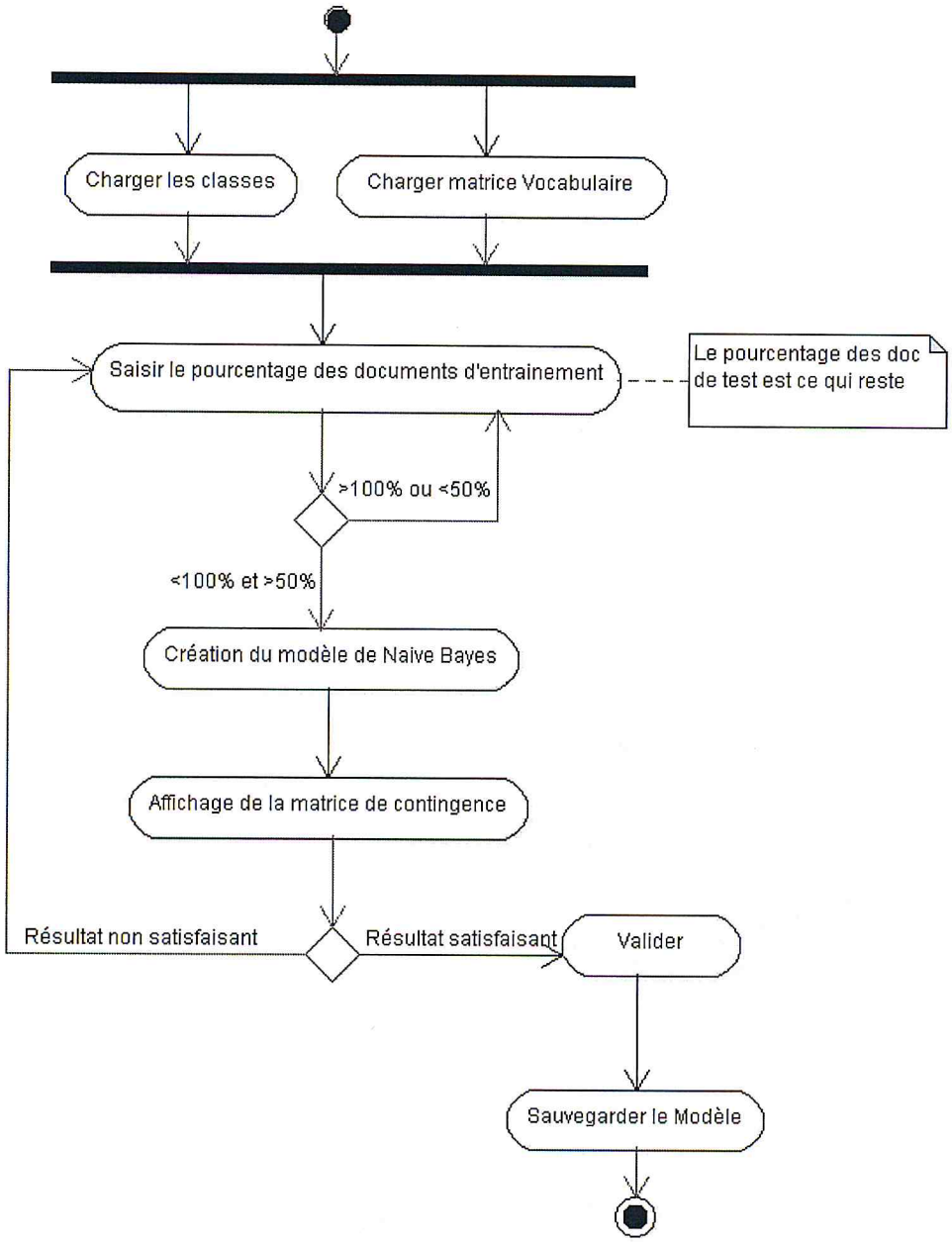


Figure 16 : *Diagramme d'activités pour le cas d'utilisation Créer le modèle de classification.*

Détails :

- **Charger les classes/ Charger vocabulaire :** les deux fichiers de données sont utiles pour la création du modèle de classification
- **Saisir le pourcentage des documents d'entrainement :**Notre modèle d'apprentissage doit avoir en entré, des documents d'apprentissage et des documents de test,dans la théorie, le pourcentage des documents d'apprentissage doit etre superieur au pourcentage des documents de test
- **Création du modèle de Bayes :** c'est l'execution de l'algorithme Naive Bayes.
- **Affichage de la matrice de contingence :** Les résultats de la classification des documents test sont affichés sous forme de matrice de contingence, qui donne une idée suffisante sur l'aptitude de prédiction du modèle Naive Bayes.
- **Valider:** l'administrateur valide le modèle.
- **Sauvegarder le modèle :** si le modèle est validé, il sera sauvegardé, sinon l'administrateur doit revoir les pourcentage des documents de test et d'entrainement donné au modèle Naive Bayes

✓ ***Cas d'utilisation Classifier une nouvelle notice***

La Figure 17 représente les activités qui ce déclenche lors du traitement d'une nouvelle notice, dans le but de la classer.

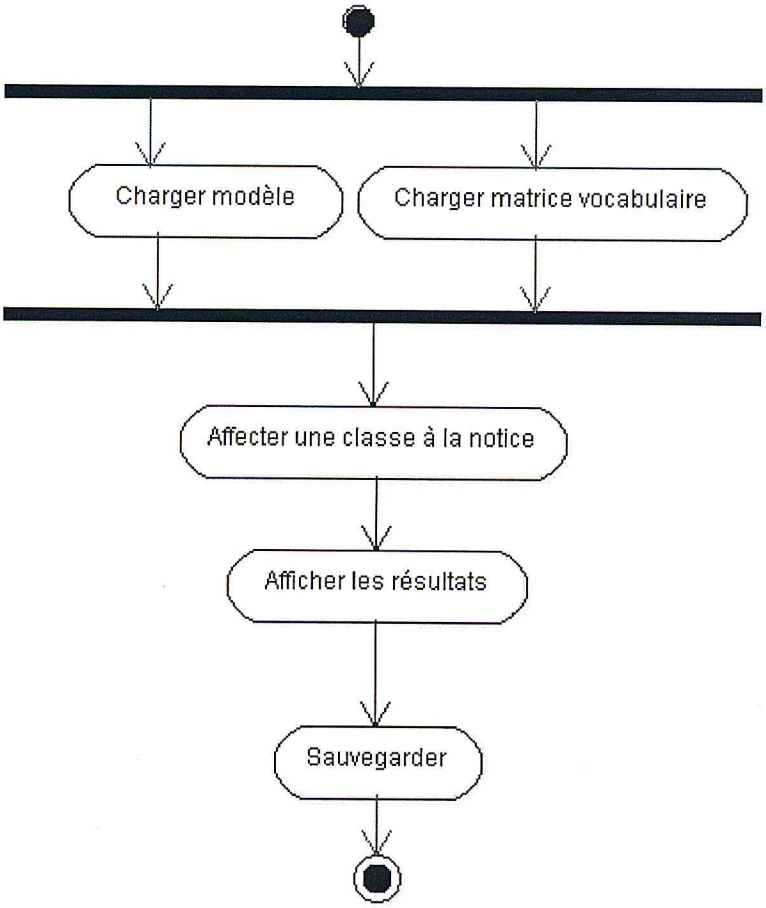


Figure 17 : *Diagramme d'activités pour le cas d'utilisation Classifier une nouvelle notice.*

Détails : une nouvelle notice passe d'abord par le diagramme d'activités **Préparer les données**, pour qu'elle soit converti en vecteur. Le modèle d'apprentissage de Bayes s'occupe de prédire sa classe.

- **Charger la matrice vocabulaire :** la matrice vocabulaire sera chargée
- **Charger le modèle :** le modèle précedement crée sera chargé.
- **Affecter la classe :**la fonction predir() de **NB** attribut la probabalité maximale d'appartenance de la nouvelle notice à une des classes.
- **Afficher les résultats :** revient à afficher la probabalité maximale d'appartenance de la nouvelle notice à une des classes du corpus.
- **Sauvegarder :** si l'administrateur est satisfait , l'action d'enregistrement de l'attribut classe est déclenchée.

- **Enregister classe dans document :** revient à stocker l'information de la classe du document.

✓ ***Cas d'utilisation s'authentifier***

Qui consiste à la vérification du login et du mot de passe de l'admin.

5-Aspect statique

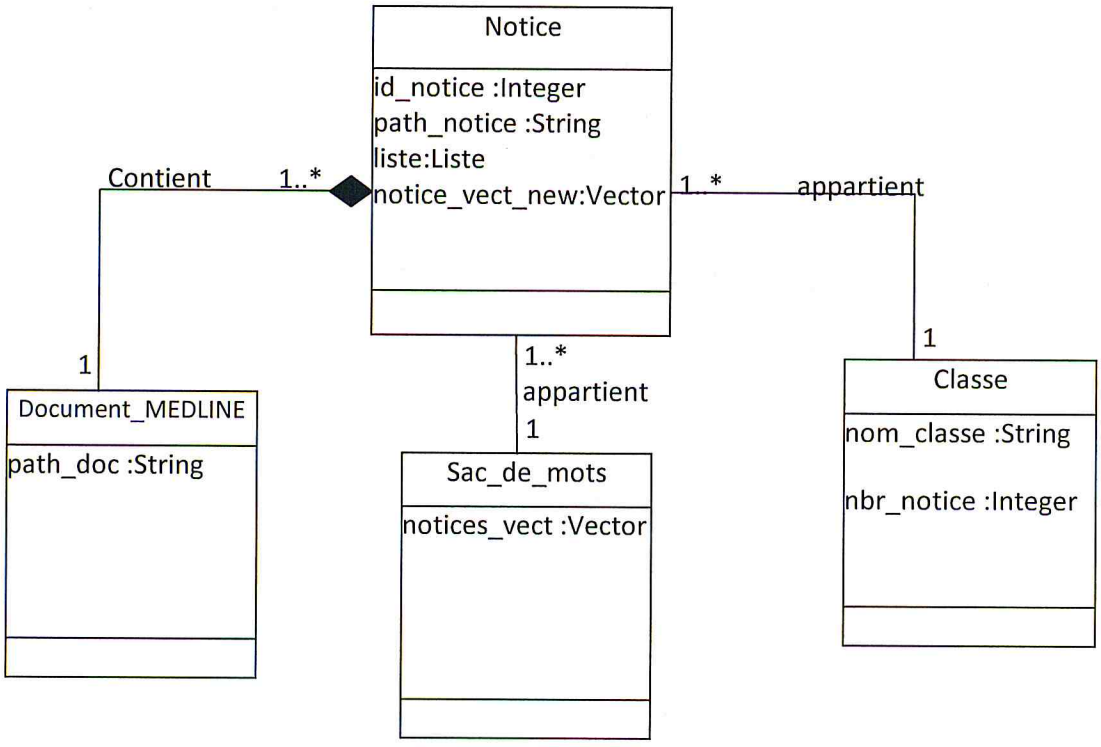
Dans cette étape, nous établissons le diagramme de classe représentant la structure des objets, afin de représenter l'aspect statique du système.

5.1-Elaboration du modèle objet

Le modèle objet identifie les relations structurelles statiques entre les objets, et met en évidence les différents types d'objets.

a) ***Diagramme des classes du système :***

Nous identifions ici les différentes classes qui composent le système.



igure 18 : *Diagramme des classes du système.*

d) ***Développement du modèle objet :***

Ici nous explicitons les différentes classes et leurs attributs.

Classe	Attributs/méthodes	Description
Document_MEDLINE	**Attributs** path_doc **Méthodes** **Cut** (Path_doc)	Chemin physique du document de données Découper le fichier en plusieurs fichiers XML.
Notice	**Attributs** id_notice path_notice liste notice_vect_new **Méthodes** List Liste = **Notice_liste** (Path_notice, Id_notice,TI,AB,MH) **Prepare_data** (notice_vect_new)	Extrait les attributs TI,AB,MH de la notice XML et transforme la notice en une liste. Détaillée dans chapitre 5 partie 1.4
Sac_de_mots	**Attributs** notices_vect **Méthodes** **Prepare_data** (Notices_vect)	Détaillée dans chapitre 5 partie 1.4

Classe	**Attributs** Nom_classe Nbr_notice **Méthodes** Clustering(Notices_vect) Build_model(Notices_vect) Classify_new_document(Vecteur)	Toutes ces fonctions sont détaillée dans chapitre 5 partie 1.4

Tableau 3 : *Représentation des classes d'objets*

6-Conclusion

Dans ce chapitre, nous avons détaillé la conception de notre système. Pour ce faire, nous avons adopté la méthode OMT tout en modélisant notre système avec le langage UML.

Dans le chapitre suivant, nous décrivons enfin le système résultant de notre conception.

Chapitre 5

Implémentation et expérimentations

Sommaire

1-Implémentation

1.1-Introduction

Nous présentons dans ce paragraphe, les outils exploités pour le développement du logiciel tels que le choix du langage de programmation, l'environnement de programmation, ainsi que quelques algorithmes implémentés.

1.2-Configuration matérielle et logicielle

- Un PC Pentium 4 à 3GHZ et 2Go de RAM.
- Argo UML.
- Microsoft Office 2007 Professionnel.
- Apache version 5.

1.3-Le langage de programmation

a. Création des programmes avec R

Le langage de programmation utilisé est Le langage R version 2.15.0 (2012-03-30) : R (ou GNU-R) est un logiciel de statistiques interactif et interprété. Il fourni un langage de commande très souple et qui est ouvert : il possède des interfaces vers d'autres programmes comme C, java, PHP et Fortran. En outre il est le proche cousin de S-plus [1]qui est un logiciel payant.

R est un logiciel gratuit de data mining destiné à l'enseignement et à la recherche, diffusé sur internet. Il implémente une série de méthodes de fouilles de données issues du domaine de la statistique exploratoire, de l'apprentissage automatique et des bases de données.

- Son premier objectif est d'offrir aux étudiants et aux chercheurs d'autres domaines (médecine, bioinformatique, marketing, etc.) une plate-forme facile d'accès, respectant les standards des logiciels actuels, notamment en matière d'interface et de mode de fonctionnement, il doit être possible d'utiliser le logiciel pour mener des études sur des données réelles.
- Le deuxième objectif est de proposer aux chercheurs une architecture leur facilitant l'implémentation des techniques qu'ils veulent étudier, de comparer les

[1] Consulter http://statwww.epfl.ch/splus/

performances de ces algorithmes. Point très important à nos yeux, la disponibilité du code source qui assure la reproductibilité des expérimentations publiées, et surtout, elle permet la comparaison et la vérification des implémentations.

Le site de diffusion de R est http://www. R-project.org ; il permet de télécharger le logiciel gratuitement, de suivre ses évolutions.

b. Création de l'interface avec PHP

PHP appartient à la grande famille des descendants du langage C, dont la syntaxe est très proche .**PHP** principalement utilisé pour produire des pages Web dynamiques via un serveur http.

c. Stockage de données dans des fichiers XLS

L'enregistrement des données se fait dans des fichiers de type XLS.

1.4-Quelques algorithmes

a. Prepare_data

Comme son nom l'indique, cette procédure prépare les données avant la modélisation, c'est l'étape préliminaire avant toute application de Text Mining. Nous avons utilisé le package "tm" de R pour l'analyse textuelle. Il est disponible via le CRAN et a un site dédié :http://tm.r-forge.r-project.org/

Début

Pour chaque document du corpus **faire**

Diviser le document en unités lexicales (mots) ;

Extraire les mots vides (stop words) ;

Supprimer les signes de ponctuation ()[]{}=:?!;-,_"+*/.",<>≥≤%«»&, ;

Supprimer les chiffres ;

Convertir les majuscules en minuscules ;

Stemmer ;

Sélectionner un ensemble constitué des k mots ayant les plus hauts, contenus en informations (N est choisi après la réduction) ;

Calculer les poids des mots caractéristiques relatifs à chaque document, en utilisant la formule tfidf ;

Représenter chaque document du corpus considéré sous forme d'un **vecteur caractéristique** constitué des N mots sélectionnés précédemment ;

Construction de la matrice Terme-Document [document, term] ;

Enregistrer la matrice ;

Fin.

b. Clustering

C'est l'étape proprement dite de l'application de l'algorithme de regroupement choisi Kmeans.

Debut

Entrer les k centres de départ;

Entrer le nombre maximum d'itérations ;

Entrer l'ensemble des points (la matrice terme-Document) ;

Appliquer Kmeans ;

Labelliser ;

Enregistrer les résultats ;

Fin.

c. Build_model

C'est l'étape de modélisation qui consiste en la construction d'un modèle de classification Naive Bayes.

Debut

Diviser l'ensemble des documents en deux sous ensembles d'une manière aléatoire, par défaut, 70% va servir pour l'entrainement et le reste pour le test du modèle à construire ;

Charger la matrice de document d'entrainement ;

Calculer les probabilités apriori P(Ci)=Nci/N ;

Ou N : nombre de document de corpus

Ci : les classes des documents d'entrainement

Nci : nombre de documents appartenant à la classe Ci

calculer la probabilité conditionnelle pour chaque mot, et pour chaque classe de document ;

Valider le modèle ;

Début

Charger les documents de test ;

Probabilité de Bayes :

Calculer les probabilités de bayes pour chaque classe Ci, sachant le document dj :

```
        Pour i =1 jusqu'à i = nombre de classes faire
            Pi=P (classe=i/document =(mot1, mot2, ……., motn))
        Fin Pour.
            Calculer la matrice de contingence ;
            Sauvegarder les probabilités calculées précédemment pour une utilisation
            ultérieure ;
    Fin.
Fin.
```

d. Classify_new_document

```
Début
        Prepare _data ;
        Charger le modèle de Bayes ;
        Probabilité de Bayes :
        Calculer les probabilités de bayes pour chaque classe Ci, sachant le
        document dj :
        Pour i =1 jusqu'à i = nombre de classes faire
            Pi=P (classe=i/document =(mot1, mot2, ……., motn))
        Fin Pour.
        Comparer les valeurs Pi .La classe à la quelle appartiendra le nouveau
        document est celle à laquelle correspond la plus grande valeur ;
Fin.
```

2-Expérimentations

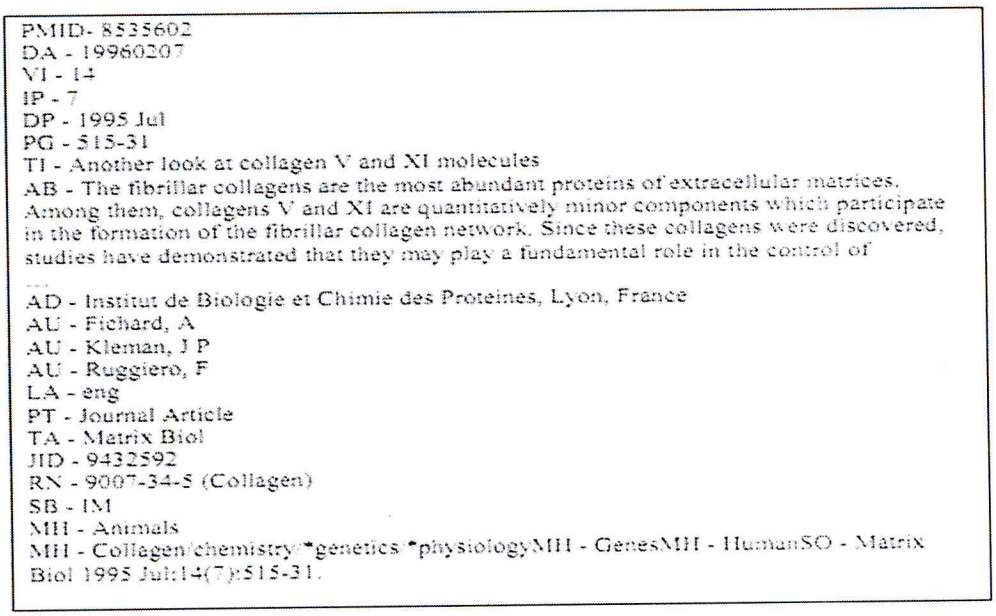
2.1-Introduction

Dans ce paragraphe nous présentons, le corpus MEDLINE, utilisé pour l'évaluation, ainsi que l'ensemble des résultats des expérimentations. Nous terminons par une conclusion générale et des perspectives.

2.2-Corpus de textes MEDLINE

Nous avons utilisé dans notre expérimentation le corpus MEDLINE (MEDical Literature Analysis and Retrieval System on LINE) qui est une banque

documentaire produite par la National Library of Medecine (NLM) située à Bethesda (MD) qui couvre les domaines biomédicaux tels que la biologie, la biochimie, la médecine clinique, la santé publique, l'éthique, la pharmacologie, l'économie liée à la santé, la toxicologie, l'odontologie, la psychiatrie et la médecine vétérinaire. Son accès via Internet s'effectue grâce à son interface nommé PubMed. Ce fonds documentaire comprenant environ 16 millions de références provenant d'environ 5 000 revues scientifiques paraissant dans 37 langues différentes. On estime que la banque s'enrichit de 2000 à 4 000 nouvelles références par jour ouvrable (soit environ 623 000 pour l'année 2006). La langue anglaise reste dominante avec environ 90 % des publications des années 2000 à 2004 et cette proportion augmente au fil des années. La figure suivante illustre un exemple de texte du corpus :



PMID- 8535602
DA - 19960207
VI - 14
IP - 7
DP - 1995 Jul
PG - 515-31
TI - Another look at collagen V and XI molecules
AB - The fibrillar collagens are the most abundant proteins of extracellular matrices. Among them, collagens V and XI are quantitatively minor components which participate in the formation of the fibrillar collagen network. Since these collagens were discovered, studies have demonstrated that they may play a fundamental role in the control of
...
AD - Institut de Biologie et Chimie des Proteines, Lyon, France
AU - Fichard, A
AU - Kleman, J P
AU - Ruggiero, F
LA - eng
PT - Journal Article
TA - Matrix Biol
JID - 9432592
RN - 9007-34-5 (Collagen)
SB - IM
MH - Animals
MH - Collagen/chemistry/*genetics/*physiologyMH - GenesMH - HumanSO - Matrix Biol 1995 Jul;14(7):515-31.

Figure 19 : *Exemple d'une notice bibliographique extraite de MEDLINE.*

Chaque document est structuré suivant un certain nombre d'attributs comme PMID (identificateur unique dans PubMed), DP (date de publication), AU (auteur), PT (type de publication), SO (source), etc. La figure 1 présente un exemple d'une notice complète tandis que la figure 2 illustre le cas d'une référence bibliographique sans résumé. On remarquera que parfois l'information est redondante ; ainsi les champs VI (numéro du volume), IP (numéro), PG (pages), TA (titre abrégé du journal) sont extraites du champ SO tandis que JID (numéro d'identification du journal) dépend de la valeur attribuée au champ

TA[2]. Ces informations peuvent être utiles pour chercher les documents écrits par un auteur particulier ou la liste des articles publiés durant une période donnée et par un institut ou un pays spécifique.

```
PMID- 8582950
DA - 19960319
VI - 10
IP - 7
DP - 1995 Jul
PG - 1628-30
TI - Potential health hazards of assisted reproduction. Problems facing the clinician
AD - Queen Elizabeth Hospital, Gateshead, Tyne, Wear, UK.
AU - Amso NN
LA - eng
PT - Journal Article
TA - Hum Reprod
JID - 8701199
SB - IM
MH - Anxiety/etiology
MH - FemaleMH - Genital Neoplasms, Female/etiologyMH - HumanMH - Oocyte
Donation/adverse effectsMH - Ovarian Hyperstimulation Syndrome/etiology/therapyMH -
Reproductive Techniques/*adverse effectsMH - Stress, Psychological/etiologySO - Hum
Reprod 1995 Jul;10(7):1628-30.
...
```

Figure 20 : *Exemple d'une notice simple MEDLINE.*

Un aspect important de la collection MEDLINE est la présence de descripteurs attribués manuellement part des experts du domaine dans lequel est écrit l'article. Ces termes sont extraits d'un thésaurus hiérarchisé (ou vocabulaire contrôlé), le MeSH[3], qui couvre l'ensemble du domaine biomédicale.

Choix des attributs

Afin de rechercher de l'information, notre système s'appuie exclusivement sur les attributs **« tire de l'article »** TI), **le résumé** (AB) et l'ensemble des **descripteurs (MH ou MeSH)** manuellement sélectionnés. Signalons que le résumé n'est pas disponible pour tous les articles et on peut estimer qu'environ 79 % des articles des années 2000 à 2004 disposent de cette information.

[2] A l'adresse http://www.nlm.nih.gov/bsd/mms/medlineelements.html, on retrouve la description complète de tous les champs d'une référence Medline.
[3] Voir le site http://www.nlm.nih.gov/mesh/.

2.3-Descriptions de l'échantillon utilisé

Un sous-ensemble des données MEDLINE a été utilisé pour les expériences, Certains détails de l'ensemble de données utilisé sont donnés ci-dessous.

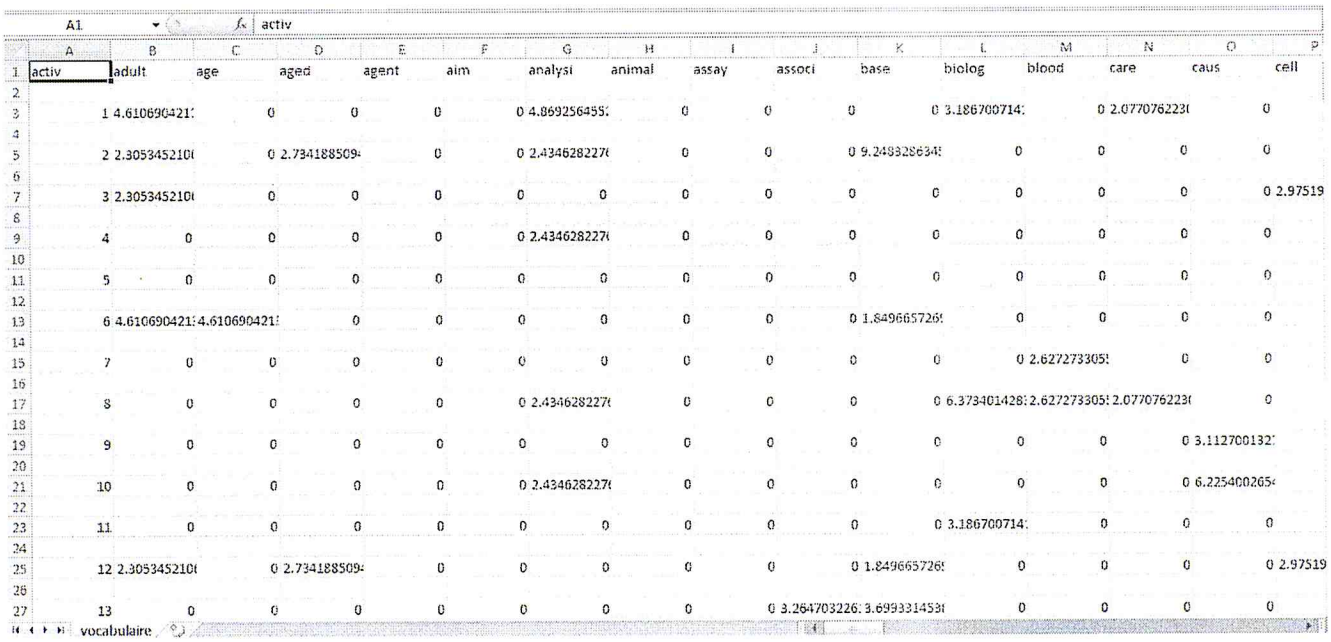
Total des documents MEDLINE : 173

Nombre approximatif de mots dans chaque document: 200

2.4-Résultats du prétraitement des textes

Tous les documents sont transformés en une matrice des fréquences (tfidf) des termes dont les colonnes correspondent à tous les termes du vocabulaire et les lignes correspondent aux document du corpus.

Après la réduction, la matrice résultante contient 117 mots distincts.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	activ	adult	age	aged	agent	aim	analysi	animal	assay	associ	base	biolog	blood	care	caus	cell
2																
3	1	4.610690421	0	0	0	0	4.869256455	0	0	0	0	3.186700714	0	2.077076223	0	
4																
5	2	2.305345210	0	2.734188509	0	0	2.434628227	0	0	0	9.249328634	0	0	0	0	
6																
7	3	2.305345210	0	0	0	0	0	0	0	0	0	0	0	0	0	2.97519
8																
9	4	0	0	0	0	0	2.434628227	0	0	0	0	0	0	0	0	
10																
11	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
12																
13	6	4.610690421	4.610690421	0	0	0	0	0	0	0	1.849665726	0	0	0	0	
14																
15	7	0	0	0	0	0	0	0	0	0	0	0	2.627273305	0	0	
16																
17	8	0	0	0	0	0	2.434628227	0	0	0	0	6.373401428	2.627273305	2.077076223	0	
18																
19	9	0	0	0	0	0	0	0	0	0	0	0	0	0	3.112700132	
20																
21	10	0	0	0	0	0	2.434628227	0	0	0	0	0	0	0	6.225400265	
22																
23	11	0	0	0	0	0	0	0	0	0	0	3.186700714	0	0	0	
24																
25	12	2.305345210	0	2.734188509	0	0	0	0	0	0	1.849665726	0	0	0	0	2.97519
26																
27	13	0	0	0	0	0	0	0	0	3.264703226	3.699331453	0	0	0	0	

vocabulaire

Figue 21 : *Matrice du vocabulaire.*

Déterminer le nombre K cluster

- **Avec ACP**

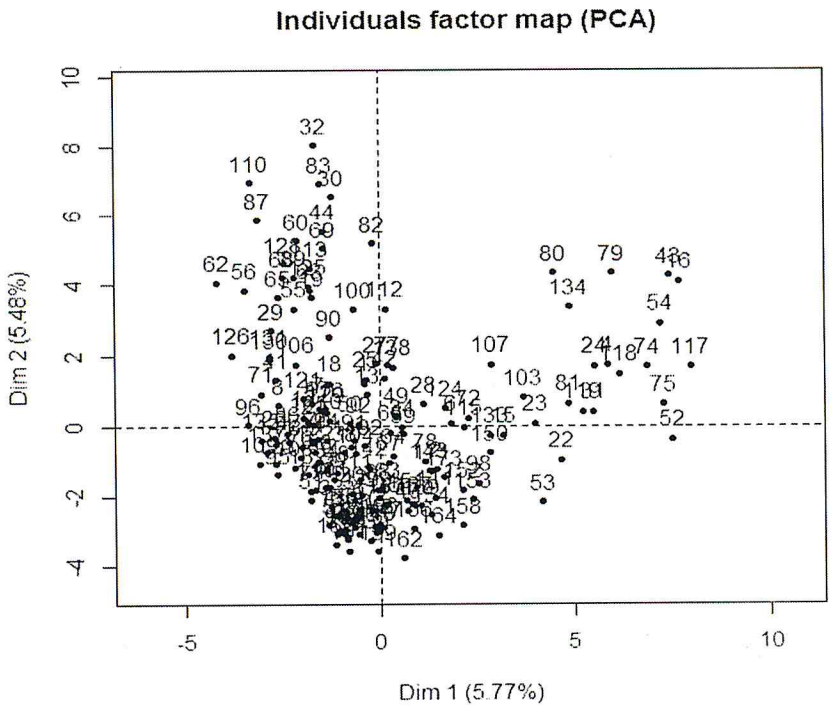


Figure 22 : L'ACP.

- **Avec La CAH**

La classification ascendante hiérarchique (CAH) des variables avec la distance euclidienne, donne le dendrogramme suivant :

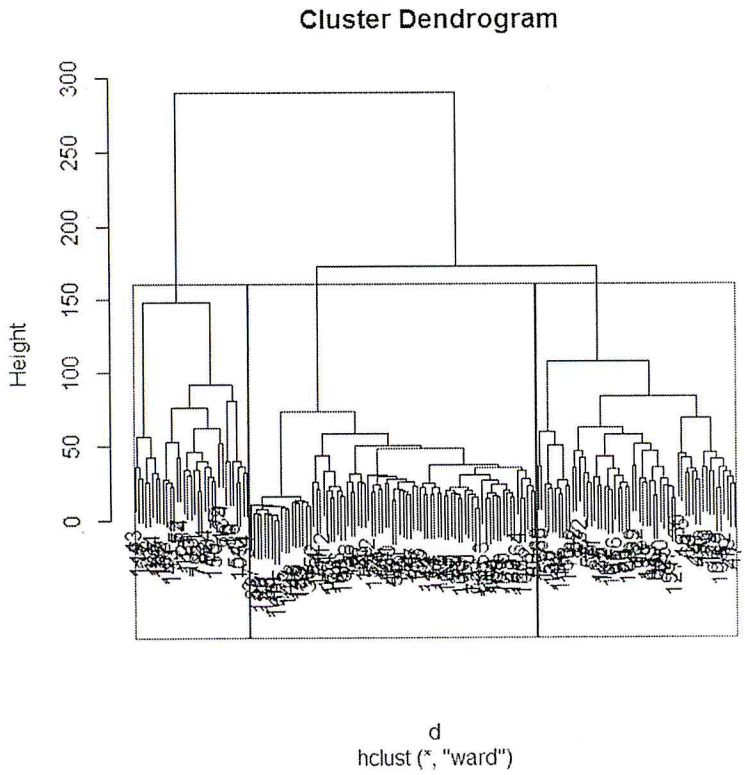


***Figure 23** : Dendrogramme avec CAH*

Nous avions vu dans le dendrogramme ainsi que dans l'ACP que les partitions en 2 ou 3 classes semblent les plus appropriées dans cette classification. Pour acquérir le nombre de classe adéquat : essayant d'utiliser une troisième méthode, la courbe d'évolution de la variation de l'inertie intra-classe (test de coude) en fonction du nombre de classes. Le premier coude apparus correspond à 2 classes de variables.

- **Avec test de coude**

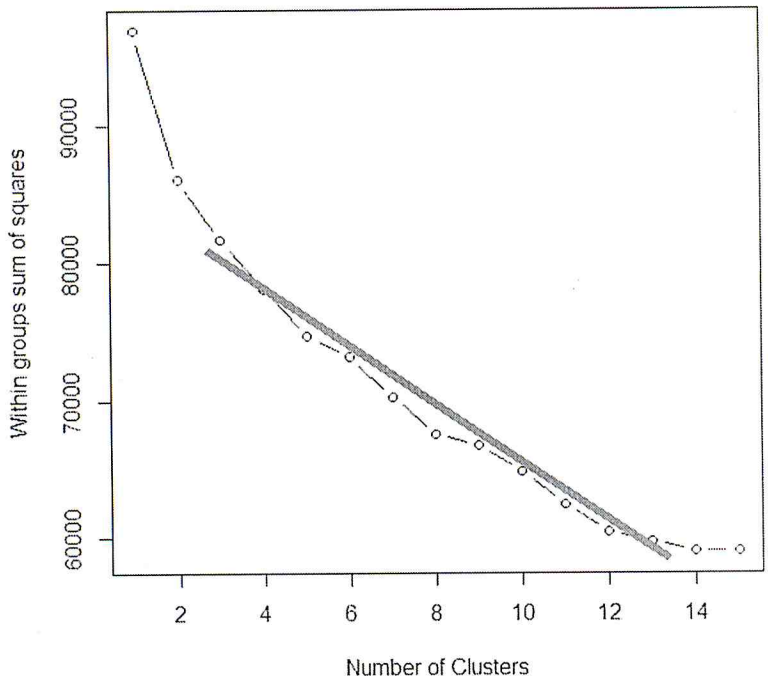


Figure 24 : *Test de coude.*

D'où nous confirmons le résultat de la méthode Scree test, à savoir **2 cluster**.

Nous exécuterons Kmeans avec K=2

2.5-Résultats de Kmeans avec K=2

Silhouette de Kmeans pour k=2

Deux groupes de documents ont été créé :

Groupe1 : Contient 151 documents

Groupe2 : Contient 22 documents

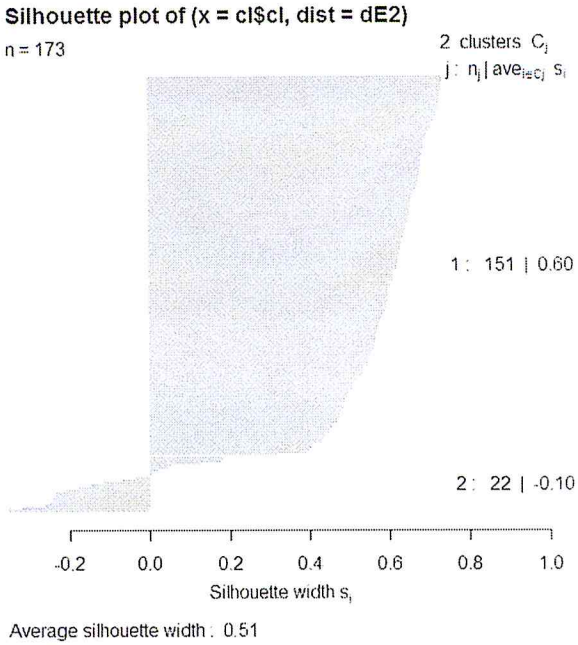


Figure 25 : *La silhouette de Kmeans.*

2.6-Résultats de labellisation

Notre algorithme de labelisation a donné les résultats suivants

Clusters	Les 10 premiers mots clés	Tentative d'appelation
Cluster 1	agnosi patient diseas syndrom idemiolog risk hypertens peut etiolog sever	Biology
Cluster 2	rat insulin cell secret express gene gluc os metabol pancreat	Diabet

Tableau 4: *Labellisation des clusters.*

2.7-Validation des résultats de Kmeans

Après plusieurs exécutions de Kmean, **le coefficient de silhouette** maximum obtenu est de **0.51** (voir **figure 25**), que nous estimons suffisant.

2.8- Résultats de Naïve Bayes

Les résultats de kmeans, qui sont des documents étiquetés sont utilisés par notre modèle. L'ensemble de données total a été divisé en deux sous-ensembles. Un sous-ensemble est utilisé pour l'entrainement de l'algorithme Naïve bayes et le reste est utilisé pour le tester. Par défaut 70% de la totalité des documents est utilisé pour l'entrainement et les 30% sont utilisées pour le test.

Nous avons utilisé la fonction **sample** du langage **R,** pour échantillonner notre ensemble.

Des documents d'entrainement: 122

Documents de test: 51

2.8.1-Apprentissage

✓ Entrainer le classifieur NB sur les documents d'apprentissage, en calculant les probabilités à priori des deux classes et les vraisemblances de tous les termes du vocabulaire relatives à ces classes.

Classes	Classe 1	Classe2
probabilités à priori	0.85	0.15

Table 5 : *Les probabilités à priori.*

2.8.2-Test

✓ Tester le classifieur NB en utilisant les documents du test, en calculant les probabilités à posteriori d'appartenance des documents test aux différentes classes.

✓ Classer les documents dans les classes qui disposent des plus grandes probabilités à posteriori

2.9-Validation des résultats de Kmeans

✓ Générer les matrices de contingence correspondantes & Calculer les mesures de performances Précision/Rappel/F-measure :

Matrice du corpus		Réel	
		OUI	NON
Classifieur	OUI	32	2
	NON	12	6

Table 6 : *Matrice de contingence globale de tout le corpus.*

P=32/32+2 R=32/32+12 F_1=2*0.94*0.72/0.94+0.72= 0,8154

Donc la F-measure du ***classifieur_medline*** *est*

$$F_1=81.54\%$$

2.10-Conclusion

Après expérimentation de notre classifieur, nous estimons que les résultats obtenus sont très satisfaisants, bien que des améliorations peuvent être apportées.

Les perspectives de notre travail sont détaillées dans la conclusion générale.

Conclusion générale et perspectives

1-Conclusion Générale

La classification de textes, devient de plus en plus necessaire pour le traitement des grandes bases de textes.

L'introduction des techniques d'apprentissage automatique a amélioré significativement le taux de bonne classification.

La combinaison entre les deux approches d'apprentissage, supervisé et non supervisé, réalisée, dans notre projet, a donné des résultats très encourageants.

L'algorithme Kmeans, utilisé pour le regroupement de notre corpus, a prouvé sa capacité de traitement de grands volumes de données, en un temps d'exécution assez rapide.

L'algorithme Naive bayes, utilisé pour la classification de nouveaux documents, a donné de bons résultats et a confirmé sa capacité d'apprentissage, même avec un petit corpus d'entrainement.

2-Perspectives

Nous citons quelques perspectives de notre travail préliminaire :

- Clusterisation de tout le corpus MEDLINE, pour avoir un catalogue complet de la base, facilitant son exploitation. Néanmoins le traitement d'un tel espace vectoriel demanderai beaucoup de mémoire et de temps de calcul; d'où la nécessité de recours aux grilles de calcul.
- Etudier la valeur du MESH dans la classification.
- Utilisation du thésaurus UMLS[1] qui est une base de vocabulaire biomédicale contrôlée, comme solution de réduction la taille du vocabulaire.

Enfin, ce projet était l'occasion de mettre en application l'ensemble des acquis de ma formation de Master « Ingénierie des logiciels», et m'a permis d'acquérir de nouvelles compétences sur le plan organisationnel.

[1] Voir http://www.nlm.nih.gov/research/umls/

Annexe

Kmeans : Comment fixer le K ?

Une des limites de Kmeans, est le choix du nombre ***k*** de classes à obtenir. Si la configuration de la population se prête parfaitement à un autre nombre de clusters, c'est dommage ! C'est pourquoi les kmeans sont souvent précédées d'une ACP qui permet de visualiser la structure d'une population et, parfois, de faire apparaître un certain nombre de nuages plus ou moins formés. Une autre solution consiste à effectuer d'abord une CAH, aussi la possibilité d'utiliser le graphe de coude. Ces trois techniques seront détaillées dans ce qui suit.

1-Analyse en composantes principales (ACP)

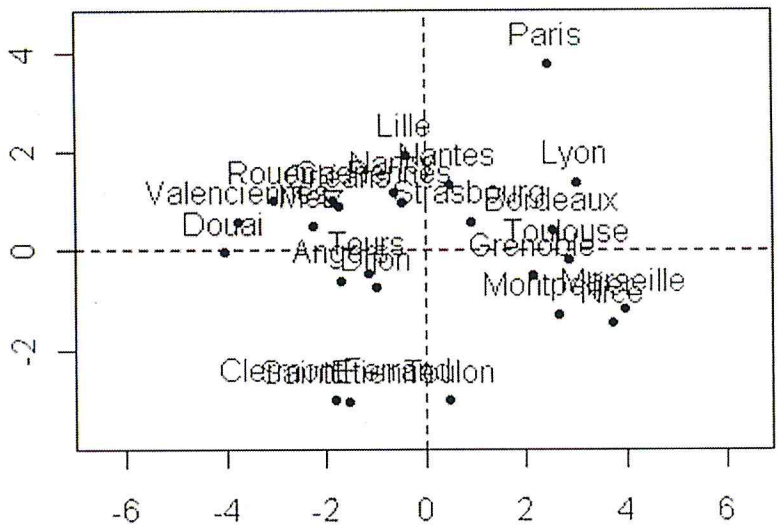
Présentation de l'ACP : [Zoo, 2004]

Voici une première manière de voir l'ACP. On dispose d'un nuage de points, dans un espace de dimension élevée, dans le quel on ne voit pas grand-chose. L'ACP va nous donner un sous-espace de dimension raisonnable, tel que la projection sur ce sous-espace retienne le plus d'information possible, i.e., tel que le nuage de points projecté soit le plus dispersé possible. Cela permet de réduire la dimension du nuage de points.

L'algorithme est le suivant. On commence par translater les données pour que le point moyen du nuage de points soit à l'origine (afin de pouvoir utiliser de l'algèbre linéaire). Ensuite, on essaye d'effectuer une rotation pour que l'écart-type de la première coordonnée soit le plus grand possible : cela revient à diagonaliser la matrice des covariances (c'est une matrice symétrique réelle (elle est même positive), elle est donc diagonalisable dans une base orthonormale), en commençant par les vecteurs propres les plus grands Le premier axe des nouvelles coordonnées correspond à une approximation du nuage de point par un sous-espace de dimension un ; si on veut une approximation par un sous-espace de dimension k, on prend les k premiers vecteurs propres.

Pour choisir la dimension de ce sous-espace, on regarde (graphiquement) les valeurs propres, et on s'arrête quand elles se mettent à décroitre rapidement (si elles décroissent très lentement, on est mal).

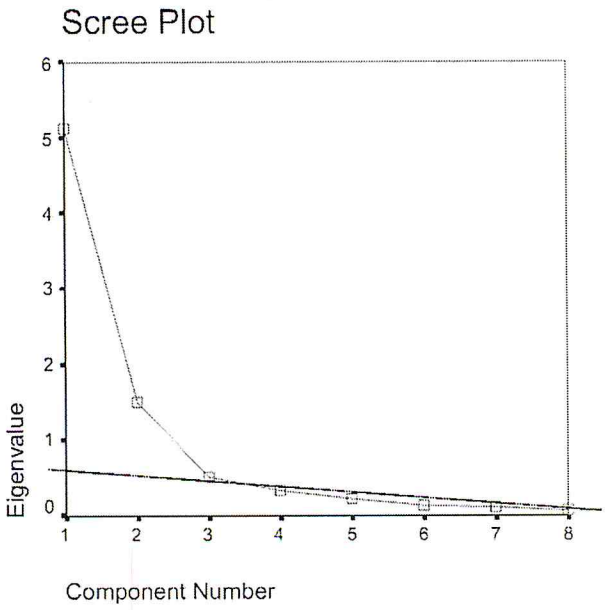
En résumé, L'ACP permet une présentation réduite du nuage des individus, ce qui donne une idée sur la dispersion de groupe d'individus **K**, qui sert d'entré à l'algorithme Kmeans.



2-Graphe de test de coude

Principe de test de coude (Scree-test en anglais)

On observe le graphique des valeurs propres et on ne retient que les valeurs qui se trouvent à gauche du point d'inflexion. Graphiquement, on part des composants qui apportent le moins d'information (qui se trouvent à droite), on relie par une droite les points presque alignés et on ne retient que les axes qui sont au dessus de cette ligne.



3-CAH : Classification Ascendante Hiérarchique

La classification ascendante hiérarchique (CAH) est une méthode de classification itérative dont le principe est simple.

1. On commence par calculer la dissimilarité entre les N objets.
2. Puis on regroupe les deux objets dont le regroupement minimise un critère d'agrégation donné, créant ainsi une classe comprenant ces deux objets.
3. On calcule ensuite la dissimilarité entre cette classe et les N-2 autres objets en utilisant le critère d'agrégation. Puis on regroupe les deux objets ou classes d'objets dont le regroupement minimise le critère d'agrégation.

On continue ainsi jusqu'à ce que tous les objets soient regroupés.

Ces regroupements successifs produisent un arbre binaire de classification (dendrogramme), dont la racine correspond à la classe regroupant l'ensemble des individus. Ce dendrogramme représente une hiérarchie de partitions. On peut alors choisir une partition en tronquant l'arbre à un niveau donné, le niveau dépendant soit des contraintes de l'utilisateur (l'utilisateur sait combien de classes il veut obtenir), soit de critères plus objectifs.

Avantages de la classification ascendante hiérarchique

La classification ascendante hiérarchique (CAH) est une méthode de classification qui présente les avantages suivants :

- On travaille à partir des dissimilarités entre les objets que l'on veut regrouper. On peut donc choisir un type de dissimilarité adapté au sujet étudié et à la nature des données.
- L'un des résultats est le dendrogramme, qui permet de visualiser le regroupement progressif des données. On peut alors se **faire une idée d'un nombre adéquat de classes dans lesquelles les données peuvent être regroupées**.

Méthode d'aggrégation pour la Classification Ascendante Hiérarchique

Plusieurs méthodes d'agrégation sont disponibles :

- Méthode de Ward (iniertie)
- Méthode de Ward (variance)
- Lien complet
- Lien simple
- Lien fort
- Lien flexible
- Lien moyen
- Lien proportionnel

Bibliographie

[Che, 2001]: Hsinchun Chen «Knowledge management systems: a text mining perspective».
[Che, 2004]: Hacene Cherfi «Etude et réalisation d'un système d'extraction de connaissances à partir de textes».

[Cri. 1999]: CRISP DM «*Cross Industry Standard Process for Data Mining*».

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

[Dom & Paz, 1997]: P.Domingos & M.Pazzani «Beyond Independence Conditions for the Optimality of the Simple Bayesian Classifier ».

[Dum & al, 1998]: S.Dumais, J.Platt, D.Heckerman, M.Sahami « Inductive learning algorithms and representations for text categorization ».

[Fel & al, 1998]: R. Feldman, M. Fresko, Y. K Kinar, Y Lindell, O. Liphstar, M. Rajman, Y. Scheler, O. Zamir « Text Mining at the Term Level ».

[For, 1965]: Edward W. Forgy. (1965). « Cluster Analysis of Multivariate Data: Efficiency Versus Interpretability of Classification ».

[Han & Kam, 2001]: *Jiawei Han and Micheline Kamber « Data Mining: Concepts and Techniques ».*

[Har, 1975]: J.A Hartigan. « Clustering algorithms».

[Hea, 2003]: Marti Hearst «What Is Text Mining? ».

[Hil, 2009]: H.Hilali. « Application de la classification textuelle pour l'extraction des règles d'association maximales ».

[Iwa, 1995]: M .IWAYAMA «Cluster-based text categorization: a comparison of category search strategy».

[Joa 1998b, Seb 2002, Yan 1999a]: T. Joachims «Making Large-Scale SVM Learning Practical».

[Joa, 1998]: T. Joachims «Text categorization with *support vector machines*».

[Lew, 2004]: D.D.Lewis « Bayesian Text Classification for Spam Filtering ».

[Mac, 1967]: J. B. MacQueen « Some Methods for classification and Analysis of Multivariate Observations, *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability* ».

[Man, 1999]: Manuel DE LAVALLEE, «La méthode OMT de Rumbaugh et l'évolution vers UML. Etude critique et comparative ».

[Mat, 2011]: Hocine Matallah « Classification Automatique de Textes Approche Orientée Agent ».

[Mou, 1996]: I.Moulinier « Une approche de la catégorisation de textes par l'apprentissage symbolique ».

[Ngo & May, 2005]: Eric Ngouana & Serge Mayaya «CLASSIFICATION BAYESIENNE NAÏVE DE TEXTES ».

[Pal & Jain, 2005]: Nikhil R.Pal and Lakhil Jain « Advanced Techniques in Knowledge Discovery and Data Mining **».**

[Rad, 2003]: RADWAN JALAM : « Apprentissage automatique et catégorisation de textes multilingues ».

[Rij, 1979]: C. J. van. Rijsbergen « Information retrieval. Butterworths ».

[Sal, 1983]: SALTON G,McGILL M.J «Introduction to modern information retrieval».

[Sal, 1989]: SALTON G,McGILL M.J« Automatic text processing: the transformation, analysis and retrieval of information by computer ».

[Seb, 1999]: F. Sebastiani «A Tutorial on Automated Text Categorization».

[Seb, 2002]: F.Sebastiani « Machine learning in automated text categorization ».

[Sim, 2005]: SIMON RÉHEL : « Catégorisation automatique de textes et cooccurrence de mots provenant de documents non étiquetés ».

[Zoo, 2004] : Vinc ent Zoonekynd « Méthodes factorielles : autour de l'Analyse en Composantes Principales (ACP) ».

Blogs :

[Fau, 2007] : C. Faure. « Introduction au text-mining».

[Gar, 2006] :G.Gardarain : «Notes de cours sur le Text Mining».

Livre

« TEXT MINING HAND BOOK, Advanced Approaches in Analyzing Instructed Data». Ronen Feldman and James Sanger 2007.