REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE ABOUBEKR BELKAID-TLEMCEN
FACULTE DES SCIENCES
DEPARTEMENT D'INFORMATIQUE

Mémoire présenté pour l'obtention du diplôme de

Magister

en

INFORMATIQUE

Option : Intelligence Artificielle et Aide à la Décision

Classification Automatique de Textes Approche Orientée Agent

Présenté par

MATALLAH Hocine

Soutenu en Février 2011 devant la commission du jury composée de :

Président	M^{r} M. BOUCHEKIF	PROFESSEUR, UNIVERSITE DE TLEMCEN
Directeur de thèse	M^{r} M.A. CHIKH	MAITRE DE CONFERENCES A, UNIVERSITE DE TLEMCEN
Examinateur	M^{me} F. DIDI	MAITRE DE CONFERENCES A, UNIVERSITE DE TLEMCEN
Invité	M^{r} M.A. ABDERAHIM	MAITRE DE CONFERENCES B, UNIVERSITE DE TLEMCEN

Résumé/ ملخـــص/ Abstract

Avec l'avènement de l'informatique et l'accroissement du nombre de documents électroniques stockés sur les divers supports électroniques et sur le Web, particulièrement les données textuelles, le développement d'outils d'analyse et de traitement automatique des textes, notamment la classification automatique de textes, est devenu indispensable, pour assister les utilisateurs, de ces collections de documents, à explorer et à répertorier toutes ces immenses banques de données textuelles.
Ainsi la catégorisation automatique de textes, qui consiste à assigner un document à une ou plusieurs catégories, s'impose de plus en plus comme une technologie clé dans la gestion de l'intelligence, les résultats obtenus sont utiles aussi bien pour la recherche d'information que pour l'extraction de connaissance soit sur internet (moteurs de recherche), qu'au sein des entreprises (classement de documents internes, dépêches d'agences, etc.).
À l'égard des différentes approches de classification automatique de textes, décrites dans l'état de l'art, se reposant sur une architecture classique basée sur un seul point de vue, nous avons introduit une nouvelle utilisation du classifieur « Naïve Bayes » avec des textes codés en «N-grammes », basée sur une architecture Multi-Agent.
L'objectif principal de nos travaux, est d'améliorer les performances et l'efficacité du modèle de classification.
Le corpus de référence Reuters, va servir à mener une étude comparative des résultats obtenus.

Mots Clés : *Catégorisation, Classification, Texte, Apprentissage, Evaluation, N-grammes, Naïve Bayes, SMA, Reuters.*

مع تطور المعلوماتية و تزايد الوثائق الإلكترونية و بالأخص منها النصوص، تأتي أدوات التحليل و المعالجة الأوتوماتيكية لهذه البيانات النصية كضرورة حتمية لمساعدة المستعملين لاستكشاف و فهرسة هذه القواعد النصية الضخمة.

و في هذا الإطار، أصبح التصنيف الأوتوماتيكي للنصوص وسيلة من الوسائل التكنولوجي ة الرئيسية لإدارة هذا النوع من الإشكاليات، و النتائج المتحصل عليها مفيدة سواء في البحث عن المعلومات أو استخراج المعرفة إما على شبكة الإنترنت (محركات البحث) ، أوفي المؤسسات.

على غرار التقنيات المختلفة المستعملة في هذا المجال المبنية على نظرية الرأي الأحادي، قمنا بتصميم و انجاز طريقة جديدة في استعمال المصنف الاحتمالي « *Naïve Bayes* » بتقنية « *N-Grams* » لتصنيف النصوص، مرتكزة على نظام الآراء المختلفة و المنسجمة « *SMA* ». الهدف المرجو و المسطر من هذا الإنجـــاز هو تطوير إمكانيات المصنف و تحسين النتائج المتحصل عليها.من أجل مقارنة النتائج، سنستعمل مرجع النصوص النموذجي المعروف «*Reuters*».

الكلمات الرئيسية : التصنيف، النص، التعلم، التقييم، رويترز.

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

Keywords : *Categorization, Classification, Text, Learning, Evaluation, N-gram, Naive Bayes, SMA, Reuters.*

A mon père et mon frère Abdelhadi

Que Dieu les accueille dans son vaste paradis

A ma chère mère

A ma chère femme

A mes enfants

A mes sœurs et mon frère

A toutes les personnes qui m'aiment

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

Remerciements

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

Entreprendre une thèse en informatique après 18 ans de rupture avec l'univers d'études et de recherche, c'était plus qu'un défi pour moi. Les premiers mois de la thèse étaient extrêmement ardus et réussir à surpasser l'épreuve c'était un vrai challenge que j'ai entamé.

Comme toute thèse, ce mémoire est le fruit de longues heures de lecture, de recherches, de réflexion et le résultat d'un effort constant, cet effort n'aurait pu aboutir sans la contribution d'un nombre de personnes que je tiens à remercier.

Tout d'abord, j'exprime ma double gratitude à Mr CHIKH Mohamed Amine, en tant que directeur de thèse pour ses conseils et orientations en dépit d'un emploi du temps chargé, et en tant que responsable de ce magister pour tous les efforts fournis durant cette année théorique qui était pleine et d'une extrême richesse d'enseignements qui m'a permis personnellement de me recycler et d'actualiser mes connaissances. Ce magister qui a été pour moi la clé et la chance de me relancer dans une deuxième carrière.

Je suis très reconnaissant à Dr BOUCHEKIF Mohamed de me faire l'honneur de présider le jury et mes plus sincères remerciements à l'égard de Mme DIDI Fedoua, et Mr ABDERAHIM Mohamed Amine de me faire l'honneur de juger mon travail.

Un grand merci aux Ingénieurs d'Etats en Informatique (Promo 2010) Semmoud A., Bouhassoune L., Mogtit S., qui m'ont accompagné dans la phase d'implémentation, leur aide a été extrêmement précieuse.

Je tiens ensuite à remercier Mr Hadjila F., Pour avoir contribué à la réflexion lors de l'élaboration du mémoire.

Je remercie chaleureusement toute l'équipe du département d'informatique et à tous ceux qui m'ont était une source d'aide ou de motivation importante, en particulier je cite parmi eux Mrs Benamar A., El Yebdri Z., Belabed A., Bentaallah M.A., etc..

Il serait trop long de tous les nommer mais je remercie chaleureusement tous mes proches, amis, et collègues de travail qui m'ont toujours soutenu et encouragé au cours de la réalisation de ce mémoire. Je pense particulièrement à ceux de la Wilaya de Tlemcen, de la Soitex et mon club de Foot-Ball C.S.R que je préside.

Enfin, ce travail de recherche n'aurait jamais été terminé sans le soutien de plusieurs personnes qui n'ont pas hésité à me donner le courage et le dynamisme pour l'accomplir. Qu'ils trouvent ici l'expression de mes sincères remerciements.

Merci à tous et à toutes.

Table des matières

Introduction

Chapitre 1 - Classification automatique de textes

Chapitre 2 - Codage des textes : Etat de l'art

Chapitre 3 - Approches de classification : Etat de l'art

Chapitre 4 - Evaluation des classifieurs

Chapitre 5 - Les Systèmes Multi-Agents

Chapitre 6 - Classification Automatique des textes Approche Orientée Agent

Figures

Tableaux

Introduction

Table des matières

1- Problématique et contexte du mémoire

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

L'information textuelle qui prend de plus en plus d'importance dans l'activité quotidienne des chercheurs et des entreprises ainsi que les besoins d'accès intelligents aux immenses bases de données textuelles et leurs manipulations qui ont augmenté très largement, d'une part.
D'autre part les limites d'une approche manuelle qui est coûteuse en temps de travail, peu générique, et relativement peu efficace, ont motivé la recherche dans ce domaine.
Ainsi la recherche des solutions opérationnelles, et la mise en œuvre d'outils efficaces pour automatiser la classification de ces documents devient une nécessité absolue. De nombreux travaux de recherche se focalisent sur cet aspect donnant ainsi un nouvel élan à la recherche dans le domaine qui connait une évolution réelle depuis les deux dernières décennies.
Comment partitionner cette masse d'information en groupes ou classes pour dégager des ressemblances par thèmes, par auteurs, par langue, ou par d'autres critères de classification ou carrément un filtrage de l'ensemble de documents utiles parmi les documents inutiles (Cas des filtres anti-spams). C'est à ce niveau que se positionne notre problématique de classification de textes.
L'objectif de la classification de textes est de rassembler les textes similaires selon un certain critère, au sein d'une même classe.
Deux types d'approches de classification automatique peuvent être distingués :
La classification supervisée et la classification non supervisée. Ces deux méthodes diffèrent sur la façon dont les classes sont générées. En effet dans le cas de la classification non supervisée, les classes sont calculées automatiquement par la machine, par contre, dans l'approche supervisée, la classification de textes consiste à rattacher un texte à une ou plusieurs catégories prédéfinies par un expert, ces catégories pouvant être par exemple le sujet du texte, son thème, l'opinion qui y est exprimée, etc... Nous disposons pour cela d'un ensemble de textes pour lesquels la catégorie est connue (corpus d'apprentissage) et qui nous servent à entraîner nos modèles, modèles qui seront testés et évalués sur d'autres documents pour lesquels la catégorie est connue également (corpus de test), le meilleur de ces modèles sera adopté par la suite pour étiqueter automatiquement des nouveaux documents de catégorie indéterminée.

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

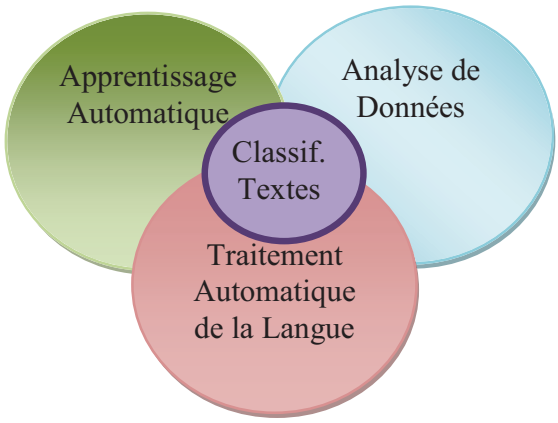


Figure 1 : Position de notre problème

2- Contribution

L'étude que nous avons mené sur les différentes approches de classification de textes conçues par un programme représentant l'expert qui est capable de résoudre le problème par lui-même ou concevoir des programmes comme des sortes de penseurs repliés sur eux-mêmes a trouvé sa limitation lorsque nous avons cherché à développer des modèles plus complexes de classification de bases de données textuelles gigantesques réalisées habituellement non pas par une seule personne mais par un groupe de personnes parfois délocalisées.
Ces limitations peuvent être ressenties facilement par une dégradation considérable des performances des meilleurs classifieurs, en temps de réponse qui augmente proportionnellement avec la taille des volumes traités et même pour la qualité des résultats.
Nous sommes ainsi naturellement conduits à chercher de meilleures performances en décentralisant le processus de classification et à donner plus d'autonomie et d'initiative aux différents modules logiciels spécialisés dans la classification et qui peuvent dialoguer pour partager leurs connaissances comme des experts humains. Le concept de système multi-agent propose un cadre de réponse à ces deux enjeux complémentaires (et à première vue contradictoires) : Autonomie et Organisation. Dans notre mémoire de magistère intitulé : « **Classification Automatique de Textes - Approche Orientée Agent** », nous essayons de réaliser un couplage entre deux grands domaines dans le monde d'informatique qui sont la classification automatique de textes et le paradigme agent.
L'optique de ce couplage est d'améliorer les performances et l'efficacité des systèmes de classification de textes.
La particularité de notre approche est le développement et la définition d'un modèle fondée sur une architecture composée de parties distinctes, chaque partie va traiter le problème de catégorisation d'une base de documents d'une façon spécifique, mais pouvant communiquer pour partager leurs connaissances.
Le paradigme Multi Agent serait ici appliqué dans plusieurs contextes :
Chaque agent dans le système fera sa propre classification pour le même texte (Autonomie).
La décision finale sera prise après un vote majoritaire (Collaboration)

Qui veut dire que le texte sera catégorisé dans la classe qui a été nommé par le plus grand nombre d'agents.
Pour l'évaluation de nos propositions, selon l'opinion qui y est exprimée, nous avons utilisé le corpus Reuters 21578 Top10 composé des dix catégories les plus importantes.

3- Organisation du mémoire

Ce mémoire va être organisé de la façon suivante : Un premier chapitre préliminaire pour définir l'ensemble des concepts de base du contexte étudié. Un état de l'art va être étalé au cours des chapitres 2, 3 et 4 des techniques employées dans les différentes phases du processus de classification automatique de textes, le cinquième chapitre fera l'objet d'une présentation générale de l'univers des systèmes multi-agents, alors que le dernier chapitre est consacré à motiver toutes les options entreprises ainsi que nos contributions méthodologiques et les résultats expérimentaux.

- Dans ***le premier chapitre***, nous présentons un aperçu sur l'histoire de la discipline, ensuite nous définissons la classification et les différents jeux de mots utilisés : classification, catégorisation ou clustering. Les différents objectifs et intérêts et attendus de la classification ainsi que les confluents avec d'autres disciplines comme le Text Mining ou Recherche d'Informations seront exposés par la suite puis nous décrivons le processus général de la catégorisation de textes avec toutes ces étapes, pour en finir avec les problèmes spécifiques aux textes lors de l'apprentissage automatique.

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

- Dans ***le troisième chapitre*** nous nous contentons d'exposer les méthodes de classification les plus utilisées dans la littérature en insistant sur les caractéristiques, les avantages et les limites de chaque méthode, mais avant ceux-ci nous allons introduire la matière en positionnant notre problème dans un cadre d'apprentissage supervisé pour que le choix de la méthode soit adéquat. Quelques mesures de similarité et formules pour calcul de distance, utilisées dans quelques techniques de classification, seront exposées en fin de chapitre.

- ***Le quatrième chapitre*** va identifier les différents indicateurs employés pour évaluer et mesurer les performances des classifieurs pour pouvoir les comparer ultérieurement.

- ***Le cinquième chapitre*** est consacré aux SMA, en commençant par répondre à la question pourquoi distribuer l'intelligence? On définira par la suite les différents concepts de base se rapportant au paradigme agent avant de présenter les différents modèles d'agents, l'apprentissage des agents et des SMA pour en conclure par les méthodologies de conception d'un SMA.

- ***Le sixième chapitre*** va synthétiser tous nos efforts fournis dans cette problématique de recherche. On l'entamera par une description générale de notre approche, puis on enchainera par les motivations et les justifications de toutes les solutions adoptées pendant tout le processus, en commençant par les n-grammes pour représenter nos textes puis Naive Bayes pour entrainer notre modèle et classer de nouveaux textes, ensuite la précision, le rappel et la F-Measure pour évaluer

notre classifieur construit et enfin les SMA pour une distribution de la classification. Cette dernière alternative orientée agent est considérée comme la marque principale de l'approche proposée. Une description de la base de texte utilisée pour l'évaluation à savoir Reuters va être donnée par la suite. Et enfin de manière à montrer les améliorations apportées par notre approche, nous décrivons les expérimentations réalisées sur des données issues de dépêches de presse Reuters-Top10, et des comparaisons vont être faites pour comparer les performances des approches classiques et distribuées.

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

Chapitre 1

Classification automatique de textes

Table des matières

1.1- Introduction

La classification automatique de textes consiste à regrouper de manière automatisée des documents qui se ressemblent suivant certains critères à savoir les critères observables tels que le type du document, l'année, la discipline, l'édition, etc... ou le critère du contenu.

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

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

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

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

Le domaine du traitement intelligent de données textuelles regroupe tout les outils et méthodes capables d'extraire des informations de textes écrits dans une langue naturelle.

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

1- Les approches issues de l'analyse de données et de la statistique étudient et chercher surtout à proposer des dispositifs aux statisticiens et aux linguistes pour leur permettre d'analyser les grandes bases de données textuelles en fournissant des informations synthétiques sur les corpus. Les logiciels d'analyse de corpus qui fournissent des listes de fréquence de mots et les représentations graphiques issues de l'analyse factorielle des correspondances font partie de cette catégorie.

2- Les approches qui proposent des systèmes de type « boite noire », ces méthodes traitent les documents de façon automatique sans intervention humaine. Elles réalisent souvent des fonctions de bas niveau : analyse lexicale, analyse syntaxique de surface, recherche d'information par mots-clés. Les moteurs de recherche popularisés avec le réseau Internet présentent un exemple typique des applications qui s'appuient sur cette approche.

Dans ce chapitre préliminaire, nous allons entreprendre notre sujet en répondant à la question pourquoi automatiser la classification ? Puis par rappeler de l'ancienneté de cette discipline, ensuite définir la classification et les différents jeux de mots utilisés dans la discipline, ensuite éclaircir la notion de classe et la notion de catégorisation déterministe et floue. Les différents objectifs et intérêts et attendus de la discipline ainsi que les confluents avec d'autres disciplines comme le Text Mining ou Recherche d'Informations seront exposés par la suite. Nous finirons par développer la démarche classique d'un système de classification automatique de textes de la représentation des documents jusqu'aux évaluations des résultats

ainsi que les différentes contraintes qui s'opposent au processus qui sont soit liées à la nature des données traitées (textuelles) soit au corpus lui-même soit aux techniques de représentation ou même le type de classifieur.

1.2- Pourquoi automatiser la classification ?

On assiste aujourd'hui à un accroissement de la quantité d'information textuelle disponible et accessible d'une manière exponentielle. D'après les derniers chiffres, on parle de plus de 200 millions de serveurs hôtes sur Internet et plus de 3 milliards de pages, la taille des corpus tests utilisés est passée de quelques mégaoctets à plusieurs Gigaoctets.
Dans les années 1996-1997, Reuters a produit un peu plus de 800 000 nouvelles en anglais par année. Si l'on ajoute aux articles écrits par les journalistes de l'agence ceux provenant d'autres sources, on arrive à un total de 5.5 millions de textes anglais par année à catégoriser. À un moment, l'organisation employait 90 personnes dédiées à l'étiquetage de ces documents. Il serait à coup sûr très intéressant de pouvoir déterminer avec précision le coût de classification. De combien de temps a besoin un humain pour associer un texte à une catégorie ? En pratique, il s'agit d'une question difficile à répondre. Assurément, plusieurs variables influencent le phénomène et les lignes qui suivent porteront sur certaines d'entre elles.
Certainement une grande partie du temps consommé pour classer un document est employé dans sa lecture, puis éventuellement à sa relecture. On peut aussi imaginer que la longueur des textes à classer est assez déterminante du temps qui va être requis pour cette opération, et sans doute, d'une personne à une autre, la vitesse de lecture varie. Une fois cette étape achevée, il faut trancher à quelle(s) catégorie(s) ce texte appartient. Au temps de réflexion exigé s'ajoute, certainement, le temps de se référer à la description des classes et éventuellement de consulter d'autres textes préalablement associés à certaines classes, pour valider la décision. D'autres facteurs interviennent également comme, comme par exemple le nombre de classes qui peut faire la différence : plus il y a de classes différentes, autrement dit plus il y a d'étiquettes possibles pour un texte donné, plus il est difficile de faire un choix parmi celles-ci. Aussi, plus la sémantique des catégories est précise, fine, détaillée, plus il faut faire attention avant d'y associer un document. À cet égard, classer des documents appartenant soit à la catégorie *«informatique»* soit à la catégorie *«mathématiques»* est vraisemblablement plus aisée que celle de classer des documents appartenant à l'une ou l'autre des catégories *«Intelligence artificielle»* , *«Génie logiciel»* et *«Système d'information»* .

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

- La réalisation manuelle de cette tâche par un expert est extrêmement coûteuse en terme de temps et personnel car il s'agit de lire attentivement chaque texte, au vu de la quantité phénoménale de textes aujourd'hui accessibles (par le biais du réseau Internet en particulier) (Moulinier, 1996), (Sebastiani, 2002)
- 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 (Moulinier 1996), (Sebastiani 2002)
- Cette opération peut être perçue comme subjective puisque basée sur l'interprétation du document, deux experts peuvent classer différemment un même document, ou encore un même expert peut classer différemment un même document soumis à deux instants différents (Clech & Zighed, 2004), (Clech, 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 plusieurs travaux de recherche se concentrent ces dernières années.

1.3- Historique de la Catégorisation de textes

C'est une discipline assez ancienne, en 1627, Gabriel Naudé propose un classement selon cinq grands thèmes : théologie, jurisprudence, histoire, sciences et arts, belles lettres. Le désir de maîtriser l'Univers se fait sentir dans la multiplication des encyclopédies. L'encyclopédie de Diderot (parue entre 1751 et 1772) est organisée selon l'ordre alphabétique avec des renvois associatifs alors que celle de Panckoucke (parue de 1776 à 1780) suit une organisation méthodique selon un ordre arborescent (Fayet & Scribe, 1997).
Le système de classification par thème, apparu dès les débuts de l'écriture et institutionnalisé à Alexandrie conduisit à la création par Dewey, en 1876, d'un système de classification « universel ». Il s'agit d'une classification documentaire de type encyclopédique.
Toutefois l'idée d'effectuer la classification de textes par des machines remonte au début des années 60 et qui a connu des progrès considérables à partir des années 90 avec l'apparition d'algorithmes beaucoup plus performants qu'auparavant.
Jusqu'au début des années 80, pour construire un classifieur, il fallait consacrer d'importantes ressources humaines à cette tâche. Plusieurs experts éditaient des règles manuellement puis les affinaient au fur et à mesure des tests. L'avènement des de l'AA s'est donc traduit par un gain de temps conséquent. Il n'est plus nécessaire par exemple de reconfigurer tout le système en cas de changement d'arborescence.
Ces évolutions technologiques et algorithmes avancées font aujourd'hui de la catégorisation un outil fiable.
Au début des années 90, les travaux proviennent essentiellement de la communauté de Recherche d'Information (RI). En effet, les méthodes de numérisation, les algorithmes de classification et les méthodologies de test ont été adaptés à la CT en particulier au cours des conférences TREC (**T**ext **RE**trieval **C**onference. http://trec.nist.gov).
La communauté d'Apprentissage Automatique (AA) s'est intéressée elle aussi à ce problème il y a une dizaine d'années en le considérant comme domaine d'application à ces algorithmes de reconnaissance des formes. Actuellement, les méthodes de numérisation de texte restent largement inspirées de la RI alors que les classifieurs les plus performants sont issus de l'AA.
Une autre communauté composée essentiellement de statisticiens et de linguistes, traite également le problème de la CT en s'appuyant sur les méthodes d'analyse de données. Le but ici n'est pas de créer un système qui classe automatiquement des documents sans intervention humaine mais d'extraire des informations synthétiques du corpus. Les problématiques traitées ici sont par exemple l'étude des genres littéraires ou la détermination de l'auteur d'un texte.

1.4- Les systèmes de classification et vocabulaire utilisé

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

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.
Les deux termes « Classification », « Catégorisation » ont des histoires et des origines très différentes. La confusion entre ces deux termes persiste depuis le temps, dans le langage

courant voire philosophique. La première définition de la classification apparaît pour la première fois dans la cinquième édition du dictionnaire de l'académie française en 1798 « Distribution en classes et suivant un certain ordre ». Le mot « Catégorisation » n'existait pas dans le dictionnaire français, contrairement au mot « catégorie », quoiqu'il puisse être défini comme étant l'action de créer des catégories ou le résultat de cette action. Ce terme vient du grec « katêgoria : qualité attribuée à un objet ». Les catégories sont définies par Aristote comme étant « les espèces les plus générales de ce qui est signifié par un mot simple ». Il rassemble dans un même groupe des éléments proches et recense dix catégories différemment à certains pythagoriciens qui voulaient opposer toutes les espèces deux à deux : masculin et féminin, pair et impair, fini et infini, statique et dynamique, etc.…
Comme tenu de l'historique de ces deux termes et leur contexte d'utilisation actuelle, nous allons essayer de distinguer entre les différentes variantes de classification de textes et le vocabulaire utilisé dans la section suivante.

1.4.1- Catégorisation (Supervisé)

Ainsi, la *catégorisation* de textes correspond à la procédure d'affectation d'une ou de plusieurs catégories ou classes prédéfinies à un texte. Elle correspond à la *classification supervisée* pour l'apprentissage automatique et à la *discrimination* en statistiques alors que la recherche d'informations utilise des termes plus proches de l'application concernée : *filtrage* ou *routage*.
Cette problématique a par ailleurs dernièrement trouvé de nouvelles applications dans les domaines du traitement du langage tels que : l'affectation de sujets en recherche d'information, l'aide de l'utilisateur pour l'indexation de documents (Hayes & Weinstein, 1990), la veille technologique, le filtrage personnalisé des documents intéressant un internaute connaissant ses préférences de sujets (catégories) (Lang, 1995), le routage de textes (tels que le courrier) et l'amélioration de la recherche sur le web (Armstrong & all, 1995), et enfin l'organisation des sources textuelles de plus en plus nombreuses, en particulier des pages web.
Aujourd'hui, cette problématique utilise largement des méthodes issues de l'apprentissage automatique et beaucoup d'algorithmes d'apprentissage supervisé lui ont été appliqués (Naïve bayes, K-plus proches voisins, arbres de décision, machines à vecteurs support, réseaux de neurones, etc...)

1.4.2- Clustering (Non supervisé)

Toutefois quand l'ensemble des catégories n'est pas donné au départ, et qu'il s'agit de le créer en regroupant les textes en classes qui possèdent un certain degré de cohérence interne, on est dans un contexte de *classification non supervisée* pour l'apprentissage automatique.
La *classification non supervisée* consiste à trouver de manière automatique une organisation cohérente à un groupe de documents homogènes pour construire des regroupements cohérents (des classes ou clusters), elle correspond en statistiques au clustering, qui est également le terme utilisé en recherche d'informations.
Le clustering consiste donc, à diviser les objets (dans notre cas des textes) en groupes sans connaître à priori leurs classes d'appartenance.
Les techniques pour réaliser de tels regroupements constituent un domaine d'étude très riche, qui a donné lieu à de multiples propositions dont le recensement n'est pas l'objet de ce document.

- ***Dans ce qui suit notre travail va être concentré sur la catégorisation de textes (la classification supervisée).***

1.5- Définition de la Catégorisation de textes

Dans sa forme la plus simple, la catégorisation de documents consiste à assigner à un texte une ou plusieurs étiquettes permettant d'indexer le document dans un ensemble prédéfini de catégories, Originellement conçue pour assister le classement documentaire d'ouvrages ou d'articles dans des domaines techniques ou scientifiques.
La Catégorisation de Textes (C.T) est le processus qui consiste à assigner une ou plusieurs catégories parmi une liste prédéfinie à un document. L'objectif du processus est d'être capable d'effectuer automatiquement les classes d'un ensemble de nouveaux textes.
La catégorisation de documents consiste à apprendre, à partir d'exemples caractérisant des classes thématiques, un ensemble de descripteurs discriminants pour permettre de ranger un document donné dans la (ou les) classe(s) correspondant à son contenu (Brown & Chong, 1998).
Principalement, les algorithmes de catégorisation s'appuient sur des méthodes d'apprentissage qui, à partir d'un corpus d'apprentissage, permettent de catégoriser de nouveaux textes. Ce type de méthodes sont dites inductives car elles induisent de la connaissance à partir des données en entrée (les textes) et des sorties (leurs classes).
Les divers travaux dans le domaine cherchent à trouver un algorithme permettant d'assigner un texte à une classe avec le plus grand taux de réussite possible sans toutefois assigner un texte à trop de classes. Dans un tel contexte, une mesure de similarité textuelle permet d'identifier la ou les catégories les plus proches du document à classer. Si cette notion de similarité sémantique est un processus souvent intuitif pour l'homme, elle résulte d'un processus complexe et encore mal compris du cerveau.
Le problème de la catégorisation peut se résumer en une formalisation de la notion de similarité textuelle, soit en d'autres termes à trouver un modèle mathématique capable de représenter la fonction de décision d'appartenance des textes aux catégories.
Nous considérons un ensemble de classes $C = \{ci\}$ et un ensemble de documents $D = \{dj\}$.
Un système de classification associe automatiquement à chaque document un ensemble de classes (0,1 ou plusieurs). Le problème de la classification a été formalisé de plusieurs manières, nous vous proposons la formalisation de Sebastiani (Sebastiani, 1999) reprise par Yang (Yang, 1999)
Deux fonctions sont définies :

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

La fonction de décision est une estimation de la fonction cible qu'on ignore. Plus cette estimation est correcte, plus le système de classification est performant.
La fonction de décision et la fonction cible attribuent à chaque couple $(dj, ci) \in D \times C$ une valeur booléenne pour indiquer si le document *dj* appartient ou non à la classe *ci* .

La fonction de décision sera définie de la manière suivante :
$\mathbf{D} : D \times C \rightarrow \{ vrai, faux \}$, $\mathbf{D}(d,c)$ = Vrai si *d* est associé à la classe *c* sinon $\mathbf{D}(d,c) = Faux$

La fonction cible sera définie de la manière suivante :
$\mathbf{C} : D \times C \rightarrow \{ vrai, faux \}$, $\mathbf{C}(d,c)$ = Vrai si *d* est associé à la classe *c* sinon $\mathbf{C}(d,c) = Faux$

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

1.6- La notion de classe pour les systèmes de classification

La notion de classe pour un système de classification a été habituellement synonyme de « thème ». Dans ce contexte, classer les documents revient à les organiser par différentes thématiques. (Par exemple : *Earn, Ship, Trade,* correspondent à des thèmes dans le corpus Reuters). Cependant, la problématique de classification a évolué en même temps que les besoins et elle s'intéresse aujourd'hui à différentes tâches pour lesquelles les catégories ne sont pas interprétables comme des thèmes : ainsi, par exemple, les tâches consistant à classer les documents par auteur, par genre, par style, par langue, ou encore selon que le document exprime un jugement positif ou négatif, etc.. Ainsi la classe va correspondre à un besoin d'information d'un utilisateur ou d'une société et n'est donc pas obligatoirement un thème unique. Nous considérerons dans la suite qu'une classe est simplement une étiquette à associer à des documents.
Dans la figure 1.1, un système de classification d'emails est représenté où les classes peuvent être de différentes natures (thèmes, messages provenant de certaines personnes, messages d'un certain type, etc…)

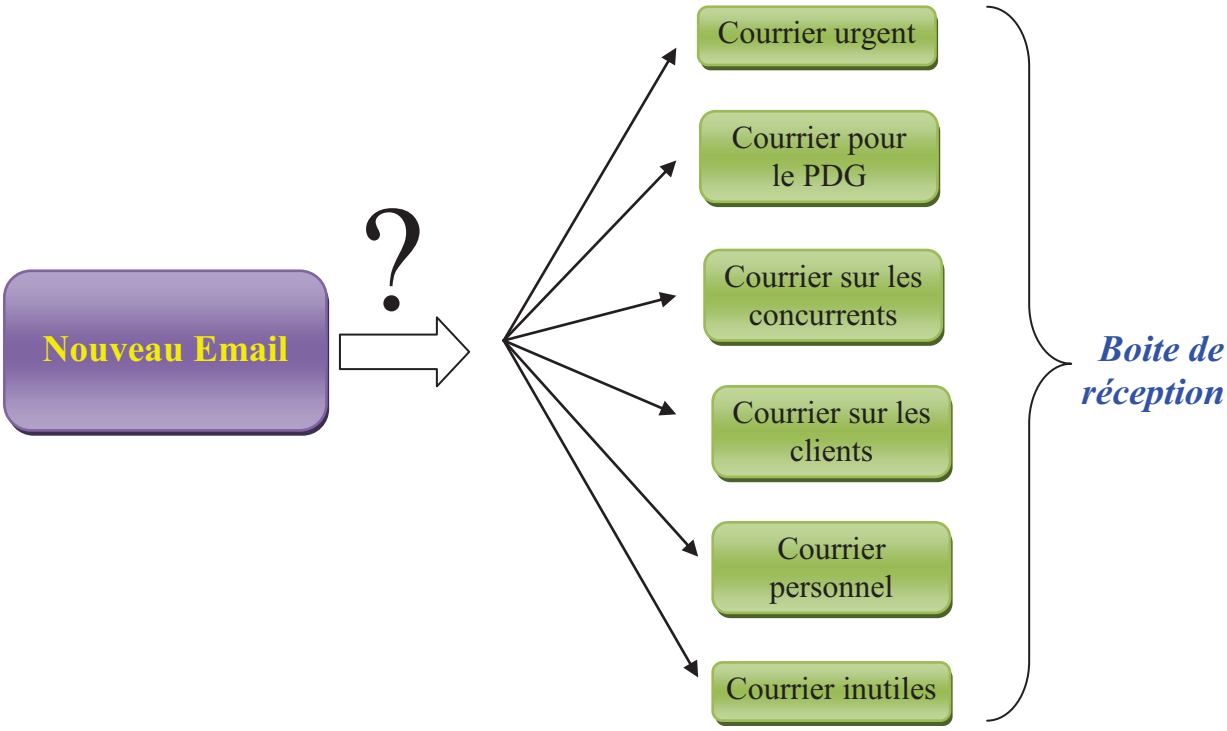


Figure 1.1 : Exemple de système de classification d'emails

Ce système organise des emails dans des boîtes aux lettres qui correspondent chacune à une classe qui sont de différentes natures (« mails urgents », « Mails du Directeur général »,etc…)

1.7- Les différents contextes de classification

Plusieurs contextes de classification se distinguent, ils influents directement sur les modèles utilisés. Ludovic DENOYER a bien résumé les différents contextes de classification dans (Denoyer, 2004) que nous avons reporté dans ce qui suit, les problématiques les plus récentes comme par exemple la classification dans une hiérarchie de classes (McCallum & all, 1998),(Koller & Sahami, 1997) ne sont pas abordés ici.

1.7.1- Classification bi-classe et multi-classes

1.7.1.1- La classification bi-classe

La classification bi-classe correspond au *filtrage*. C'est une problématique pour laquelle le système de classification répond à la question : *« Le texte appartient-il à la catégorie C ou non (i.e. ou à sa catégorie complémentaire ¬C ? »* (Par exemple, un document est il autorisé aux enfants ou non).

Cependant quand il s'agit d'effectuer une classification multi-classe qui permet de transmettre le document vers le ou les catégories(s) le(s) plus approprié(s), on parle alors de *routage. C*ette classification muti-classes, selon le cas, peut être disjointes ou non.

1.7.1.2- La classification multi-classes disjointes

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

1.7.1.3- La classification multi-classes

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

1.7.2- Catégorisation déterministe et floue

1.7.2.1- Catégorisation déterministe

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

1.7.2.2- Catégorisation floue ou le ranking

Contrairement aux cas précédents, on peut également souhaiter dans certains cas d'avoir simplement une évaluation des classes les plus adéquates -dans l'ordre- pour y classer le texte. Ce qu'on peut appeler par classification floue ou ranking.
Ce type de classification va permettre à l'utilisateur d'être plus indulgent si le texte est "proche" du thème que si le texte n'a absolument rien à voir avec celui-ci dans le cas ou ce dernier est incorrectement attribué à la classe.
Le ranking est une problématique de classification dans laquelle le système, au lieu d'associer un texte à une classe catégoriquement, il ordonne les classes par ordre de pertinence pour un texte donné.
Les méthodes qui évaluent une distance d'un texte à une catégorie permettent facilement ce type de classement de même pour les approches qui estiment des probabilités d'appartenance d'un texte à une classe.

Ludovic DENOYER dans (Denoyer, 2004) donne quelques exemples d'applications dans lesquelles ce système de classification est sollicité :

- Le ranking de pages Web pour une thématique définie par un internaute.
- Le filtrage avec un rajustement de seuil de tolérance, le seuil étant ajusté par rapport aux scores de ranking.
- Proposer à un utilisateur un classement d'experts compétents pour évaluer un projet.

Dans ce cas spécifique, une fonction de score est définie de la manière suivante :

$$\textbf{SCORE} : \quad D \times C \longrightarrow [0,1]$$

Cette fonction nous renseigne sur le degré d'appartenance d'un texte à une classe donnée. Ainsi, plus *SCORE*(*d*,*c*) est proche de 1, plus le document *d* est proche à être attribué à la classe *c* et inversement, plus cette valeur est proche de 0, plus le document est loin d'être attribué à la classe. Le calcul de cette fonction de score nous permet alors d'organiser les classes dans l'ordre pour y classer le texte et donc de savoir par exemple quelle est la classe la plus probable à être sélectionnée par rapport aux autres.
Pratiquement, tous les algorithmes de classification calculent un score entre un texte et une classe. C'est le cas de toutes les approches probabilistes, particulièrement le classifieur Naïve Bayes. Toutefois, ces systèmes peuvent être aussi utilisés pour la classification déterministe. Dans ce cas, il est fondamental d'adopter une stratégie transformant la fonction de score en une fonction de décision. Pour cela, la stratégie habituelle consiste à utiliser un seuil L_C tel que :

$$\begin{cases} \text{si SCORE } (d, c) > L_c \text{ alors } \mathbb{D}(d,c) = \text{vrai} \\ \text{sinon } \mathbb{D}(d,c) = \text{faux} \end{cases}$$

1.8- Objectifs et intérêts

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

Nous pouvons citer six applications typiques qui sont :

- Le classement automatique de différents communiqués de presse, ou messages sur des forums en différentes matières (« Les actualités de la région », « la bourse », « culture », etc..), (Exemple : Une boite propose un système de tri d'informations dans des flots de dépêches d'agence de presse AFP ou Reuters etc.. ou pages web. Chaque matin les nouvelles importantes sont faxées à différentes entreprises).
- Indexation automatique sur des catégories d'index de bibliothèques : aide à la classification thématique des différentes rédactions dans une bibliothèque.
- La gestion de bases documentaires (mémoire d'entreprise). Ce système peut être utilisé pour présenter l'information à l'utilisateur selon des catégories thématiques, ce qui facilite la navigation.
- Sauvegarde automatique de fichiers dans des répertoires.
- Les filtres internet en général, et en particulier les filtres anti-spams.
- Le classement automatique des emails, et particulièrement la redirection automatique de courriers des clients et fournisseurs en fonction de leur contenu vers les personnes compétentes dans une entreprise (Service commercial, livraison, service après vente, approvisionnements, etc..) ou vers des répertoires prédéfinis dans un outil de

messagerie, ou encore le tri de courriers électroniques dans différentes boîtes aux lettres personnelles et possibilité d'envoi de réponses automatiques.

1.9- Classification de textes et Text Mining

Le Text Mining est une technique permettant d'automatiser le traitement de gros volumes de contenus texte pour en extraire les principales tendances et répertorier de manière statistique les différents sujets évoqués ainsi découvrir des connaissances et des relations à partir des documents disponibles.
L'outil de Text Mining va générer de l'information sur le contenu du document. Cette information n'était pas présente, ou implicite, dans le document sous sa forme initiale, elle va être rajoutée, et donc enrichir le document.

Les besoins en Text Mining peuvent être :

- Recherche d'information
- Correction orthographique/grammaticale
- Traduction automatique
- Résumé automatique
- Question/réponse (interfaces en langage naturel)
- La veille technologique

Et notamment

- **La Classification automatique des documents**

Toutes ces applications sont étroitement liées.

1.10- Classification de textes et Recherche d'informations

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

- La recherche d'informations (RI), aussi appelée recherche documentaire (RD), est la problématique la plus ancienne de ce domaine, elle consiste à trouver, dans une importante base de documents, les documents pertinents correspondant à des requêtes qui peuvent être de différentes natures (liste de mots clefs, langage naturel, langage spécifique comme le SQL par exemple etc.).
La RI correspond à la tâche classique d'interrogation par des requêtes, aujourd'hui démocratisée par le Web avec des moteurs tels que Google ou Altavista ou encore la recherche informatisée de documents dans de sources bibliothécaires. Beaucoup de modèles ont été développés et continuent aujourd'hui de l'être, ces modèles peuvent être ensemblistes, algébriques, statistiques (Miller & all, 1999).
La recherche d'informations est généralement effectuée en indexant préalablement tous les documents de la base selon les mots qu'ils contiennent ; la recherche consiste à trouver, le plus rapidement possible, les documents ayant des mots communs avec la requête de l'utilisateur.

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

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

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

SALTON recommandait, à la fin des années 60, le regroupement des documents des corpus pour permettre une recherche d'information plus rapide en ne calculant plus les distances entre la requête et chaque document mais seulement entre la requête et chaque classe : « Clearly in practice it is not possible to match each analysed document with each analysed search request because the time consumed by such operation would be excessive » (Salton, 1968).
Une autre étude menée par (Bellot, 2002) montre que la classification automatique permet d'améliorer l'efficacité des systèmes de recherche. Un système de recherche documentaire, comme on a vu précédemment, donne, en réponse à une requête, une liste de documents. La liste des documents trouvés est souvent si longue que les utilisateurs ne peuvent l'examiner entièrement et laissent de côté certains documents pertinents mal classés. L'étude a démontré qu'une classification automatique des seuls documents retrouvés permet d'améliorer la qualité de la recherche documentaire.

1.11- Démarche à suivre pour la catégorisation de textes

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

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

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

La figure 1.2 illustre la démarche de catégorisation de textes avec ses trois étapes qui peuvent être schématisées comme suit :

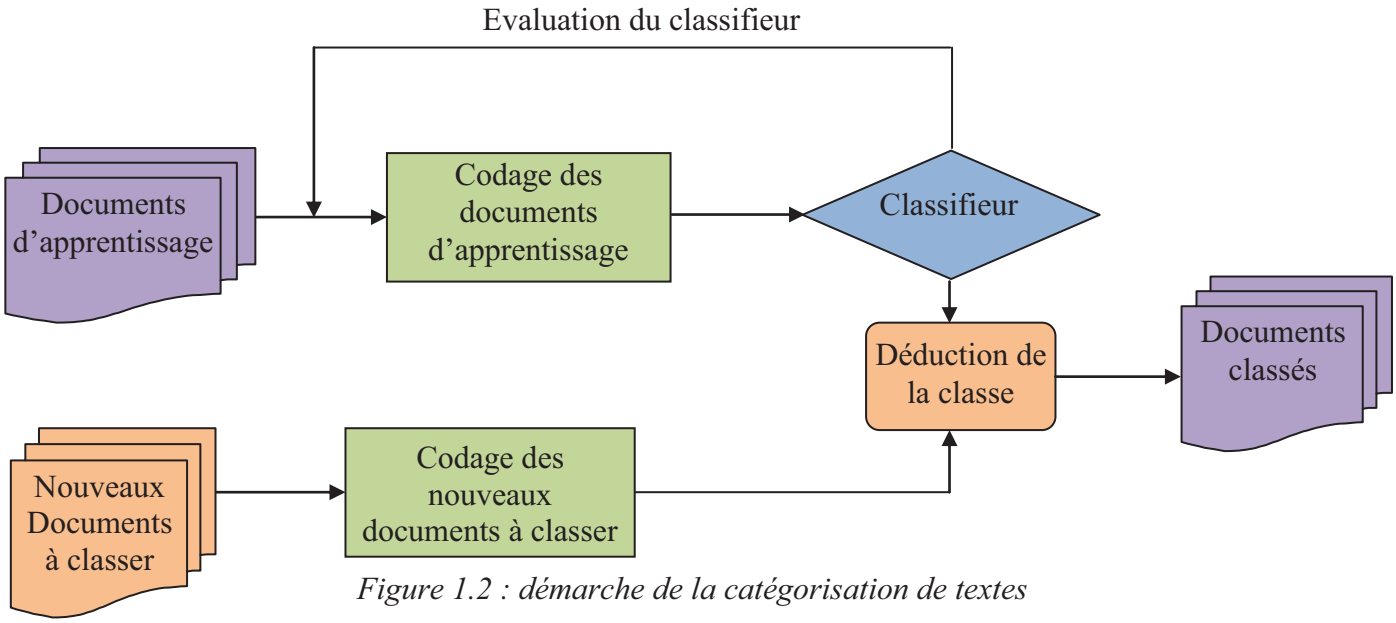


Figure 1.2 : démarche de la catégorisation de textes

Toutes ces étapes seront développées dans les chapitres 2,3 et 4.

1.12- Problèmes de la catégorisation de textes

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

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

1.12.1- Redondance(Synonymie)

La redondance et la synonymie permettent d'exprimer le même concept par des expressions différentes, plusieurs façons d'exprimer la même chose.
Cette difficulté est liée à la nature des documents traités exprimés en langage naturel contrairement aux données numériques. LE FEVRE illustre cette difficulté dans l'exemple du chat et l'oiseau : *mon chat mange un oiseau, mon gros matou croque un piaf et mon félin préféré dévore une petite bête à plumes* (Lefèvre, 2000).
La même idée est représentée de trois manières différentes, différents termes sont utilisés d'une expression à une autre mais en fin compte c'est bien le malheureux oiseau qui est dévoré par ce chat.
Lors d'une représentation vectorielle d'un document, ces termes sont représentés séparément, et les occurrences du concept sont dispersées. Il est alors important de rassembler ces termes en un groupe sémantique commun.
Pour y remédier, il est alors intéressant de concevoir une ontologie afin de cerner les sens des termes, naturellement, cela engendre des coûts supplémentaires pour sa réalisation et sa maintenance.

1.12.2- Polysémie (Ambiguïté)

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

1.12.3- L'homographie

Deux mots sont dits homographes si 'ils s'écrivent de la même façon sans forcément avoir la même prononciation. L'homographie est une sorte d'ambigüité supplémentaire. (Ex : avocat en tant que fruit et avocat en tant que juriste)
L'homographie et l'ambiguïté génère du bruit qui va causer une dégradation de précision (indicateur nécessaire pour mesurer la performance du classifieur). Il sera alors préférable d'ôter ces ambigüités.

1.12.4- La graphie

Un terme peut comporter des fautes d'orthographe ou de frappe comme il peut s'écrire de plusieurs manières ou s'écrire avec une majuscule. Ce qui va peser sur la qualité des résultats. Parce que si un terme est orthographié de deux manières dans le même document (Ghelizane, Relizane), la simple recherche de ce terme avec une seule forme graphique néglige la présence du même terme sous d'autres graphies, ce qui va influencer les résultats puisque les différentes graphies vont être traitées séparément. Néanmoins du point de vue pratique, le fait qu'un terme inconnu est proche d'un autre terme prouve qu'il a été mal orthographié.
Toujours dans ce contexte, (Loupy & El-Bèze, 2000) et (Loupy, 2000) affirment que la graphie peut donner une information relative au sens du terme employé. Prenons par exemple le cas pour le mot Histoire dont la majuscule indique qu'il s'agit de la discipline étudiant le passé et non d'un roman ou une blague. La prise en compte de toutes ces variations morphologiques pour la classification automatique de textes n'est pas étudiée dans ce mémoire.

1.12.5- Les variations morphologiques

Les conjugaisons, pluriels, influent négativement sur la qualité des résultats puisque les différentes variations morphologiques vont êtes considérées séparément et chacune va être prise comme un élément à part comme par exemple les trois termes : maître, maîtresse, maîtriser sont traités indépendamment quoique en réalité ça pivote sur la même idée. Pour y remédier soit on applique la lemmatisation ou le stemming, à notre texte soit carrément on opte pour une représentation en n-grammes qui peut nous éviter ces pré-traitements, tout ceux-ci va être étalé dans le chapitre 2.

1.12.6- Les mots composés

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

1.12.7- Présence-Absence de termes

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

1.12.8- Complexité de l'algorithme d'apprentissage

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

1.12.9- Sur-apprentissage

Le nombre de termes très important et très varié qui ne se répètent dans tous les textes va causer énormément de creux dans le tableau de grande dimension (textes*termes) qui peut provoquer du sur-apprentissage qui s'explique par le fait que le modèle n'arrive pas à bien classer les nouveaux textes, pourtant il l'a bien fait dans la phase d'apprentissage en classant correctement les textes de la base d'apprentissage.
Pour limiter le sur-apprentissage, on doit sélectionner des termes pour réduire la dimensionnalité. D'après les expériences antérieures, le nombre de termes doit être limité par rapport au nombre de textes de la base d'apprentissage.
Quelques auteurs recommandent d'utiliser au moins 50 à100 fois plus de textes que de termes.
En général le nombre de textes d'apprentissage est limité, c'est pour cela on cherche à agir sur le nombre des termes utilisés en les diminuant, pour éviter ce sur-apprentissage.
Sans bien sûr pénaliser le système en supprimant des termes pertinents (Sebastiani, 2002).

1.12.10- Subjectivité de la décision

Parmi les problèmes classiques usuels dans le domaine de l'apprentissage supervisé c'est la subjectivité de la décision prise par les experts qui décident de la classe à laquelle le texte va être attribué.
Certainement après la lecture du texte à classer, l'expert va trancher à quelle(s) catégorie(s) ce texte appartient en se basant sur le contenu sémantique et le contexte du texte et même en consultant d'autres textes préalablement associés à certaines classes, pour valider la décision prise qui ne peut être que subjective.

Les experts humains ne lisent pas de la même manière ! Ne réfléchissent pas de la même manière ! Donc ne classent pas de la même manière !
Ainsi un même document peut être classé différemment par deux experts, ou encore un même document peut être classé différemment par le même expert, soumis à deux instants différents (Clech & Zighed, 2004)
D'après les expériences : Lorsque deux experts humains doivent déterminer les classes d'une collection de textes, il y a souvent désaccord sur plus de 5 % des textes. Il est donc illusoire de rechercher une classification automatique parfaite.

1.13- Conclusion

La catégorisation de textes s'est avérée au cours des dernières années comme un domaine majeur de recherche pour les entreprises comme pour les particuliers.
Ce dynamisme est en partie dû à la demande importante des utilisateurs pour cette technologie. Elle devient de plus en plus indispensable dans de nombreuses situations où la quantité de documents textuels électroniques rend impossible tout traitement manuel.

La catégorisation de textes a essentiellement progressé ces dix dernières années grâce à l'introduction des techniques héritées de l'apprentissage automatique qui ont amélioré très significativement les taux de bonne classification.

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

Chapitre 2

Codage des textes : Etat de l'art

Table des matières

2.1- Introduction

L'explosion de la quantité d'informations textuelles provoquée par l'évolution à grande échelle des outils de communication essentiellement Internet qui est sorti de l'aspect réservé à un milieu restreint à un aspect de vulgarisation au grand public, a rapidement, fait sentir le besoin de recherche de mécanismes et outils de traitement automatique des quantités d'informations diffusées sur le Web.
Ainsi, avec les bases de données multimédia, les dépêches d'agences de presse, les publications scientifiques, les bibliothèques électroniques, etc... Qui sont consultés habituellement sur le réseau, on dispose de plus en plus de grandes masses de documents non ou faiblement structurées, en particulier les documents textuels qui sont considérés comme étant des documents non structurés, surnommés « documents plats » par quelques auteurs, c'est-à-dire comme une séquence ou un ensemble de mots sans informations complémentaires sur le document.
Différents formats HTML, SGML, XML, DOC, PDF, … peuvent être des moyens pour stocker et représenter ces documents.
Le manque de structure au sein de ces collections volumineuses rend difficile l'accès à l'information qu'elles contiennent, d'où la nécessité aujourd'hui, de chercher comment structurer automatiquement ces corpus pour les rendre utilisables d'une façon rapide et optimale pour y faciliter leurs traitements automatiques et notamment la classification.
Pour pouvoir y appliquer les différentes techniques et algorithmes d'apprentissage, une transformation de ces documents non ou peu structurés est indispensable.
La transformation ou le codage de ces documents est une préparation à « l'informatisation » de ces derniers, chaque type de documents comme les images, les vidéos et notamment les textes dispose de ses propres techniques de codage.
Plusieurs approches de représentation des documents textuels ont été proposées dans ce contexte, la plupart étant des méthodes vectorielles.
Les principales méthodes de représentation de textes n'utilisent pas d'information grammaticale ni d'analyse syntaxique des mots : seule la présence ou l'absence de certains mots est porteuse d'informations.

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

2.2- Le texte

La numérisation est une opération qui s'est élargie pour atteindre toutes formes de documents et notamment les textes, et ce dans le but de leur exploitation sur les réseaux. Cet élargissement a entraîné derrière lui beaucoup de travaux qui ont un rapport surtout avec le formatage et la normalisation des textes qui ont été développés pour être à la fois rapide et efficace suite au développement de l'Internet.
Depuis les débuts de la numérisation des données textuelles, le texte a été considéré, et c'est encore vrai aujourd'hui dans la plupart des cas, comme tout simplement une séquence de caractères. Ces caractères peuvent être représentés dans différents espaces de codage, le plus courant étant le codage ASCII admettant 256 caractères différents, mais en dépit ce codage ne prenait pas en charge les langues comme l'arabe ou le chinois. Afin de pouvoir représenter ces langues, différentes normes de codages sont crées et plus largement utilisées aujourd'hui comme la norme UNICODE qui permet la représentation de 65536 caractères.

Pour la plupart des langues (occidentales et orientales font partie), l'espace de codage au niveau caractère n'est pas un espace très informatif car un caractère seul ne présente pas une information sémantique riche. Un texte est plutôt considéré comme une séquence de mots (un

mot lui même étant une séquence de caractères) et représenté dans un espace de mots dont la dimension est plus grande que celle du caractère (Le nombre de caractères possibles est limité mais en revanche le nombre de mots qu'on peut avoir est énorme), mais dont chaque dimension est beaucoup plus informative.
Ainsi la représentation informatique de ces textes nécessite un traitement spécifique.
Mais Très vite, les méthodes de se sont heurtées au fait qu'un texte n'est pas un sac dans lequel seraient mélangées en vrac ses propres éléments. Le moins qu'on puisse dire sur un texte qu'il est une chaîne linéaire, donc un espace ordonné. (« *Voile du bateau* » et « *Bateau à voile* » ont des sens complètement différents).
Evoquer la composition d'un texte fait appel à deux définitions de la composition : il s'agit à la fois de déterminer les unités qui vont constituer le texte, tels les atomes qui composent les molécules, et de constituer un texte c'est-à-dire de distribuer, d'organiser ces unités afin d'atteindre certaines idées, comme une molécule qui possède certaines propriétés en raison de sa structure.
Plusieurs approches de représentation dans cet espace sont proposées dans la littérature. Nous détaillons par la suite ces différentes représentations.

2.3- Prétraitements

Nous allons aborder ultérieurement les différentes méthodes de représentation des documents. Ces représentations sont toutes effectuées à base de mots qui sont eux-mêmes une séquence de caractères. Il est donc nécessaire d'effectuer, au préalable du codage d'un document dans un espace de mots, une transformation permettant le passage de l'espace du caractère à un espace de mots.
Le prétraitement des textes est une phase capitale du processus de classification, puisque la connaissance imprécise de la population peut faire échouer l'opération.
Après la première opération que doit effectuer un système de classification à savoir la reconnaissance des termes utilisés, nous devons expurger le plus possible les informations inutiles des documents afin que les connaissances gardées soient aussi pertinentes qu'il se peut. En effet dans les documents textuels de nombreux mots apportent peu (voir aucune) d'informations sur le document concerné. Les algorithmes dits de *"*Stop Words*"* s'occupent de les éliminer. Un autre traitement nommé *"*Stemming*"* permet également de simplifier les textes tout en augmentant leurs caractères informatifs comme d'autres méthodes qui proposent de supprimer des mots de faible importance.
Toutes ces transformations et méthodes font partie de ce qu'on appelle le prétraitement. Plusieurs d'entre elles sont spécifiques à la langue des documents (on ne fait pas le même type de prétraitement pour des documents écrits en anglais qu'en français ou encore en arabe).

Le prétraitement est généralement effectué en six étapes séquentielles :

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

2.3.1- La segmentation

La première opération que doit effectuer un système de classification est la reconnaissance des termes utilisés. La segmentation consiste à découper la séquence des caractères afin de regrouper les caractères formant un même mot.

Habituellement, cette étape permet d'isoler les ponctuations (reconnaissance des fins de phrase ou de paragraphe), ensuite découper les séquences de caractères en fonction de la présence ou l'absence de caractères de séparation (de type « espace », « tabulation » ou « retour à la ligne »), puis regrouper les chiffres pour former des nombres (reconnaissance éventuelle des dates), de reconnaitre les mots composés.
Eventuellement, nous pouvons unifier les écritures en lettre majuscules ou en lettres minuscules avant ou après les opérations déjà indiquées.
C'est un traitement de surface assez simple dans le principe, mais particulièrement difficile à réaliser de manière exacte sur les documents ayant beaucoup de bruits et des représentations assez variées.
Notons que pour des corpus multilingues, une technique de segmentation moins intuitive a été proposée : la segmentation en n-grammes.

2.3.2- Suppression des mots fréquents ou élimination des "Mots Outils"

Les mots qui apparaissent le plus souvent dans un corpus sont généralement les mots grammaticaux, mots vides (empty words) ou mots outils (stop words) : les articles, les prépositions, les mots de liaisons, les déterminants, les adverbes, les adjectifs indéfinis, les conjonctions, les pronoms et les verbes auxiliaires etc.., qui constituent une grande part des mots d'un texte, mais malheureusement sont faiblement informatifs, sur le sens d'un texte puisqu'ils sont présents sur l'ensemble des textes.
A titre d'exemple on peut citer en dans la langue Française, le cas des articles « le », « la », « les » ou de certains mots de liaison « ainsi », « toutefois » etc..
Ou en Anglais : Les prépositions (about, after,through.), les déterminants (the, no, one.), les conjonctions (though, and, or.), les adverbes (above,almost, yet.), les pronoms (who, another, few.) et certains verbes (are, can, have, may,will.).
Et en Arabe : … حروف الجر، حروف العطف، أسماء الإشارة، أخوات كان، أخوات إن، الخ

Ces termes très fréquents peuvent être écartés du corpus pour en réduire la dimension. Cette possibilité de réduire la taille des entrées de l'index en éliminant les mots vides s'explique par le fait que ces termes sont présents dans la quasitotalité des documents et ont donc un pouvoir discriminant faible en comparaison avec d'autres termes.
D'après la loi de Zipf (Voir Section 2.6.1.4). Leur élimination lors d'un pré-traitement du document permet par la suite de gagner beaucoup de temps lors de la modélisation et l'analyse du document.
Ces mots doivent être supprimés de la représentation des textes pour deux raisons :

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

Une répartition des mots outils par rapport les mots utiles dans un corpus est représentée dans la figure 2.1.

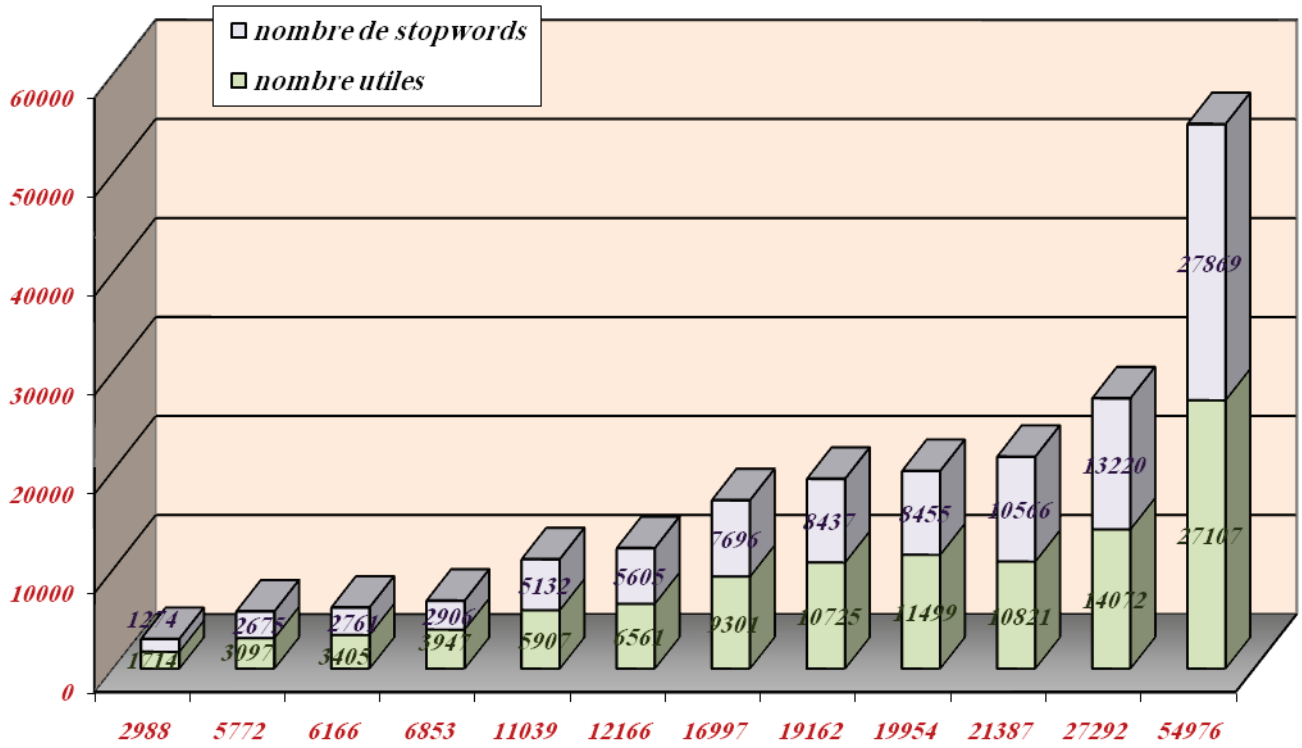


Figure 2.1 : Répartition des mots utiles et des mots vides dans un corpus

L'élimination systématique du corpus des mots vides peut se faire par l'intermédiaire d'une liste prédéfinie de mots pour chacune des langues étudiées.
Cependant, l'établissement d'une telle liste peut poser des problèmes. D'une part, il n'est pas facile de déterminer le nombre de mots exacts qu'il faut inclure dans cette liste. D'autre part, cette liste est intimement liée à la langue utilisée et n'est donc pas transposable directement à une autre langue.
Par exemple Sahami.M dans dans sa thése de PHD (Sahami, 1999) définit une liste de 570 mots courant en anglais, plus une liste de 100 mots très fréquents sur le web.
Comme on peut les écarter en fixant un seuil maximal de fréquence, pour ne pas sélectionner les mots présents dans une grande partie du corpus.
Une autre manière d'éliminer les mots vides d'un texte passe par l'utilisation d'un étiqueteur syntaxique (Part of Speech Tagger) – Les mots sont écartés en fonction de leur étiquette syntaxique sans avoir besoin de liste prédéfinie.

Enfin, un dernier point concernant les opérateurs de négation (ex : pas, ne, non) qui peuvent être supprimés sans gravité. Dans un contexte de classification de textes, une notion affectée par un opérateur de négation reste inchangée contrairement à une négation dans un contexte de recherche d'information qui peut être déterminante pour les résultats attendus. Dans le cadre d'une recherche documentaire, le but à atteindre pour l'utilisateur de rechercher l'information en lien avec la requête. En revanche, dans le cade d'une classification de textes en plusieurs catégories, les opérateurs de négation ne vont guère influencer les résultats puisque l'on cherche à distinguer les thèmes les uns des autres. Par exemple les deux phrases suivantes : il est malade et il n'est pas malade traitent toutes les deux le même sujet de santé, et le terme malade, avec ou sans négation, est un terme décrivant cette notion de santé. En évidence, elles ont un sens opposé mais sont toutes liées au sujet de santé.

2.3.3- Suppression des mots rares

En général, les auteurs cherchent également à supprimer les mots rares, qui n'apparaissent qu'une ou deux fois sur un corpus, afin de réduire de façon appréciable la dimension des vecteurs utilisés pour représenter les textes, puisque, d'après la loi de Zipf (Voir Section 2.6.1.3), ces mots rares sont très nombreux.
D'un point de vue linguistique, la suppression de ces mots n'est pas nécessairement justifiée : certains mots peuvent être très rares, mais très informatifs. Néanmoins, ces mots ne peuvent pas être utilisés par des méthodes à bases d'apprentissage du fait de leur très faible fréquence ; il n'est pas possible de construire de statistiques fiables à partir d'une ou deux occurrences ;
Une des méthodes communément retenues pour supprimer ces mots consiste à ne considérer que les mots dont la fréquence totale est supérieure à un seuil fixé préalablement.

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

2.3.4- Le traitement morphologique

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

Plusieurs traitements morphologiques existent :

- ***Le stemming*** ou ***la désuffixation*** regroupe sous un même terme (stem) les mots qui ont la même racine. L'extraction des stems se fait par la technique de racinisation (ou stemming) qui utilise à la place des dictionnaires, des algorithmes simples basées sur des règles de remplacement de chaînes de caractères pour supprimer les suffixes les plus utilisés.
Le stemming est un traitement linguistique moins approfondie que la lemmatisation, ayant deux avantages : Plu rapide que la lemmatisation (algorithmes simples ne faisant pas référence aux dictionnaires et règles de dérivation) et la possibilité de traiter les mots inconnus sans traitement spécifique. (Clech, 2004)
Néanmoins, sa précision et sa qualité sont naturellement inférieures, du fait qu'elle ne gère que les règles principales et ne peut pas prendre en compte les nombreuses exceptions des règles de dérivations. Par exemple, en français l'une des règles préconise de supprimer le « e » final de chaque mot, le mot *« fraise »* est alors transformé en « *frais »* ce qui suppose une relation entre les deux mots qui n'existe pas. Qui fait de cette opération dépendante de la langue, nécessitant une adaptation pour chaque langue utilisée.
Plusieurs stemmers ont été développés pour déterminer les racines lexicales, l'algorithme le plus couramment utilisé pour la langue anglaise est celle de PORTER (Porter, 1980).

- ***La lemmatisation*** conserve, non pas les mots eux-mêmes, mais leur racine ou lemme. Ce principe permet de prendre en compte les variations flexionnelles (singulier/pluriel, conjugaisons,…) ou dérivationnelles (substantifs, verbes, adjectifs,…) en regroupant sous le même terme tous les mots de la même famille et donc d'améliorer la classification.
La lemmatisation est donc une tâche plus compliquée à mettre en œuvre que la recherche de racines, puisqu'elle elle s'appuie sur des outils de TALN, ce qui nécessite beaucoup de

ressources linguistiques (dictionnaires, règles de dérivation, etc..). De plus les résultats contiennent encore des erreurs à cause des problèmes de polysémie (ambiguïté) et d'incomplétude des dictionnaires.
Un algorithme efficace, nommé TreeTagger (Schmid, 1994) a été développé pour les langues anglaise, française, allemande et italienne. Cet algorithme utilise des arbres de décision pour effectuer l'analyse grammaticale, puis des fichiers de paramètres spécifiques à chaque langue.

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

2.3.5- Le traitement syntaxique

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

2.3.6- Le traitement sémantique

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

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

2.4- 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, les expressions ou les collocations, comme d'autres qui préfèrent les techniques des n-grams, etc…
Dans la section suivante, nous allons définir les différentes sortes de termes, utilisés dans la littérature, pour la représentation d'un document texte.

2.4.1- Représentation en « sac de mots » « bag of words »

Le choix des mots comme descripteurs d'un document c'est le choix le plus intuitive, ainsi un texte sera représenté dans l'espace des mots par un vecteur dont chaque composante correspond au nombre d'apparition d'un mot dans le document, cette représentation est connue par « sac de mots » , « bag of words » .
(Salton & McGill, 1983), (Lewis, 1992), (Dumais & all, 1998), (Yang, 1999), (Vinot & all, 2003), (Pessiot & all, 2004), (Pothin & Richard, 2007) (Trinh, 2008), (Gotab, 2009), (Jégou & all, 2010), et bien d'autres ont préféré l'utilisation des mots comme descripteurs pour le codage des documents.
Pour clarifier la notion de mot, Y. Gilly dans son ouvrage « Texte et fréquence » (Gilly, 1988) l'a considéré comme étant une séquence de caractères appartenant à un dictionnaire, ou formellement, comme étant une suite de caractères séparés par des espaces ou des caractères de ponctuations (Cette définition n'est pas valable pour toutes les langues).

Ces descripteurs ont un vrai privilège de posséder un sens explicite cependant la représentation en « bag of words » affiche plusieurs anomalies :

- Le problème des mots composés comme : Arc-en-ciel, peut-être et le problème des sigles comme : APN, FAF, IBM.
- La présence parmi les descripteurs, des mots outils, qui constituent une grande part des mots d'un texte, mais qui portent peu d'informations utiles pour classer un texte.
- La distinction des mots d'une même famille en raison de leur variation morphologique (ex : écrire, écriture, écrit, écrivain,…) est en général un handicap, car chaque variation a une fréquence très faible alors que les regrouper permettrait d'avoir des fréquences importantes et d'amoindrir le phénomène d'imprécision des fréquences évoqué dans (Jalam ,2003).
- Enfin cette représentation est un regroupement en vrac de tous les mots du document « sac de mots » sans prendre en compte les combinaisons et l'ordre des mots dans la phrase entraine une perte dans la sémantique du texte.

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

Marionnaud: Union et Etudes Investissement franchit 5% des droits de vote
PARIS, 31 juil (AFP) - La société Union et Etudes Investissement (caisse Nationale de Crédit Agricole) a franchi en hausse le seuil de 5% des droits de vote du groupement français de parfumerie Marionnaud et détient désormais 292.157 actions, soit 8,09% du capital et 5,05% des droits de vote, a indiqué vendredi le Conseil des Marchés Financiers.
Ce franchissement de seuil résulte de l'acquisition de 11.460 actions, précise le CMF.

Mot	Occur.	Mot	Occur.	Mot	Occur.
a	2	détient	1	le	3
acquisition	1	en	1	marchés	1
actions	2	et	4	marionnaud	2
agricole	1	études	2	nationale	1
caisse	1	financiers	1	parfumerie	1
capital	1	franchi	1	précise	1
ce	1	franchissement	1	résulte	1
cmf	1	franchit	1	seuil	2
conseil	1	français	1	société	1
crédit	1	groupement	1	soit	1
de	9	hausse	1	union	2
des	4	indiqué	1	vendredi	1
droits	3	investissement	2	vote	3

du	2	l	1		
désormais	1	la	1		

Tableau 2.1 : Exemple de la représentation en « sac de mots »
Les chiffres et dates sont supprimés de la représentation

2.4.2- Représentation des textes par des collocations

Cette approche proposée ici, consiste à regrouper certains mots (collocations) afin d'obtenir des descripteurs ou expressions plus porteurs de sens au lieu d'utiliser des mots isolés composant le texte.
Identifier des collocations consiste à trouver des mots qui "vont ensemble" et qu'il est naturel de trouver proches dans le langage. Pour former ces groupes de mots, on n'a pas besoin de syntagmes nominaux, juste des paires de mots qui peuvent être séparés par des mots vides

Le but n'est pas ici de chercher à analyser les textes d'un point de vue syntaxique, mais les représenter selon un ensemble d'usages de la langue, qui ont une influence sur le système classification. (Par exemple : *repas-bien-garni*, *parler-en-connaissance-de-cause, tout-à-fait-normal*).

Rémi Lavalley, Patrice Bellot et Marc El-Bèze dans (Lavalley & all, 2009), expliquent pourquoi le fait de considérer une suite de mots comme une seule unité informative permet d'améliorer les performances d'un système de classification dans leur article « Interactions entre le calcul de collocations et la catégorisation automatique de textes ».
Tout d'abord pour augmenter la significativité du terme : par exemple, si nous arrivons à repérer l'expression « *effet particulièrement désagréable* » dans un texte, on pourra préjuger que la critique est négative, alors qu'un système classique aurait pris les mots séparément et aurait pu décider autrement, par exemple :
– *effet* : fait pencher vers une critique positive (comme dans l'expression "*cette odeur fait bon effet*") ;
– *désagréable* vers une critique négative.
Ainsi, en traitant l'expression dans son intégralité nous pensons accroître son pouvoir discriminant.
La seconde raison qui pousse à penser que l'on peut améliorer les résultats, selon (Lavalley & all, 2009), vient du fait qu'on peut envisager de créer des collocations propres à une catégorie pendant la phase d'apprentissage.
Pour extraire les collocations présentes dans le corpus d'apprentissage, ils se sont appuyés sur la méthode du Rapport de Vraisemblance.
Néanmoins, parmi les grands problèmes dans cette approche est de savoir lesquelles garder : toutes n'ayant pas la même pouvoir discriminant sur la classification finale, certaines peuvent en effet se recouper.
Les algorithmes d'extraction des collocations, appliquent les règles dans l'ordre dans lequel se trouvent les mots (parcours gauche-droite de la phrase).
Un autre problème posé est de savoir où s'arrêter (nombre de termes associés), car créer des combinaisons trop grandes entraîne des problèmes de fréquence faible (faible probabilité d'apparition de ces collocations).
Aussi, ils ne traitent que des règles assemblant les mots deux à deux (et pas directement trois à trois ou quatre à quatre par exemple)
De plus, il faut trouver un moyen de trouver des collocations "à trous" (mots non obligatoirement consécutifs).

Enfin, malgré qu'il s'agit d'une méthode nouvelle, pour laquelle de nombreuses améliorations sont envisageables, comme même, il existe un certain nombre de travaux de développement de méthodes pour trouver des collocations évoquées dans (Lavalley & all, 2009), par exemple celle de Yu J., Jin Z.,Wen Z.(2003),ou celle de Smadja F. A., McKeown K. R(1990) ou encore Seretan V., Nerima L., Wehrli E (2004), pour une méthode utilisant des filtres syntaxiques appliqués au corpus du Web).

2.4.3- Représentation des textes par des phrases

Ces techniques sont venues pour remédier à la déstructuration syntaxique causée par la représentation en "sacs de mots". Les résultats fournis par ce type de représentation « sac de mots » se basent finalement sur des mots éparpillés composant des textes qui sont très éloignés de ceux qu'ils sont censés représenter. Mais les techniques de traitement statistique (Bayes, Markov...) s'apprêtent mal des représentations à partir de phrases en raison de leurs caractéristiques irrégulières et exceptionnelles (longueur, redondance, bruit, structure compliquée...). Beaucoup de chercheurs s'y sont cassé les dents ayant abouti souvent à des solutions dérivées plus simples.
Logiquement, une telle représentation doit obtenir de meilleurs résultats que ceux obtenus via les mots en raison de la richesse sémantique de la phrase, cependant leurs propriétés statistiques ne permettent pas de définir des hypothèses statistiques fiables (Lewis, 1992) car, le grand nombre d'assemblages possibles de mots engendre des faibles fréquences et trop aléatoires, ne permettant pas d'approximer le risque réel de manière correcte grâce au risque empirique.
L'utilisation de "sac de phrases" entraîne évidemment un problème de taille (pour n mots il y existe potentiellement n^k combinaisons de longueur k).
Pour y remédier, on ne considère pas toutes les séquences possibles mais on tente d'effectuer une sélection des phrases, en privilégiant celles qui sont sémantiquement riche. Dans la phrase "*Le gentil lapin orange mange la carotte bleue*" par exemple, on peut dire que des séquences comme "*gentil lapin orange*", "*carotte bleue*", "*lapin orange*", ... sont porteuses de sens. Alors que les séquences "*orange mange*", "*le gentil*"…etc. sont insignifiantes.
Une autre approche de (Caropreso & all, 2001) qui proposent d'utiliser des phrases statistiques comme descripteurs au lieu des phrases grammaticales qui ont amélioré considérablement la performance du classifieur. Une phrase statistique est définie par (Jalam ,2003), comme une collection de mots adjacents (mais pas forcément classés) qui apparaissent ensembles mais qui ne respectent pas forcement les règles grammaticales.
Une autre étude menée par (Scott & all, 1999), démontre que ce type de représentation améliore la qualité des résultats par rapport aux méthodes de type « sac de mots » lorsque les documents étudiés sont limités en nombre et taille. De plus, ce type de représentation présente de grandes variations dans la qualité de ses résultats en fonction du type de documents à classifier. En effet, lors de ces tests il a utilisé une base de documents issus de Reuters et une base de textes de chansons folkloriques américaines. Les résultats sont bons sur Reuters mais très mauvais sur les textes de chansons.

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

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

L'opération de stemming expliquée précédemment, vise de ramener un mot à une de ses parties qui le caractérisent (racine) ainsi que tous les mots qui lui sont linguistiquement liés plutôt que considérer les mots entiers (on parle de *stem* en anglais).

Ainsi, le stem de numériseur serait *numéris* regroupant aussi : *numériseur, numériseurs*, *numérisée, numériser, numérisation, numérisations*, etc... On voit donc bien, l'intérêt du stemming puisque qu'il rassemble plusieurs mots de significations très proches dans un même groupe. La lemmatisation n'aurait pas pu regrouper *numériseur* et *numérisation* dans le même ensemble. Malheureusement, cette opération n'étant pas basée sur des contraintes linguistiques puissantes, peut conduire, à une amplification du bruit et des confusions sémantiques, en regroupant par erreur des mots de différentes significations, peuvent être générées. Comme la racine lexicale *port* qui regroupe dans le même ensemble le verbe *porter* et le nom *port* (Lieu pour les bateaux) alors que sémantiquement sont très distincts. De plus, le terme œil et son pluriel yeux ne pourront jamais être ramenés au à la même racine avec une simple opération de stemming. Pour éviter ces confusions, cette opération doit être utilisée avec précaution. (De Loupy, 2000)
La représentation des textes par ces stems peut apporter des résultats supérieurs à ceux obtenus par les lemmes (que nous allons voir dans ce qui suit), démontré et approuvé par de Loupy, et bien meilleur que le codage de type « sac de mots » ou chaque variation d'un mot est considéré comme une nouvelle composante du vecteur. Alors, on peut facilement imaginer combien on va gagner en question de dimensionnalité en optant pour les stems comme descripteurs.

2.4.5- Représentation des textes avec des lemmes (lemmatisation)

La lemmatisation décrite auparavant, consiste à utiliser l'analyse grammaticale afin de remplacer les verbes par leur forme infinitive et les noms par leur forme au singulier.
La substitution des mots par leur lemme réduit également l'espace des descripteurs comme pour les racines, et permet de représenter par un même descripteur des termes de même signification. Par exemple, le remplacement des mots *banking, bank*, *banks*, par l'unique racine *bank* semble être rentable tout comme le remplacement des formes conjuguées *rebondit* et *rebondi* par le lemme *rebondir*.
Les mêmes confusions d'ambiguïté peuvent être entrainées en représentant par un même descripteur des mots avec des sens différents, comme par exemple *glace* qui peut être *un miroir* ou « *une glace aux chocolats"*.
L'ambiguïté peut être causée aussi par le simple remplacement de la forme plurielle d'un mot par sa forme singulier comme *actions* qui est représenté par le descripteur *action*. Dans un contexte économique, le mot *actions* se réfère couramment à des actions d'entreprises et n'a rien à voir avec la notion *action* employée par exemple dans la phrase : « Le plan d'*action* de l'état ».

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

Une autre approche pour coder les documents émerge : les n-grammes (Shannon, 1948). On définit un n-gramme (n-gram) par est une séquence de n caractères : bi-grammes pour n=2, tri-grammes pour n=3, quadri-grammes pour n=4, etc..On n'a plus besoin de chercher les délimiteurs(les espaces ou les caractères de ponctuations) comme c'était le cas pour les mots.
Quelques auteurs admettent les n-grammes comme une chaine non ordonnée de caractères; par exemple un tri-grammes peut être constitué du 2éme, 4éme et 1ér caractère, d'autres auteurs n'autorisent pas ce désordre. Pour notre cas, on va admettre qu'un n-grammes désignera une chaîne de n caractères consécutifs.
Pour un texte quelconque, les n-grammes correspondants sont générés en faisant déplaçer un masque de n caractères sur tout le texte. Ce déplacement s'effectue caractère par caractère, à chaque déplacement la séquence de n caractères est enregistrée, l'ensemble de ces séquences constitue l'ensemble des n-grammes représentant le texte. (Miller & all, 1999)

Par exemple, pour générer les 3-grammes de la phrase "Tu es libre", on obtient : "Tu " , "u e" , " es" , "es " , "s l" , " li" , "lib" , "ibr" , "bre".
Historiquement, les n-grammes étaient conçus pour la reconnaissance de la parole, pour prédire l'apparition de certains caractères en fonction des autres caractères mais par la suite, le concept n-grammes a été bénéfique, pour le domaine de recherche d'information et la classification de textes, avec plusieurs travaux qui ont démontré que cette segmentation ne faisait pas perdre d'information.
Il est à noter qu'il existe une autre utilisation de cette notion, où le n-gramme est une suite de n mots et dont l'intérêt est de détecter les liens entre les mots pour en déduire quel mot va apparaître conditionnellement à la présence des n-1 mots précédents (Brown & all, 1992) (dans ce qui suit nous allons considérer le n-gramme dans le sens de séquence de n caractères).
Ainsi pour un alphabet de 26 lettres on obtient $26^2 = 676$ bi-grams ou $26^4 = 456\ 976$ quadri-grams, concernant les mots, pour un dictionnaire de 20 000 mots on obtient $20\ 000^2 = 400$ millions de bi-grams et $20\ 000^3 = 8\ 000$ milliards de tri-grams.
Plusieurs spécialistes récents dans le domaine, ont employé ce type de codage pour repésenter leurs documents, par exemple : (Ralaivola, 2006), (Gotab, 2009), (Généreux, 2010). Cette technique a servi pour coder, même de textes en langue chinoise (Wei, 2009).

Enfin, notons que la technique des n-grammes est tojours trés utilisée, pour représenter les textes, en raison de ses avantages qui vont être dévoilés dans le chapitre 6, qui nous ont incité nous aussi, d'ailleurs, pour l'adopter dans notre étude.

2.4.7- Représentation des textes par des combinaisons de termes

Au lieu de prendre les termes un par un comme descripteurs, l'idée ici est de combiner linéairement des termes pour améliorer la qualité des résultats. L'intérêt est corriger les anomalies liés aux ambigüités et redondances du vocabulaire en combinant plusieurs termes pour avoir des nouvelles variables artificielles, jouant le rôle de nouveaux « termes » (Jalam ,2003).
Une approche typique a été proposé par Deerwester, S.Dumais et autres dans (Deerwester & all, 1990), appelée Latent Semantic Indexing (LSI) appuyée sur l'Analyse des Correspondances Factorielles, introduite dans plusieurs recherches dans le cadre du traitement des données textuelles.
On va revenir sur cette méthode ultérieurement dans le cadre de la sélection de termes dans la section 2.5.3.3 puisque c'est une, parmi les techniques de réduction de dimensionnalité

2.4.8- Représentation des textes basée sur les concepts

Les approches précédentes n'extraient pas la sémantique d'un document mais simplement une comparaison morphologique. Si on peut supposer que chaque terme a un sens, il est plus difficile de prouver que deux documents étant composés des mêmes termes aient forcément le même sens. Les auteurs proposent donc, une nouvelle approche de représentation textuelle « plus sémantique » basée non pas sur les termes présents sur le texte à traiter mais sur les concepts correspondants. Ainsi, au lieu de définir un espace vectoriel dont chaque composante représente un terme (mot, stem, lemme, ou n-gram), on projette l'ensemble de termes du texte sur un ensemble fini de concepts.
Un concept peut représenter un objet matériel, une notion, une idée (Uschold & King, 1995). Trois éléments constituent la notion de concept { terme(s), notion, objet(s)}, un terme ou plusieurs, une notion et un ensemble d'objets. La notion, également appelée intention du concept, contient la sémantique du concept, exprimée en termes de propriétés et d'attributs.

L'ensemble d'objets, également appelé extension du concept, regroupe les objets manipulés à travers le concept ; ces objets sont appelés instances du concept. Par exemple, le terme « stylo », a pour intention « instrument nécessaire pour écrire ou dessiner », et a plusieurs réalisations : « marqueur, stylo à bille, stylo à encre, etc... ».
Un concept est ainsi doté d'une sémantique référentielle (celle imposée par son extension) et d'une sémantique différentielle (celle imposée par son intension).
Un concept ayant une extension vide est appelé concept générique, ces concepts génériques correspondent généralement à des notions abstraites (par exemple, la « vérité »).
Le stylo qui est lui-même composé d'un bouchon et de l'encre et autre chose montre que cette notion ne peut se définir qu'en utilisant d'autres concepts comme « bouchon », « encre » etc..
Les concepts manipulés dans un langage donné sont organisés au sein d'un réseau de concepts liés par des propriétés conceptuelles sous forme d'ontologie lexicale appelée thesaurus.
Un thesaurus est un ensemble de termes normalisés basé sur une structuration hiérarchisée. Les termes y sont organisés de manière conceptuelle et reliés entre eux par des relations sémantiques.
L'avantage d'une telle méthode est en particulier de réduire les problèmes d'ambigüité et de synonymie dans le vocabulaire et de la construction syntaxique (restituer l'ordre des termes). Cette nouvelle approche de représentation permet donc, une factorisation des termes par regroupement de leur champ sémantique. (Jaillet & all, 2003). En effet plusieurs synonymes seront représentés par un seul concept, d'où une réduction de dimensionnalité considérable dans l'espace de codage.

Une expérience a aboutit, celle du Langage Universel d'Echanges (Universal Networking Language) défini dans (Uchida & Zhu, 1999). UNL est un formalisme permettant de modéliser la sémantique de chaque texte par un graphe. Toute expression en langage naturel peut être modélisée en UNL puis traduite dans n'importe quelle langue cible. En UNL, chaque phrase d'un document est définie par un hyper graphe ou les nœuds sont les concepts et les arcs orientés les relations entre les concepts. Puisque la comparaison de graphes n'est pas toujours évidente, une méthode de représentation de ces graphes pour être exploitée dans un processus de catégorisation est décrite dans (Shah & all, 2002).
Par ailleurs, d'autres approches utilisent une représentation de type conceptuelle, c'est le cas de WCM (Word Category Map) (Kohonen & all, 2000).

Cependant, les deux inconvénients majeurs de ce type de codage restent, que les noms propres du texte ne sont pas pris en compte (Absents du thésaurus puisque ces derniers sont sémantiquement vides par définition), et le coût excessif pour la conception, la réalisation et la maintenance d'une telle solution appuyée sur les ontologies.

2.5- Sélection de descripteurs

2.5.1- Besoin de la sélection de descripteurs

Pour une problématique de classification, l'ensemble des descripteurs est constitué de l'ensemble des termes du corpus, un terme pouvant être un mot, un stem ou un n-gramme, etc.., ce qui peut représenter plusieurs centaines de milliers de termes, même après les prétraitements appliquées dans la première phase qui ont procédé à l'élimination des mots les plus fréquents et les plus rares, soit parce qu'ils n'étaient pas discriminants (Mots vides très faiblement informatifs), soit parce qu'ils n'étaient pas exploitables statistiquement (très faible fréquence), le nombre de termes s'avère encore très élevée (De quelques dizaines de milliers à plusieurs centaines de milliers de termes). Parmi l'ensemble des descripteurs restants, on peut penser que tous ne sont pas nécessairement discriminants voire nuisible pour le système.

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

La sélection de descripteurs est un des principaux enjeux du processus, puisque du choix des descripteurs et de la connaissance précise de la population va dépendre la mise au point du classifieur. L'information nécessaire à la construction d'un bon modèle de prédiction peut être disponible dans les vecteurs d'entrées mais une sélection inappropriée de descripteurs ou d'exemples d'apprentissage peut faire échouer l'opération. (Zighed & Rakotomalala, 2002).

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

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

- Pour réduire le temps de calcul : Plus le nombre d'entrées est grand, plus le nombre de paramètres à déterminer est élevé, ce nombre intervient dans l'expression de la complexité de l'algorithme qui va exiger un temps de traitement plus important. (Pour les modèles tels que les réseaux de neurones, le nombre de poids du réseau croît linéairement avec le nombre de descripteurs utilisés en entrée du modèle).

- Pour diminuer le sur-apprentissage : Comme les bases d'apprentissage sont limitées, des associations inattendues peuvent apparaître entre des descripteurs non informatifs et des classes ; elles peuvent avoir une influence négative sur la qualité du modèle. Il faut alors disposer d'une base d'exemples plus grande afin de diminuer le sur-apprentissage résultant du nombre trop important de paramètres, dont certains sont de très faible fréquence, par rapport aux textes du corpus d'apprentissage : on ne peut pas construire des règles stables à partir de quelques apparitions d'un terme dans l'ensemble d'apprentissage.

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

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

2.5.2- Le nombre de descripteurs conservés

Les méthodes de sélection de descripteurs fournissent, en général, une liste de descripteurs classées par degré d'importance, cette notion d'importance qui dépend de la méthode de classement; l'intérêt des différentes approches de réduction de dimensionnalité est d'avoir un ensemble de descripteurs plus réduit mais informatif. Il reste ensuite à fixer le nombre de descripteurs à garder dans cet ensemble. (Stricker, 2000)
La méthode de classification va être forcément, très décisive dans le seuil à fixer pour le nombre de descripteurs à conserver. Comme par exemple, dans un réseau de neurones, réduire la dimension des vecteurs d'entrées est très recommandé alors qu'une approche SVM est capable de traiter des listes plus longues de termes.
Nous cherchons donc, à supprimer des termes de la représentation des textes, tout en sachant que chaque suppression de terme entraîne une perte d'information ; il faut trouver le bon compromis entre, d'une part, la nécessité de réduire l'espace des descripteurs avec moins de redondances possibles et, d'autre part, le nécessité de garder suffisamment d'informations.

Plusieurs chercheurs dans le domaine ont essayé de réaliser ce bon compromis, comme par exemple (Dumais & all, 1998) construit son modèle à base des SVM en prenant en considération seulement 300 termes sur le corpus Reuters, par contre (Joachims, 1998) pense autrement en considérant que tous les termes du corpus, fournis après les prétraitements (Presque 10000 termes), sont informatifs et sa conservation en entrée dans son modèle est nécessaire, sauf que les résultats fournis sont moins bons que pour (Dumais & all, 1998), ce qui amène à dire que les 10000 termes gardés par Joachims n'étaient pas tous utiles.
Plusieurs auteurs, dans leurs travaux, ont proposé différents nombres de descripteurs pour représenter les textes, de 180 à 100 jusqu' aux 20 premiers descripteurs, sans atteindre les qualités des classifieurs. Une synthèse des différentes expérimentations portantes sur le nombre adéquat de descripteurs retenus, qui n'impliquent pas qu'une grande dimension est nécessaire pour avoir des meilleurs résultats, est évoquée dans (Jalam ,2003).

2.5.3- Les méthodes de sélection de descripteurs

La majorité des méthodes sont basées sur le calcul d'une statistique pour chaque terme qui représente son importance pour le document où il figure ou pour le corpus complet, puis à sélectionner les termes les plus importants. Il existe plusieurs formules statistiques pour mesurer la quantité d'information apportée à partir du nombre du nombre d'apparitions du terme dans la classe et hors de la classe.
Dans cette phase, il s'agit aussi de générer un profil pour chaque catégorie. Le profil d'une catégorie doit contenir tout les termes qui caractèrisent cette catégorie par rapport aux autres. Pour mesurer ces statistiques et construire ces profils, il est nécessaire d'utiliser une méthode de sélection de termes.
En effet, il existe plusieurs méthodes de selection de termes :

2.5.3.1- Principales méthodes

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

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

  La Fréquence-document du terme T : $\mathcal{P}$ (T)

- Le *Gain d'Information* : C'est une mesure nécessaire pour prédire la catégorie d'un document selon la présence ou l'absence d'un mot dans un texte, on peut interpréter cette statistique par la quantité d'information apportée par la présence ou l'absence d'un terme dans un document. Un gain d'information important indique que le terme contient plus d'information pour le texte, en revanche, une perte d'information indique que le terme contient moins d'information nécessaire pour classer les textes avec ce terme.

Le Gain d'Information apporté par T=$\mathrm{P}(T,C)\ log\ \frac{P(T,C)}{P(T)P(C)} + \mathrm{P}(T,C)\ log\ \frac{P(\overline{T},C)}{P(\overline{T})P(C)}$

- L'*Information Mutuelle* : Représente la corrélation entre deux variables aléatoires; pour notre cas, les deux variables sont le terme et la classe ; Cette mesure a été fréquemment utilisée pour la catégorisation de textes pour effectuer la sélection de descripteurs, utilisée par (Lewis, 1992), (Moulinier, 1996) et (Dumais et all, 1998).

  L'Information Mutuelle du terme T pour la classe C = $\log\ \frac{P(T,C)}{P(T)P(C)}$

- Le *Chi-deux univarié* ($\mathcal{X}^2_{\text{univarié}}$) : mesure le degré d'indépendance de deux variables aléatoires (présent ou absent), ici le terme (*T*) et la classe (*C*). Cette mesure se calcule à partir d'une table de contingence (2x2), indiquant le nombre de textes associés à la classe *C* où le terme *T* est soit présent (i textes dans le tableau), soit absent (k textes dans le tableau). La même chose est faite pour les textes non associés à *C,* nous calculons j et l. Ainsi de suite chaque terme candidat aura son tableau.

	Classe *C* présente	Classe *C* absente
Terme *T* présent	i	j
Terme *T* absent	k	l

Tableau 2.2 : Table de contingence selon le nombre de documents

Ainsi la formule du $\mathcal{X}^2_{\text{univarié}}$ sera comme suit :

$$\mathcal{X}^2_{\text{univarié}}(\mathrm{T,C}) = \frac{(i+j+k+l)\ (il-jk)^2}{(i+k)(j+l)(i+j)(m+l)}$$

Cette mesure a été utilisée pour la sélection des descripteurs dans (Schütze et all, 1995) et (Wiener et all, 1995).

- Le *Chi-deux mutivarié* ($\mathcal{X}^2_{\text{mutivarié}}$) est une méthode supervisée permettant la sélection de termes en prenant compte de leurs fréquences dans chaque classe comme l'univarié ajoutées aux interactions termes/termes et termes/classes. L'idée consiste à extraire les meilleurs termes qui caractérisent une classe par rapport aux autres. Les principales caractéristiques de cette méthode sont évoquées dans (Jalam ,2003) et (Clech, 2004).
  Un tableau de contingence (termes - classes) de dimension N x M sera construit, N étant le nombre total de termes, M le nombre de documents.
  Cette mesure a été utilisée pour la sélection des descripteurs dans (Jalam ,2003) dans le cadre de la catégorisation de textes multilingues.

2.5.3.2- Inconvénient commun (Association de termes)

Chacune de ces techniques sélectionne un terme pour son d'importance, mais deux termes peuvent ne pas avoir d'importance pour la classification du document pris indépendamment, alors que la présence simultanée de ces deux termes peut avoir un rôle important. Par exemple, les deux termes *droits* et *homme* ne sont pas des termes très informatifs pris un par un, mais l'association de ces deux mots compose le concept de *droits de l'homme* qui a une signification très précise, reste à prendre en considération la distance entre les termes : *droits* et *homme* qui peuvent être présents dans le même document séparément sans la présence d'une interaction entre ces deux termes. Aucune des méthodes présentées ci-dessus ne résout ce problème qui représente un défaut commun à toutes les statistiques précédentes, une

méthode dite « l'orthogonalisation de Gram-Schmidt » issue des méthodes utilisées pour trouver la solution des moindres carrés d'un problème linéaire par rapport à ses paramètres résout ce problème partiellement. Cet algorithme itératif classe les termes par ordre décroissant de leur pouvoir discriminant, du plus important au moins important tout en tenant compte de ceux déjà classés. Une description détaillée de cette procédure et son application peut être trouvée dans (Dumais & Chen, 2000).
Ajoutons au problème d'association de termes, des problèmes de synonymie et polysémie du vocabulaire qui ne sont pris en charges par ces formules statistiques classiques.

2.5.3.2- Autres approches

Les méthodes de représentation des textes à base de concepts ou les combinaisons de termes, sont naturellement des techniques pour diminuer le nombre de termes, qui peuvent solutionner les problèmes de synonymie et polysémie. Ainsi comme on a vu précédemment, le texte sera représenté de telle manière que le descripteur ne sera plus un terme simple mais une combinaison de termes ou il va correspondre à un concept sémantique. Cela dépasse donc la racinisation qui ne cherche qu'à regrouper les mots de même famille et non pas les mots de même sens.
Une première approche appelée *Term Clustering* (Lewis, 1992) consiste à regrouper plusieurs termes pour former un nouvel attribut. Chaque attribut est donc censé représenter un concept sémantique. Cette association de plusieurs termes avec un concept permet de gérer la synonymie. La polysémie des mots est également prise en compte en permettant à un terme d'appartenir à plusieurs groupes.
Une autre technique nommée *Indexing by Latent Semantic Analysis* proposée par S.Deerwester et S.Dumais (Deerwester & all, 1990). LSI est basée sur le principe d'une structure latente des termes qu'on peut retrouver à l'aide de l'analyse factorielle, qui décompose la matrice d'occurrence $[M_{ij}]$ en valeurs singulières (M_{ij} est le nombre d'occurrences du terme *j* dans le document *i*). La décomposition revient à changer la représentation par un changement de base. Chaque descripteur est donc représenté par une combinaison linéaire de termes. Seulement les *j* axes de plus grandes valeurs singulières seront préservés. LSI a été utilisée pour sélectionner les entrées d'un réseau de neurones dans (Wiener & all, 1995).
Ces deux méthodes, LSI et Term Clustering, améliorent les performances de quelques pourcents par rapport les autres méthodes de sélection de termes moins complexes, mais elles nécessitent un temps de calcul supplémentaire pendant l'apprentissage et impossibilité de traiter un nouveau document sans relancer tout le processus. L'utilisation de ces deux techniques dépend du contexte de l'application et de la vitesse des changements susceptibles d'intervenir dans le corpus.

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

(Jalam ,2003) a rappelé ce qui a été évoqué par (Sebastianni, 2002) sur la réduction des dimensions qui peut être localement ou globalement :

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

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

2.6- Pondération ou calcul de poids

Le tableau (termes x documents) est constitué par le nombre d'apparitions du terme dans le document du corpus. Cette information de base doit être pondérée en fonction de divers paramètres liés au document lui même (ex : le nombre de termes par document) ou au corpus en intégralité (ex : le nombre de termes du corpus). L'intérêt de cette pondération est mieux exploiter l'information contenue dans le document pour améliorer les performances d'un système de classification de textes (SPARCK-JONES, 1972).

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

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

Voici, un petit aperçu sur les pondérations les plus habituellement utilisées signalées dans (Clech, 2004):

- Commençant par le choix le plus simple, qui ne s'intéresse que sur la présence ou la non-présence d'un terme dans le texte, il consiste à utiliser une pondération binaire : 1 si le terme est présent une ou plusieurs fois dans le document, 0 dans le cas contraire .Cette représentation binaire est historiquement la plus ancienne et la plus simple.
  Néanmoins, cette fonction est moins utilisée pour les méthodes statistiques, car ce codage supprime de l'information qui peut être utile : l'apparition du même mot plusieurs fois dans un texte peut constituer un élément de décision important.
- Les représentations fréquentielles sont aujourd'hui les plus utilisées et sur lesquelles notre étude va être basée. Plusieurs variantes s'illustrent :
  - Une première approche consiste à utiliser seulement le nombre d'occurrences des termes. Cette pondération ne peut être valable que dans les documents de même taille, sinon elle avantage les termes qui se répètent souvent dans les documents les plus longs.
  - Une autre approche simple consiste à utiliser la fréquence d'un terme par rapport au nombre de termes composant le texte.
  - La pondération TFxIDF corrige la fréquence du terme(Term Frequency) en fonction de sa fréquence dans le corpus(Inverse Document Frequency). La correction se fait en multipliant du rapport des n textes du corpus sur le nombre de documents contenant le terme. Le logarithme est utilisé pour lisser les résultats.
  - Le TFC normalise le TFxIDF en fonction de l'ensemble des termes du document.
  - La pondération LTC est du TFC sur lequel on applique du logarithme afin d'atténuer les différences de nombre d'occurrences pour diminuer les effets des différences de fréquences.
- La pondération basée sur l'entropie est la plus performante comparée à six autres méthodes, approuvée par les expérimentations de (Dumais, 1991) et confirmée par d'autres auteurs récémment. Selon l'auteur l'entropie devance le TFxIDF sur les cinq corpus testés. Néanmoins, cette méthode est assez complexe du fait qu'elle fait intervenir l'ensemble des autres textes, sollicitant un temps de traitement plus grand.

D'autres pondérations sont en pleine étude :

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

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

- D'autres représentations plus riches existent notamment dans le domaine du Traitement de la Langue Naturelle (TALN) comme par exemple les représentations qui prennent en compte le rôle du mot dans une phrase (Nom, Verbe, Sujet, etc…). Ces représentations s'avèrent efficaces, cependant, les modèles qui les utilisent sont peu performants de ceux qui travaillent dans des espaces plus « simples ». Ces dernières ne sont pas présentées ici, pour plus d'informations sur celles-ci, il est intéressant de se reporter à (Chandra, 1998).

Un processus de classification automatique de textes employant des méthodes essentiellement statistiques peut-être représenté selon :

- Le modèle vectoriel
- Le modèle probabiliste
- Représentation séquentielle

2.6.1- Le modèle vectoriel

2.6.1.1- Représentation binaire

Historiquement, la première représentation d'un texte était sous forme de vecteur binaire, et malgré l'apparition de nouvelles formules pour pondérer les termes, cette façon de représenter un document, est restée toujours largement utilisée en raison du bon compromis fourni entre performance et complexité. Effectivement, en raison de sa simplicité, son temps de traitement est faible et en contrepartie ses résultats ne sont pas mauvais. Elle est appelée représentation « par mots clés ». La méthode consiste à transformer le texte en un vecteur dont les éléments renseignent sur la présence (valeur égale à 1) ou l'absence (valeur égale à 0) d'un terme dans le texte. Deux exemples de textes sont indiqués dans la figure 2.2, leurs vecteurs binaires correspondants sont représentés dans le tableau 2.3

« ce groupe est le groupe de la mort du mondial 2010»	« cette équipe est une équipe favorite au mondial 2010»
Document 1	*Document 2*

Figure 2.2 : Deux exemples de documents

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

2.6.1.2- Représentation fréquentielle

Cette représentation consiste à présenter le texte sous forme de vecteur dont les éléments renseignent non seulement sur la présence ou l'absence d'un terme comme dans un vecteur binaire mais aussi informe sur le nombre de présences du terme dans le texte.
Ainsi, un document est transformé en un vecteur dont les composantes vont correspondre au nombre d'occurrences des termes dans le document.
Pour chaque document, un poids est attribué à chacun des termes qu'il contient. Une matrice « documents /termes » représente l'ensemble des documents (un vecteur est associé à chaque document, les composantes des vecteurs sont les poids des termes) (Salton & McGill, 1983).

Cette méthode conçoit le calcul du poids proprement dit des termes.
Aucune analyse linguistique n'est utilisée, ni aucune notion de distances entre les mots.
Ainsi les deux inconvénients majeurs de cette représentation, sont la non prise en charge des interactions des termes entre eux traduit par une indépendance de ces termes d'une part et d'autre part la déstructuration syntaxique du document causée par le fait que le modèle ne permet pas de conserver l'ordre des mots.
Un exemple de cette représentation est montré dans le tableau 2.3

2.6.1.3- Représentation fréquentielle normalisée

Du point de vue statistique**,** la représentation fréquentielle confronte un problème majeur du fait qu'un texte long sera représenté par un vecteur dont la norme sera supérieure à celle de la représentation d'un document plus court. Il est donc recommandé de normaliser la représentation fréquentielle par rapport à la taille du document. Ainsi le poids du terme sera le nombre d'occurrences de ce dernier dans le texte sur le nombre d'occurrences de tous les termes du texte. Le tableau 2.3 contient une représentation de ce type.

2.6.1.4- Vecteur TF-IDF

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

a- Loi de Zipf :

La répartition des fréquences des termes dans un corpus a été étudiée empiriquement par Zipf (Zipf, 1949). Zipf est parti d'un principe général avant qu'il énonce cette loi mathématiquement. Cette loi réaffirme que la distribution des occurrences des mots dans un corpus donné n'est pas uniforme, certains mots apparaissent très fréquemment, tandis que d'autres apparaissent très rarement.
Les termes les plus informatifs d'un corpus de textes ne sont pas :

- Les termes qui se répètent souvent dans le corpus, qui sont généralement des mots outils. Les mots les plus fréquents en français sont les mots grammaticaux comme *le*, *la*, *les*, *et*... Sur le corpus Reuters, les cinq mots qui se répètent le plus sont : the, of, to, and, in. Cependant les mots non outils qui apparaissent fréquemment contiennent sûrement des informations fortes sur la sémantique du texte.
- Ni les termes les moins fréquents du corpus présents dans un seul ou quelques textes rédigés par des auteurs utilisant un vocabulaire très particulier ou même des termes issus de fautes d'orthographe.

Ces deux observations précédentes ne se contredisent pas et peuvent être récapitulées de la manière suivante : ***« Un mot est informatif dans un document si il y est présent souvent mais qu'il n'est pas présent trop souvent dans les autres documents du corpus ».*** (Denoyer, 2004)

La figure 2.3 illustre de manière graphique la loi de Zipf, qui montre clairement l'évolution de l'importance des mots par rapport leurs fréquences dans un corpus.

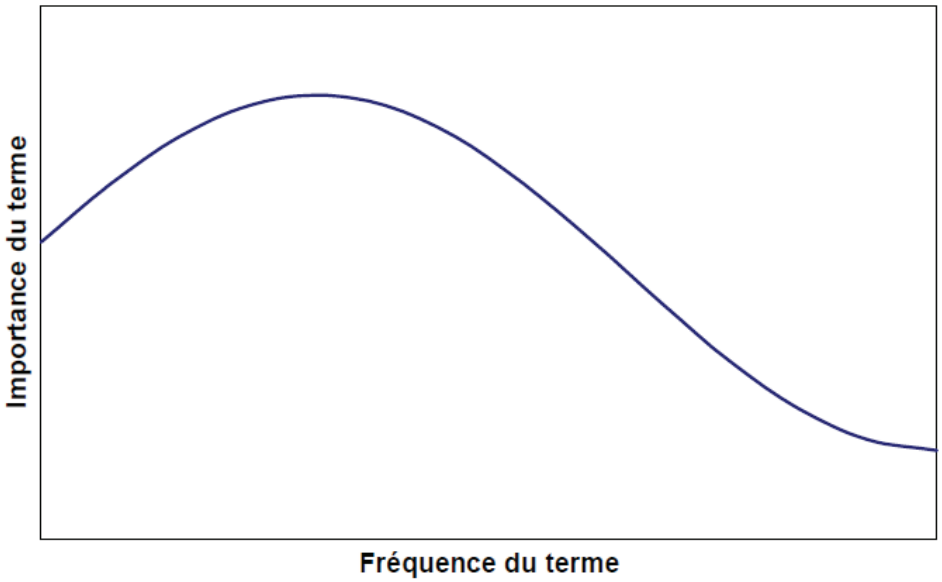


Figure 2.3 : Loi de Zipf

« Un mot est important si il n'est ni trop fréquent ni trop rare ».

Les observations de cette loi sur la distribution des fréquences des termes nous conduit à la suppression des mots fréquents et des mots rares en prenant en considération un seuil fixé préalablement, qui fait de cette loi une première méthode de réduction de dimensionnalité de l'espace des descripteurs. Donc il s'agit d'éliminer des termes inutiles pour les algorithmes d'apprentissage. Cette phase du processus est critique, car les mots écartés sont irrécupérables. De_ce_fait il faut être très attentif pour ne pas supprimer des mots sémantiquement riches.

b- Représentation TF-IDF :

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

$$\mathbf{L_{zipf}\ (T) = Poids\ dans\ le\ document\ *\ Poids\ dans\ le\ corpus}$$

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

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

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

*Term Frequency * Inverse Document Frequency*

c- Variantes du TF-IDF :

Il est admis que l'occurrence d'un terme est une information importante, mais cependant, on considère généralement que *TF.IDF* ne doit pas être l'identité. Si, par exemple, un mot apparaît deux fois dans un texte, son importance n'est pas nécessairement deux fois plus

grande que s'il n'apparaissait qu'une seule fois. Pour améliorer la pondération des termes plusieurs variantes du *TF* ont été proposées :
Le *TF* peut être égal au nombre d'occurrence du terme, à son log, au log de son log, etc. L'utilisation du logarithme permet de diminuer l'importance des termes fortement répétés. En effet un terme à plus d'importance s'il passe de 1 à 2 occurrences que s'il passe de 20 à 21 occurrences. Contrairement au *TF*, un large consensus existe pour l'*IDF*.
Le modèle le plus classique est celui pour lequel la première valeur est égale à la fréquence du mot dans le document notée $TF(d, w_i)$ pour *term frequency* et la seconde valeur est égale à $Log\left(\frac{nbre\_doc}{DF(w_i)}\right)$ où nbre_doc est le nombre de documents du corpus et $DF(w_i)$ est le nombre de documents qui contiennent le mot *i (DF* signifie *document frequency)*.
Particulièrement dans ce cas là, cette représentation sera appelée représentation *TF-IDF* , Elle correspond à la représentation suivante :

$$TFIDF(d, w_i) = TF(d, w_i) \times Log\left(\frac{nbre\_doc}{DF(w_i)}\right)$$

Le tableau 2.3 inclut un exemple de représentation *TF-IDF*.

En général, on utilisera une version du vecteur *TF-IDF* normalisé, comme pour les représentations fréquentielles, afin d'éviter les problèmes posés par les différentes longueurs de textes. On aura ainsi, une représentation *TF-IDF* normalisée.
Plusieurs normalisations sont proposées, parmi elles, une mesure de pondération appelée *TFC* qui est similaire à celui de *TF-IDF* mais qui corrige les longueurs des textes par la normalisation en cosinus, afin de ne pas favoriser les documents les plus longs.

$$TFC(t_k, d_j) = \frac{TF \times IDF(t_k, d_j)}{\sqrt{\sum_{s=1}^{|r|} (TF \times IDF(t_s, d_j))^2}}$$

D'autres pondérations sont aussi utilisées, comme par exemple le LTC (Buckley & all, 1994) dont l'intérêt est la réduction des effets des différences de fréquences, ou encore le codage à base d'entropie utilisé par (Dumais, 1991).

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

	Doc1	***Doc2***	***Doc1***	***Doc2***	***Doc1***	***Doc2***	***DF***	***Doc1***	***Doc2***
Ce	1	0	1	0	0.09	0	1	0.09	0
Cette	0	1	0	1	0	0.11	1	0	0.11
Groupe	1	0	2	0	0.18	0	2	0.09	0
Equipe	0	1	0	2	0	0.22	2	0	0.11
Le	1	0	1	0	0.09	0	1	0.09	0
Est	1	1	1	1	0.09	0.11	2	0.045	0.055
De	1	0	1	0	0.09	0	1	0.09	0
La	1	0	1	0	0.09	0	1	0.09	0
Une	0	1	0	1	0	0.11	1	0	0.11
Mort	1	0	1	0	0.09	0	1	0.09	0
Favorite	0	1	0	1	0	0.11	1	0	0.11
Du	1	0	1	0	0.09	0	1	0.09	0
Au	0	1	0	1	0	0.11	1	0	0.11
Mondial	1	1	1	1	0.09	0.11	2	0.045	0.055
2010	1	1	1	1	0.09	0.11	2	0.045	0.055
Vocabulaire	***Vecteur binaire***		***Vecteur Fréquentiel***		***Vecteur Fréquentiel Normalisé***		***DF et Vecteur TF-IDF***		

Tableau 2.3 : Représentations vectorielles des documents de la figure 2.2

2.6.2- Le modèle probabiliste

La pondération des termes dans ce modèle est estimée par les probabilités d'apparitions des termes dans les textes. Ces techniques admettent l'indépendance entre les différents termes pour une simplification des calculs. Malgré que cette supposition soit peu réaliste, elle donne, néanmoins, des résultats intéressants. (Robertson & Sparck-Jones, 1976).
Dans l'approche du modèle probabiliste, le coefficient de similarité entre un document et les différentes classes du corpus est la probabilité que le document soit assigné à la classe.
Dans le modèle probabiliste, on considère que les documents sont générés par tirage aléatoire des différents termes qui les composent, les valeurs de probabilité de chaque tirage, sont estimées à partir des occurrences trouvées sur les documents du corpus. Cela revient en général à estimer la probabilité d'apparition du terme T sachant que le document appartient à la classe C.
Parmi les modèles probabilistes les plus utilisés le modèle Naïve Bayes, que nous verrons par la suite dans la section 3.2.6.

2.6.3- Représentation séquentielle

La représentation séquentielle d'un texte est une représentation sémantiquement plus riche et conceptuellement plus simple mais elle exige des modèles plus complexes comme les modèles de Markov Cachés. Le texte dans un tel codage, n'est pas représenté par un vecteur dans un espace donné, mais par une séquence de mots. Cette représentation est en fait une représentation naturelle d'un texte mais à cause de sa conception séquentielle qui nécessite le développement de modèles plus évolués, beaucoup moins d'applications se sont imposés par rapport aux représentations vectorielles.
De plus, il s'est avéré que, bien que cette représentation fût plus informative pour sa conservation de l'ordre des mots dans le texte, les classifieurs basés sur ce codage ne livraient

pas toujours de meilleurs résultats que des classifieurs basés sur des codages plus simples. Enfin, de par la nature non vectorielle de cette représentation, le stockage et l'indexation des textes restent une tâche beaucoup plus compliquée que dans les représentations précédentes. Notons néanmoins qu'il est possible à partir d'une représentation séquentielle de construire une représentation vectorielle tandis que l'inverse n'est pas vrai. (Denoyer, 2004)

2.7- Conclusion

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

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

Plusieurs techniques de sélection des descripteurs ou réduction de dimensionnalité sont proposées dans la littérature, une bonne partie de ces approches sont étalées dans ce chapitre.

Une fois la liste des descripteurs arrêtés, un degré d'importance ou poids est attribué à tous les termes présents dans la représentation vectorielle ou probabiliste puisque chaque terme possède un certain nombre d'occurrences dans le document ou dans le corpus qui est différents aux autres.

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

Chapitre 3

Approches de classification : Etat de l'art

Table des matières

3.1- Introduction

Avant de présenter les approches les plus utilisées dans la littérature, en exposant les principes de base, les particularités de chaque méthode de classification de textes (section 3.3), nous allons introduire ce chapitre par un petit rappel sur le principe de base de l'apprentissage automatique et particulièrement le supervisé, après nous pourrons conclure facilement que la problématique de catégorisation de textes se situe bien dans le cadre de l'apprentissage supervisé, et pour en finir cette introduction, nous donnerons un aperçu sur la façon avec laquelle les documents sont classés, le chapitre qui va suivre définit les formules les plus connues pour le calcul de distance qui va être utilisée dans plusieurs méthodes de classifications comme les SVM, kPPV ou Rocchio .

3.1.1- L'apprentissage automatique

Puisque l'approche manuelle de classification de textes est coûteuse en temps de travail, peu générique, et relativement peu efficace, l'autre solution a été admise, qui consiste à faire apprendre automatiquement à l'ordinateur, sur la base d'un corpus de textes qui servent d'exemples, les paramètres de la fonction de classement.
Ainsi depuis une quinzaine d'années la classification de textes a été considérée comme un problème d'apprentissage automatique et est rapidement devenue un champ d'essai sollicité par les différentes techniques de classification.
De toute façon, quelle que soit l'approche retenue, une des particularités de cette tâche est la très grande dimensionnalité de l'espace dans lequel les textes sont représentés, qui comprend généralement plusieurs milliers de termes.
L'apprentissage automatique s'intéresse aux méthodes inductives permettant d'acquérir des connaissances à partir d'observations d'un phénomène. Cette connaissance peut être exploitée pour des tâches de décision ou de prévision : c'est le cadre de l'apprentissage supervisé ; ou à des fins d'analyse exploratoire ou de structuration d'un ensemble de données : c'est le cadre de l'apprentissage non-supervisé (Yvon, 2006).
Le contexte de notre étude se situe dans le premier cas.

3.1.2- L'apprentissage supervisé

Le cadre général de l'apprentissage supervisé consiste, à partir de l'observation d'un ensemble de couple de données de la forme [(x(i), y(i)], (i = 1 ---> n), à induire la valeur de y pour de nouvelles valeurs de x. Dans un cadre probabiliste, chaque x(i) représente une observation d'une variable aléatoire X.
Suivant les valeurs de la variable aléatoire Y, deux cas de figures peuvent être distingués : lorsqu'elle prend des valeurs discrètes, on parle de catégorisation, lorsque ses valeurs sont discrètes, de régression.
Le cadre statistique de la catégorisation supervisée considère le problème de l'apprentissage comme celui de l'induction d'une fonction f. Sous l'hypothèse que les observations sont indépendantes et uniformément distribuées selon une loi de probabilité inconnue P, la meilleure fonction (hypothèse) sera celle qui a une espérance de risque minimum (Yvon, 2006).
Ce cadre général est bien connu et plusieurs outils statistiques ont été utilisés dans le domaine pour résoudre ce problème : réseaux de neurones, régression logistique, Formule de Bayes, arbres de décisions, k-plus proche voisins, machines à vecteurs de support (SVMs), boosting, et bien d'autres. Ces modèles se généralisent plus ou moins directement aux situations dans lesquelles Y prend plus de deux valeurs (catégorisation multiclasses).

3.1.3- La catégorisation est un problème de classification supervisée

Pour construire un filtre relatif à une classe donnée, il faut donc disposer de couples (Document, Classe), ces exemples de chaque classe, préalablement étiquetés constituent le corpus d'apprentissage.
On fait appel aux méthodes d'apprentissage supervisées pour ajuster un modèle qui crée une association entre les documents d'entrée et les classes de sortie. Ainsi, par ces méthodes d'apprentissage, il est possible de construire un modèle de classification, à partir de ces exemples connus à priori (Document, Classe). (Jalam, 2003)
Ce qui affirme clairement que la catégorisation de textes est bien un problème de classification supervisée.

3.1.4- Comment classer ?

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

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

3.2- Différents modèles de classifieurs

Historiquement, différentes générations d'algorithmes de classification automatique de textes se sont succédé. Une part d'amélioration a été apportée par chaque nouvelle génération par rapport aux antécédentes. Parmi les premières, nous trouverons certainement les approches sémantiques dont l'handicap principal résidait dans le coût excessif puisque plusieurs experts sont chargés dans des intervalles de temps importants, pour mettre à jour les plans de classement.
Pour apaiser à ces limitations, plusieurs boites ont conçues et commercialisées des technologies de catégorisation automatique basées sur des approches purement statistiques.

Plusieurs générations de techniques statistiques ont depuis été développées et qui ont confirmé leurs performances en obtenant de meilleurs résultats.

Actuellement, plusieurs approches de catégorisation coexistent. Nous citons parmi les plus utilisées le modèle probabiliste Naïve Bayes, ou les modèles vectoriels de Machine à Vecteur de Support, les k-plus-proches voisins, Rocchio, ainsi que des modèles à base de règles ou d'arbre de décision ou encore des approches fondées sur les réseaux de neurones qui ont été proposées. Chacun de ces modèles possède certains avantages et certains inconvénients. Dans ce qui suit une liste plus ou moins exhaustive des différents modèles et de leur intérêt respectif sera présentée. Une description plus détaillée sera accordée à l'approche naïve bayesienne dans le chapitre 6 qui fera l'objet des modèles de classification dans notre approche proposée.

3.2.1- Machines à Vecteurs Support – SVM

3.2.1.1- Présentation de l'approche

L'algorithme SVM (Support Vector Machine) est une méthode d'apprentissage supervisée relativement récente introduite pour résoudre un problème de reconnaissance de formes à deux classes (Vapnik, 1995). Le principe de SVM a été proposé par Vapnik à partir de la théorie du risque empirique.
La méthode SVM est un classificateur linéaire utilisant des mesures de distance.
En ce qui concerne son application à la problématique de catégorisation de documents, l'algorithme repose sur une interprétation géométrique simple est l'idée générale est de représenter l'espace des exemples (ici des documents) dans un espace vectoriel où chaque document étant un point dans cet espace et de trouver la meilleure séparation possible de cet espace en deux classes. L'espace de séparation est une surface de décision appelée marge, défini par les points « vecteur support ». Ces points se trouvent au minimum de marge. La marge se présente alors comme la plus courte distance entre un vecteur de support et "son" hyperplan. La marge se définit comme la plus petite distance entre les exemples de chaque classe et la surface séparatrice S :

$$\text{marge}(S) = \sum_{c_j \in C} \min_{x_i \in c_j} (d(x_i, S))$$

Ainsi la décision s'appuie sur les SVM pour couper l'espace en deux : d'un côté, ce qui est dans la catégorie, de l'autre côté, ce qui n'y est pas.
l'approche par SVM permet donc de définir, par apprentissage, un hyperplan dans un espace vectoriel qui sépare au mieux les données de l'ensemble d'apprentissage en deux classes, minimisant le risque d'erreur et maximisant la marge entre deux classes. La qualité de l'hyperplan est déterminée par son écart avec les hyperplans parallèles les plus proches des points de chaque classe. Le meilleur hyperplan est celui qui a la marge la plus importante.
SVM a été étendu pour les points ne pouvant être séparées de manière linéaire (par exemple notre cas des vecteurs de documents), en transformant l'espace initial des vecteurs de données à un espace de dimension supérieure dans lequel les points deviennent séparables linéairement. Nous trouvons dans (Joachims, 1998) une application efficace des SVM.
La figure 3.1 montre une telle séparation dans le cas d'une séparation linéaire par un hyperplan.

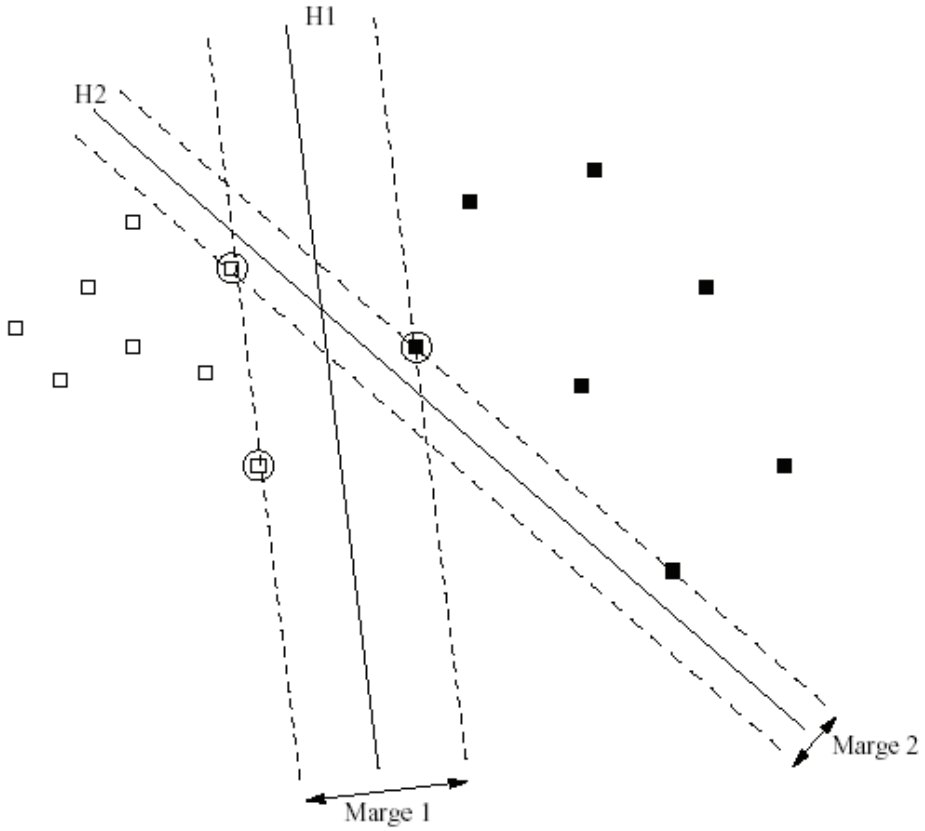


Figure 3.1: Exemples d'hyperplans séparateurs en dimension deux. Les vecteurs de support sont encerclés

Dans l'exemple de la figure 3.1, les exemples des deux classes peuvent être séparés par un hyperplan, le problème est dit linéairement séparable. Les deux hyperplans H_1 et H_2 sont tous les deux des séparateurs acceptables, mais l'hyperplan H_1 a une plus grande marge et sera donc préféré. Pour calculer l'hyperplan optimal et donc la marge, seuls les exemples les plus proches de la zone-frontière sont mis à contribution. L'apprentissage consiste à déterminer ces exemples appelés vecteurs de support. Tous les autres peuvent être écartés et n'interviennent plus dans les calculs.

Le problème se traduit mathématiquement en un problème d'optimisation quadratique : trouver l'hyperplan (w,b) (b est la distance à l'origine de l'hyperplan) qui minimise la norme de w sous les contraintes :

$$\forall d_i,\ c_i(w.d_i - b) \leq 1$$

Avec d_i le i^{éme} document, de classe c_i (+1 ou –1). Si les exemples ne sont pas linéairement séparable, on peut les plonger conceptuellement dans un espace de dimension plus grande (la dimension peut même être infinie) par une fonction de transformation appelée *noyau* (kernel). Dans cet espace, les exemples seront plus facilement séparables. Une propriété de l'algorithme est qu'il ne requiert pas les coordonnées de chaque exemple mais seulement les produits scalaires de chaque couple d'exemples, qui restent calculable une fois les exemples plongés dans un nouvel espace même de dimension infini.

Si cela ne suffit pas pour rendre les exemples séparables, il est possible d'ajouter encore un terme correctif qui autorise un nombre limité d'exemples à être mal classés. Pendant l'apprentissage, on cherchera à rendre ce terme le plus petit possible. Un paramètre de l'algorithme permet de donner plus ou moins d'importance à ce terme correctif. Dans sa

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

3.2.1.2- Critiques de l'approche

Actuellement, l'algorithme SVM semble tres Prometteuse et considéré parmi les plus performants pour la catégorisation en raison de sa modélisation simpliste et rapide à calculer par une machine étant donné que SVM est un Classificateur linéaire qui correspond à une équation polynomiale de degré p :

$a_{1x}\,x+a_{1y}\,y+a_{1z}\,z$ pour la catégorie C1

$a_{2x}\,x+a_{2y}\,y+a_{2z}\,z$ pour la catégorie C2

Seulement elle introduit des concepts complexes peu adaptés aux corpus de grandes tailles non fixes, sans oublier de rappeler de son faible pouvoir descriptif puisque les coefficients ne sont pas interprétables intuitivement par des humains.

3.2.2- Rocchio

3.2.2.1- Présentation de l'approche

La méthode Rocchio parue dans (Rocchio, 1971), est un classifieur linéaire basée sur le calcul des mesures de distance, il fait partie des premières techniques de classification supervisée. Plusieurs améliorations se sont apportés sur le modèle mais la définition présentée ici est celle de la version initiale.

Dans la phase d'apprentissage de la méthode, les représentations vectorielles, vont permettre le codage de chaque catégorie par un vecteur dont lequel figure tout les termes générés avec leur nombre d'occurrence. Le processus représente donc les classes par des profils prototypiques correspondants à des vecteurs dans un espace vectoriel similaire aux documents. Ces profils sont donc des listes de termes pondérés générées pendant l'apprentissage de même pour les vecteurs correspondants aux textes qui sont aussi générés durant cette phase. Le profil d'une catégorie doit contenir tous les termes qui caractèrisent cette catégorie par rapport aux autres.
Bien évidemment, le vecteur d'une classe sera calculé ici uniquement en fonction des documents du jeu d'apprentissage. Pour construire les profils, il est nécessaire d'utiliser une méthode de sélection et réduction de termes.
Chaque terme du corpus représente une dimension de l'espace et le codage des vecteurs se fait soit par une fonction booléenne soit par une fonction du nombre d'occurrences d'un terme dans le document. Plusieurs pondérations des termes se présentent, la plus utilisée est TFIDF.

Dans la phase de classification, il s'agit de comparer le profil (vecteur) du nouveau document à classer à tous les profils des classes déjà calculés dans l'étape d'apprentissage.
Cette comparaison équivaut au calcul d'une fonction de similarité ou de distance entre les vecteurs représentant les classes et le vecteur correspondant au nouveau document. Elle permet d'ordonner les classes en fonction de leur distance du document. Par conséquent, le principe de catégorisation Rocchio se résume à assigner le document à la classe dont la distance euclidienne entre le vecteur du document et le vecteur de la classe est la plus courte.

3.2.2.2- Critiques de l'approche

Rocchio est caractérisée par sa modélisation simpliste et rapide à calculer par une machine et malgré l'ancienneté et la simplicité du modèle, la méthode a confirmé par ses performances qui sont concurrentielles à celles obtenues par d'autres techniques plus sophistiquées, avec un gain de temps d'apprentissage pour une machine très considérable.
Plusieurs auteurs dans leurs travaux ont démontré la résistance de Rocchio au bruit : même avec 50% des exemples bruités, les performances sont assurées. Les différentes expérimentations basées cette approche ont donné aussi de très bons résultats sur les tâches de routage (filtre anti-spams par exemple). (Vinot & Yvon, 2002).
En revanche, l'inconvénient principal de la méthode reste sa faiblesse d'expressivité et son pouvoir descriptif très réduit qui fait du modèle une fonction de décision non interprétable intuitivement par les humains.

3.2.3- Méthode du centroïde

3.2.3.1- Présentation de l'approche

L'algorithme du centre de gravite ou « centroïde » est une variante de l'algorithme Rocchio bien connu dans le domaine, c'est une méthode géométrique utilisant les mesures de distance pour classer les documents.
La phase d'apprentissage consiste à calculer un centroïde pour représenter les classes. Le vecteur centroïde d'une catégorie sera représenté par la moyenne des vecteurs des textes qu'elle contient, qui correspond au barycentre des différents textes préalablement assignés à cette classe. La méthode du centroïde calcule la distance entre les centres de gravité (au sens géométrique) des catégories.

La catégorisation d'un nouveau document se fait par le calcul de similarité entre les vecteurs centroïdes des catégories et le vecteur du nouveau document. La catégorie du centroïde le plus proche est attribuée dans le cas de classification multi-classes disjointes sinon des scores sont calculés pour les classes puis un seuillage est utilisé pour choisir les classes à attribuer.

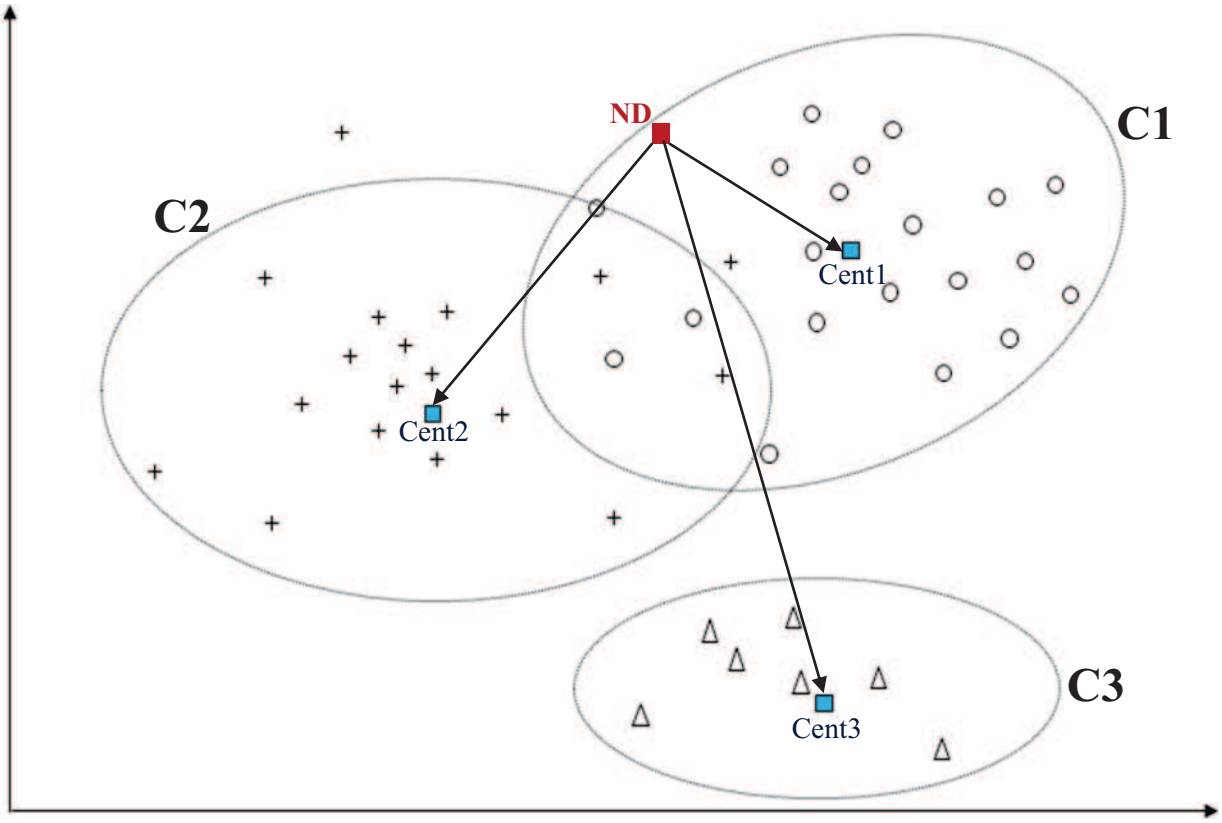


Figure 3.2: Exemple de la méthode du centroïde
Le nouveau document ND est attribué à la classe C1 du centroïde Cent1 le plus proche

3.2.3.2- Critiques de l'approche

Comme la méthode considère un vecteur représentatif pour chaque classe, elle ne fonctionne que si cette classe est bien définit par un prototype. Mais, puisque les comparaisons ne sont faites qu'avec les vecteurs centroïdes, l'algorithme est plus efficace que k-PPV (Section 3.2.4) pour la catégorisation en temps réel.
La méthode de la comparaison directe ou méthode du centroïde est caractérisée par son efficacité (rapidité), sa robustesse et son interprètabilité, en revanche son optimisation est difficile dans le cas de nombreuses classes.
Elle est très utilisée dans le domaine du traitement du Signal et d'image. Elle consiste à calculer le centre de gravité correspondant des spectres ou des pixels pour divers traitements comme la segmentation par exemple.

3.2.4- K plus proches voisins - kPPV

3.2.4.1- Présentation de l'approche

La méthode des k plus proches voisins ou The k-NN classification (k-Nearest Neighbors) est un classifieur à base d'instances qui fait partie des méthodes géométriques utilisant des mesures de distance.
k-NN est un algorithme classique d'apprentissage qui a été longtemps à la base des algorithmes de catégorisation des documents, elle a été employée avec succès dans de domaine de classification et a engendré toute une famille de classifieurs connus sous le nom de classifieurs paresseux (lazy learns). Dans ces systèmes, le seul traitement effectué au cours de la phase d'apprentissage est la mémorisation des exemples sous une forme optimale de

façon à pouvoir les extraire ensuite rapidement. Chaque texte est représenté dans un espace vectoriel, dont chacun des axes représente un descripteur. Tous les calculs sont reportés à la phase de classification (d'où le terme de paresseux).
k-NN est un algorithme de catégorisation dans lequel les classes ne sont pas représentées sous forme de texte "prototype" (profil de catégorie).
La partie classification est en contrepartie plus coûteuse en temps : Le classifieur calcule la similarité du nouveau texte à catégoriser avec l'ensemble des autres exemples du corpus d'apprentissage, dont les catégories sont déjà connues, puis il sélectionne les k documents les plus proches du document à classer. Ensuite, pour affecter la catégorie, les relations entre ces k documents et les catégories sont évaluées et un score est calculé par catégorie afin d'évaluer la pertinence de la catégorie au document. La catégorie (ou les catégories) ayant le score le plus élevé (celle qui contient le plus de textes voisins) est affectée au document (Yang, 2001).
Voici son algorithme général :

Paramètres : le nombre k de voisins
Données : un ensemble d'exemples classés (document, classe)

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

Sortie : la classe de D est C

Traduit en langage naturel: on va affecter au nouvel élément à traiter la classe la plus représentée dans ses k plus proches voisins. On cherche les « k plus proches voisins » dans les documents déjà répertoriés du nouveau document à classer. L'assignation du document peut être à plusieurs classes à la fois. Ainsi, chaque classe suffisamment représentée dépassant un certain seuil sera associé au document. Le seuil minimum acceptable pour qu'un document se voit assigner à une classe (en cas de multi-catégories) se calcul à partir des documents d'entraînements et représente une des difficultés de la mise en œuvre de cet algorithme.
Pour que cet algorithme soit efficace, il faut utiliser une bonne mesure de similarité entre les documents, notamment afin que les attributs non discriminant ne soient pas pris en compte et que ceux qui sont discriminants le soient. Le modèle vectoriel a l'avantage de fournir une représentation dans laquelle les termes peu informatifs ont un poids faible.
La similarité en cosinus convient donc parfaitement et les k-PPV sont bien adaptés à ce modèle.
La phase d'apprentissage est souvent effectuée hors-ligne avant la mise en exploitation du logiciel. Il est donc acceptable de laisser l'algorithme tourner pendant un temps important. En revanche, pendant la phase de classification, les calculs doivent être très rapides (pour traiter un grand nombre de documents) voire quasiment instantanés si un utilisateur attend les résultats. L'algorithme des k-PPV à l'inconvénient de déporter tous les calculs qui pourraient être fait pendant la phase d'apprentissage à la phase d'exploitation. Ce problème est d'autant plus important dans le cas où les documents sont représentés dans un espace de très grandes dimensions (le calcul de la similarité prend dans ce cas un temps non négligeable). Ils sont donc peu efficaces dans le cas du texte. En revanche, comme chaque document contient peu de termes, il est possible d'utiliser la méthode des index inversé (au lieu de stocker pour chaque document la liste des termes présents, on conserve pour chaque terme la liste des documents qui le contiennent), mais cela reste parfois insuffisant.
Dans les k-PPV, il n'existe pas de solution efficace pour choisir une valeur pour le paramètre k. ce choix relève d'un compromis. Si k est trop petit, le nombre d'exemples qui prennent part à la décision est faible et les exemples bruités peuvent alors jouer un rôle néfaste important. Si k est trop grand, l'hypothèse de localité n'est plus respectée car des exemples très éloignés du

document sont sélectionnés pour participer au vote. Dans le domaine textuel, la valeur optimale pour k dépend du corpus et de l'application. D'après les travaux réalisés jusqu'à présent, la meilleure classification est obtenue avec une valeur de k comprise entre 10 et 50.

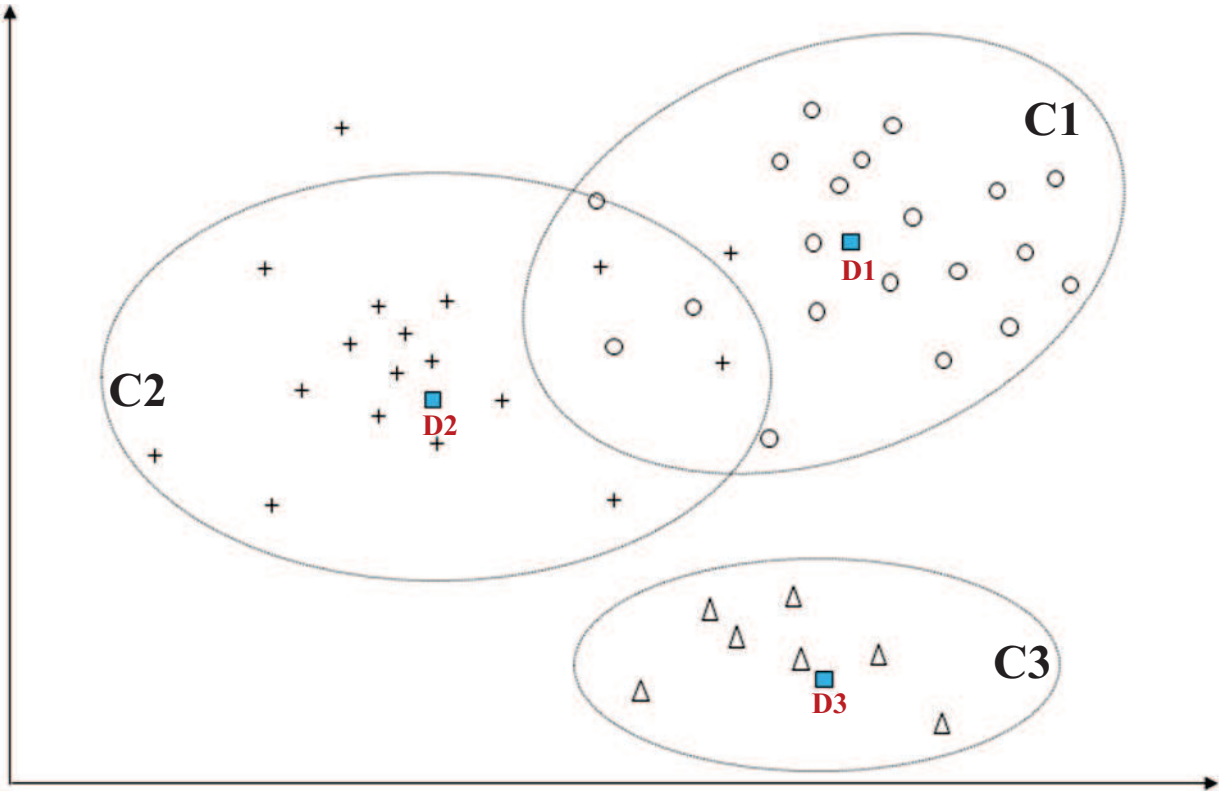


Figure 3.3 : Exemple de la méthode k-PPV
Les trois nouveaux documents D1, D2 et D3 sont associés respectivement aux classes des k plus proches voisins C1, C2 et C3.

Il existe néanmoins plusieurs versions de k-PPV. Ces versions peuvent diverger, principalement, sur la politique de classification. Pour choisir la classe la plus proche, on peut procéder de plusieurs manières : Certaines méthodes choisiront comme classe candidate pour un texte, la classe majoritairement retrouvée sur ses k plus proches documents voisins. D'autres pondèreront les classes candidates en fonction de la position du texte dans le classement (des k plus proches voisins), avant de réaliser notre choix final. Dans (Yang, 1999) nous pouvons trouver une description plus détaillée des différentes versions de k-PPV.

3.2.4.2- Critiques de l'approche

Selon les tests sur l'ensemble des techniques statistiques pratiqués par (Yang, 1999) dans le domaine de classification, kPPV est actuellement une des méthodes les plus efficaces en qualité de classement en raison de sa performance très stable puisque elle donne toujours de bons résultats tout en étant simple et facile à implémenter.
Néanmoins, Un ensemble de limitations des plus proches voisins découle, voici quelques limites qui affectent la catégorisation de textes :

1. Le principal problème que présente cette méthode est sa prédiction longue : du point de vue temps d'exécution, les algorithmes de type plus proches voisins sont longs en phase de classification, puisqu'on sauvegarde et on fait appel à tous les exemples du corpus d'apprentissage (le document à classer est comparé par calcul de similarité à l'ensemble des documents de la base). Et pour chaque nouvel élément à traiter, il faut entièrement tout recalculer, ce qui implique un coût de traitement important.

2. Par ailleurs, kPPV est sensible aux documents (exemples) négatifs dans la base d'apprentissage.
3. Fait partie des algorithmes locaux, car au moment de prise de décision on n'utilisent qu'une faible partie des exemples à chaque classification.(K plus proche voisins)
4. Les kPPV sont sensibles au choix de la fonction de similarité de l'algorithme.
5. Le principe du kPPV s'appuie sur la proximité entre les individus, néanmoins l'utilisation de ce seul critère parait insuffisante. En effet du point de vue cognitif lorsqu'un terme x a pour plus proche voisin y, on s'attend normalement que y est réciproquement a pour plus proche voisin x mais malheureusement ce n'est pas le cas puisque dans notre exmple, z qui est le plus proche voisin de y.

Figure 3.4 : Principe des kPPV

3.2.5- Arbres de décision

3.2.5.1- Présentation de l'approche

Les arbres de décision, utilisés dés les années 60 en statistique, sont des outils supervisés. Ils sont devenus, depuis une vingtaine d'années des outils très populaires pour générer des règles de classification et plus généralement des règles de prédictions. On parle ainsi d'arbre de classification lorsque la variable à prédire est catégorielle et que ses valeurs représentent donc des classes.
Les arbres de décision ont été utilisés pour permettre la catégorisation des documents dans un certain nombre de classes prédéfinies. L'apprentissage des probabilités d'attribution d'un texte à une classe est réalisé sur des textes étiquetés manuellement. L'arbre de décision généré, sur l'ensemble des documents du corpus d'apprentissage, permet de décider à quelle classe appartient chaque nouveau document du corpus de test. Chaque feuille de l'arbre contient la probabilité d'appartenance à l'une ou l'autre des classes. Suivant les réponses aux questions posées au document à classer, celui-ci est « dirigé » vers telle ou telle feuille de l'arbre. Le document est alors attribué à la classe de plus forte probabilité.(Bellot, 2000)

Selon Gilbert Ritschard, Simon Marcellin et Djamel A. Zighed dans (Ritschard & all, 2009), un partitionnement récursif des données est le principe de base sur lequel s'appuie la méthode pour avoir des ensembles cohérents par rapport à la variable à prédire. Un enchaînement hiérarchique de règles est généré. Chaque feuille (nœud terminal) de l'arbre caractérise une règle, de type « Si condition Alors résultat », dont la prémisse est définie par les conditions d'embranchement le long du chemin menant du nœud initial à la feuille, la conclusion de la règle correspondant à la classe assignée à la feuille. L'ensemble de toutes ces règles représente le modèle de prédiction.
En effet, Pour construire l'arbre de décision, il faut chercher lequel des attributs (termes) qu'il faut tester à chaque nœud, c'est un processus récursif. Pour décider de l'attribut qu'il faut tester à chaque étape, on utilise un calcul statistique qui détermine dans quelle mesure cet attribut sépare bien les exemples Oui/Non. On crée alors un nœud contenant ce test, et on crée autant de descendants que de valeurs possibles pour ce test.
Ainsi, si on teste la présence d'un mot, les valeurs possibles sont « Présent/Absent ». A chaque fois, on aura donc deux descendants pour chaque nœud.
On répète ce processus en associant à chaque descendant le reste des exemples qui satisfont le test du prédécesseur.

La phase d'apprentissage consiste à la construction de l'arbre de décision avec sa racine, ses nœuds et toutes ses feuilles représentant l'ensemble des règles.
En ce qui concerne la phase de classification : pour classer un nouveau document, il suffit de simplement de parcourir l'arbre selon les réponses aux différents tests pour le document.
L'évaluation de la qualité de l'arbre se fonde le plus souvent sur le taux d'erreur de classification. Ce taux de cas mal classés par les règles, calculé sur les données d'apprentissage ou des données test est évidemment pertinent comme choix des conclusions et validation des règles issues d'arbres (Pisetta & all, 2007).

3.2.5.2- Architecture d'un arbre de décision

Un arbre de décision a pour but d'expliciter une catégorisation dans le cas où les textes classés sont décrits par un ensemble de termes qui représentent les propriétés caractéristiques des textes. Un arbre de décision est un arbre dont les nœuds sont :

- Soit des feuilles contenant des objets appartenant tous à une même catégorie. Les feuilles représentent donc les classes d'une même catégorie C (sous ensembles de C).
- Soit des nœuds de décision qui partitionnent les données suivant plusieurs sous ensembles, chaque sous-ensemble correspondant à un résultat d'une fonction de test, cette fonction caractérisant le nœud de décision. (Mari & Napoli, 1996)

3.2.5.3- Algorithme de construction

L'algorithme d'apprentissage de construction des arbres de décision prend en entrée un ensemble de textes déjà classés sous forme de couples (texte, classe) et présente en sortie un arbre de décision. L'algorithme procède de façon descendante : il démarre du nœud initial (racine), et récursivement, génère les nœuds fils jusqu'aux feuilles.
L'algorithme de construction d'un arbre de décision est basé sur l'hypothèse diviser et conquérir (divide and conquer). Il s'agit de partitionner un corpus C d'apprentissage en plusieurs sous ensembles C_1, C_2,..,C_N, où chaque C_i correspond à un résultat du test utilisé. La procédure est répétée récursivement sur tous les sous-ensembles jusqu'à ce que tous les nœuds terminaux ne contiennent que des objets d'une même classe. (Mari & Napoli, 1996).

D'aprés (Breiman et all, 1984) nous devons définir :

- Un ensemble de questions à poser aux individus (dans notre cas c'est les textes) ; nous choisissons des questions telles que chaque individu peut y répondre par l'affirmative ou la négative ; suivant sa réponse, l'individu est transféré dans le nœud fils correspondant à la réponse « oui » ou dans le nœud fils correspondant à la réponse « non » (Une question Q de la forme : « *les individus contiennent-ils le terme* x *?* » est posée à chaque individu du nœud *S*. Cette question permet de subdiviser le nœud *S* en deux nœuds fils $S_{(OUI)}$ et $S_{(NON)}$ qui comprennent respectivement les individus de *S* qui contiennent et qui ne contiennent pas le terme *x*.
- Une règle pour déterminer les questions à poser aux individus ;
- Un critère d'arrêt déterminant l'ensemble des feuilles de l'arbre.

Le critère de sélection d'une question est souvent fonction du gain en entropie observé avant et après l'affectation des individus dans de nouvelles partitions suivant la question considérée (Quinlan, 1986), (Quinlan, 1993). L'entropie est utilisée comme critère de sélection des questions pour subdiviser les nœuds de l'arbre. Les individus à répartir dans les feuilles de l'arbre sont, ici, des documents.

Ainsi pour induire l'arbre, il faut rechercher à chaque nœud l'éclatement qui produirait le meilleur gain en termes pour la sélection des règles, pour cela deux mesures statistiques sont utilisées : l'entropie et le gain d'information.

3.2.5.4- L'entropie et le gain d'information

L'entropie permet de mesurer l'homogénéité des exemples. Si l'entropie vaut 0, alors tous les exemples appartiennent à la même classe (par exemple Oui). Si l'entropie vaut 1, alors c'est qu'il y a autant d'exemples positifs que d'exemples négatifs.
L'entropie est définie par :

Entropie(S) = -p/N log2 p/N - n/N log2 n/N
Où S est l'ensemble des exemples (**S**amples), de taille **N**,
p (exemples **p**ositifs) est le nombre d'exemples classés Oui,
et n (exemples **n**égatifs) est le nombre d'exemples classés Non dans l'ensemble **S** des **N** exemples.
N.B : on admet 0 log 0 = 0.

Le gain d'information peut être traduit par la quantité d'information apportée à la classe par le terme choisi. Une valeur importante du gain d'information apporté par le terme implique une quantité d'informations importante pour classer les documents avec cet attribut qui est nécessaire.
La mesure appelée gain d'information calcule la réduction attendue de l'entropie des exemples si un attribut particulier est utilisé. L'algorithme de classification calcule cette valeur pour chaque attribut et choisit alors celui qui réduit le plus l'entropie, c'est à dire celui qui permettra le plus nettement possible de séparer les exemples qui restent.

Information Gain(S,A) = Entropie(S) - Somme sur les valeurs de A de (|Sv| * Entropie(Sv) / |S|)
Où S = l'ensemble des exemples,
A est l'attribut utilisé,
Sv = le sous-ensemble de S dont l'attribut a la valeur v.

3.2.5.5- Évaluation des arbres de décision

Les arbres de décisions ont l'avantage d'offrir à la fois un bon pouvoir explicatif (de meilleurs performances de classification), généralement du même niveau des meilleurs techniques de classification et un bon pouvoir descriptif, dans la mesure où on peut interpréter l'arbre comme un ensemble de règles de décision. Pour cela, il suffit de partir de la racine de l'arbre au plus haut et de former tous les chemins possibles pour obtenir une feuille d'une classe.
Nous obtenons donc des règles logiques, sous forme normale disjonctive (SI .. ALORS ..SINON...).

- Si (x=1) et (y=0) alors C_1
- Si (x=1) et (y=1) alors C_2
- Si ((x=0) et (y=1)) ou ((x=0) et (z=1)) alors C_3
  etc...

Ainsi le modèle aura la forme de fonction de décision la plus interprétable par des humains.
D.A. Zighed, et R. Rakotomalala dans (Zighed & Rakotomalala, 2000) évoquent les principales caractéristiques des arbres de décision à savoir : Performance du classifieur du point de vue rapidité et qualité des résultats, tous les types de données sont supportés, capable d'être développés dans tous les environnements de fouille de données et sans oublier bien sûr leur pouvoir descriptif (Lisibilité du résultat)**.**

Cependant, les arbres de décision ont comme principal inconvénient d'être souvent grands, c'est à dire de posséder beaucoup de feuilles puisque on crée un nœud de décision pour chaque résultat possible d'un test. Lorsque l'arbre est très développé, ça devient illisible et l'on perd ainsi en pouvoir explicatif. Il faut alors recourir à des méthodes automatiques d'extraction de règles à partir de l'arbre (Mari & Napoli, 1996).
Ajouté à d'autres inconvénients qui peuvent être soulignés comme leur sensibilité au nombre de classes important qui peut dégrader les résultats, et la non-évolutivité dans le temps, puisque un changement dans les exemples d'entrées exige le relancement de la phase d'apprentissage sur le tout le corpus (anciens exemples et nouveaux exemples). Sans oublier de souligner l'inconvénient causé par le fait que les règles ou arbres de décision ne peuvent prendre que des décisions binaires (Moutarde, 2008).

Les arbres de décision sont des instruments privilégies d'exploration de données, que ce soit en termes de classement ou de description. Dans ce dernier cas, plutôt que de prédire la valeur de la réponse, il s'agit de repérer les profils typiques des individus appartenant à chacune des classes de la variable à prédire. Nous inversons en quelque sorte le problème en cherchant à caractériser les profils propres à la classe, plutôt que la classe à partir du profil (Pisetta & all, 2007). Notons enfin, que le domaine d'utilisation des arbres de décision dépasse le cadre de l'apprentissage supervisé puisque il y a des applications de classification par arbres de décision non supervisés exploités dans le cadre de la recherche documentaire (Bellot, 2000).

3.2.6- Les approches neuronales

3.2.6.1- Présentation de l'approche

Les réseaux de neurones artificiels sont habituellement utilisés pour des tâches de classification. Par analogie avec la biologie, ces unités sont appelées neurones formels.
Un neurone formel est caractérisé par :

- Le type des entrées et des sorties ;
- Une fonction d'entrée ;
- Une fonction de sortie.

Le connexionnisme peut être défini comme le calcul distribué d'unités simples, regroupées en réseau. Un réseau de neurone est un ensemble d'éléments ou unités extrêmement simples (neurones) se comportant comme des fonctions de seuil, suivant une certaine architecture ; Chaque neurone prend en entrée une combinaison des signaux de sortie de plusieurs autres neurones, affectés de coefficients (les poids) ;
L'apprentissage s'effectue sous le contrôle des associations prédéfinies entre documents (entrées du réseau) et classes (sorties du réseau) qui fixent le comportement du réseau souhaité. La différence entre le comportement réel et désiré est une erreur qui sera à la base de l'apprentissage sous la forme d'une fonction de coût ou d'un signal d'erreur. Dans ce cas, l'apprentissage s'effectue en réajustant chaque fois les poids Wi.
Donc les algorithmes d'apprentissage permettent de calculer automatiquement les poids qui correspondent en réalité à des paramètres permettant de définir les frontières des classes.
Une structuration en couches effectue en cascade différents traitements sur un ensemble de données. Ces données sont présentées sur une couche terminale, appelée couche d'entrée; elles sont ensuite traitées par un nombre variable de couches intermédiaires ou couches cachées. Le résultat est exposé sur l'autre couche terminale, la couche de sortie.

3.2.6.2- Le perceptron

Historiquement, le perceptron est le premier réseau efficace qui a été proposé et étudié en détail. Il comprend une couche d'entrée et une couche de sortie. Les connexions entre ces deux couches sont bidirectionnelles et variables et. Les neurones ont des sorties binaires. Les neurones de la couche de sortie réalisent une fonction à seuil. C'est la couche de sortie qui assure l'opération de classification (Leray, 2006).
Les connexions sont ajustées par un apprentissage supervisé selon le principe de correction d'erreurs : si un neurone de la couche de sortie n'est pas dans l'état désiré, toutes les connexions du neurone avec ceux de la couche d'entrée sont augmentées ou diminuées selon le type d'action que le neurone de la couche d'entrée effectuait sur le neurone de sortie en question, dans le but de favoriser une réponse connue au préalable.

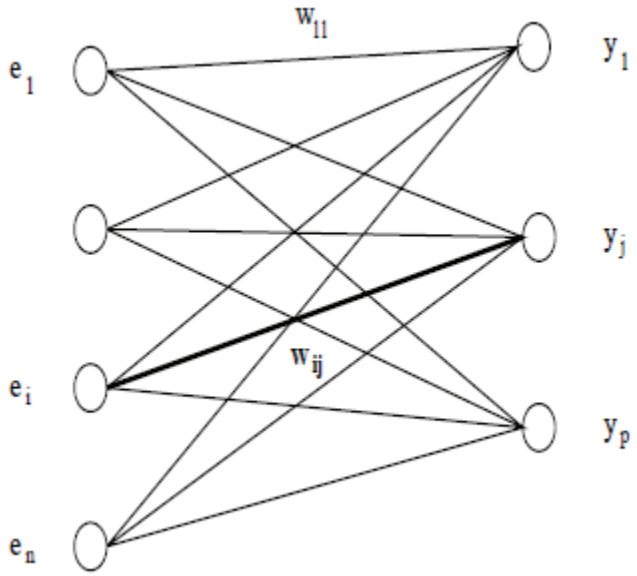


Figure 3.5 : Perceptron monocouche

D'autres réseaux de type perceptron multicouche sont également très utilisés pour les tâches de classification.

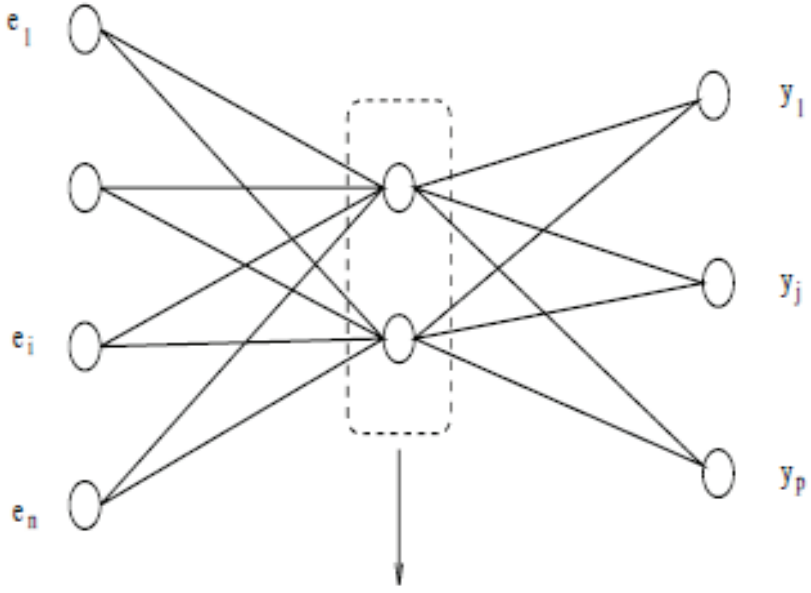


Figure 3.6 : Perceptron multicouche

3.2.6.3- Autres réseaux à couches

Les réseaux à fonction à base radiale (Radial Basis Function ou RBF) sont des réseaux à trois couches. la couche d'entrée qui renvoie les exemples d'entrées sans changement , la couche cachée RBF qui contient les neurones RBF est en ensemble de neurones réalisant une convolution avec une fonction gaussienne, et une troisième couche de sortie qui réalise simplement une combinaison linéaire des neurones de la couche cachée.
Chaque couche est complètement connectée à la suivante.

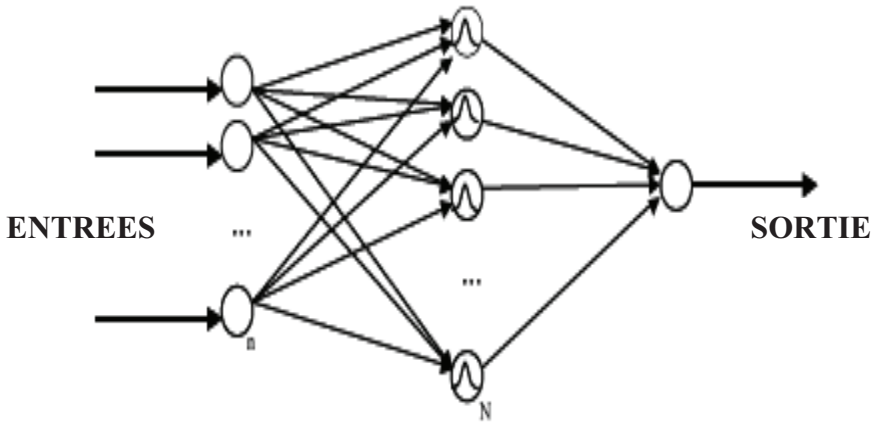


Figure 3.7: Radial Basis Function (RBF)

Les facteurs principaux d'un réseau RBF à ajuster c'est : Le nombre de neurones de la couche cachée RBF, les poids des liaisons des neurones RBF et les neurones de sortie, la position des centres des gaussiennes correspondantes aux neurones RBF, la largeur de ces gaussiennes.

3.2.6.4- Classification à base des réseaux de neurones

Plusieurs travaux, dans le domaine de classification automatique de textes, à base des approches neuronales ont été évoqués.
Une approche fondée sur les réseaux de neurones a été proposée dans la thèse de (Wiener, 1993) dont les résultats ont été repris dans (Wiener & all, 1995). Deux architectures neuronales sont proposées et testées sur le corpus Reuters.
La première architecture est un perceptron multi-couche avec une couche de neurones cachés et un neurone de sortie; un réseau de neurones différent est construit pour chaque classe.
Pour la deuxième architecture, les classes du corpus Reuters sont regroupées en cinq grands ensembles (*agriculture, energy, foreign exchange, government, metals*). Un réseau est ensuite utilisé pour déterminer à quel ensemble appartient un document, puis cinq réseaux différents sont construits pour déterminer, à l'intérieur d'un ensemble, la classe exacte du texte. Cette architecture a l'avantage de permettre à chacun des cinq réseaux d'être spécialisé et d'utiliser une représentation particulièrement adaptée pour distinguer des classes proches.
Cette deuxième architecture améliore les résultats, mais elle nécessite un découpage manuel des classes pour déterminer les ensembles et n'est réalisable que sur un corpus pour lequel le nombre de classes est connu à l'avance et n'évolue pas.
Dans l'ensemble de cette étude, le surajustement est limité en considérant un terme de pénalisation dans la fonction de coût conjointement avec la méthode de l'arrêt prématuré ;

Cette étude montre notamment que, même lorsque le nombre de neurones cachés est nul, le modèle peut être surajusté. La mise en œuvre d'une procédure d'arrêt prématuré limite ce surajustement et améliore significativement les résultats.

3.2.6.5- Critiques de l'approche

A partir des expériences à bases des réseaux de neurones, il est possible de tirer plusieurs conclusions :

- D'après les études précédentes, l'ajout de neurones cachés n'améliore pas vraiment les performances du classifieur.
- Il est nécessaire de se protéger du surajustement, même pour le modèle sans neurone caché, par une méthode de régularisation qui peut prendre la forme d'un terme de pénalisation dans la fonction de coût ou d'une procédure d'arrêt prématuré.
- La combinatoire des combinaisons linéaires et des fonctions de seuil appliquées récursivement donne un classificateur complexe.
- Et enfin une caractéristique des réseaux connexionnistes qui représente leur inconvénient principal c'est leur très faible pouvoir descriptif puisque un réseau de neurone est une sorte de boîte noire non interprétable par ses utilisateurs.

3.2.7- Naïve Bayes

L'algorithme Naïve Bayes (NB), est une autre méthode bien utilisée en apprentissage, elle est également employée dans la classification automatique de textes. NB est très connu pour son efficacité, sa facilité d'implémentation et ses résultats considérables. Pour toutes ces raisons et bien d'autres avantages, nous l'avons adopté dans nos travaux de recherches.
Une description détaillée sera accordée ultérieurement dans la chapitre 6 (section 6.3.2).

3.2.7.1- Description de l'approche

L'algorithme Naïve Bayes (NB) est une méthode basée sur le modèle probabiliste, il vise à estimer la probabilité conditionnelle d'une catégorie sachant un document et affecte au document la (ou les) catégorie(s) la (les) plus probable(s). Ainsi ce classifieur va donc tenter d'estimer la probabilité qu'un document d fasse partie de la classe c.
Un modèle probabiliste de classification est un modèle qui permet le calcul pour chaque classe $c \in C$ et chaque document $d \in D$ la probabilité $\mathbb{P}(c / d)$ que le document soit assigné à cette classe.
Le nom Naïve Bayes découle du fait que l'algorithme utilise le théorème de Bayes sans toutefois prendre en compte les dépendances existantes entre les variables (Dans notre cas les termes c'est des mots ou n-grammes, etc..); de ce fait, ses suppositions sont dites naïves. Donc la partie naïve de ce modèle est l'hypothèse d'indépendance des termes, c'est à dire que la probabilité conditionnelle d'un terme sachant une catégorie est supposée indépendante de cette probabilité pour les autres termes.
Cette hypothèse fait que la catégorisation par NB est plus efficace que la complexité exponentielle des approches bayesiennes non naïves qui utilisent des combinaisons de mots comme prédicteurs, (McCallum & all, 1998) (Yang & Liu, 1999), (Hertzmann, 2004), (Chethan & all, 2007).

Dans un contexte probabiliste bayésien général, le choix de la classe d'une nouvelle instance ***A*** est réalisé par la règle du maximum a posteriori (MAP). L'estimation des probabilités est réalisée en général par maximisation de la vraisemblance conditionnelle de l'ensemble d'apprentissage par la formule $\mathbb{P}(\boldsymbol{B})*\mathbb{P}(\boldsymbol{A/B})$ qui fait apparaître deux termes : la vraisemblance

de l'observation ***A*** conditionnellement à ***B*** et la probabilité a priori de ***B***. Dans un cadre supervisé, ces deux distributions sont estimées à partir des exemples d'apprentissage.
Si l'on considère un problème à deux classes c_1 et c_2, L'apprentissage consiste à déterminer la distribution de probabilité connaissant les données d'apprentissage : une probabilité fixée a priori, et, une fois que les données d'apprentissage ont été observées, cette probabilité a priori est transformée en probabilité a posteriori grâce au théorème de Bayes.
Le théorème de Bayes permet donc de calculer les probabilités a posteriori connaissant les distributions des observations a priori.
Notons que la probabilité a posteriori est la probabilité conditionnelle $P(c_j/d_i)$ de la classe ci connaissant d_i. En d'autres termes, quand on a un exemple d'entrée d_i (dans ce cas d_i est un texte), $P(c_j/d_i)$ représente la probabilité que d_i appartienne à la classe c_j.
Pour la probabilité a priori c'est la probabilité conditionnelle $P(d_i/c_j)$. Cela représente la probabilité d'avoir d_i comme entrée quand on sait que l'on est dans la classe c_j.
La classification d'un exemple s'obtient alors par estimation de $P(c_j/d_i)$ *;* la probabilité connaissant l'exemple d_i que celui-ci fasse partie de la classe c_j. Le choix optimal (pour minimiser le taux d'erreur) est de mettre l'exemple dans la classe qui à la plus forte probabilité a posteriori. Comme cette probabilité n'est pas connue, il faut l'estimer à partir des données contenues dans le corpus d'apprentissage.
Fondés sur l'idée qu'on peut estimer la probabilité qu'un document appartient à une classe en connaissant la probabilité qu'une classe correspond à ce document.
La formule de bayes permet « d'inverser » la probabilité conditionnelle :

$$P(c_j|d_i) = \frac{P(c_j) \times P(d_i|c_j)}{P(d_i)}$$

Comme le but est de discriminer les différentes classes (il suffit d'ordonner les $P(c_j/d_i)$ pour toutes les classes ; il est inutile d'obtenir la valeur exacte), on peut alors supprimer le terme $P(d_i)$ qui est le même pour toutes les classes. $P(c_j)$ est la probabilité a priori qui est le plus couramment estimée par le pourcentage d'exemples (Dans ce cas pourcentage de documents) appartenant à la classe c_j dans le corpus d'apprentissage.

$$P(c_j) = \frac{N(c_j)}{N}$$

Quant à la classification, l'estimation $P(d_i / c_j)$ est calculée à partir des deux formules suivant le modèle utilisé en remplaçant les probabilités par leur estimateur et **la classe c_j ayant la probabilité la plus élevée est choisie.**

3.2.7.2- 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. (Hilali, 2009)

Le fonctionnement de naïve bayes est relativement similaire à celui de rocchio. Chaque classe est décrite par un profil qui gère un coefficient par terme (P_{jk} pour Rocchio, $P(t_k/c_j)$ pour Naïve Bayes). Tous ces coefficients sont ensuite regroupés pour former une valeur de pertinence (un degré de similarité pour Rocchio, une probabilité pour Naïve Bayes).
Notons aussi que ce modèle ne prend pas en compte l'ordre des mots dans la modélisation des documents. Intuitivement, cela apparaît comme une hypothèse très forte et peu réaliste. Si pour la tâche de classification, cette hypothèse peut paraître représenter un compromis entre la puissance du modèle et sa simplicité, elle rend impossible différentes tâches comme le résumé automatique, l'extraction de passage, etc. Dans les années 80, (Jelinek & Mercer, 1980) ont proposé d'abandonner cette hypothèse d'indépendance (ou de dépendance à l'ordre 0) afin de prendre en compte localement l'ordre des mots. Les modèles développés alors reposent sur une hypothèse d'indépendance à un ordre supérieur. Cette hypothèse s'écrit :
$P(t_i/t_{i-1},....,t_1, c_j) = P(t_i/t_{i-1},....,t_{i-n}, c_j)$ ou n représente l'ordre de la dépendance.
Dans le cas des données textuelles, les probabilités $P(t_i/t_{i-1},....,t_{i-n}, c_j)$ correspondent à la probabilité de voir dans un document le mot t_i précédé de la séquence de mots $t_{i-1},....,t_{i-n.}$

Notons enfin, qu'un certain nombre de modifications à l'algorithme ont été proposées dans le but d'élargir son utilisation et améliorez ses performances.

3.2.8- Les méthodes mixtes et Boosting

3.2.8.1- Présentation de l'approche

Plusieurs auteurs dans leurs travaux récemment réalisés, ont proposé des modèles hybrides pour la classification de données textuelles pour combiner les résultats de classifieurs simples (et pas très bons) pour donner des classifieurs plus complexes (et bien meilleurs) :

- Procédures de vote : Plusieurs classifieurs s'entrainent sur le même corpus pour tenter de classer chacun de sa manière un même nouveau document, la classe qui a été choisit par le plus grand nombre de classifieurs va être attribuée à ce document.
- Boosting : Est une instance des classifieurs par comités dont le principe est d'associer les résultats de plusieurs classifieurs pour obtenir un résultat plus intéressant. Les différents classifieurs peuvent correspondre à différents algorithmes ou au même algorithme utilisé avec différents sous-échantillons du corpus d'apprentissage. Dans cette approche, les différents classifieurs sont appris séquentiellement, à chaque étape, le corpus d'origine est échantillonné suivant une distribution qui favorise le tirage des documents d'entrainement mal classés par le classifieur construit à l'étape précédente. Les classifieurs se focalisent ainsi de plus en plus sur les exemples difficiles à catégoriser. Lors de la phase de test, tous les classifieurs sont utilisés et un vote pondéré par les taux de performance de chaque système est effectué, comme dans les procédures de vote. Le boosting est souvent utilisé avec un algorithme peu performant mais très rapide (surnommé Weak Learner) avec lequel on construit plusieurs centaines de classifieurs. L'algorithme garantit que le taux d'erreur sur le corpus d'apprentissage peut être rendu aussi petit que l'on veut en augmentant simplement le nombre de classifieurs construits.

3.2.8.2- Evaluation de l'approche

Les expériences menées montrent que, en pratique, les performances en généralisation sont très bonnes et les risques d'avoir du sur-apprentissage est très minime. En revanche le problème d'efficacité des résultats en terme de temps peut être posé dans le cas où on opte

pour un nombre important de classifieurs pour améliorer les résultats. Comme SVM, NB, kPPV, le boosting a été adapté à la catégorisation de textes avec succès.

3.2.9- Autres méthodes

Il y a d'autres approches plus ou moins utilisées dans le cadre de classification de textes mais elles sont surtout orientées vers la recherche d'information, résumé automatique, etc...

- Comme par exemple la régression logistique :

L'entrée du classifieur logistique est une représentation des phrases en un vecteur de scores, Les paramètres sont appris en maximisant la log-vraisemblance binomiale des documents d'apprentissage. La régression logistique a déjà été utilisée avec succès en résumé automatique (Usunier & all, 2005) et a montré de bonnes performances en tant qu'algorithme de combinaison de caractéristiques.

- Ou les Modèles de Markov Cachés (MMC) :

Lorsque le nombre de classes est connu, ce dernier problème peut également être résolu en utilisant des Modèles de Markov cachés. Un modèle de Markov caché est un modèle probabiliste dont le but est de modéliser un processus séquentiel. Nous considérons ici uniquement le cas des MMCs dans des espaces discrets, ce qui correspond à l'utilisation la plus courante de ces modèles avec des données textuelles.

Pour construire des regroupements en k classes, on suppose de nouveau que chaque séquence est issue d'un parcours dans un graphe à k états (un par classe), chaque état étant caractérisé par un modèle de génération de x qui lui est propre.

On définit un MMC comme un triplet $\{A, B, \Pi\}$: L'ensemble des états possibles du processus, l'ensemble des observations possibles, une matrice de probabilités initiales.

Il est habituel de représenter les MMC sous forme graphique.

Les MMCs ont été utilisés pour différentes problématiques de la RI et du traitement du langage naturel. (Miller & all, 1999) pour la recherche documentaire et (Amini, 2001) qui a développé un modèle d'extraction d'information (professions, personnes, etc..) dans des documents textes.

- Ou encore les Réseaux bayésiens :

Les réseaux bayésiens se présentent comme un formalisme de représentation graphique des dépendances conditionnelles entre des variables aléatoires. Historiquement, ils ont été développés dans le cadre de la prise en compte de l'incertain pour la prise de décisions. Ils représentent une extension du classifieur Naïve Bayes.

Un réseau bayésien est un graphe orienté sans cycles $G = (X, U)$ où X est l'ensemble des sommets et U l'ensemble des arcs qui définissent la fonction pa(Xi). La fonction pa(Xi) renvoie pour toute variable X_i l'ensemble de ses parents.

La phase dite d'apprentissage consiste à l'estimation des différentes probabilités conditionnelles à priori à partir d'un corpus d'apprentissage. L'apprentissage de ces paramètres est souvent complexe et coûteux en terme de calculs.

La phase dite d'inférence correspond au calcul de la probabilité d'une observation quelconque. Cette inférence peut-être exacte ou estimée.

3.3- Mesures de similarité et formules pour calcul de distance

Plusieurs méthodes de classification présentées précédemment et particulièrement les méthodes géométriques s'appuient sur le principe des mesures de distance.

Les bons résultats de ces méthodes sont démontrés dans plusieurs travaux, en revanche la faiblesse des méthodes utilisant les mesures de distance apparaît dans le cas des espaces de travail importants (grand nombre de documents, par exemple dans le cas du web). Les

performances de ces techniques diminuent considérablement non pas en qualité des résultats mais en rapidité des calculs.
Dans ce contexte, il existe plusieurs variantes de distance, et divers mesures de similarités entre documents ou classes qui peuvent être utilisés, dont l'influence sur les performances d'un système de catégorisation est démontré.
Dans ce qui suit, on va évoquer les différents choix possibles pour la distance, pour ensuite présenter les mesures de similarités les plus utilisés dans les domaines de la recherche d'information et la classification.

3.3.1- Calcul de distance

3.3.1.1- 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 :

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

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

x, y, z sont des éléments de l'espace E. (Saporta, 1990)

Dans le contexte de catégorisation, ces éléments sont soit des textes soit des classes.

3.3.1.2- Variantes de distance

Une formule générale est connue pour mesurer les distances dans les espaces vectoriels c'est la grandeur de Minkowski :

$$D_k(x, y) = \sqrt{(\sum_i | xi - yi |^k)^{1/k}}$$

A partir de cette formule très générale, plusieurs distances connues en pratique sont déduites :

- Comme la distance euclidienne, dans le cas où k = 2, exprimée par :

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

- Où la distance de Manhattan, dans le cas où k = 1, formulée par :

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

- Où aussi la distance de Chebyshev, dans le cas où $k = \infty$, définie par :

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

Les résultats fournis par un classifieur peuvent varier suivant l'utilisation de telle ou telle distance, comme par exemple dans les kPPV, Radwan JALAM dans (Jalam, 2003) confirme bien que le plus proche voisin peut varier selon la distance utilisée, ainsi la distance

euclidienne favorise bien les voisins dont tous les descripteurs sont assez proches, cependant la distance de Manhattan permet de tolérer une distance importante sur l'un des descripteurs.

3.3.2- Mesures de similarité

La problématique de classification automatique de textes peut se résumer en une formalisation de la notion de similarité textuelle, soit en d'autres termes : construire un modèle mathématique capable de représenter, pour ensuite comparer, la sémantique des textes. Si cette notion de similarité sémantique est un processus souvent intuitif pour l'homme, elle résulte d'un processus complexe et encore mal compris du cerveau. Le but de la recherche sur la catégorisation est donc de trouver un algorithme permettant d'attribuer les nouveaux textes à une classe avec le plus petit taux d'erreur possible, sans toutefois associer un texte à trop de classes. Dans un tel contexte, une mesure de similarité textuelle permet d'identifier la ou les catégories les plus proches du texte à classer. S'il existe plusieurs techniques pour la représentation des textes, il en est de même pour les formules adoptées comme élément de mesure. Quatre d'entre elles se sont illustrées dans le domaine : Nombre de mots communs entre un document et une classe, Okapi, Cosinus et Kullback&Liebler. La première mesure est basique qui n'est qu'une simple comparaison entre les profils des objets (texte ou classe) pour extraire les intersections entre ses objets (Nombre de mots en commun). La seconde est une variante de la mesure Okapi testée avec succès par (Wilkinson & all, 1996) (Une description peut être trouvée dans (Bellot, 2000). Les deux autres mesures à savoir Cosinus et Kullback&Liebler sont présentées dans ce qui suit.

3.3.2.1- Cosinus

Dans les modèles vectorielles, la mesure du cosinus est la plus utilisée pour définir la similarité entre un texte et une classe (Sebastiani, 2002) en raison de la stabilité de ses résultats sur des corpus variés. La similarité de deux éléments (dans notre cas documents) qu'on veut comparer, peut alors être définie par le cosinus de l'angle séparant les vecteurs des deux éléments (Salton & McGill, 1983). Par rapport à un simple produit scalaire, cette mesure présente l'avantage de normaliser les scores de chaque objet en fonction de sa taille, elle-même pondérée par le poids des termes. Le résultat renvoyé est facilement exploitable ensuite car c'est une valeur située entre 0 et 1. La valeur 1 indiquant une similarité maximum (les deux objets sont identiques) et 0 une similarité nulle (les deux objets n'ont absolument rien en commun). Cette mesure est égale au produit scalaire divisé par les deux vecteurs sont qui sont déjà normalisés.

$$\mathrm{Cosinus}(i,j) = \frac{\sum_{w \in i \cap j} \mathrm{TFIDF}_{w,i} \times \mathrm{TFIDF}_{w,j}}{\sqrt{\left(\sum_{w \in i} \mathrm{TFIDF}_{w,i}^2\right)} \times \sqrt{\sum_{w \in j} \mathrm{TFIDF}_{w,j}^2}}$$

Avec : ***w*** un terme, ***i*** et ***j :*** les deux objets (profils documents ou classes) à comparer. $TFIDF_{w,i}$ le poids du terme ***w*** dans ***i*** et $TFIDF_{w,j}$ le poids du terme ***w*** dans ***j***.

Ce qui peut ce traduire de la façon suivante :
« ***Plus on a de termes communs et plus ces termes communs ont des pondérations fortes, plus la similarité sera proche de 1, donc forte et vice versa. »***

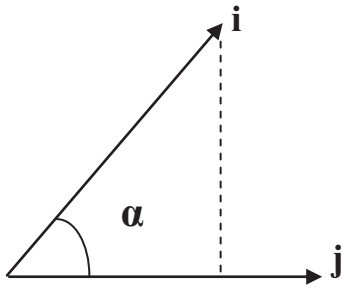


Figure 3.8 : La mesure de similarité Cosinus
i et j sont respectivement les vecteurs représentant le texte et la classe

3.3.2.2- Kullback&Liebler (la mesure d'entropie relative)

Kullback et Liebler ont étudié en 1951 une mesure statistique d'information appelée fonction de discrimination en prenant en considération deux distributions de probabilité.
La mesure Kullback&Liebler connu aussi sous le nom de l'entropie relative est une mesure qui calcule la divergence entre deux distributions de probabilité. En effet la divergence entre deux probabilités P et Q sur un ensemble fini X est définie comme suit :

$$D(P\|Q) = \sum_{x\in X} P(x) \times \log \frac{P(x)}{Q(x)}$$

Il faut noter que cette divergence n'est pas symétrique ($D(P\|Q) \neq D(Q\|P)$).
Donc la divergence symétrique Kullback&Liebler est définie comme suit :

$$D(P\|Q) = \sum_{x\in X} \left(\left(P(x) - Q(x)\right) \times \log \frac{P(x)}{Q(x)} \right)$$

Tout de même une version symétrisée de la distance de Kullback-leibler pour faciliter la comparaison avec les autres mesures de similarité existe dans (Haddad, 2002)
Cette mesure de similarité a été utilisée dans différents domaines tel que le traitement des langages naturels (Carpinto & all, 2001), la recherche d'information pour l'identification des thèmes (Bigi & all, 2000) ainsi que la reconnaissance de la parole (Dagan, 1999) et (Dagan & all, 2005).
Dans le contexte de catégorisation de textes, cette mesure est utilisée pour calculer la distance entre le profil du texte et le profil de la classe comme suit :

$$KLD(c_i, d_j) = \sum \left\{ \left(P(t_k, c_i) - P(t_k, d_j)\right) \times \log \left(\frac{P(t_k, c_i)}{P(t_k, d_j)} \right) \right\}$$

Dans son calcul, quatre cas sont pris en considération :

- ($t_k \in d_j$) et ($t_k \in c_i$) i.e : le terme t_k apparaît dans le profil de la catégorie et dans le profil du document.
- ($t_k \in d_j$) et ($t_k \notin c_i$) i.e : le terme t_k apparaît dans le profil du document mais n'apparaît pas dans le profil de la catégorie.
- ($t_k \notin d_j$) et ($t_k \in c_i$) i.e : le terme t_k n'apparaît pas dans le profil du document mais apparaît dans le profil de la catégorie.

- ($t_k \notin d_j$) et ($t_k \notin c_i$) i.e : le terme t_k n'apparaît pas dans le profil du document et n'apparaît pas non plus dans le profil de la catégorie.

La probabilité d'apparition d'un terme t_k dans un profil de catégorie est définie comme suit :

$$\begin{cases} P(t_k, c_i) = \dfrac{tf(t_k, c_i)}{\sum_{x \in c_i} tf(t_x, c_i)} & \textbf{\textit{Si le terme } t_k \textit{ apparaît dans le profil de } c_i.} \\ \varepsilon \text{ (epsilon)} & \textbf{\textit{Sinon.}} \end{cases}$$

De même, La probabilité d'apparition d'un terme t_k dans le profil du document est définie comme suit :

$$\begin{cases} P(t_k, d_j) = \dfrac{tf(t_k, d_j)}{\sum_{x \in c_i} tf(t_x, d_j)} & \textbf{\textit{Si le terme } t_k \textit{ apparaît dans le profil de } d_j.} \\ \varepsilon \text{ (epsilon)} & \textbf{\textit{Sinon.}} \end{cases}$$

Où :

- $P(t_k, c_i)$ est la probabilité conditionnelle d'un terme dans une catégorie.
- ε est une probabilité accordée aux termes qui n'apparaissent ni dans le document, ni dans la catégorie ;

Pour chaque catégorie, il est nécessaire de normaliser la distance, parce que les catégories sont de tailles différentes. Par conséquent, nous utiliserons la distance Kullback&Liebler normalisée :

$$KLD^*(c_i, d_j) = \frac{KLD(c_i, d_j)}{KLD(c_i, 0)}$$

Où : $KLD(c_i, 0)$ représente la distance entre la catégorie et un document vide.
Finalement, après avoir calculer la distance $KLD^*(c_i, d_j)$ entre le document à catégoriser et toutes les catégories, le document sera assigné à la catégorie la plus proche :

$$H_{KLD^*}(d_j) = \arg\min_{c_{i \in C}} KLD^*(c_i, d_j)$$

3.3.2.3- Synthèse sur les mesures de similarité

Pour la mesure de similarité entre documents, le modèle vectoriel préconise généralement la similarité dite du cosinus correspondant au cosinus de l'angle entre deux vecteurs, plusieurs études ont constaté la nette supériorité de cosinus par rapport aux autres mesures dans la majorité des cas. Dans (Jones & Furnas, 1987) une analyse géométrique des différentes mesures de similarité existantes est effectuée et elle conclue que le cosinus semble être le choix le plus judicieux car le moins sensible aux cas particuliers qui génèrent des comportements illogiques avec les autres mesures. Néamoins un dysfonctionnement de cosinus dans certains cas a déjà été constaté par (Wilkinson & all, 1996), qui proposent de combiner différentes mesures pour améliorer les résultats. En effet Wilkinson propose de combiner linéairement les mesures Cosinus, Okapi et NbMotsCommuns mais en obtenant seulement de légères améliorations. Comme nous pouvons citer aussi un autre élément de mesure appelé Coefficient de Cohérence recommandé par Salton, ce dernier a proposé un procédé d'extraction limité aux expressions de deux mots et se base sur la cooccurrence des termes utilisant un coefficient de cohérence qui représente la proportion des cas de cooccurrence de deux termes.

3.4- Conclusion

La phase de catégorisation de textes et le choix de technique d'apprentissage (Chapitre 3) c'est le cœur du processus de classification automatique de textes, elle est située entre une phase primordiale, de préparation des documents et catégories à l'informatisation (codage des documents), pour le bon fonctionnement du processus (Chapitre2), et une autre phase d'évaluation du ou des classifieurs utilisés aussi importante pour l'amélioration des performances du système (Chapitre 4).
Beaucoup d'approches différentes ont été utilisées pour la catégorisation de textes. Une des questions récurrentes est : quelle est la meilleure méthode pour la catégorisation de textes ?
Il existe, en pratique, plusieurs méthodologies, pour tenter de répondre à cette question, qui vont être évoqués dans le chapitre suivant, mais habituellement plusieurs paramètres peuvent influencer le choix de la technique de classification qui est relatif aux attentes des utilisateurs du système. Si l'efficacité est une priorité on peut privilégier un système rapide simple avec des résultats moins performants, et si l'exactitude des résultats et minimiser les erreurs est plus important que l'aspect temps de classification on peut pencher vers d'autres méthodes plus complexes et plus performantes, par contre si on veut avoir un système avec un pouvoir descriptif important qui peut être interprété intuitivement par son utilisateur, alors on va favoriser des modèles compréhensibles tels que les arbres de décision.
Finalement on peut dire qu'il ya des classifieurs plus performants que d'autres mais ce qui est sûr qu'il n'y a pas de classifieur parfait.

Chapitre 4

Evaluation des classifieurs

Table des matières

4.1- Introduction

Différentes approches décrites dans le chapitre 3, ont été utilisées pour la catégorisation de textes offrant ainsi, aux développeurs dans le domaine plusieurs issues, qui amène à poser une question très récurrente sur le choix du meilleur algorithme pour la classification automatique de textes.
Pour pouvoir répondre à cette question, il faut bien disposer de critères et tests utiles pour mesurer et évaluer les performances d'un classifieur pour pouvoir les comparer par la suite, afin, éventuellement, opter pour un classifieur ou un autre.

Mais qu'est ce que l'***évaluation*** ? L'évaluation consiste à mesurer la différence entre un résultat attendu et un résultat obtenu. Aucune métrique n'est associée, mais en général, on utilise des indicateurs compris entre 0 et 1 pour en faciliter l'interprétation. (Nakache & Metais, 2005).

Néanmoins, l'évaluation des performances d'un système de classification n'est pas toujours triviale et elle dépend de l'utilisation finale de ce système, certains éléments sont très subjectifs et difficilement automatisables.
Notons qu'il n'est pas facile de juger un système de catégorisation de textes s'il est performant ou moins performant qu'un autre. Plusieurs facteurs entrent en jeu qui rend cette évaluation relative, de la base textuelle à classer, de l'approche adoptée pour représenter les textes, de l'algorithme d'apprentissage opérée, et enfin du juge humain puisque l'attribution finale d'un document à une classe dépend du centre d'intérêt des utilisateurs de ses systèmes.

Ainsi, des mesures différentes, pour des systèmes dédiés à la classification déterministe ou « dure » et pour des systèmes dédiés à la classification floue ou ranking, sont proposées, qui s'intéressent chacune d'elles à un aspect de classification.
Dans ce chapitre nous allons présenter les méthodologies existantes pour aborder une vraie comparaison entre les classifieurs, pour ensuite dévoiler les mesures de performance souvent utilisées dans la littérature, et enfin achever par un description brève d'autres critères de performances non mesurables.

4.2- Méthodologies de comparaison de classifieurs

Il existe, en pratique, plusieurs méthodologies pour tenter de répondre à la question : quel est la meilleure méthode pour la catégorisation de textes ?

4.2.1- Différentes approches sur le même corpus

La première solution consiste à comparer différentes méthodes mises en œuvre par différents auteurs sur le même corpus, néanmoins, du point de vue pratique, comme le confirme Radwan JALAM dans (Jalam, 2003), on est confronté à pas mal de problèmes, parmi lesquels :

4.2.1.1- Même corpus avec des découpages différents

Les différents auteurs n'utilisent pas exactement le même découpage du corpus, par exemple pour Reuters seulement, il y a plus de six versions différentes, qui se distinguent par le nombre de leurs classes et la répartition des documents sur le corpus d'apprentissage et le corpus de test. Pour Reuters-21578 qui est souvent utilisé, (Joachims, 1998), (Schapire & all, 1998), (Yang & Liu, 1999) considèrent 90 catégories, (Dumais & all, 1998) en considèrent 118, d'autres travaillent carrément sur Reuters-top10 comme dans (Turenne, 2000) ou (Denoyer, 2004) ou (Yvon, 2006), qui trient les dix meilleurs catégories (Mini corpus utilisé dans nos expérimentations). De plus, la plupart des auteurs considèrent 3299 documents sur la

base de test, mais (Yang & Liu, 1999) en considèrent uniquement 3019 en supprimant tous les documents de la base de test qui n'appartiennent à aucune catégorie. Finalement, ces légères différences de découpage rendent difficiles les comparaisons à travers ces publications.

4.2.1.2- Les différentes techniques de représentation de textes

Les différentes alternatives offertes, pour le choix de descripteurs, afin de coder un texte, ainsi que les diverses méthodes de réduction de dimensionnalité utilisées par les différents auteurs peuvent embrouiller la comparaison de deux classifieurs s'exerçant sur le même corpus.

4.2.1.3- Les différentes mesures utilisées pour l'évaluation

Les mesures de performance utilisées dans les différentes expérimentations ne sont pas les mêmes (une description de quelques mesures est présentée dans la section suivante), ainsi les différents critères de performance peuvent être estimés de différentes façons empêchant une comparaison efficace entre les classifieurs.

4.2.2- Différentes approches par le même auteur

Une autre approche, plus crédible de point de vue scientifique, souvent proposée est l'utilisation de plusieurs méthodes par le même auteur, et automatiquement le corpus, le découpage de ce dernier, les techniques de codage, et les mesures de performance sont semblables pour toutes les méthodes. (Yang & Liu, 1999) comparent ainsi les kPPv, les SVM, les réseaux de neurones, et d'autres approches.
(Dumais & all, 1998) proposent également plusieurs comparaisons en mettant en opposition Les SVM, l'algorithme de Rocchio, les arbres de décision, et les réseaux bayesiens.

4.2.3- Difficultés approuvées pour juger les capacités d'une méthode

Les comparaisons présentées évaluent plus les compétences des auteurs dans l'exploitation des différentes approches de l'état de l'art les méthodes, plus que les capacités des méthodes elles-mêmes.
Le problème vient du fait que toutes ces méthodes sont délicates à mettre en œuvre et leurs performances dépendent fortement de leurs différentes utilisations.
Par exemple, l'implémentation des machines à vecteurs supports proposées par (Dumais & all, 1998) obtient nettement de meilleurs résultats que celle proposée par (Joachims, 1998).
Les réseaux de neurones testés par (Yang & Liu, 1999) sont des perceptrons multi-couches avec une couche cachée comportant 64 neurones, 1000 descripteurs en entrées et 90 neurones de sorties correspondant aux 90 catégories ; ils considèrent un seul réseau pour l'ensemble des catégories comportant plus de 64000 poids. Il n'est pas surprenant, dans ces conditions, que les performances obtenues ne soient pas très bonnes.
Il reste aussi difficile d'extrapoler les performances sur d'autres corpus et applications. Les résultats sont extrêmement dépendants du type des textes et des classes (en particulier de leur nombre). Il n'existe pas, à l'heure actuelle d'analyse de la performance des algorithmes en fonction des spécificités des corpus.
Ces différentes remarques prouvent que le succès d'une méthode dépend d'un ensemble de paramètres et certaines conditions non liées seulement, aux algorithmes d'apprentissage eux mêmes, mais aussi aux différents choix opérés pendant tout le processus, et qui peuvent intervenir et influencer les résultats obtenus. Par conséquent, il est extrêmement difficile de tirer des conclusions définitives sur une approche.

4.2.4- TREC

Il nous semble que la conférence TREC (Décrite en annexe) est une bonne solution pour comparer différentes méthodes, car chaque participant propose des solutions qu'il connaît bien avec des algorithmes dont il a pu tester l'efficacité. Le corpus est évidemment identique pour tout le monde, ainsi que les méthodes d'évaluation et la répétition annuelle de cette conférence permet de juger les approches sur le long terme.
De plus la conférence TREC a l'avantage de proposer un état de l'art à un instant donné contrairement aux comparaisons faites à partir des publications pour lesquelles le décalage dans le temps peut rendre certaines conclusions obsolètes.

4.3- Mesures de performance de classifieurs

4.3.1- Classification déterministe à deux classes

Nous considérons ici un problème simple de classification pour lequel nous nous intéressons à une classe unique C et nous voulons évaluer un système qui nous indique si un document peut être associé ou non à cette classe *C.* Ce problème est un problème de classification à deux classes (C et *non C* noté ¬ C). Si on peut maitriser ce problème simple, on pourra fusionner par la suite, les mesures de performance de plusieurs systèmes bi-classes afin d'obtenir une mesure de la performance d'un classifieur multi-classes.

4.3.1.1- 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 4.1), 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	$\mathbf{VP_i}$	FP_i
	Non	FN_i	$\mathbf{VN_i}$

Tableau 4.1 : Matrice de contingence de la classe C_i

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

4.3.1.2- Précision et Rappel

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

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

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

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

Le rappel mesure la capacité d'un système de classification à détecter les documents correctement classés. Cependant, un système de classification qui considérerait tous les documents comme pertinents obtiendrait un rappel de 100%. Un rappel fort ou faible n'est pas suffisant pour évaluer les performances d'un système. Pour cela, on définit la précision.

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

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

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

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

Ces deux indicateurs pris l'un indépendamment de l'autre ne permettent d'évaluer qu'une facette du système de classification : la qualité ou la quantité. Les courbes de *précision* vs *rappel* permettent de mieux comprendre le comportement du classifieur, et de visualiser l'évolution de la précision en fonction du rappel pour les 11 niveaux standard [0-0,1-0,2-...-1].

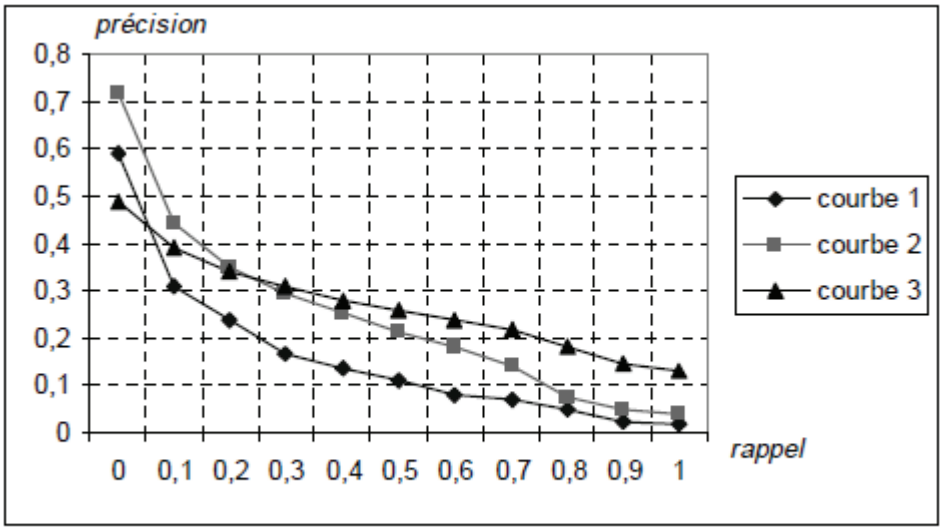


Figure 4.1 : Courbe Rappel-Précision pour trois classifieurs

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

4.3.1.3- Bruit et silence

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

$$Bruit\ (B)\ = 1 - Précision(P)\ = \frac{FP_i}{VP_i + FP_i}$$

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

$$Silence\ (S)\ = 1 - Rappel(R) = \frac{FN_i}{VP_i + FN_i}$$

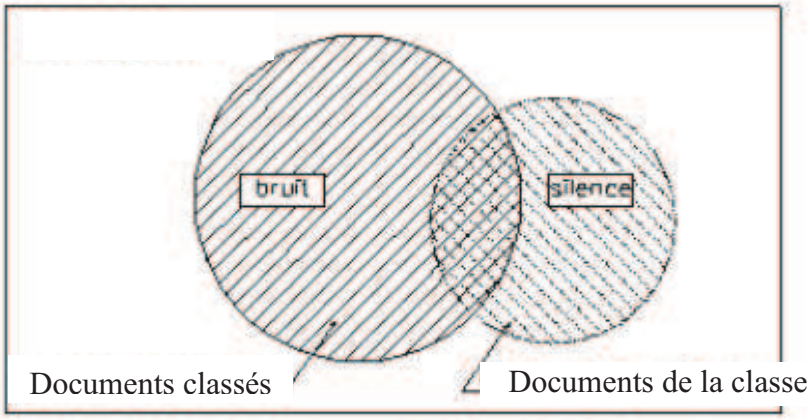


Figure 4.2 : Notions de bruit et de silence

4.3.1.4- Taux de succès et taux d'erreur

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

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

4.3.1.5- Taux de chute et la spécificité

Deux autres indicateurs peuvent être utilisés pour mesurer la performance d'un classifieur :

$$Taux\ de\ chute = \frac{FP_i}{FP_i + VN_i}$$

$$Spécificité = \frac{VN_i}{FP_i + VN_i}$$

4.3.1.6- L'overlap et la généralité

$$Overlap = \frac{VP_i}{VP_i + FP_i + FN_i}$$

$$Généralité = \frac{VP}{VP + VN + FP + FN}$$

4.3.1.7- F-measure

Observés conjointement, les indicateurs les plus célèbres à savoir le rappel et la précision, sont une estimation courante de la performance d'un système de classification. Cependant plusieurs mesures ont été développées afin de synthétiser cette double information. Nous ne

retiendrons ici la mesure F_β décrite dans (Van Rijsbergen, 1979) . La ***F***-mesure est la mesure de synthèse communément adoptée depuis les années 80 pour évaluer les algorithmes de classification de données textuelles à partir de la précision et du rappel. Elle est employée indifféremment pour la classification (Non supervisé) ou la catégorisation (Supervisé), pour la problématique de recherche d'information ou de classification.
Elle permet donc, de combiner, selon un paramètre β, rappel et précision.
On définit la mesure F_β comme la moyenne harmonique entre le rappel et la précision :

$$F_\beta = \frac{(\beta^2 + 1) * précision * rappel}{\beta^2 * précision + rappel}$$

Pour utiliser cette mesure, il est donc nécessaire de fixer préalablement un seuil de décision pour le classement, et de calculer la valeur de F_β pour ce seuil.
Le paramètre β permet de choisir l'importance relative que l'on souhaite donner à chaque quantité. On choisit en général de donner la même importance aux deux critères, donc habituellement, la valeur de β est fixée à 1 et la mesure est ainsi notée F_1 (noté ***F)*** qui s'écrit :

$$F_1 = \frac{2 * P * R}{P + R}$$

Une des propriétés intéressante de cette mesure est le fait que, si $P = R = X$, alors $F = X$; cette mesure a alors une interprétation simple.

Afin de mieux comprendre le fonctionnement de ces 3 mesures (précision, rappel et mesure F), nous détaillons dans les tableaux 4.2 les performances de plusieurs classifieurs utilisés sur deux classes C et ¬ C de respectivement 100 et 200 documents.

Classifieur \ Expert	C	¬ C
C	100	200
¬ C	0	0
Rappel = 100 %		
Précision = 33 %		
F_1 = 50 %		

Tous les textes sont classés dans C

Classifieur \ Expert	C	¬ C
C	0	0
¬ C	100	200
Rappel = 0 %		
Précision = 100 %		
F_1 = 0 %		

Aucun texte n'est classé dans C

Classifieur \ Expert	C	¬ C
C	100	0
¬ C	0	200
Rappel = 100 %		
Précision = 100 %		
F_1 = 100 %		

Classifieur parfait

Classifieur \ Expert	C	¬ C
C	0	200
¬ C	100	0
Rappel = 0 %		
Précision = 0 %		
F_1 = 0 %		

Classifieur « le pire »

Tableaux 4.2 : Différents classifieurs et les mesures rappel, précision et F_1 associées

Les deux classifieurs du haut représentent des classifieurs « radicaux » qui classent tous les textes dans C, soit aucun n'est classé dans C. Les deux autres classifieurs sont des classifieurs, soit parfait – les textes sont correctement classés – soit dramatique – les textes sont toujours mal classés.

4.3.2- Classification déterministe à plusieurs classes

Pour la catégorisation à plusieurs classes de textes, une approche commune consiste à couper le processus de catégorisation en sous-problèmes. Chaque sous-problème concerne uniquement une classe et l'objectif est alors de juger si le nouveau texte appartient ou n'appartient pas à cette classe par opposition aux autres.

Pour la catégorisation multi-classes de textes, nous avons un ensemble de classes C = (C_1,..., $C_{/C/}$) où /C/ est le nombre de classes (|C| > 2). Nous notons N*i* le nombre de documents de C_i.
Pour chacune des classes, nous pouvons calculer comme précédemment le rappel, la précision et la mesure F_1, notés respectivement R_i , P_i et $F1_i$. Nous pouvons donc obtenir des mesures globales pour le système à |C| classes en moyennant ces mesures par classe.
La précision et le rappel globaux, c-à-d, sur toutes les classes peuvent être calculés à travers une moyenne des résultats obtenus pour chaque catégorie.

Cependant, si les classes ne possèdent pas le même nombre de documents, ces moyennes risquent de ne pas refléter la performance du classifieur pour les grandes classes. Les résultats de chaque catégorie peuvent être combinés de deux manières :

- On peut calculer un score pour chaque catégorie à partir de sa matrice de contingence puis déterminer la moyenne des scores sur l'ensemble des catégories **(macro-averaging)**. Dans ce cas, toutes les catégories interviennent de la même manière dans le calcul du score final quelque soit le nombre de documents qu'elles contiennent.

- Une autre possibilité est de créer une table de contingence globale pour toutes les catégories **(micro-averaging)** : le contenu d'une cellule de cette table correspond à la somme des valeurs de la même cellule dans la table de chaque catégorie (Yang, 1999).

4.3.2.1- Matrice de contingence globale

		Expert	
	$C = \{c_1, c_2, ..., c_{\|C\|}\}$	C_i	$\neg C_i$
Classifieur	C_i	$VP = \sum_{i=1}^{\|C\|} VP_i$	$FP = \sum_{i=1}^{\|C\|} FP_i$
	$\neg C_i$	$FN = \sum_{i=1}^{\|C\|} FN_i$	$VN = \sum_{i=1}^{\|C\|} VN_i$

Tableau 4.3 : Table de contingence globale

4.3.2.2- La micro-moyenne

Les mesures de type *micro moyenne* (ou *micro*) correspondent à une moyenne qui pondère chaque classe par son effectif.
La micro-moyenne (traduction de micro-averaging) calcule les mesures rappel et précision de façon globale : si l'on considère les tables de contingences associées à chaque catégorie, cela revient à sommer les cases VP, FP, FN et VN de chaque catégorie pour obtenir la table de contingence globale (voir le tableau 4.3).

Les différentes mesures sont ensuite calculées à partir des valeurs cumulées. La micro-moyenne accorde donc des poids importants aux catégories ayant beaucoup d'exemples. La performance du classifieur dépend surtout de sa capacité à traiter les catégories les plus fréquentes. Ainsi, la précision micro-moyenne et le rappel micro-moyenne sont estimés comme suit :

$$\boldsymbol{P} = \frac{VP}{VP + FP} = \frac{\sum_{i=1}^{|C|} VP_i}{\sum_{i=1}^{|C|} \left(VP_i + FP_i\right)}$$

$$\boldsymbol{R} = \frac{VP}{VP + FN} = \frac{\sum_{i=1}^{|C|} VP_i}{\sum_{i=1}^{|C|} \left(VP_i + FN_i\right)}$$

4.3.2.3- La macro-moyenne

Les mesures de type *macro moyenne* correspondent à une moyenne qui ne prend pas en compte la taille des classes.
La macro-moyenne (traduction de macro-averaging) évalue d'abord indépendamment chaque catégorie. Ensuite, la performance globale du classifieur est calculée en faisant la moyenne des mesures individuelles. Les différentes catégories ont alors la même importance. La précision et le rappel macro-moyenne sont calculés comme suit :

$$\boldsymbol{P} = \frac{\sum_{i=1}^{|C|} P_i}{|C|} \quad , \quad \boldsymbol{R} = \frac{\sum_{i=1}^{|C|} R_i}{|C|}$$

Ainsi, les mesures de type *micro moyenne* permettent d'obtenir une estimation du système en privilégiant les classes de grande taille tandis que les mesures de type *macro moyenne* donnent une information quant aux performances d'un système sur les petites classes.

Le tableau 4.4 résume les 6 mesures obtenues :

Mesure du rappel	*Rappel macro moyenne*	$\frac{\sum_{i=1}^{\|c\|} rappel_i}{\|C\|}$
	Rappel micro moyenne	$\frac{\sum_{i=1}^{\|C\|} n_i * rappel_i}{\sum_{i=1}^{\|C\|} n_i}$
Mesure de la précision	*Précision macro moyenne*	$\frac{\sum_{i=1}^{\|c\|} précision_i}{\|C\|}$
	Précision micro moyenne	$\frac{\sum_{i=1}^{\|C\|} n_i * précision_i}{\sum_{i=1}^{\|C\|} n_i}$
Mesure F_1	*F1 macro moyenne*	$\frac{\sum_{i=1}^{\|c\|} F_{1_i}}{\|C\|}$
	F1 micro moyenne	$\frac{\sum_{i=1}^{\|C\|} n_i * F_{1_i}}{\sum_{i=1}^{\|C\|} n_i}$

Tableau 4.4 : Les mesures de performances en classification multi-classes
Cette figure illustre les formules de calcul pour les mesures micro-moyenne et macro-moyenne

4.3.2.4- Une mesure issue de TREC : l'utilité

Les fonctions d'utilité ont été introduites dans le cadre de la tâche de filtrage, lors de la compétition TREC, décrite brièvement en annexe.
L'idée consiste à donner un nombre positif de points au système pour chaque document correctement classés et à retirer des points négatifs pour chaque document incorrectement classés. L'utilité est donc de la forme :

$$U = a.VP + b.FP$$

Où *VP* est le nombre de documents correctement classés, et *FP* est le nombre de documents incorrectement classés. Les coefficients *a* et *b* varient selon l'importance relative que l'on souhaite donner à chaque terme. Les valeurs les plus couramment utilisées sont a = 3, b = -2 et a = 3, b= -1.
L'évaluation de l'utilité ne nécessite que l'observation des documents classés ; elle est donc plus facilement calculable que le rappel.
Néanmoins, cette mesure présente quelques inconvénients qui ne sont pas détaillés dans notre mémoire, qui font qu'elle est peu utilisée en dehors de la conférence TREC (Voir Annexe).

4.3.3- Classification floue ou Ranking

Certains systèmes de classification, et notamment les classifieurs probabilistes ou ceux basés sur le calcul de distance, trient les catégories les plus adéquates dans l'ordre pour y classer le

texte. Les catégories sont classées soit par les distances croissantes ou par probabilités décroissantes.
Ils existent des mesures de performances adéquates à ces systèmes, inspirées de la recherche d'information adaptée à la classification; citons parmi lesquelles la technique du « 11-point average précision : **Précision moyenne sur 11 points** ».
Cette approche consiste à évaluer la précision et le rappel pour chacun des 11 seuils de 0 % à 100 % par pas de 10 % {0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0}, puis de calculer les précisions et rappel moyens. La moyenne de ces 11 valeurs de précision estime la capacité de catégoriser un document. La moyenne de ces résultats obtenus pour les différents textes de l'ensemble de test permet d'évaluer la capacité globale du classifieur sur ce corpus.

4.4- Autres critères de comparaison de classifieurs

La majorité des travaux de développement concernant la classification et la recherche d'information s'appuient principalement, pour tirer des conclusions sur les classifieurs, sur des notions de rappel et de précision. Or, il existe d'autres critères pour évaluer et comparer deux systèmes. Si ces deux informations sont très utiles pour l'évaluation des performances des systèmes de classification, cela ne renseigne rien, par exemple sur la complexité ou la facilité d'utilisation du système, le temps de réponse, l'effort fourni par l'utilisateur, ou la présentation du résultat, ou encore d'autres facteurs d'évaluation des classifieurs qui sont introduits par la communauté d'Apprentissage Automatique indépendants de l'adéquation aux données d'apprentissage. Citons parmi eux :

- ***Compéhensibilité*** : Le modèle est-il compréhensible ? Le système donne-t-il des réponses permettant de comprendre pourquoi un document a été classé dans une certaine catégorie ou bien s'agit-il d'une fonction numérique calculée à partir de données servant d'exemples (Boite noire) ? La distinction principale entre induction (apprentissage) numérique et apprentissage symbolique inductif réside dans l'expression de la fonction f; l'apprentissage symbolique produit des expressions compréhensibles, telles que des règles de production ou des arbres de décision.
- ***Simplicité*** : apprécie le taux de simplicité des résultats d'apprentissage produits par le classifieur.
- ***Intelligibilié*** : évalue le degré d'intelligence du classifieur.
- ***Le temps de réponse et d'indexation*** : est aussi un point qui peut être fondamental.
- ***L'encombrement du système et les ressources en mémoire requises*** : l'espace alloué en mémoire vive et sur le disque dur qui doit être prise en compte dans de nombreux cas.

On peut trouver dans la littérature tous ces critères d'évaluation mais pas avec le même degré d'importance. Par exemple, (Dumais & all, 1998) ont mis en compétitions cinq techniques d'apprentissage selon trois critères :

- Training efficiency (Efficacité d'apprentissage) : Le temps moyen nécessaire pour l'apprentissage.
- Classification efficiency (Efficacité de classement) : Le temps moyen nécessaire pour la classification de nouveaux documents.
- Effectiveness (Capacité d'apprentissage) : L'aptitude de traitement des grands corpus d'apprentissage.

4.5- Conclusion

Nous avons montré dans ce chapitre que les mesures absolues de performances ont une portée limitée. Cette limitation est due, d'une part, à l'impossibilité de définir précisément la notion

de pertinence, et d'autre part, à l'impossibilité d'obtenir des corpus de grande taille totalement et correctement étiquetés.
Il est nécessaire de mesurer les performances d'un filtre sur un ensemble de thèmes pour d'une part limiter l'impact des erreurs d'annotations et d'autre part, pour juger globalement une approche sur des thèmes de difficultés différentes. Plusieurs indicateurs de mesures sont proposés dans la littérature et la recherche d'autres mesures plus fiables n'a pas cessé et reste toujours une matière intéressante pour les chercheurs.
Néanmoins toutes les mesures présentées dans ce chapitre traitent toutes les erreurs avec la même importance, alors que du point de vue de l'utilisateur, cette assertion n'est pas vraie.

Il est cependant très difficile de prendre ces informations en considération, puisqu'il existe une grande part de subjectivité dans ces appréciations, et que finalement la seule vraie mesure est la satisfaction de l'utilisateur.

Chapitre 5

Les Systèmes Multi-Agents

Table des matières

5.1- Introduction

5.1.1- Historique

Durant la première génération des programmes informatiques, l'ordinateur était chargé de réaliser des tâches prises en charge habituellement par un homme comme par exemple la classification automatique d'une population qui requiert de l'intelligence artificielle. Ce remplacement progressif de l'homme par une machine s'est accompagné d'une identification de la machine à l'humain, un programme représentant l'expert capable de résoudre le problème par lui-même. Cette façon de concevoir les programmes comme des sortes de penseurs repliés sur eux-mêmes a trouvé sa limitation lorsqu'on a cherché à développer des applications plus complexes réalisées habituellement non pas par une seule personne mais par un groupe de personnes parfois délocalisées.
L'informatique devient ainsi de plus en plus diffuse et distribuée dans de multiples objets et fonctionnalités qui sont amenés à coopérer. La décentralisation est donc la règle et une organisation coopérative entre modules logiciels est un besoin. De plus, la taille, la complexité et l'évolutivité croissante de ces nouvelles applications informatiques font, de cette vision centralisée, rigide et passive (contrôlée explicitement par le programmeur), atteindre ses limites.
La machine devait alors être identifiée non plus uniquement à un humain mais à une société organisée d'humains. En particulier, les concepteurs de système industriels complexes ont constaté que le savoir-faire, les compétences et les connaissances diverses sont détenues par des individus différents qui, au sein d'un groupe, communiquent, échangent leurs connaissances et collaborent à la réalisation d'une tâche commune. Les méthodes de réalisation d'applications informatiques se sont alors concentrées sur les aspects organisationnels des logiciels et sur la représentation des communications entre ses différents composants.
Ainsi une nouvelle manière de penser a surgit, donnant naissance à ce qu'on appelle l'intelligence artificielle distribuée.

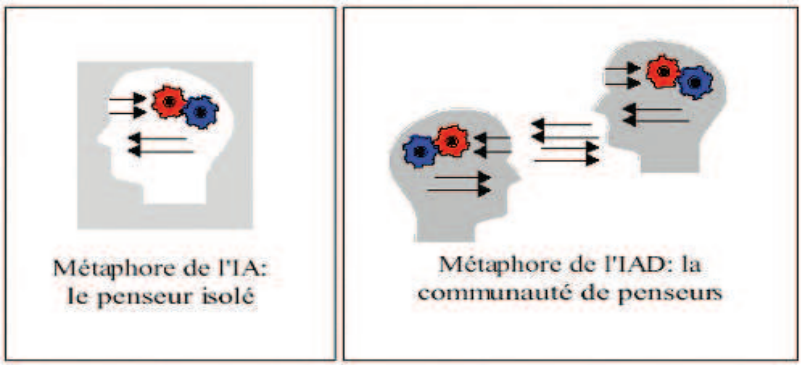


Figure 5.1 : I.A versus l'I.A.D

Une réponse à la question pourquoi distribuer l'intelligence ? la définition des concepts de base des SMA ainsi que les différents modèles d'agents(Architecture) sont étalés dans la section suivante.

5.1.2- Pourquoi distribuer l'intelligence?

Pourquoi cherche-t-on à créer des intelligences collectives? Pourquoi vouloir à tout prix prendre un point de vue local? Pourquoi tout simplement vouloir distribuer l'intelligence?

L'Informatique « moderne » contient des systèmes de plus en plus répartis parallèlement, qui peuvent donner des éléments de réponse à ces questions :

L'accroissement des capacités informatiques

- Connectivité (internet, WWW, Satellite…)
- puissance de calcul
- Puissance de transmission (vitesse, bande passante…)
- Interface (visualisation, vocal…).
- Informations (taille, complexité, modalité).
- Ressources (distribuées, hétérogènes partagées)

L'accroissement de l'hétérogénéité

- Interactivité (collecticiels, environnement)
- données/connaissances
- BDD multimédia, scientifiques numériques
- Syntaxe, sémantique, structuration
- Régionalisation des informations
- Interface multimodales (données, voix, image, geste, vidéo, langage…).
- Communication (satellite, communication mobiles, réseaux).

L'accroissement des besoins applicatifs

- Traitement intégrés, omniprésents, dynamiques (sujets, information, activité).
- Système d'information hétérogène, étendus, fortement intégrés, très complexes.
- De nouveaux utilisateurs (population globale, communautés d'intérêt, multiples intérêts).
- De nouveaux champs (domaines de connaissance application).
- De nouvelles opportunités d'information, de cout, de transaction, de valorisation.

Les problèmes sont physiquement distribués

Réseaux, contrôle aérien, robotique, etc...

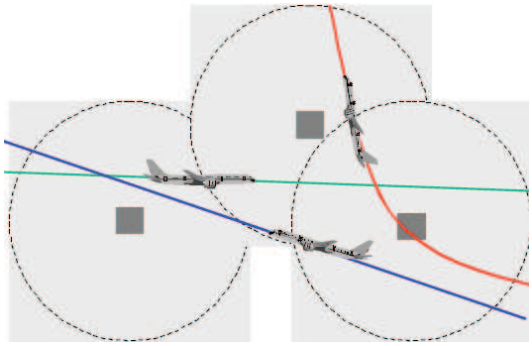


Figure 5.2 : Distribution physique

Les problèmes sont fonctionnellement très distribues et hétérogènes

Concevoir un produit industriel aussi élabore qu'une voiture de course, qu'un avion de ligne ou qu'un lanceur de satellites réclame l'intervention d'un grand nombre de spécialistes, qui ne possèdent qu'une vision locale de l'ensemble des problèmes poses par la réalisation du système: nul en effet n'est suffisamment savant ou qualifie pour produire une telle réalisation a lui tout seul. L'ensemble des problèmes est trop vaste pour un seul individu. Une voiture de formule 1, par exemple, fait intervenir un grand nombre d'experts pour sa mise au point: il y a le spécialiste des moteurs, celui des châssis, celui des pneumatiques, l'ingénieur en chef et le pilote. Toutes ces personnes mettent leurs connaissances en commun pour essayer de faire la meilleure voiture possible.

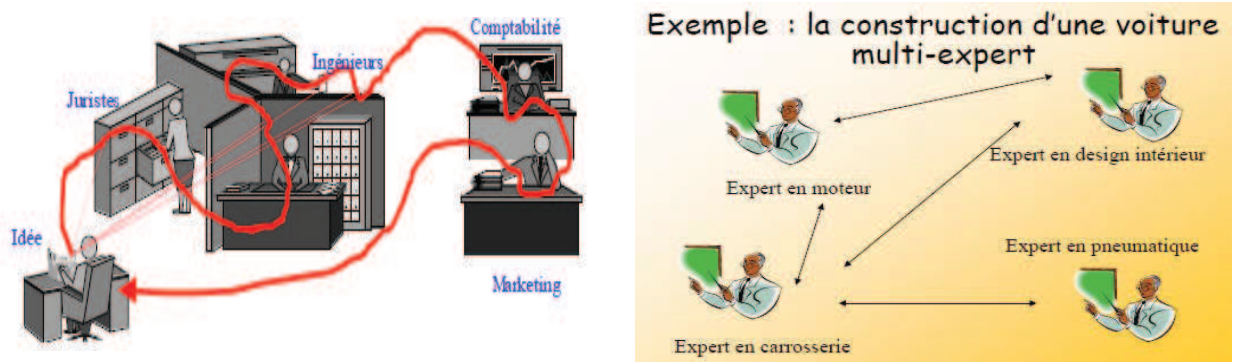


Figure 5.3 : Distribution fonctionnelle

Les réseaux imposent une vision distribuée

A l'heure des réseaux interplanétaires (Internet), ou toute l'information et la puissance de traitement est repartie sur un nombre Très grand de sites, il faut penser en termes de systèmes ouverts, c'est-`a-dire en termes d'interopérabilité radicale des systèmes informatiques. Il faut considérer que l'espace informatique n'est qu'une gigantesque toile d'araignée (WWW) dans laquelle tout ordinateur, qu'il soit fixe, ou mobile, est connecte `a l'ensemble des autres ordinateurs du monde entier. Les SMA se présentent ainsi comme des candidats sérieux pour la construction d'architectures ouvertes, distribuées, hétérogènes et souples, capables d'offrir une grande qualité de service dans un travail collectif, sans imposer une structure a priori.

La complexité des problèmes

Impose une vision locale. Lorsque les problèmes sont trop vastes pour être analyses globalement, les solutions fondées sur des approches locales permettent souvent de les résoudre plus rapidement. Par exemple, la régulation de trafic aérien est un problème global complexe, difficile à résoudre du fait du grand nombre de paramètres qui sont mis en jeu et de l'ensemble des contraintes qui doivent être satisfaites, des approches locales permettent de résoudre efficacement ce type de problème.

Les systèmes doivent pouvoir s'adapter à des modifications de structure ou d'environnement

Savoir concevoir des systèmes informatiques efficaces, fiables et corrects ne suffit plus. Devant les défis de la complexité, il faut aussi penser à l'adaptabilité d'un logiciel à des modifications du contexte de travail (changement de système d'exploitation, de gestionnaire de bases de données, d'interfaces graphiques, ajouts d'autres logiciels, etc.), et sa nécessaire capacité d'´evolution face à des besoins sans cesse accrus (ajouts de fonctionnalités, modifications de l'utilisation, intégration à d'autres logiciels, etc.). Dans ce cadre, les SMA, de par leur nature distribuée, parce qu'ils supposent toujours un raisonnement local, qu'ils permettent l'intégration et l'apparition ou la disparition d'agents en cours même de fonctionnement, constituent des architectures particulièrement aptes à prendre en compte l'évolutivité et l'adaptation nécessaires au fonctionnement du système.

Le génie logiciel va dans le sens d'une conception en termes d'unités autonomes en interactions

L'histoire du développement des logiciels montre que la réalisation de programmes informatique suit une démarche visant à la réalisation de systèmes conçus comme des ensembles d'entités de plus en plus distribuées, mettant en jeu des composants davantage individualises et autonomes. Le récent développement des langages à objets dans tous les secteurs de l'informatique est là pour en témoigner: le génie logiciel passe par la réalisation de

modules autonomes capables d'interagir les uns avec les autres, même, et surtout, lorsque ceux-ci sont conçus par des personnes, des équipes ou des entreprises différentes. Il faut donc associer la fluidité des calculs, la distribution des traitements et l'hétérogénéité des réalisations. Les SMA ont ici un rôle essentiel `a jouer en s'inscrivant comme les possibles successeurs des systèmes à objets, en ajoutant à la localité des comportements l'autonomie et la répartition des prises de décision. On peut ainsi déjà parier que le génie logiciel de demain sera "orienté agent", comme celui d'aujourd'hui qui est "oriente objet" (Ferber, 1995).
dans l'espace : il est difficile de décomposer le système en parties déboguables indépendamment ; si on coupe un système en deux on le "tue".
dans le temps : il est difficile d'analyser des systèmes en fonctionnement en continu ; si on arrête un système pour l'observer on le "tue".

Tous ces facteurs peuvent justifier la nécessite de cette distribution d'où des systèmes :
– de plus en plus complexes
– répartis sur des sites de plus en plus nombreux
– constitués de logiciels en interaction entre eux ou avec des êtres humains d'où une volonté
D'intégrer :
De faire inter opérer :
De faire coopérer des logiciels existants.

5.1.3- Qu'est que l'intelligence artificielle distribuée (IAD) ?

Les systèmes experts (qui sont l'émanation la plus visible de l'IA dans l'entreprise) ne peuvent donc prétendre qu'à un rôle mineur dans les procédures de travail informatisé :

- unicité de leur expertise,
- unicité de leur point de vue,
- rigidité de leur capacité d'interaction.

« L'IAD est l'étude, la conception et la réalisation de systèmes multi-agents, c'est à-dire de systèmes dans lesquels des agents intelligents qui interagissent, poursuivent un ensemble de buts ou réalisent un ensemble d'actions » (Wies, 1995).
« L'intelligence, comme la science, n'est pas une caractéristique individuelle que l'on pourrait séparer du contexte social dans lequel elle s'exprime » (Latour, 1989), (Latour, 2006), (Latour & Lemonnier , 1994), (Lestel, 1986), (Lestel & all, 1994).
« Un être humain ne peut se développer convenablement s'il ne se trouve pas entouré d'autres êtres de son espèce. Sans un entourage adéquat, son développement cognitif est très limité, et le simple apprentissage d'une langue articulée lui devient proprement impossible s'il n'a pas été plongé dans une culture humaine dés sa première enfance. En d'autres termes, les autres sont indispensables à notre développement cognitif et ce que nous appelons "intelligence" est autant dû aux bases génétiques qui définissent notre structure neuronale générale qu'aux interactions que nous pouvons avoir avec le monde qui nous entoure et, en particulier, avec la société humaine » (Ferber, 1995).
L'IA Distribuée ajoute donc la dimension sociale à l'IA classique.
Les capacités intellectuelles d'un être humain proviennent :
De ses prédispositions génétiques ;
Des interactions avec ses semblables (accointances) ;
Des interactions avec son environnement.
Les capacités d'une machine "intelligente" devraient donc provenir :

- de ses possibilités d'inférence ;
- des interactions avec les autres machines ;
- des interactions avec son environnement.

L'IAD est née de la difficulté d'intégrer dans une même base de connaissances, l'expertise, les compétences et la connaissance de différentes entités qui communiquent et collaborent pour réaliser un but commun (Erceau & Ferber, 1991). L'IAD consiste à distribuer l'expertise au sein d'une société d'entités, appelées *agents* dont le contrôle et les données sont distribués (Guessoum, 1996). Ces agents, qui sont relativement indépendants et autonomes interagissent dans des modes simples ou complexes de coopération pour accomplir un objectif global, notamment la résolution de problèmes complexes. N.Skarmeas définit l'IAD, dans «Agents as Objects with Knowledge Base State» (Skarmeas, 1998), comme étant un domaine concerné par les systèmes ouverts et distribués dont les entités présentent une sorte d'intelligence et qui essayent d'accomplir des buts qui peuvent être implicites ou explicites. L'IAD a donné naissance à deux principaux domaines :

La ***résolution des problèmes distribués*** (DPS), qui s'intéresse à la décomposition d'un problème complexe en sous-problèmes et à sa résolution par des entités distribuées logiques et physiques.
Les ***systèmes multi-agents*** (SMA) qui sont concernés par la coordination du comportement des agents, qui peuvent être vus comme des entités intelligentes et autonomes (Skarmeas, 1998).

SMA «versus» DPS

DPS (résolution distribuée de problème) proche de l'IA classique
•Répartir à la conception du travail nécessaire à la résolution d'un problème parmi un ensemble d'agents.
•Orienté par le problème à traiter.
•Approche descendante, centrée Agent.
⇒contrôle centralisé, statique, modulaire

SMA (système multi-agent)
•Ensemble d'agents autonomes en interaction
•Coordonner le comportement de cet ensemble d'agents (pré-existants) pour résoudre collectivement un problème
•Dans un environnement complexe et évolutif
•Approche ascendante, centrée sociale
⇒résolution « émergente » : à **l'exécution**

(Labidi & Lejouad 1993) et (Oliveira, 1998) affirment que l'IAD était supposée apporter une solution à des problèmes spécifiques tels que :

- ✓ la modélisation et distribution de la connaissance parmi plusieurs agents ;
- ✓ la génération de plans d'actions où la présence d'autres agents doit être considérée ;
- ✓ la résolution de conflits entre agents et la maintenance de la cohérence des décisions et plans d'actions ;
- ✓ la résolution des problèmes de communication pour permettre les interactions entre agents ;
- ✓ la résolution des problèmes spécifiques concernant l'organisation des SMAs

On voit que les systèmes multi-agents se positionnent au carrefour de la programmation (ce sont des logiciels), de l'intelligence artificielle (leur autonomie de décision), et des systèmes répartis (leur décentralisation). Historiquement, on peut replacer le concept d'agent et de système multi-agent dans l'histoire de l'intelligence artificielle et de manière duale dans l'histoire de la programmation.

5.1.4- Le monde est ouvert

La programmation classique est fermée

- Dans l'espace : le programmeur a une connaissance globale du logiciel à construire. Le principe même de l'analyse descendante d'application ou de la spécification d'application est de partir du haut (où on voit tout) puis de décomposer en parties à programmer.
- Dans le temps : bien que la notion de cycle de vie d'un logiciel introduise une dose de dynamique, il s'agit plutôt de corriger et de maintenir un logiciel spécifié une fois pour toutes.
- Dans la modalité : il existe une volonté de développer les applications de manière la plus homogène possible : mêmes personnes, mêmes logiciels de développement.
- Dans la sémantique : Les applications ont une sémantique globale et statique.
- Dans la complexité : Les applications sont bien délimitées et conçues de manière analytique. Ceci facilite le découpage en parties pour leur mise au point et l'étude de leur comportement qui est considéré comme devant être totalement prédictible.

A l'opposé, la nouvelle programmation est ouverte

L'ouverture est une propriété inhérente des systèmes d'information actuels. Elle n'est voulue par personne mais c'est un état des choses que l'on ne peut plus se permettre d'ignorer. Plutôt que d'essayer de l'enrayer, il vaut mieux essayer de la maîtriser voire en tirer avantage. Cette ouverture s'exprime :

- Dans l'espace : les systèmes d'information actuels sont intrinsèquement distribués (vision locale obligatoire) et se développent de manière agrégative.
- Dans la modalité : les deux points précédents ont pour conséquence directe que les systèmes d'information actuels sont de plus en plus hétérogènes : environnements de programmation ; points de vues adoptés sur un même sujet : fonctionnel, matériel, structurel etc. ; applications hybrides.
- Dans la sémantique : les multiples points de vue engendrent autant de mondes sémantiques hétérogènes les uns aux autres.
- Dans la complexité : les systèmes actuels présentent des interactions très intriquées ce qui les rend difficiles à prédire et à analyser

5.1.5- Domaines d'intérêts

- Intelligence Artificielle Distribuée (IAD)
- Bases de Données Distribuées (BDD)
- Systèmes d'Information Coopératifs (SIC)
- Génie logiciel
- Aide à la décision
- Apprentissage automatique

5.2- Concepts de base

5.2.1- Agent

5.2.1.1- Définitions

Ces dix dernières années, le concept d'agent a été utilisé et étudié dans plusieurs domaines. Toutefois, il n'y a encore aucun consensus, entre les différents chercheurs, quant à la définition même du mot « agent ». Selon H.S.Nwana (Nwana, 1996) et (Nwana & Ndumu,

2000) et rapporté par M.Raza (Raza, 2009), il y a au moins deux raisons qui permettent d'expliquer cette difficulté.

-La première réside dans le fait que les chercheurs, dans le domaine des agents, ne sont pas à l'origine de ce terme comme l'on été, par exemple, les chercheurs dans le domaine de la logique floue. En effet, le terme agent a été et continue d'être utilisé dans la vie de tous les jours par des personnes travaillant dans des domaines très différents.

Par exemple, on parle d'agent de voyage, d'agent immobilier, d'agent d'assurance, etc.

-La deuxième raison est que même dans la communauté des chercheurs sur les agents logiciels, le mot « agent » est utilisé pour décrire des systèmes très différents les uns des autres. Pour ajouter à la confusion, les chercheurs sont allés même jusqu'à inventer plusieurs synonymes au mot « agent ». Ils ont ainsi inventé, par exemple, « knowbots » (robots à base de connaissances), « softbots » (robots logiciel), taskbots (robots à base de tâche), « userbots » (robots pour utilisateur), robots, agents personnels, agents autonomes, assistants personnels, etc. Il est vrai qu'une telle prolifération de termes trouve sa justification dans le fait que les agents peuvent prendre différentes formes physiques (robot ou agent logiciel) et qu'ils peuvent aussi jouer plusieurs rôles.

Cela dit, il est tout de même important de s'entendre sur une définition du terme agent pour que les exposés qui suivent dans ce mémoire aient un sens. La définition que nous avons adoptée, et qui semble couvrir les caractéristiques des agents que nous avons développés, est celle proposée par Jennings, Sycara et Wooldridge (Jennings & all, 1998) :

A ce propos, Carl Hewitt fit remarquer (lors du troisième international workshop sur l'IAD), que la question qu'est ce qu'un agent ? est aussi embarrassante pour la communauté informatique que la question qu'est ce que l'intelligence ? est embarrassante pour la communauté d'intelligence artificielle.

Etant donné les origines diverses du concept agent, nous ne pouvons pas donner une seule définition au terme agent. En effet, plusieurs définitions ont été proposées par différents auteurs pour clarifier ce concept.

Déf1 : « les agents peuvent être vus comme des unités intelligentes et autonomes » (Skarmeas, 1998)

Déf2 : « les objets qui pensent » (Magendaz, 1995)

Déf3 : Un agent est un système informatique, situé dans un environnement, qui agit d'une façon autonome et flexible pour atteindre les objectifs pour lesquels il a été conçu.

Déf4 : Un agent est une entité logicielle ou physique à qui est attribuée une certaine mission qu'elle est capable d'accomplir de manière autonome et en coopération avec d'autres agents.

Déf5 **:** Un agent intelligent tout ce qui perçoit son environnement à l'aide de ses capteurs et qui agit sur son environnement à l'aide de ses effecteurs. (Chaib-Draa & all, 2001)

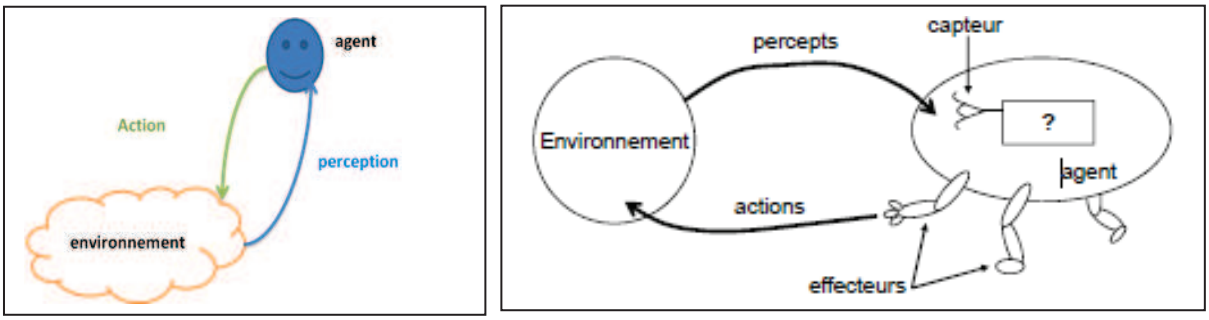


Figure 5.4 : L'environnement d'un agent

Déf6 : Le terme agent représente une entité intelligente qui agit de manière rationnelle et intentionnelle en fonction de ses buts et de l'état courant de ses connaissances (Demazeau & Müller, 1991).
Déf7 : un agent est entité dotée d'un état mental, qui représente ses connaissances, croyances, intentions et engagements vis-à-vis de lui-même et des autres agents. (Shoham, 1993), (Cohen & Levesque, 1995), (Wooldridge & Jennings, 1994), (Wooldridge, 1999) (Wooldridge, 2002).
Déf8 : Selon Jacques (Ferber, 1995) Qu'est ce qu'un agent? Comme dans tous les domaines porteurs, le terme agent est utilisé de manière assez vague. Cependant on peut dégager une définition minimale commune qui est approximativement la suivante:
: On appelle agent une entité physique ou virtuelle

a. qui est capable d'agir dans un environnement,
b. qui peut communiquer directement avec d'autres agents,
c. qui est mue par un ensemble de tendances (sous la forme d'objectifs individuels ou d'une fonction de satisfaction, voire de survie, qu'elle cherche `a optimiser),
d. qui possède des ressources propres,
e. qui est capable de percevoir (mais de manière limitée) son environnement,
f. qui ne dispose que d'une représentation partielle de cet environnement (et éventuellement aucune),
g. qui possède des compétences et offre des services,
h. qui peut éventuellement se reproduire,
i. dont le comportement tend à satisfaire ses objectifs, en tenant compte des ressources et des compétences dont elle dispose, et en fonction de sa perception, de ses représentations et des communications qu'elle reçoit.

Chacun des termes de cette définition est important

Une entité physique est quelque chose qui agit dans le monde réel: un robot, un avion ou une voiture sont des exemples d'entités physiques.
En revanche, un composant logiciel, un module informatique sont des entités virtuelles, car elles n'existent pas physiquement.
Les agents sont capables d'agir, et non pas seulement de raisonner comme dans les systèmes d'IA classique. L'action, qui est un concept fondamental pour les systèmes multi-agents, repose sur le fait que les agents accomplissent des actions qui vont modifier l'environnement des agents et donc leurs prises de décision futures.
Ils peuvent aussi communiquer entre eux, et c'est d'ailleurs là l'un des modes principaux d'interaction existant entre les agents.
Ils agissent dans un environnement
Les agents sont doués d'autonomie. Cela signifie qu'ils ne sont pas dirigés par des commandes venant de l'utilisateur (ou d'un autre agent), mais par un ensemble de tendances qui peuvent prendre la forme de buts individuels `a satisfaire ou de fonctions de satisfaction ou de survie que l'agent cherche à optimiser. On pourrait dire ainsi que le moteur d'un agent, c'est lui-même. C'est lui qui est actif. Il a la possibilité de répondre par l'affirmative ou le refus à des requêtes provenant des autres agents. Il dispose donc d'une certaine liberté de manœuvre, ce qui le différencie de tous les concepts semblables, qu'ils s'appellent "objets", "modules logiciels" ou "processus". Mais l'autonomie n'est pas seulement comportementale, elle porte aussi sur les ressources. Pour agir, l'agent a besoin d'un certain nombre de ressources: énergie, CPU, quantité de mémoire, accès à certaines sources d'informations, etc. Ces ressources sont `a la fois ce qui rend l'agent non seulement dépendant de son environnement, mais aussi, en étant capable de gérer ces ressources, ce qui lui donne une certaine indépendance vis-à-vis de lui. L'agent est ainsi à la fois un système ouvert (il a

besoin d'éléments qui lui sont extérieurs pour survivre) et un système fermé (car les échanges qu'il a avec l'extérieur sont très étroitement réglementés).
Les agents n'ont qu'une représentation partielle de leur environnement, c'est-`a dire qu'ils n'ont pas de vision globale de tout ce qui se passe. C'est d'ailleurs ce qui se passe dans les réalisations humaines d'envergure (la fabrication d'un Airbus par exemple) dans lesquelles personne ne connaît tous les détails de la réalisation, chaque spécialiste n'ayant qu'une vue partielle correspondant à son domaine de compétence.
L'agent est ainsi une sorte "d'organisme vivant" dont le comportement, qui se résume à communiquer, à agir et, éventuellement, à se reproduire, vise à la satisfaction de ses besoins et de ses objectifs à partir de tous les autres éléments (perceptions, représentations, actions, communications et ressources) dont il dispose.

5.2.1.2- Des Objets aux Agents

D'un point de vue informatique, l'approche multi-agent peut être considérée comme une évolution du paradigme orienté-objet. Du point de vue conceptuel, un objet est simplement une structure de données à laquelle sont associées des fonctions. Les agents sont des entités autonomes, ce qui signifie que leur comportement ne dépend pas d'une pression extérieure, contrairement aux objets.
-Agent: entité autonome interagissant avec son environnement
-Objet: entité passive possédant un état et sur lequel on peut effectuer des opérations.
(Chaib-draa, 2010)
-Un agent est à un degré d'abstraction plus élevé qu'un objet.
Un agent peut être constitué de plusieurs objets.
-C'est un paradigme de programmation mettant en évidence l'autonomie et les interactions.
(Programmation orientée-agent)

Similarités :

Possèdent un «état interne»,
Des unités de comportement modulaires (méthodes/compétences),
Communiquent par envoi de messages,
Peuvent agir pour modifier leur état,

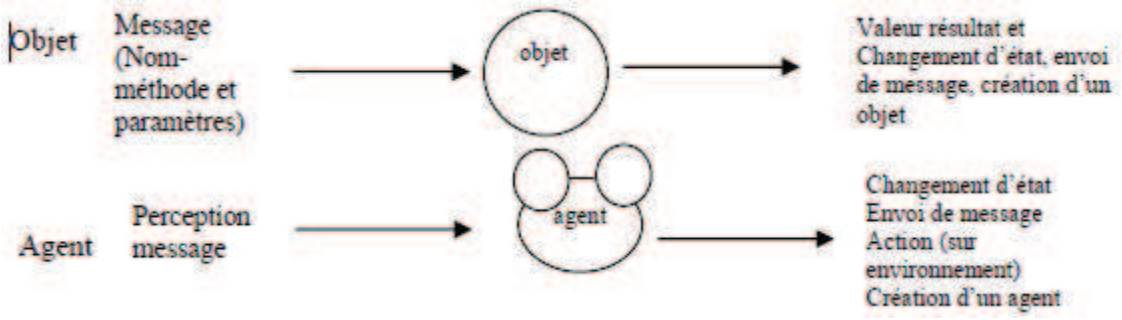


Figure 5.5 : Objet «versus» Agent

Différences entre objets et agents (Briot & Demazeau, 2001-2002)

Objet	**Agent**
Pas d'autonomie: l'objet qui reçoit un appel de méthode exécute celle-ci (pas de proactivité, de réactivité)	**Autonomie de contrôle**: l'agent décide de son comportement en fonction de son état, croyances, connaissances, perceptions de l'environnement, requêtes des autres

Peu de socialité: interaction rigide (pas d'évolution dans le temps) et simple	**Socialité**: composante très importante, complexité des interactions, des organisations
	Notion d'environnement importante et plus complexe

Tableau 5.1 : Différences entre objets et agents

5.2.2- Système Multi-Agents

5.2.2.1- Qu'est-ce qu'un système multi-agents ?

Précédemment on a présenté des systèmes où il n'y avait qu'un seul agent mais, dans la plupart des situations réelles, l'agent n'est pas seul dans son environnement, il y a d'autres agents présents autour de lui. Il nous faut donc aborder des systèmes où plusieurs agents doivent interagir entre eux pour effectuer leurs tâches. De tels systèmes sont appelés « systèmes multiagents » et ils possèdent les caractéristiques principales (Jennings & all, 1998) suivantes :
– chaque agent a des informations ou des capacités de résolution de problèmes incomplètes, donc chaque agent a un point de vue limité ;
– il n'y a pas de contrôle global du système ; les données sont décentralisées ;
– les calculs sont asynchrones.
La définition d'un système multi-agents (avec son acronyme SMA, et MAS pour « multi-agent system » en anglais) est plus immédiate : «Un ensemble d'agents qui agissent (et interagissent) dans un environnement commun ». Nous ne faisons que suivre ici la définition usuelle du terme système : « un ensemble organisé d'éléments ». Cela signifie que dans un système multi-agent, il existe une ou plusieurs organisations qui structurent les règles de cohabitation et de travail collectif entre agents (définition des différents rôles, partages de ressources, dépendances entre tâches, protocoles de coordination, de résolution de conflits, etc.). Dans un même système, il existe en général plusieurs organisations et un même agent peut appartenir à plusieurs simultanément. Des exemples d'organisations d'agents dans le monde réel sont une organisation économique telle qu'une entreprise, mais aussi une organisation animale telle qu'une fourmilière.
Selon (Ferber, 1995) : On appelle système multi-agent (ou SMA), un système composé des éléments suivants:

1. Un environnement E, c'est-`a-dire un espace disposant généralement d'une métrique.
2. Un ensemble d'objets O. Ces objets sont situés, c'est-`a-dire que, pour tout objet, il est possible, à un moment donné, d'associer une position dans E. Ces objets sont passifs, c'est-à-dire qu'ils peuvent être perçus, crées, détruits et modifiés par les agents.
3. Un ensemble A d'agents, qui sont des objets particuliers (A _ O), lesquels représentent les entités actives du système.
4. Un ensemble de relations R qui unissent des objets (et donc des agents) entre eux.
5. Un ensemble d'opérations Op permettant aux agents de A de percevoir, produire, consommer, transformer et manipuler des objets de O.
6. Des opérateurs chargés de représenter l'application de ces opérations et la réaction du monde à cette tentative de modification, que l'on appellera les lois de l'univers.

5.2.2.2- Utilité des systèmes multi-agents

Certains domaines requièrent l'utilisation de plusieurs entités, par conséquent, les systèmes multiagents sont très bien adaptés à ce type de situations. Par exemple, il y a des systèmes qui sont géographiquement distribués comme la coordination entre plusieurs frégates, le contrôle

aérien, les bases de données coopératives distribuées, etc. Tous ces domaines sont par définition distribués, par conséquent, les systèmes multiagents procurent une façon facile et efficace de les modéliser.
Une autre situation, où les systèmes multiagents sont requis, est lorsque les différents systèmes et les données qui s'y rattachent appartiennent à des organisations indépendantes qui veulent garder leurs informations privées et sécurisées pour des raisons concurrentielles. Par exemple, la majorité des missions maritimes se font maintenant en collaboration avec plusieurs pays. Donc, il y a plusieurs bateaux de pays différents qui doivent agir ensemble. Pour concevoir un système qui permettrait aux bateaux de se coordonner, il faudrait avoir des informations sur toutes les caractéristiques de chacun des bateaux. Par contre, aucun pays ne voudra donner ces informations puisqu'elles sont considérées comme des secrets militaires. De plus, les façons de faire diffèrent d'un pays à l'autre. Une solution à ce problème est de permettre à chaque pays de concevoir ses propres agents qui représenteront adéquatement ses buts et ses intérêts. Par la suite, ces agents pourront communiquer ensemble pour coordonner l'ensemble de la mission. Dans ce cas, les agents ne se transmettront que les informations nécessaires à une bonne coordination des bateaux.
Même si le domaine ne requiert pas l'utilisation des systèmes multiagents, il existe tout de même de bons avantages à utiliser ce type de système. Ainsi, ils peuvent s'avérer bien utiles pour des problèmes possédant de multiples méthodes de résolution, de multiples perspectives et/ou de multiples résolveurs. En particulier, les systèmes multiagents sont très utiles pour modéliser le raisonnement humain à l'intérieur de grandes simulations de combats aériens.
Ils possèdent également les avantages traditionnels de la résolution distribuée et concurrente de problèmes:
– La modularité permet de rendre la programmation plus simple. Elle permet, de plus, aux systèmes multiagents d'être facilement extensibles, parce qu'il est plus facile d'ajouter de nouveaux agents à un système multiagents que d'ajouter de nouvelles capacités à un système monolithique.
– La vitesse est principalement due au parallélisme, car plusieurs agents peuvent travailler en même temps pour la résolution d'un problème.
La fiabilité peut être également atteinte, dans la mesure où le contrôle et les responsabilités étant partagés entre les différents agents, le système peut tolérer la défaillance d'un ou de plusieurs agents. Si une seule entité contrôle tout, alors une seule défaillance de cette entité fera en sorte que tout le système tombera en panne.
Finalement, les systèmes multiagents héritent aussi des bénéfices envisageables du domaine de l'intelligence artificielle comme le traitement symbolique (au niveau des connaissances), la facilité de maintenance, la réutilisation et la portabilité.

5.2.2.3- Un premier exemple

Comme premier exemple, considérons le projet d'envoyer un robot sur une planète lointaine pour l'explorer. Le délai de communication avec la Terre est tel qu'il est difficile d'envisager un contrôle direct à partir du sol. Il est donc indispensable de doter un tel robot d'autonomie et d'initiative, de manière à prendre des décisions en fonction de la situation locale avec une réactivité suffisante. Ceci n'empêche pas bien sûr de maintenir une communication régulière avec le sol pour recevoir des informations et si besoin réajuster la mission.
Plutôt que d'envoyer un seul robot multi-spécialisé (locomotion, vision, prélèvement d'échantillons, analyse, communication avec la Terre, etc.), il peut être intéressant d'y substituer plusieurs robots. Ils pourront être spécialisés (par exemple, un robot spécialisé dans les prélèvements, un autre dans les analyses - ce dernier étant éventuellement immobile -, etc.). Mais ils devront coopérer et coordonner leurs activités à partir de leurs connaissances et comportements locaux. Une telle spécialisation selon les expertises rend une conception et

une construction plus aisées qu'un robot unique qui doit savoir tout faire. Une telle organisation offre également une meilleure efficacité potentielle (du fait des tâches pouvant être effectuées en parallèle). De plus si l'on assure une certaine redondance (par exemple plusieurs robots d'analyse), cela assure une certaine robustesse à l'ensemble du système en cas de panne d'un de ses éléments. Ceci est donc un premier exemple de système multi-agent. Dans ce cas, les agents sont robotiques (on parle de robotique collective). Un autre exemple est celui de la surveillance de réseaux où l'on délègue des tâches de surveillance, détection d'anomalies, diagnostic et réparations à différents agents logiciels coopératifs.

5.2.2.4- Vue intuitive d'un Agent dans un SMA

Un SMA peut-être :
Ouvert : les agents y entrent et en sortent librement (ex: un café)
Fermé : l'ensemble d'agents reste le même (ex: un match de football)
Homogène : tous les agents sont construit sur le même modèle (ex: une réunion de travail, une colonie de fourmis)
Hétérogène : des agents de modèles différents, de granularité différentes (ex: l'organisation hospitalière)

5.2.2.5- Variables globales et locales et les SMA

En programmation classique les variables globales sont accessibles à tous et les variables locales ne sont accessibles à personne.

```
Global X, Y
FuncfooLocal a,b
… a …
… X …
Funcbar Local u,v,w
… u,v…
… X …
```

Dans un SMA rien n'est complètement global

L'environnement étant vaste (sans compter le problème de la représentation récursive –je sais que je sais que je sais... —) et ouvert, il n'est pas possible en un lieu donné (par exemple, dans un agent) de stocker toute la représentation du monde. Par contre, un agent peut se déplacer ou encore interagir avec d'autres agents qui sont dans son voisinage pour explorer l'environnement.

Dans un SMA rien n'est complètement local

Pour un agent donné, toutes ses entités (informations, processus, buts, …) sont locales mais elles restent accessibles à l'introspection par d'autres agents. Le moyen d'accéder à cette information passe par les interactions entre les agents.

5.2.2.6- Niveaux d'organisation

En reprenant la classification proposée par G.Gurvitch (Gurvitch, 1963), maintenant traditionnelle en sociologie (Rocher, 1968), on peut distinguer trois niveaux d'organisation dans les systèmes multi-agents:

a- Le niveau micro-social, o`u l'on s'intéresse essentiellement aux interactions entre agents et aux différentes formes de liaison qui existent entre deux ou un petit nombre d'agents. C'est à ce niveau que la plupart des études ont été généralement entreprises en intelligence artificielle distribuée.

b- Le niveau des groupes, o`u l'on s'intéresse aux structures intermédiaires qui interviennent dans la composition d'une organisation plus complète. A ce niveau, on étudie les différenciations des rôles et des activités des agents, l'émergence de structures organisatrices entre agents et le problème général de l'agrégation des agents lors de la constitution d'organisations.

c- Le niveau des sociétés globales (ou populations) où l'intérêt se porte surtout sur la dynamique d'un grand nombre d'agents, ainsi que sur la structure générale du système et son évolution. Les recherches se situant dans le cadre de la vie artificielle se situent assez souvent à ce niveau.

5.2.3- Propriétés d'un agent intelligent

Un agent peut être caractérisé par plusieurs propriétés, nous pouvons citer parmi elles :

5.2.3.1- Autonomie

Wooldridge et Jennings (Wooldridge & Jennings, 1994) définissent l'autonomie comme étant la capacité pour un agent d'opérer de manière autonome sans une intervention directe d'humains ou d'autres agents et de contrôler ses actions et son état interne. Quant à C.CastelFranchi dans (Castelfranchi, 1995), définit un agent autonome comme étant un agent qui a ses propres buts et qui est capable de décider des buts à poursuivre, comment les atteindre et résoudre les conflits internes relatives aux buts choisis. Pour P.R.Cohen, H.J.Levesque dans (Cohen & Levesque, 1995) et Y.Shoham dans (Shoham 1993), un agent est capable :

- d'agir selon ses intentions ;
- d'adopter les buts qu'il croit réalisables et d'abandonner ceux qu'il croît
- irréalisables ;
- de planifier ses propres actions et de tenir compte de celles des autres ;
- de raisonner non seulement sur ses connaissances mais aussi sur celles des autres.

5.2.3.2- Réactivité

Les agents perçoivent leur environnement et réagissent aux changements qui s'y produisent (Wooldridge & Jennings, 1994).
La réactivité signifie aussi la capacité qu'a un agent de modifier son comportement lorsque les conditions environnementales changent (Oliveira, 1998).
Un objet est une entité passive(ou réactive), si personne ne demande la valeur d'un attribut ou n'active une méthode de l'objet, il ne se passe rien (Sansonnet, 2002).

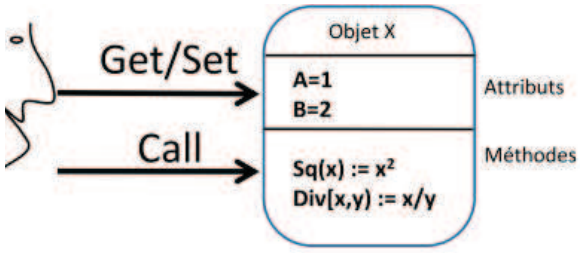


Figure 5.6 : Réactivité

5.2.3.3- Proactivité

Les agents n'agissent pas simplement en réponse à leur environnement, ils sont capables d'exhiber un comportement guidé par des buts en prenant des initiatives (Wooldridge & Jennings, 1994). La proactivité est la capacité d'un agent d'anticiper des situations et de changer son cours d'action pour les éviter (Oliveira, 1998).
Un agent possède, en plus des attributs et méthodes, des processus internes qui fonctionnent même en l'absence de sollicitations externes. Un agent peut donc agir même si personne ne lui demande rien.

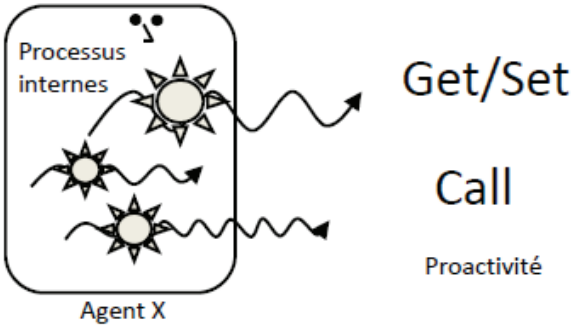


Figure 5.7 : Proactivité

5.2.3.4- Adaptabilité

Face à un environnement perpétuellement changeant, un agent doit constamment modifier le plan qu'il poursuit pour atteindre un but. Pour cela :
Il doit, de manière continue, percevoir et évaluer la situation (contexte) de son action.
Construire des représentations en cours même de fonctionnement (c'est à dire être capable d'apprendre).
Élaborer des plans dynamiques qui lance des processus internes ou au contraire les stoppent.
L'adaptabilité est la capacité d'un agent de s'adapter à l'environnement dans lequel il est situé (Oliveira, 1998). Un agent adaptatif est un agent capable de contrôler ses aptitudes (communicationnelles et comportementales) selon l'environnement dans lequel il évolue et selon l'agent avec lequel il interagit (Guessoum, 1996).
L'adaptabilité est une propriété très importante pour un agent qui évolue dans un environnement dynamique.

5.2.3.5- Sociabilité

La sociabilité est la capacité d'un agent de s'intégrer dans un environnement peuplé d'agents avec qui il échange des messages pour accomplir un but (Oliveira, 1998).
Nous reviendrons sur cette propriété lorsque nous aborderons les propriétés d'un SMA car elle intègre d'autres propriétés telles que la communication, la coopération et la délégation.

5.2.3.6- Apprentissage

L'apprentissage est une propriété assez particulière car un agent n'est pas forcément une entité capable d'apprendre. Les agents peuvent avoir à apprendre lorsqu'ils réagissent et/ou interagissent avec leur environnement externe (Nwana, 1996).
L'apprentissage est une propriété qui fournit aux systèmes la capacité d'acquérir la compréhension de certains comportements au cours du temps, sans nécessiter que ces comportements soient programmés manuellement (Guessoum & all, 2006).

Un agent a la faculté d'apprendre s'il est capable d'utiliser de nouvelles connaissances pour modifier son comportement (Cheikhrouhou & all, 1998).
Bien que l'apprentissage soit un attribut clé de l'intelligence (Nwana, 1996), il existe très peu d'agents qui ont cette capacité.

5.2.3.7- Sécurité

La sécurité est une propriété importante, notamment dans le contexte de ce travail, car elle permet de garantir que lorsque l'on interagit avec un agent, que cet agent n'a pas été corrompu par un virus, par de fausses croyances ou par des connaissances qui n'ont pas de sens (Oliveira, 1998).

5.2.4- Propriétés des systèmes multi-agents

5.2.4.1- Interactions entre agents

Jacques Ferber donne la définition suivante de l'interaction : " Une interaction est la mise en relation dynamique de deux ou plusieurs agents par le biais d'un ensemble d'actions réciproques. Les interactions sont non seulement la conséquence d'actions effectuées par plusieurs agents en même temps, mais aussi l'élément nécessaire à la constitution d'organisations sociales. " (Ferber, 1995).
Les systèmes multiagents ont surtout l'avantage de faire intervenir des schémas d'interaction sophistiqués. Ils peuvent ainsi coexister, être en compétition ou coopérer.
S'ils ne font que coexister, alors chaque agent ne considère les autres agents que comme des composantes de l'environnement, au même titre que toutes les autres composantes.
Si les agents ont une représentation physique, les autres agents ne seront vus que comme des obstacles que l'agent doit éviter. Il s'ensuit qu'il n'y a aucune communication directe entre les agents. En fait, il peut y avoir une certaine forme de communication indirecte parce que les agents peuvent se percevoir les uns les autres. Le but visé n'est toutefois pas de communiquer avec l'autre. Ces informations ne servent qu'à mieux éviter les autres agents. Par exemple, si l'on considère une personne marchant dans une foule d'étrangers, elle communique avec les autres à l'aide de gestes ou de mouvements, mais uniquement dans le but de pouvoir circuler sans accrocher tout le monde. (Chaib-Draa & Gageut, 2002)
S'ils sont en compétition, alors le but de chaque agent est de maximiser sa propre satisfaction, ce qui se fait généralement aux dépens des autres agents. La situation de compétition la plus fréquente se produit lorsque plusieurs agents veulent utiliser ou acquérir la même ressource. Les agents doivent donc pouvoir communiquer entre eux pour résoudre le conflit. Cette communication prend habituellement la forme d'une négociation. Les agents se transmettent des propositions et des contre-propositions jusqu'à ce qu'ils arrivent à une entente ou qu'ils se rendent compte qu'une entente est impossible. Ce type de communication demande un protocole de négociation et un langage de communication de haut niveau du type de KQML (Finin & Fritzson, 1994) ou FIPA-ACL (Fipa, 2000) pour permettre une certaine structure dans la négociation.
S'ils sont en coopération, alors le but des agents n'est plus seulement de maximiser sa propre satisfaction mais aussi de contribuer à la réussite du groupe. Les agents travaillent ensemble à la résolution d'un problème commun. Dans ce type de système, les agents communiquent ensemble, à l'aide de messages plus ou moins sophistiqués, dans le but d'améliorer la performance du groupe. Ils peuvent s'échanger des informations sur l'environnement pour augmenter leurs perceptions individuelles, ou bien se transmettre leurs intentions pour que les agents puissent avoir une idée de ce que les autres font. Somme toute, les communications servent aux agents à améliorer leur coordination, c'est-à-dire à organiser la résolution du

problème de telle sorte que les interactions nuisibles soient évitées et/ou que les interactions bénéfiques soient exploitées.

Nous avons abordé la *sociabilité* dans la section précédente mais en réalité cette propriété est centrée sur les interactions entre agents (Guessoum, 1996). Nous allons donc voir maintenant les différentes propriétés qui sont impliquées dans la sociabilité.

5.2.4.2- Coopération

Pour Ferber (Ferber, 1995), pour que plusieurs agents soient dans une situation de coopération, il faut que l'une des deux conditions suivantes soit vérifiée :
1. l'ajout d'un nouvel agent permet d'accroître différentiellement les performances du groupe
2. l'action des agents sert à éviter ou à résoudre des conflits potentiels ou actuels.

La coopération entre agents peut être (Oliveira, 1998) :
Une approche multiagents pour la gestion de sécurité

- soit ***implicite*** et dans ce cas les agents ont un but commun implicite qu'ils doivent atteindre en exécutant des actions indépendantes. Dans ce type de coopération, la communication entre les agents est facultative.
- soit ***explicite*** et dans ce cas, les agents exécutent des actions qui leur permettent d'achever non seulement leurs propres buts mais aussi les buts des autres agents.

Ce type de coopération nécessite une communication entre les agents pour prendre connaissance des buts des autres agents.

5.2.4.3- Coordination

Dans un système multi-agents, la coordination des actions des différents agents permet d'assurer une cohérence du système. Il existe plusieurs mécanismes de coordinations parmi lesquels, nous retrouvons : l'organisation, la planification et la synchronisation (Guessoum, 1996), (Oliveira, 1998). Nous définissons ces trois mécanismes.

a- L'organisation :
Une organisation représente un groupe d'agents qui travaillent ensemble afin de réaliser une ou plusieurs tâches (Guessoum, 1996). De nombreux travaux sur l'organisation ont été proposés. T.Bouron (Bourron, 1992) les a regroupés en deux classes où l'organisation est définie :

- Soit comme une structure externe aux agents et elle est représentée par un objet ou un agent.
- Soit comme un objet abstrait dont la représentation est distribuée parmi les membres de l'organisation.

Parmi les travaux existants, nous pouvons citer le *réseau contractuel* introduit par R.Smith (Smith, 1980). Le *réseau contractuel* est un mécanisme d'allocation de tâches fondé sur la notion d'appel d'offre (Ferber, 1995).
Dans ce modèle, un agent peut avoir deux rôles par rapport à une tâche (Guessoum, 1996) : *manager* ou *contractant*. L'appel d'offre s'effectue en quatre étapes :

- ***appel d'offre*** : le *manager* décompose une tâche en sous-tâches. Il recherche ensuite des *contractants* pour les réaliser en faisant une annonce de tâches à tous les agents du système ;
- ***envoi de propositions*** : les agents élaborent une proposition et font une offre au *manager* ;
- ***attribution du marché*** : le *manager* évalue les propositions, attribue la tâche à un ou plusieurs agents et informe les autres agents de son choix ;

- ***établissement du contrat*** : l'agent ou les agents sélectionnés deviennent *contractants* et informent le *manager* qu'ils s'engagent à réaliser la tâche.

b- La planification :

La planification multiagents est une autre approche de coordination dans les systèmes à base d'agents. Afin d'éviter des actions conflictuelles ou inconsistantes, les agents construisent un plan multi-agents qui détaille toutes les actions futures et les interactions nécessaires pour atteindre leurs buts (Oliveira, 1998).
La planification dans les SMAs, se décompose en trois étapes (Ferber, 1995) :

- la construction de plans,
- la synchronisation et coordination des plans,
- l'exécution de ces plans.
  Il existe deux types de planification, dans les SMAs :
- la ***planification centralisée*** pour agents multiples qui suppose l'existence d'une vue globale du plan (Guessoum, 1996). Dans cette approche, il existe un seul agent capable de planifier et d'organiser les actions pour l'ensemble des agents (Ferber, 1995).
- la ***planification distribuée*** où chaque agent planifie individuellement ses actions en fonction de ses propres buts (Ferber, 1995). Les différents agents se communiquent ensuite leurs plans partiels afin de détecter et d'éviter les conflits éventuels.

c- La synchronisation :

La synchronisation est le "bas niveau" de la coordination où sont implémentés les mécanismes de base permettant aux différentes actions de s'articuler correctement. Elle permet de synchroniser l'enchaînement des actions des différents agents (Ferber, 1995).
J. Ferber répertorie deux types de synchronisation :

- La ***synchronisation par mouvement*** utilisée lorsque plusieurs éléments doivent se déplacer ensemble. Dans ce type de synchronisation, il s'agit de coordonner le rythme et le positionnement dans le temps d'actions en fonction des événements qui se produisent ;
- La ***synchronisation d'accès à une ressource*** utilisée lorsque plusieurs agents doivent partager une ressource.

5.2.4.4- La compétition

La compétition entre agents peut avoir plusieurs sources. Les buts des agents peuvent être incompatibles : dans une situation de jeu, chacun cherche à être gagnant mais la réalisation de ce but pour un des joueurs rend impossible la réalisation du but des autres joueurs. Les ressources dont les agents ont besoin peuvent être rares et l'utilisation d'une ressource par un des agents peut empêcher un autre agent de réaliser son but. Par exemple, si l'imprimante est utilisée je dois attendre pour imprimer mon article et je ne pourrai pas le poster avant le départ du courrier. La compétition crée des situations de conflit qui peuvent être résolues par la lutte ou par la coordination (Drogoul, 1993).

5.2.4.5- Délégation

La délégation est la capacité d'un agent à exécuter des tâches pour le compte d'un tiers. Elle permet ainsi à un agent, qui ne peut pas atteindre ses buts par manque de ressources, de compétences ou dans le but d'alléger sa tâche, de demander à d'autres agents d'achever des buts pour son compte. Cette propriété est cruciale lorsque la coordination est supportée par une structure organisationnelle (Oliveira, 1998) et que l'environnement du SMA varie et

nécessite par conséquent une redistribution et mise à jour des tâches que doivent accomplir les différents agents (ce qui se traduit par une délégation de nouveaux buts). Dans une organisation hiérarchique par exemple, lorsque le SMA doit atteindre de nouveaux objectifs, elle permet à une entité gestionnaire de déléguer des sous-tâches à des agents sous-jacents et de les modifier.
La délégation est beaucoup utilisée dans la gestion de réseau où elle permet une flexibilité du système qui l'utilise (Magendaz, 1995).

5.2.4.6- Communication

Dans un SMA, les agents communiquent entre eux en s'échangeant des informations via un langage de communication agent (Wooldridge & Jennings, 1994). Le langage, le plus utilisé aujourd'hui est KQML (Knowledge Query and Manipulation Language) qui est un langage de haut niveau utilisant une liste de types de messages, appelés performatifs (Ferber, 1995). Les agents peuvent également communiquer en utilisant d'autres mécanismes tels que le mécanisme de tableau noir (Guessoum 1996).

5.2.4.7- Une Recherche de Compromis

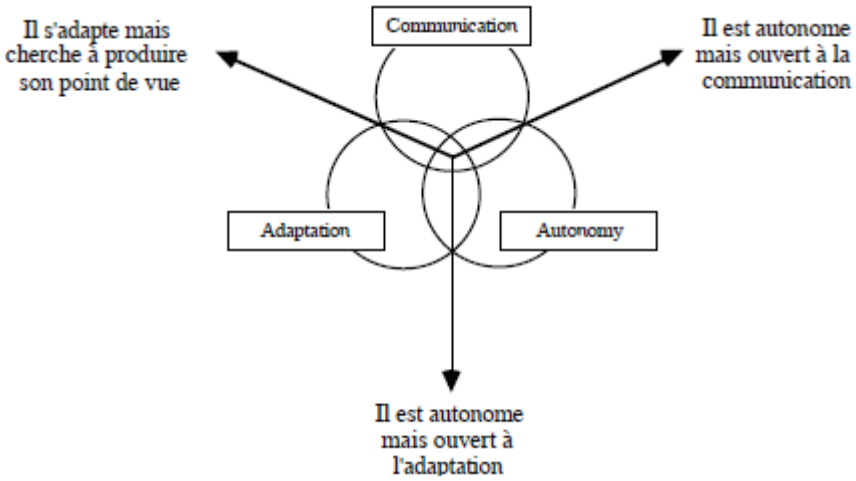


Figure 5.8 : Le compromis recherché (SMA)

5.3- Les différents modèles d'agents (Architecture)

Faut-il concevoir les agents comme des entités déjà "intelligentes", c'est-à-dire capables de résoudre certains problèmes par eux-mêmes, ou bien faut-il les assimiler à des êtres très simples réagissant directement aux modifications de l'environnement?
Deux grandes écoles émergent, celles des *Agents cognitifs* et celles des *réactifs* :

- les systèmes ***délibératifs***, dits aussi ***cognitifs***,
- les systèmes ***réactifs***. (Boissier, 2001)

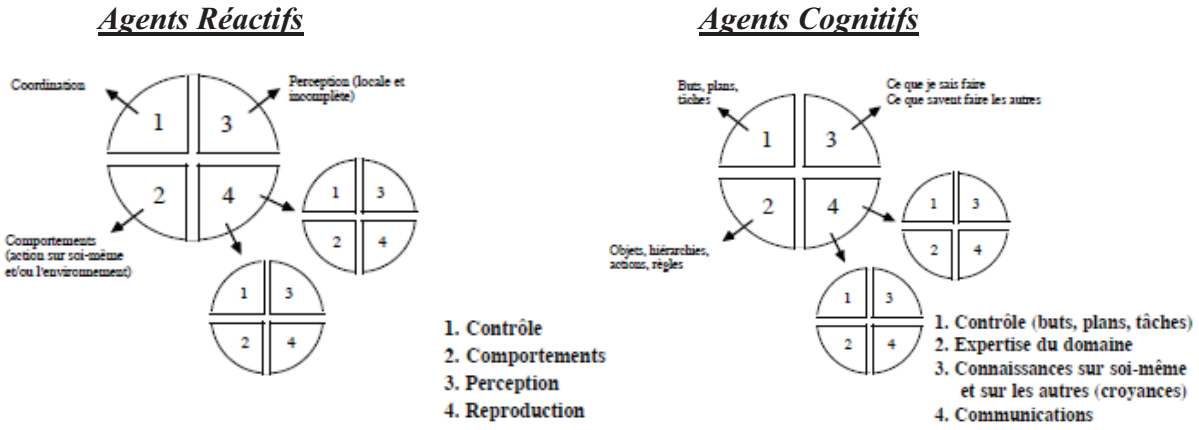


Figure 5.9 : Agents réactifs et cognitifs

Ces deux conceptions ont donné lieu à deux écoles de pensée. La première, l'école "cognitive", est la plus représentée dans le domaine que l'on appelle intelligence artificielle distribuée (IAD) car elle trouve son origine dans la volonté de faire communiquer et coopérer des systèmes experts classiques. Dans ce cadre, un système multi-agent est composé d'un petit nombre d'agents "intelligent". Chaque agent dispose d'une base de connaissance comprenant l'ensemble des informations et des savoir-faire nécessaires à la réalisation de sa tâche et à la gestion des interactions avec les autres agents et avec son environnement. On dit aussi que les agents sont "intentionnels", c'est-`a-dire qu'ils possèdent des buts et des plans explicites leur permettant d'accomplir leurs buts. Dans ce cadre, les problèmes de coopération ressemblent étonnamment à ceux de petits groupes d'individus, qui doivent coordonner leur activité et sont parfois amenés à négocier entre eux pour résoudre leurs conflits (Bond & Gasser, 1988), (Demazeau & Müller, 1991), (Chaib-Draa & all, 1992), (Briot & all, 2006).
Les analogies sont alors sociales et nombre de chercheurs dans ce domaine s'appuient sur les travaux de sociologie et en particulier sur la sociologie des organisations et des petits groupes.
L'autre tendance, l'´ecole "réactive", prétend au contraire qu'il n'est pas nécessaire que les agents soient intelligents individuellement pour que le système ait un comportement global intelligent (Deneubourg & all, 1991), (Ferber & Drogoul, 1992).Des mécanismes de réaction aux événements, ne prenant en compte ni une explicitation des buts, ni des mécanismes de planification, peuvent alors résoudre des problèmes qualifiés de complexes. L'exemple le plus manifeste d'organisation émergente est celle de la fourmilière (Corbara & all, 1993): alors que toutes les fourmis se situent sur un plan d'´egalité et qu'aucune d'entre elles ne possède de pouvoir d'autorité stricte sur les autres, les actions des fourmis se coordonnent de manière que la colonie survive et fasse donc face à des problèmes complexes tels que ceux posés par la recherche de nourriture, les soins à donner aux œufs et aux larves, la construction de nids, la reproduction, etc.
Cependant, dans la littérature, sont répertoriés quatre différents types de modèles d'agents, à savoir :

– les agents à réflexes simples,
– les agents conservant une trace du monde,
– les agents ayant des buts,
– les agents utilisant une fonction d'utilité. (Russell & Norvig, 1995)

Les deux premiers types d'agents sont considérés comme des agents réactifs et les deux derniers types sont considérés comme des agents délibératifs ou cognitifs.
Il existe un grand nombre de typologies d'agents mais celle que nous retenons, compte tenu du problème à modéliser, est fondée sur le processus de prise de décision de l'agent.
Elle distingue trois types d'agents : réactifs, cognitifs ou délibératifs et hybrides.

5.3.1- Les agents réactifs

L'idée d'architectures d'*agents réactifs* a été essentiellement introduite par Brooks (Brooks 1991). Les architectures *réactives*, dites *aussi comportementales*, se caractérisent par des agents qui ont la capacité de réagir rapidement à des problèmes simples, qui ne nécessitent pas un haut niveau de raisonnement, comme son nom l'indique, un agent réactif ne fait que réagir aux changements qui surviennent dans leur environnement ou aux messages provenant des autres agents. Autrement dit, un tel agent ne fait ni délibération ni planification, il se contente simplement d'acquérir des perceptions et de réagir à celles-ci en appliquant certaines règles prédéfinies. Étant donné qu'il n'y a pratiquement pas de raisonnement, ces agents peuvent agir et réagir très rapidement.
Il convient de remarquer que les humains aussi utilisent cette manière d'agir. Dans plusieurs situations, il est souvent préférable de ne pas penser et de réagir immédiatement.
Par exemple, lorsqu'une personne met la main sur une plaque très chaude, elle ne commence pas à se demander si c'est chaud, si ça fait mal et s'il faut ou non qu'elle retire sa main. Dans ce cas, elle retire sa main immédiatement, sans réfléchir, et c'est cette rapidité de réaction qui lui permet de diminuer les blessures. Cet exemple montre bien que ce type de comportement réflexe est essentiel pour les êtres humains. De la même manière, il est aussi essentiel pour les agents s'ils veulent pouvoir agir dans le monde réel.

En effet, leurs décisions sont essentiellement basées sur un nombre très limité d'informations et sur des règles simples de type *situation - action* (Müller, 1996). Ce type d'architectures ne nécessite aucune représentation symbolique de l'environnement réel. Ces agents ne disposent que d'un protocole de communication et d'un langage de communication réduits (Guessoum, 1996). Dans un système à base d'agents réactifs, l'intelligence émerge de l'interaction des agents avec leur environnement.
Les deux sous-sections suivantes décrivent deux modèles qui peuvent servir à la conception d'agents réactifs.

5.3.1.1- Agents à réflexes simples

Ce type d'agent agit uniquement en se basant sur ses perceptions courantes. Il utilise un ensemble de règles prédéfinies, du type **Si** *condition* **alors** *action*, pour choisir ses actions. Par exemple, pour un agent en charge du contrôle de la défense d'une frégate, on pourrait avoir la règle suivante :
Si *missile-en-direction-de-la-frégate* **alors** *lancer-missile-d'interception*
Comme on peut le constater, ces règles permettent d'avoir un lien direct entre les perceptions de l'agent et ses actions. Le comportement de l'agent est donc très rapide, mais peu réfléchi. À chaque fois, l'agent ne fait qu'exécuter l'action correspondant à la règle activée par ses perceptions.

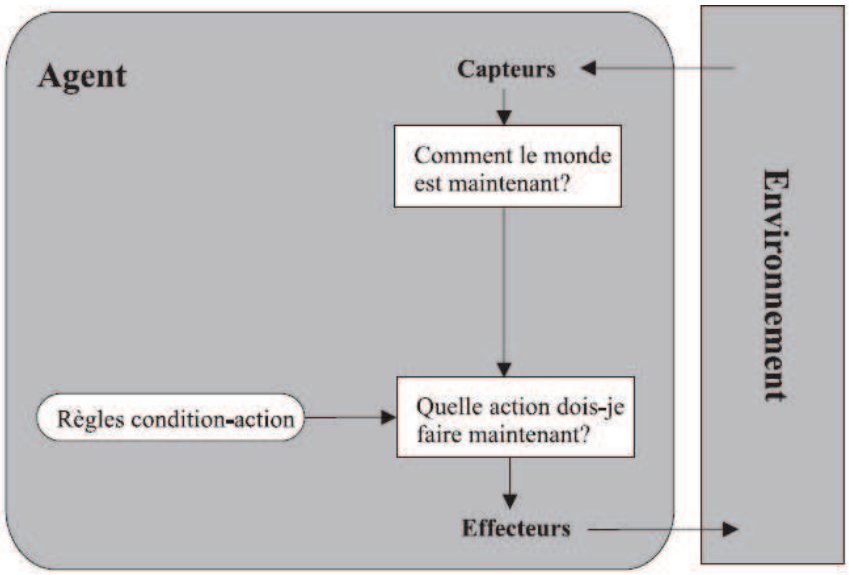


Figure 5.10 : Schéma d'un agent à réflexes simples (Russell & Norvig, 1995)

La figure montre l'architecture d'un agent à réflexes simples. Les rectangles représentent l'état interne de l'agent dans son processus de décision et les ovales représentent les informations qui sont utilisées dans le processus. L'agent se bâtit une représentation du monde à l'aide de ses perceptions lui venant de ses capteurs. Par la suite, il utilise ses règles pour choisir l'action qu'il doit effectuer selon ce qu'il perçoit de l'environnement.

5.3.1.2- Agents conservant une trace du monde

Le type d'agent qui a été décrit précédemment, ne peut fonctionner que si un tel agent peut choisir ses actions en se basant uniquement sur sa perception actuelle. Par exemple, si le radar de la frégate détecte un missile et que l'instant d'après, il le perd de vue, dû à un obstacle, cela ne signifie nullement qu'il n'y a plus de missile. Dès lors, l'agent en charge du contrôle de la frégate doit tenir compte de ce missile dans le choix de ses actions et ce, même si le radar ne détecte plus le missile.
Le problème que nous venons de mentionner survient parce que les capteurs de l'agent ne fournissent pas une vue complète du monde. Pour régler ce problème, l'agent doit maintenir des informations internes sur l'état du monde dans le but d'être capable de distinguer deux situations qui ont des perceptions identiques, mais qui, en réalité, sont différentes. L'agent doit pouvoir faire la différence entre un état où il n'y a pas de missile et un état où le missile est caché, même si ses capteurs lui fournissent exactement les mêmes informations dans les deux cas.
Pour que l'agent puisse faire évoluer ses informations internes sur l'état du monde, il a besoin de deux types d'information. Tout d'abord, il doit avoir des informations sur la manière dont le monde évolue, indépendamment de l'agent. Par exemple, il doit savoir que si un missile avance à une vitesse de 300 m/s, alors 5 secondes plus tard, il aura parcouru 1500 mètres. L'agent doit avoir ensuite des informations sur la manière dont ses propres actions affectent le monde autour de lui. Si la frégate tourne, l'agent doit savoir que tout ce qui l'environne tourne aussi. Il doit donc mettre à jour la position relative de tous les objets autour de la frégate.

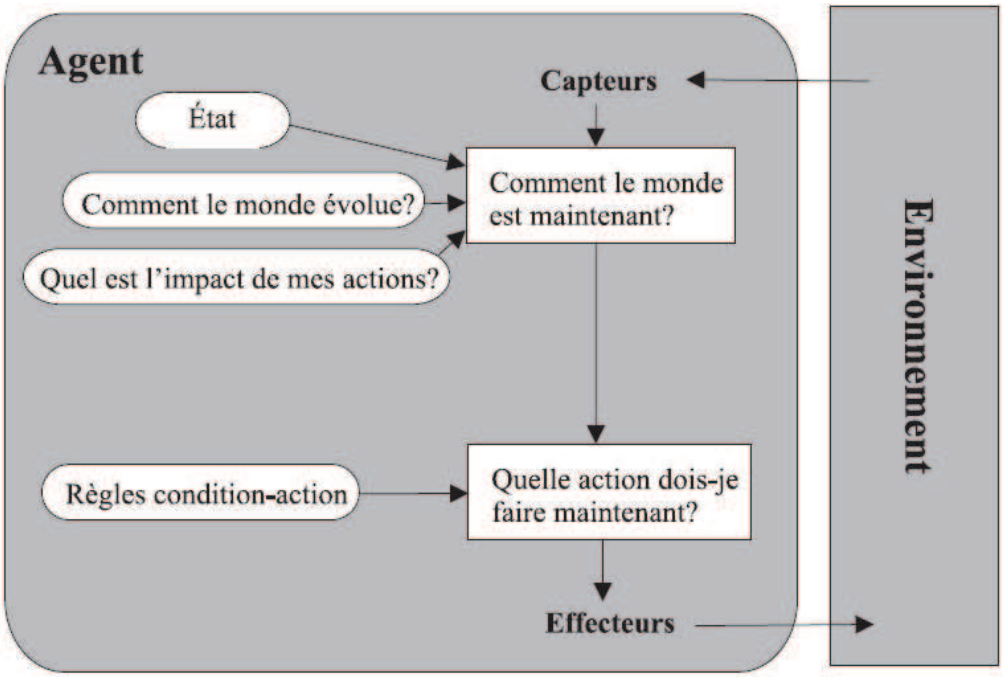


Figure 5.11 : Schéma d'un agent conservant une trace du monde (Russell & Norvig, 1995)

On peut voir, sur la figure, la structure d'un agent conservant une trace du monde. Il utilise ses informations internes (état précédent du monde, l'évolution du monde et l'impact de ses actions) pour mettre à jour ses perceptions actuelles. Par la suite, il choisit ses actions en se basant sur cette perception « améliorée » du monde qui l'entoure.

Les principales inspirations de ce type d'agents sont :

- Insectes sociaux, mammifères sociaux
- fourmis, termites, abeilles, guêpes, loups, rats, primates, oiseaux, poissons
- Eco-systèmes
- Phytosociologie
- Processus physico-chimiques

5.3.2- Les agents délibératifs

Les ***agents délibératifs*** ont la capacité de résoudre des problèmes complexes. Ils sont ainsi capables de raisonner sur une base de connaissances, de traiter des informations diverses liées au domaine d'application et des informations relatives à la gestion des interactions avec d'autres agents et avec l'environnement (Guessoum, 1996). Ces agents maintiennent une représentation interne de leur monde et un état mental explicite qui peut être modifié par un raisonnement symbolique (Müller, 1996).

Les SMAs délibératifs ont deux problèmes majeurs (Guessoum, 1996) :

- la traduction de l'univers de l'agent en une description symbolique ;
- la représentation symbolique des informations de l'univers complexe des entités et des processus ainsi que la manière de raisonner sur ces informations.

Les agents délibératifs sont des agents qui effectuent une certaine délibération pour choisir leurs actions. Une telle délibération peut se faire en se basant sur les buts de l'agent ou sur une certaine fonction d'utilité. Elle peut prendre la forme d'un plan qui reflète la suite d'actions que l'agent doit effectuer en vue de réaliser son but.

5.3.2.1- Agents ayant des buts

Dans la section précédente, les agents utilisaient leurs connaissances sur l'état actuel de l'environnement pour choisir leurs actions. Toutefois, dans plusieurs situations, cela peut s'avérer insuffisant pour prendre une décision sur l'action à effectuer. Par exemple, on ne peut pas se fier uniquement à l'état actuel de l'environnement pour déterminer la direction que la frégate doit prendre, tout simplement parce que cela dépend aussi de l'endroit où on veut se rendre.
Donc, l'agent a besoin, en plus de la description de l'état actuel de son environnement, de certaines informations décrivant ses buts. Lesquels buts peuvent être vus comme des situations désirables pour l'agent, par exemple, l'arrivée au port d 'Alger.
Par la suite, l'agent peut combiner les informations sur ses buts avec les informations sur les résultats de ses actions pour choisir les actions qui vont lui permettre d'atteindre ses buts. Cela peut être facile lorsque le but peut être satisfait en exécutant seulement une action, mais cela peut aussi être beaucoup plus complexe si l'agent doit considérer une longue séquence d'actions avant d'atteindre son but. Dans ce dernier cas, il doit utiliser des techniques de planification pour prévoir les actions devant l'amener à son but.
Contrairement aux agents réactifs, les agents délibératifs, qui raisonnent sur les buts, tiennent compte d'une certaine projection dans le futur. Ils se posent des questions comme « Qu'est-ce qui va arriver si je fais telle ou telle action ? » et « Est-ce que je serai satisfait si cela se produit ? ». Bien entendu, l'agent raisonnant sur ses buts prend, en général, beaucoup plus de temps à agir qu'un agent réactif. Il offre en revanche beaucoup plus de flexibilité. Par exemple, si nous voudrions changer de destination, il faudrait changer toutes les règles de l'agent réactif, tandis que pour l'agent ayant des buts, nous ne changerions que le but.

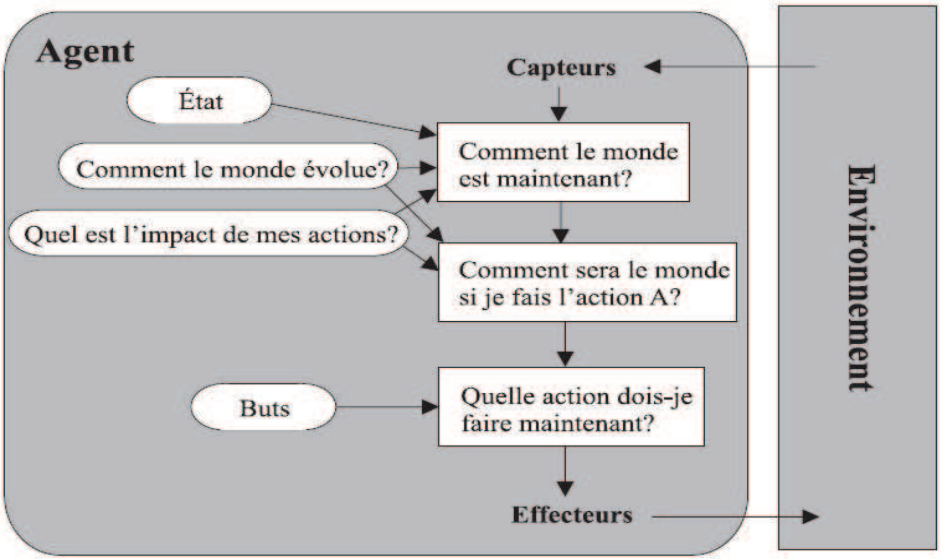


Figure 5.12 : Schéma d'un agent ayant des buts (Russell & Norvig, 1995)

La figure montre la structure d'un agent basé sur les buts. Comme on peut le constater, il est identique à l'agent réactif gardant une trace de l'environnement, sauf qu'il se projette dans le futur pour voir l'impact de ses actions et qu'il choisit ses actions en se basant sur ses buts, contrairement à l'agent réactif qui ne faisait qu'appliquer des règles prédéfinies pour relier ses perceptions à ses actions.

5.3.2.2- Agents utilisant une fonction d'utilité

Dans plusieurs situations, les buts ne sont pas suffisants pour générer un comportement de haute qualité. Par exemple, s'il y a plusieurs chemins possibles pour atteindre le port, certains

seront plus rapides et d'autres plus dangereux. Dans cette situation, l'agent raisonnant uniquement sur ses buts n'a pas de moyens pour choisir le meilleur chemin, son seul but étant de se rendre à destination. Cela se produit car les buts ne procurent qu'une simple distinction entre les états où l'agent est satisfait ou non. En fait, l'agent doit plutôt s'appuyer sur une manière plus fine d'évaluer les états pour être en mesure de reconnaître pour chacun des états son degré de satisfaction. Pour cela, on dit que l'agent va préférer un état à un autre si son utilité est plus grande dans le premier état que dans le deuxième.
Généralement, l'utilité est une fonction qui attribue une valeur numérique à chacun des états. Plus l'état a une grande valeur, plus il est désirable pour l'agent. Dès lors, la spécification d'une fonction d'utilité permet à l'agent de prendre des décisions rationnelles dans deux types de situations où le raisonnement sur les buts échoue. Ainsi, par exemple, lorsqu'il y a des buts conflictuels qui ne peuvent pas être satisfaits en même temps (par exemple, la vitesse et la sécurité), la fonction d'utilité spécifie le compromis approprié entre les différents buts. De même, lorsqu'il y a plusieurs buts possibles, mais qu'aucun d'eux ne peut être atteint avec certitude, la fonction d'utilité permet de pondérer la chance de succès avec l'importance de chacun des buts.

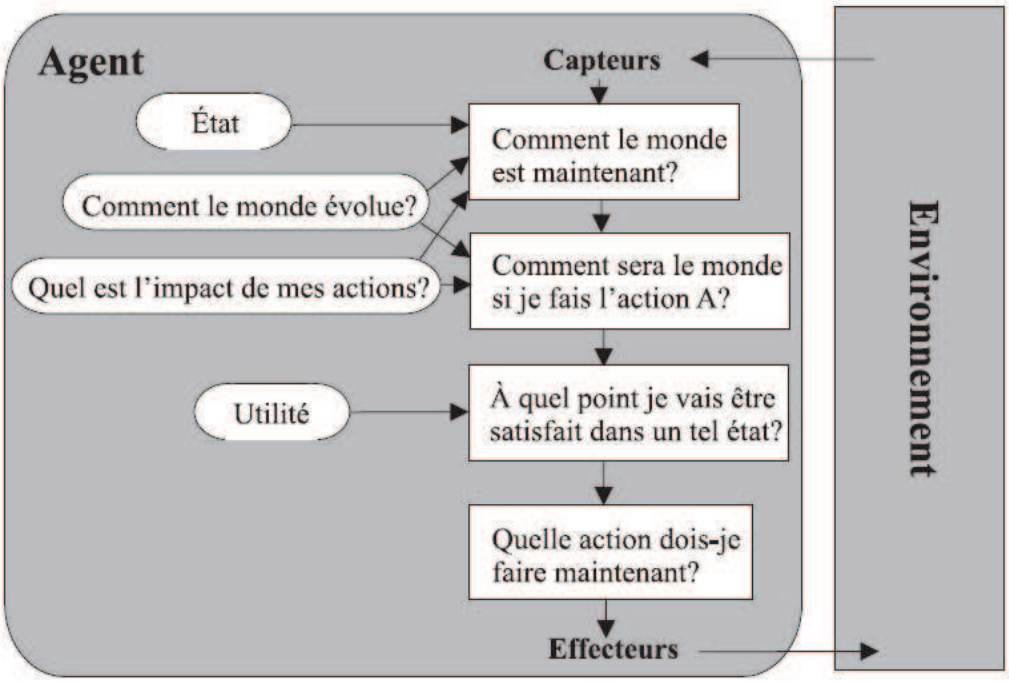


Figure 5.13 : Schéma d'agent basé sur l'utilité (Russell & Norvig, 1995)

La figure montre le schéma d'un agent basé sur l'utilité. On peut voir que l'agent utilise la fonction d'utilité pour évaluer la pertinence d'une action. Il choisit donc les actions qui l'amèneront dans les états ayant la plus grande valeur d'utilité pour lui.

5.3.2.3- Le modèle BDI

L'idée de base de l'approche BDI est de décrire l'état interne d'un agent en termes d'*attitudes mentales* et de définir une architecture de contrôle grâce à laquelle l'agent peut sélectionner le cours d'action de ses attitudes mentales. Il a été défini trois attitudes mentales de base qui sont : les *croyances* (beliefs) les *désirs* (desires) et les *intentions* (intentions). Dans des approches BDI plus pratiques tel que IRMA par exemple (Müller, 1996), il a été montré que ces trois attitudes mentales n'étaient pas suffisantes, une extension a alors été proposée en rajoutant la notion de *buts* (goals) et de *plans* (plans). Nous allons maintenant définir de manière informelle ces différents concepts :

- Les ***croyances*** décrivent l'état de l'environnement du point de vue d'un agent (Ferber, 1995). Elles expriment ce que l'agent croît sur l'état courant de son environnement (Müller, 1996).
- Les ***désirs*** sont une notion abstraite qui spécifie les préférences sur l'état futur de l'environnement d'un agent. Une caractéristique importante des *désirs* est qu'un agent peut avoir des *désirs* inconsistants et qu'il n'a donc pas à croire que ses désirs sont réalisables (Müller, 1996).
- Les ***buts*** représentent les engagements d'un agent pour atteindre un ensemble d'états de l'environnement (Müller, 1996). A partir de la définition précédente, on peut dire qu'un *désir* est une étape dans le processus de création d'un *but*. Si un *désir* d'un agent est poursuivi de manière consistante, il devient l'un des *buts* qui indiquent les options de gestion qu'a un agent (Oliveira, 1998). Cependant, il n'y a, à ce moment là, aucun engagement pour l'exécution de cours d'actions spécifiques. La notion d'engagement d'atteindre un *but* décrit la transition des *buts* aux *intentions*. De plus, il est nécessaire, que l'agent croît que ses *buts* peuvent être atteints (Müller, 1996).
- Les ***intentions*** représentent les actions que l'agent s'engage à exécuter. A partir du moment où les agents sont limités par leurs ressources, ils peuvent leur arriver de ne pas pouvoir poursuivre tous leurs *buts*. Même si l'ensemble des *buts* créés est consistant, il est nécessaire que l'agent choisisse un certain nombre de *buts* pour lesquels il s'engage. C'est ce processus qui est appelé la formation des *intentions*. Ainsi, les *intentions* courantes d'un agent sont décrites par un ensemble de *buts* sélectionnés avec leur état de traitement (Müller, 1996).
- Les ***plans*** jouent un rôle très important pour une implémentation pragmatique des *intentions*. Les *intentions* représentent des *plans* partiels d'actions que l'agent s'engage à exécuter pour atteindre ses buts. Par conséquent, il est possible de structurer les intentions en plans plus étendus, et de définir les intentions d'un agent comme les *plans* qui sont couramment adoptés (Müller, 1996).
  Plusieurs applications ont employé ce modèle d'agent, comme par exemple K.Boudaoud dans sa nouvelle approche pour détecter les intrusions dans les réseaux (Boudaoud, 2002)

5.3.3- Les agents hybrides

Les sections précédentes ont présenté deux types d'architectures : réactive et délibérative.

- Les architectures purement réactives ont un comportement assez simpliste,
- Alors que les architectures délibératives utilisent des mécanismes de raisonnement qui ne sont pas faciles à manipuler et qui ne sont pas suffisamment réactifs.

Chacune de ces architectures est appropriée pour un certain type de problème.

Pour la majorité des problèmes cependant, ni une architecture complètement réactive, ni une architecture complètement délibérative n'est appropriée. Comme pour les humains, les agents doivent pouvoir réagir très rapidement dans certaines situations (comportement réflexe), tandis que dans d'autres, ils doivent avoir un comportement plus réfléchi.
Afin d'apporter une réponse à ces imperfections, une architecture conciliant à la fois des aspects réactifs et délibératifs est requise. On parle alors d'architecture hybride, dans laquelle on retrouve généralement plusieurs couches logicielles (Müller, 1996). L'idée principale est de structurer les fonctionnalités d'un agent en deux ou plusieurs couches hiérarchiques qui interagissent entre elles afin d'atteindre un état cohérent de l'agent.

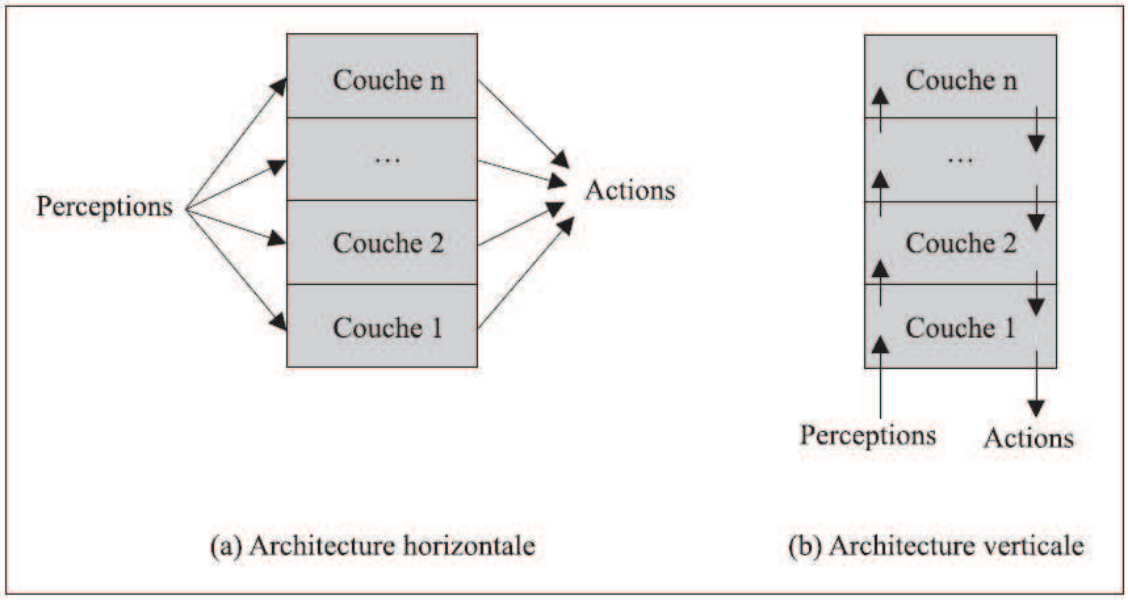


Figure 5.14 : Architectures d'agents en couches (Jennings & all, 1998)

Les couches peuvent être arrangées verticalement (seulement une couche a accès aux capteurs et aux effecteurs) ou horizontalement (toutes les couches ont accès aux entrées et aux sorties), voir figure ci-dessus.
Une approche hybride structurée en couche présente plusieurs avantages (Müller, 1996) :

- Elle permet de modulariser un agent ; ainsi les différentes fonctionnalités sont clairement séparées et reliées par des interfaces bien définies ;
- Elle permet une conception de l'agent plus compacte, ce qui augmente sa robustesse et facilite son "débogage".
- Elle accroît les capacités de l'agent car les différentes couches peuvent s'exécuter en parallèle.
- Elle augmente la réactivité de l'agent car il peut raisonner dans un monde symbolique tout en surveillant son environnement et en réagissant en conséquence.
- Elle réduit les connaissances nécessaires à une couche individuelle pour prendre ses décisions. Par exemple, une couche réactive n'aurait à utiliser que les informations concernant l'état courant de l'environnement alors qu'une couche délibérative aurait à utiliser des informations plus complexes relatives aux attitudes manipulées (croyances, buts, intentions, etc...)

Plusieurs utilisations à base de cette architecture, comme par exemple dans le domaine des transports et la gestion urbaine (Laichour, 2002), (Fayech, 2003) et (M.M.Ould Sidi & all, 2005).

5.4- Apprentissage des agents et des SMA

5.4.1- Apprentissage des Agents

5.4.1.1- Définitions et Différentes formes d'apprentissage

L'apprentissage est un processus d'acquisition de nouvelles compétences. Après apprentissage, un individu ou un groupe d'individus est capable d'un comportement efficace dans des situations qu'il ne savait pas gérer préalablement.
On peut considérer trois modes d'acquisition de compétences :

- Par construction ou transmission génétique : l'instinct est un savoir-faire non appris au cours de la vie. Il est inné ou héréditaire. De même un système artificiel peut être conçu doté de certaines comportementales.
- Par instruction : le sujet apprend à partir d'exemples ou de conseils, il cherche à reproduire le comportement d'un 'modèle' ou d'un 'professeur'.
- Par expérience : le sujet apprend par processus d'essais d'erreurs.

Philippe Beaune évoque trois aspects d'apprentissage :

- Apprentissage numérique ou symbolique
  - Analyse de données, réseau de neurones, algorithmes génétiques, renforcement...
- Apprentissage déductif ou inductif
  - Reformulation, compilation...
  - Généralisation, classification.…
- Classification selon la stratégie
  - Par cœur, par instruction, par analogie, à partir d'exemples, à partir d'observations… (Beaune, 1999)

L'apprentissage est un processus dynamique. D'après G.Clergue, l'apprentissage apparaît comme une trajectoire dans l'espace des phases des systèmes cognitifs du sujet, l'émergence d'un nouveau concept apparaît comme une transition de phase entre un état cognitif ancien et le nouvel état (Clergue, 1997)

L'élaboration d'un modèle d'apprentissage suppose de répondre à certaines questions :

- Qui est en train d'apprendre ?
- Qu'est ce qui est appris : un nouveau concept, un savoir, un savoir-faire, un savoir être?
- Quelles sont les sources de l'apprentissage ? si ce sont des exemples, ceux-ci sont-ils choisis au hasard, sont-ils positifs ou négatifs, simples ou difficiles, comment sont-ils décrits ?
- L'agent (l'apprenant) participe-t-il activement à son apprentissage : peut-il interagir avec son 'professeur' ou d'autres agents (apprenants) peut-il procéder à des expérimentations ?
- La source d'information est-elle bruitée ?
- Dans quel environnement se trouve l'apprenant ?
- De quelles connaissances ou compétences l'agent (l'apprenant) dispose-t-il au départ ?
- Quels sont les méthodes ou outils mis en œuvre pour l'apprentissage ?sont-ils génériques ou spécifiques au domaine, à l'agent (l'apprenant) ?
- Quelles sont les critères et méthodes d'évaluation des résultats de l'apprentissage ?

5.4.1.2- Apprentissage des agents

L'idée derrière l'apprentissage, c'est que les perceptions de l'agent ne devraient pas être utilisées seulement pour choisir des actions, elles devraient être aussi utilisées pour améliorer l'habilité de l'agent à agir dans le futur. L'apprentissage, pour un agent, est très important car c'est ce qui lui permet d'évoluer, de s'adapter et de s'améliorer.
Selon Russel et Norvig dans (Russell & Norvig, 1995), un agent apprenant peut être divisé en quatre composantes, comme le montre la figure ci-dessous :
– Le ***critique*** indique au module d'apprentissage à quel point l'agent agit bien.
Pour cela, il emploie un standard de performance fixe. Ceci est nécessaire, parce que les perceptions ne fournissent pas d'indications relatives au succès de l'agent. Par exemple, un programme qui joue aux échecs peut percevoir qu'il a mis l'autre joueur échec et mat, mais il a besoin d'un standard de performance pour savoir que c'est une bonne action. Il est important également que le standard de performance soit à l'extérieur de l'agent pour que

l'agent ne puisse pas le modifier dans le but de l'ajuster à son comportement. S'il pouvait le modifier, l'agent ajusterait son standard de performance pour obtenir plus de récompenses pour ses actions, au lieu de modifier ses actions dans le but de s'améliorer.
– Le ***module d'apprentissage*** utilise une certaine rétroaction sur les actions de l'agent pour déterminer comment le module de performance devrait être modifié pour, on l'espère, s'améliorer dans le futur.
– Le ***module de performance*** est vu comme étant l'agent au complet lorsqu'il n'y a pas d'apprentissage. C'est-à-dire que ce module peut prendre une des quatre formes que nous avons présentées à la section 5.3 : agent à réflexes simples, agent conservant une trace du monde, agent ayant des buts et agent basé sur l'utilité. Mais, peu importe la manière dont il est construit, son but demeure toujours de choisir des actions à effectuer en se basant sur les perceptions de l'agent.
– Le ***générateur de problèmes*** donne des suggestions d'actions amenant l'agent à faire de l'exploration. Si on laissait le module de performance choisir tout le temps les actions, il choisirait toujours les meilleures actions selon ce qu'il connaît. Par contre, si l'agent veut explorer un peu, il peut choisir des actions sous-optimales à court terme, mais qui pourraient l'amener à prendre de meilleures décisions à long terme. Par exemple, lors d'une simulation, un agent contrôlant une frégate pourrait suggérer une nouvelle stratégie de défense pour voir si elle est plus efficace que la stratégie actuelle.

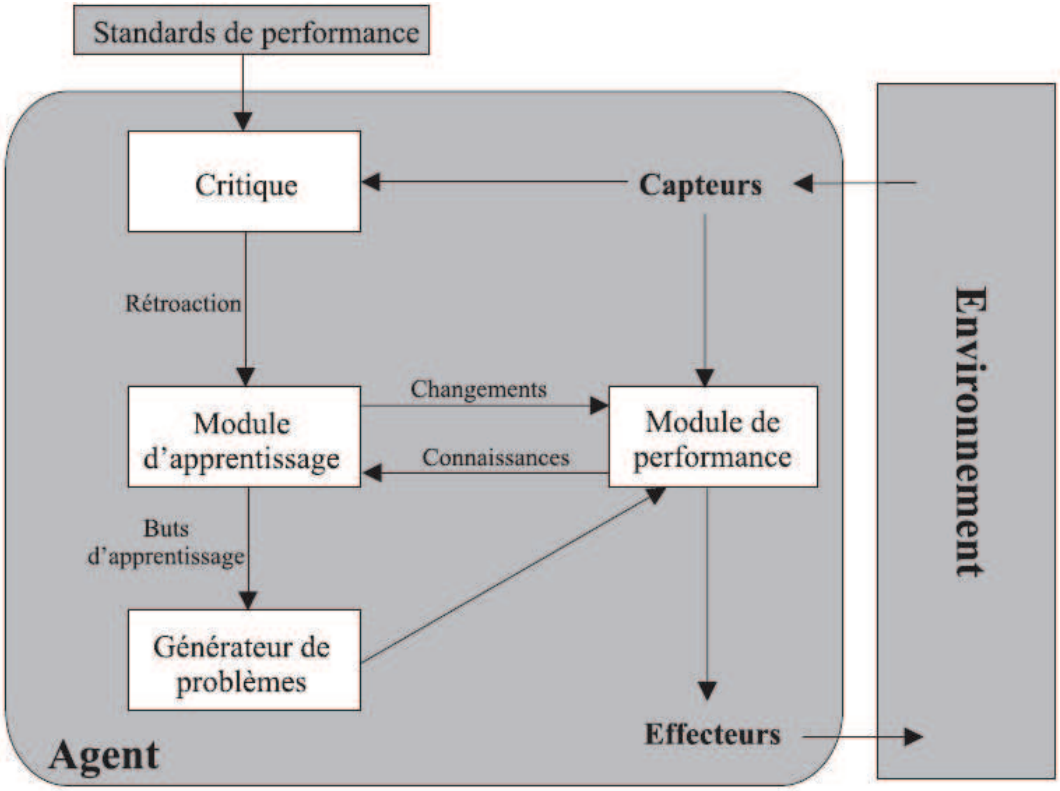


Figure 5.15 : Modèle général d'agent apprenant (Russell & Norvig, 1995)

Illustration pour l'agent taxi apprenant

- Module de performance
  - Connaissances et procédures pour choisir les actions.
- Critique
  - Observe l'agent et donne des informations au module d'apprentissage
- Module d'apprentissage
  - Modifie le module de performance.

- Générateur de problèmes
  - Identifie les possibilités d'amélioration et suggère des expérimentations.

Précédemment, nous avons vu un exemple de réflexe inné chez l'être humain, c'est-à-dire retirer notre main lorsque l'on se brûle, mais les êtres humains peuvent aussi apprendre de nouveaux réflexes. Par exemple, on peut penser à la conduite automobile.
Au début, la conduite est très difficile, car on doit penser à toutes les actions que l'on fait, mais plus on se pratique, moins on réfléchit et plus la conduite devient un réflexe.
Pour les agents, on peut penser à appliquer la même chose. C'est-à-dire, lorsqu'un agent fait face à une situation pour la première fois, il doit délibérer plus longtemps pour choisir ses actions. Mais, avec un module d'apprentissage, plus l'agent effectue des tâches similaires, plus il devient rapide. Son comportement passe graduellement d'un état délibératif, à un état réactif. L'agent a donc appris à exécuter une tâche. D'un point de vue plus technique, on peut dire que l'agent a, en quelque sorte, compilé son raisonnement dans une certaine forme d'ensemble de règles qui lui permettent de diminuer son temps de réflexion. Ce type d'apprentissage peut être très utile pour des agents hybrides.
Ce n'est qu'une façon dont les agents peuvent apprendre, il en existe plusieurs autres. En fait, on considère que toute technique qui permet à un agent d'améliorer son efficacité est une technique d'apprentissage.

5.4.1.2- L'apprentissage par renforcement

Il est tout de même bon de mentionner que la technique la plus utilisée avec les agents est l'apprentissage par renforcement (par essais erreurs) approprié au cas où l'environnement serait inconnu et changeant. Ce type d'apprentissage convient lorsqu'il n'existe pas de modèle ou de professeur capable d'indiquer à l'apprenant ce qu'il doit faire. Cependant l'agent peut apprendre en interagissant avec l'environnement et à partir de son expérience. Pour cela, on donne à l'agent un certain renforcement après chacune de ses actions. Si le renforcement est positif, cela signifie que l'agent a bien agi, celui-ci tentera donc de refaire les mêmes actions.
Si, par contre, le renforcement est négatif, alors l'agent saura qu'il a mal agi et il tentera d'éviter ces actions dans le futur. C'est un apprentissage, par essais et erreurs, qui permet à l'agent de s'améliorer avec le temps. (Dutech & all, 2003)
Bien entendu, pour utiliser ce type d'apprentissage, on doit être dans un environnement autorisant les erreurs. Par exemple, dans le domaine des frégates, on ne peut se permettre d'erreurs, car on ne voudrait pas qu'une frégate reçoive dix missiles avant d'apprendre comment les intercepter adéquatement. Dans ce type d'application, une des solutions est d'utiliser des simulations, procurant ainsi un environnement sécuritaire pour l'apprentissage des agents avant de les implémenter dans le véritable environnement.

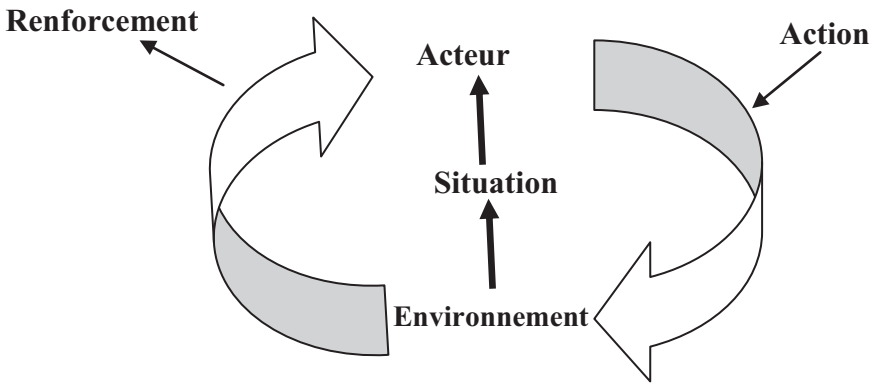


Figure 5.16 : Schéma de principe de l'apprentissage par renforcement

5.4.2- Apprentissage des SMA

L'apprentissage est une composante importante des systèmes multiagents. Ces systèmes évoluent généralement dans des environnements complexes (c'est-à-dire larges, ouverts, dynamiques et non prévisibles) (Sen & Weiss, 1999). Pour de tels environnements, c'est très difficile et même quelque fois impossible de définir correctement et complètement les systèmes à priori, c'est-à-dire lors de la phase de conception, bien avant leur utilisation.
Ceci exigerait de connaître à l'avance toutes les conditions environnementales qui vont survenir dans le futur, quels agents seront disponibles à ce moment et comment les agents disponibles devront réagir et interagir en réponse à ces conditions. Une manière de gérer ces difficultés est de donner à chaque agent l'habileté d'améliorer ses propres performances, ainsi que celles du groupe auquel il appartient.
Il est à noter que l'apprentissage dans un système multiagents comprend l'apprentissage dans un système mono-agent parce qu'un agent peut apprendre en solitaire et de façon complètement indépendante des autres agents. Mais aussi, il l'étend bien au delà dans la mesure où les activités d'apprentissage d'un agent peuvent être influencées considérablement par les autres agents et que plusieurs agents peuvent apprendre de manière distribuée et interactive comme une seule entité cohérente. Dans un environnement multiagents, les agents peuvent apprendre grâce aux autres. (Pesty & all, 2001)
Par exemple, un agent A, qui voudrait savoir comment se rendre à un certain endroit, pourrait demander à un autre agent B s'il connaît un bon chemin. Si B connaît un bon chemin, il peut le transmettre à A, permettant ainsi à l'agent A d'apprendre un bon chemin grâce à l'agent B.
D'un autre côté, les agents peuvent aussi apprendre à propos des autres. Par exemple, un agent peut regarder un autre agent agir dans certaines situations et, à l'aide de ces informations, il pourrait construire un modèle du comportement de l'autre agent. Ce modèle pourrait lui servir pour prédire les actions de l'autre agent dans le futur. Cette information pourrait l'aider à mieux se coordonner ou à mieux collaborer avec l'autre agent.
Une autre facette importante est l'apprentissage des interactions (coordination, collaboration, communication, etc.) entre les agents. Par exemple, si une frégate canadienne doit travailler avec un bateau de guerre anglais, elle doit pouvoir apprendre une manière efficace de se coordonner avec cet allié en tenant compte de ses capacités. En fait, elle doit pouvoir le faire pour toutes les possibilités, c'est-à-dire avec n'importe quel nombre de bateaux provenant de n'importe quel pays et dans n'importe quelle situation. Les agents doivent donc pouvoir adapter leurs mécanismes de coordination et de communication pour que l'ensemble des bateaux puissent agir de manière cohérente et ainsi se défendre le plus efficacement possible. Cette adaptabilité est une forme d'apprentissage de vie sociale.

5.5- Méthodologies de conception d'un SMA

5.5.1- Problématique

- ***Problèmes à résoudre :***
  – répartir les tâches et les connaissances
  – coordonner les agents
  – gérer les conflits et le maintien de la cohérence
  – permettre la communication
- ***Deux aspects à traiter :***
  – Aspects microscopiques (orientés agent)
  Comment construire un agent capable d'agir de manière autonome, quelles sont ses représentations et ses comportements ?

– Aspects macroscopiques (orientés système)
Comment construire une organisation capable d'agir de manière coopérative ?
Quels sont ses moyens de communication et de coordination ?

5.5.2- Méthodologie

Afin de guider le processus de conception des SMAs, plusieurs méthodologies de conception ont été proposées. Bien qu'en réalité, il existe très peu de travaux sur ce sujet, nous pouvons en retenir trois : (Gutknecht & Ferber 1999),(Wooldridge, 1999) et (Kinny & Georgeff, 1997).

O.Gutknecht et J.Ferber ont proposé dans (Gutknecht & Ferber 1999) puis évoqué dans (Ferber & all, 2009) et (Ferber & all, 2010), un cadre méthodologique centré sur trois concepts organisationnels :

1. l'***agent*** qui est définit comme une entité autonome communicante qui joue des *rôles* au sein de différents *groupes*. L'architecture interne de l'agent n'est pas définie afin de permettre au concepteur de choisir le modèle le plus adapté à son application ;

2. le ***groupe*** qui est défini comme un moyen d'identifier par regroupement un ensemble d'agents. Un groupe peut être fondé par n'importe quel agent et un agent peut être un membre d'un ou plusieurs groupes ;

3. le ***rôle*** qui est une représentation abstraite d'une fonction, d'un service ou d'une identification d'un agent au sein d'un groupe. Un rôle peut être attribué à plusieurs agents et un agent peut avoir plusieurs rôles.

Ce modèle est volontairement restreint afin de pouvoir s'intégrer à d'autres méthodologies. Au-dessus de ces trois concepts de base, ils spécifiene un modèle structurel, utilisé comme outil de conception, où ils définissent les notions de :

1. ***structure de groupe*** qui est une description abstraite d'un groupe qui définit les rôles et leurs interactions et qui est représentée par un tuple : $S = <R, G, L>$ où :

- R est l'ensemble des rôles identifiés dans le groupe,
- G est le graphe d'interactions entre les rôles,
- L est le langage d'interaction.

2. ***structure organisationnelle*** qui est l'ensemble des groupes qui définit un modèle d'organisation multi-agents. Elle est représentée par un couple $O = <S, Rep>$ où :

- S est un ensemble de structure de groupes,
- Rep est le graphe des représentants où chaque arc réunit deux rôles appartenant à deux structures de groupe différentes. Un représentant entre deux structures de groupes est un agent qui aurait simultanément un rôle dans un groupe et un second rôle dans un autre groupe (Gutknecht & Ferber 1999).

M. Wooldridge propose une méthodologie, appelée *Gaia*, qui traite les niveaux *macro* (social) et *micro* (agent) de conception. Tout comme l'approche précédente, la méthodologie est indépendante de l'architecture interne de l'agent. Le processus méthodologique de *Gaia* implique l'utilisation d'un ensemble de modèles définis dans deux phases : *analyse* et *conception* (Wooldridge, 1999).

5.5.2.1- Phase d'analyse

Durant la phase d'analyse, sont définies les entités abstraites qui définissent le modèle organisationnel du système. Ce dernier comprend deux modèles :

1. le ***modèle de rôles*** qui définit les rôles clés du système. Un rôle est caractérisé par deux types d'attributs :

- les *permissions* ou *droits* identifiant les ressources qui peuvent légitimement être utilisées pour remplir un rôle. Elles définissent également les limites d'utilisation de ces ressources ;
- les *responsabilités* qui définissent les fonctionnalités d'un rôle.

1. le ***modèle d'interactions*** qui définit les relations de dépendances entre les différents rôles. Dans ce modèle, pour chaque type d'interaction inter-rôle, est identifié un ensemble *de définitions de protocoles.*

Les définitions de protocoles sont composées des attributs suivants :

- le *but* : une brève description de la nature de l'interaction,
- l'*initiateur* : le ou les rôles responsables de l'initiation de l'interaction,
- le *correspondant :* le ou les rôles avec qui l'initiateur interagit,
- les *entrées* : les informations utilisées par l'initiateur,
- les *sorties* : les informations fournies par le correspondant durant l'interaction,
- le *traitement* : une brève description de tout traitement que l'initiateur du protocole exécute durant l'interaction.

5.5.2.2- Phase de conception

Durant la phase de conception, trois modèles, contenant les entités concrètes du système, sont générés :

1. le ***modèle d'agent*** qui identifie :

Les *types d'agents* qui seront utilisés pour l'implémentation du système,

Les *instances d'agents* qui traduiront ces types d'agents à l'exécution ;

2. le ***modèle de services*** qui définit les principaux services associés à chaque type d'agent. Un service est un bloque cohérent d'activités que l'agent s'engage à accomplir ;

3. le ***modèle de communication*** qui définit les liens de communication entre les types d'agent. Ce modèle est un simple graphe où les noeuds représentent les types d'agents et les arcs les chemins de communication.

Contrairement à la méthodologie de (Gutknecht & Ferber, 1999), la notion de *structure organisationnelle* n'est pas définie dans Gaia.

L'approche, de D.Kinny et M.Georgeff dans (Kinny & Georgeff, 1997), ne fait pas de distinction entre les phases d'analyse et de conception. Elle propose un ensemble de modèles pour définir d'un *point de vue externe* et d'un *point de vue interne* des systèmes multi-agents construits sur l'*architecture BDI*.

Point de vue externe :

Du point de vue externe, un SMA est défini par deux modèles qui sont indépendants de l'architecture BDI :

1. le ***modèle d'agent*** qui décrit les relations hiérarchiques entre les différentes classes d'agents abstraites et concrètes. Il identifie également les instances d'agents qui peuvent exister dans le système ainsi que leur multiplicité ;

2. le ***modèle d'interactions*** qui décrit les responsabilités d'une classe d'agent, les services fournis et les interactions entre les classes d'agents. Ce modèle définit la syntaxe et la sémantique des messages utilisés pour :

- les communications inter-agents,
- les communications entre les agents et les autres éléments du système tels que les interfaces utilisateurs.

Point de vue interne :

Du point de vue interne, sont décrites les attitudes mentales de chaque classe d'agent par l'intermédiaire de trois modèles :

1. le ***modèle de croyances*** qui définit les informations sur l'environnement et l'état interne d'un agent et les actions qu'il peut exécuter ;
2. le ***modèle de buts*** qui décrit les buts qu'un agent doit atteindre et les événements auxquels il peut répondre ;
3. le ***modèle de plans*** qui décrit les plans possibles qu'un agent peut utiliser pour atteindre ses buts.

5.5.2.3- Les étapes de réalisation d'un SMA

La démarche à suivre pour réaliser un SMA est résumée par (Ferber, 1999) :

1. Déterminer :
   - Les agents
   - L'environnement
2. Décrire les lois de l'environnement
3. Identifier les perceptions et les influences (actions) produites par les agents
4. Déterminer les variables internes et capacités des agents
5. Définir les comportements des agents:
   - Si les agents sont cognitifs: décrire la relation entre croyances, buts et actions
   - Si les agents sont réactifs: décrire les stimuli, les tropismes (attraction, répulsion, évitement) ainsi que les tâches (combinaisons d'actions élémentaires)

5.5.3- Plates-formes de développement

Les environnements de développements des SMA, les plus connus sont :

- JADE (Java Agent DEvelopement) est une plate-forme de développement de systèmes multi agents, crée par le laboratoire TILAB et répond aux spécifications FIPA (Foundation for Intelligent Physical Agents).
- ZEUS est un environnement intégré qui utilise la méthodologie Rôle Modeling pour la construction rapide d'applications à base d'agents collaboratifs
- MadKit est une plate-forme multi agents modulaire écrite en Java et construite autour du modèle organisationnel Agent/Groupe/Rôle.
- KQML (KnowledgeQueryandManipulation Language) : Langage de communication entre agents ACL (Agent communication Language).

Et d'autres moins connus :

- ADELPHE, AgentBuilder, Madkit, SamllTalk, LISP, LISP, C++, Prolog, SMALTALK, AgentBuilder, Mask, Mercure, MACE, DIMA, SWORM, Cormas, geamas, Mice, FIPA-OS, LEAP, Actalk, DECAF, JACK, MAGES IV, Magique, Mocah, OSAKA, Pandora II, Alaadin, JAF, Maleva Agent Oriented Programming in Java ,Java Agent Services.

  (Boissier & all, 1999)

Les plates-formes fournies comme logiciels libres:

- JADE, MACE, ZEUS, et MADKIT pour les agents cognitifs,
- Et SWORM pour les agents réactifs.

(Guessoum & Occello, 2001) récapitulent les différents environnements de programmation dans le tableau suivant :

Catégorie	Environnements
Outils pour la simulation	Sworm, Cormas, geamas, Mice
Outils pour l'implémentation	Actalk, DECAF, JACK, MAGES IV, Magique, Mocah, OSAKA, Pandora II
Outils pour la conception	Alaadin, JAF, Maleva
Outils pour la conception et l'implémentation	AgentBuilder, DIMA, Mask, Mercure, MACE
Outils pour la conception, l'implémentation et la validation	Jade, Zeus, Madkit, MAST

Tableau 5.2 : Environnements de développement

5.6- Conclusion

L'intelligence artificielle distribuée est née pour résoudre les problèmes de complexité des gros programmes de l'intelligence artificielle, comme par exemple les problèmes de classification : l'exécution est alors distribuée mais le contrôle reste centralisé. Contrairement aux SMA, où chaque agent possède un contrôle total sur son comportement. Pour résoudre un problème complexe, il est plus simple de concevoir des programmes relativement petits (les agents) en interaction, qu'un seul gros programme de rôle similaire. L'autonomie permet au système de s'adapter dynamiquement aux changements imprévus qui interviennent dans l'environnement.

Le fait le plus remarquable dans la montée en puissance de ces nouveaux outils tient sans doute à la transversalité de leur diffusion dans le champ scientifique. En effet, les SMA ont trouvé des terrains d'expérimentation dans de multiples sciences techniques mais aussi dans le champ des sciences de l'Homme et de la Société.

Ainsi, la technologie multi-agent semble être la solution pour le développement des logiciels de demain (Gates, 1999).

Pour nous avons fait appel à cette technologie dans un contexte très pointue de classification automatique de textes et les résultats qui vont être confirmés ultérieurement son bénéfiques.

Chapitre 6

Classification Automatique des textes : Approche Orientée Agent

Table des matières

6.1- Introduction

Après l'étude faite sur l'état de l'art des différentes approches pour la construction de modèles de catégorisation de textes ainsi que les divers codages et techniques pour la représentation des textes, développés dans la littérature, nous avons adoptés pour nos travaux :

La méthode des N-grams pour représenter notre corpus pour son indépendance des différentes langues et son non exigence des traitements linguistiques préalables et d'autres avantages qui vont être décrits dans la section suivante.

Pour l'algorithme d'apprentissage et classification, le modèle d'indépendance conditionnelle (Naïve Bayes classifier) a été utilisé pour sa simplicité d'une part, et d'autre part, comme tous les modèles probabilistes, il s'appuie sur une base théorique précise.

Pour pouvoir confronter les différents résultats fournis par les classifieurs construits classiques ou Muti-Agents, nous allons utiliser les mesures de rappel et précision ainsi que la F-measure (F_1) pour évaluer les performances de ces modèles.

Pour la partie applicative, au lieu de procéder par une démarche classique, nous proposons de distribuer l'intelligence dans une approche qui consiste à déléguer la tâche de classification de textes à un Système Multi-Agents.

Ce chapitre va contenir le produit fini de nos travaux de recherche. Nous commençons par décrire notre approche d'une façon générale, ensuite nous allons motiver et justifier tous les choix des solutions adoptées durant toutes les phases du processus, en commençant par la méthode de représentation de textes choisie, suivie par l'algorithme d'apprentissage et classification utilisée qui va être détaillé, ainsi que l'intérêt de solliciter un SMA pour un tel traitement. La base de texte utilisé pour l'apprentissage et la classification sera décrite par la suite avant d'enchainer par une présentation des résultats expérimentaux et l'influence de quelques facteurs dans ces résultats et nous terminons par discuter les résultats obtenus et une conclusion.

6.2- Description générale de l'approche

Dans le but de présenter à l'utilisateur une application performante, en matière de qualité de résultats et rapidité d'exécution, il est intéressant de distribuer le processus de classification en définissant une architecture composée de parties distinctes, chaque partie va traiter le problème de catégorisation de textes d'une façon spécifique, mais pouvant communiquer pour partager leurs connaissances.

L'idée générale de la présente architecture se présente de la manière suivante :

Notre système de catégorisation de textes va être composé de plusieurs modules logiciels (Agent), le nombre d'agents va être fixé après l'expérimentation et évaluation qui va solliciter 3, 9, 21, 33, 61, 99 et 181 agents, après une étude qui va porter sur le compromis qualité-des résultats, efficacité en termes de temps d'exécution. Deux types d'agents délibératifs (cognitifs), vont être utilisés, des agents classifieurs pour catégoriser les documents en première manche et un agent administrateur central qui va recueillir tous les suffrages pour consolider les votes et éventuellement trancher pour la catégorie qui va être élue.
Dans nos travaux, nous allons varier le "N" des N-grammes, de même pour le nombre d'agents qui va être renforcé au fur et à mesure, jusqu'à stabilisation des résultats en matière de qualité et de rapidité, à chaque variation de paramètres les mesures d'évaluation de performances rappel, précision et F-measure (F_1) seront nos marques pour jauger nos classifieurs.

- Dans la phase d'apprentissage chaque classe va être répartie sur le nombre d'agents choisis. Par conséquent notre base de d'apprentissage est scindée en plusieurs sous bases secondaires réduites en nombre de textes par classe. Chaque base secondaire contiendra un échantillon de toutes les classes de la base principale Reuters. Chaque agent possèdera sa propre base sur laquelle il exercera son apprentissage.
  Comme on a vu précédemment dans la section 3.2.8, les différents classifieurs peuvent correspondre à différents algorithmes ou au même algorithme utilisé avec différents sous-échantillons du corpus d'apprentissage, notre approche se situe dans le deuxième cas qui va procéder avec le même algorithme de classification « Naïve Bayes », avec différents mini-corpus d'apprentissage.
- Dés que l'apprentissage soit terminé, la phase de classification des documents test est lancée : Chaque texte va être traité par tous les agents qui utilisent le modèle Naïve Bayes, une probabilité va être accordée pour l'appartenance du texte à chaque classe, le texte sera catégorisé dans la classe qui possède la plus grande probabilité.
  Chaque agent dans le système fera sa propre classification pour le même texte (Autonomie).
  La décision finale sera prise après un vote majoritaire (Collaboration), le texte sera catégorisé dans la classe qui a été nommé par le plus grand nombre d'agents, et puisque le nombre d'agents a été choisi impair volontairement, l'égalité parfaite dans le vote est évitée, et une catégorie prendra toujours l'ascendant sur les autres.
- Après une étude portée sur les résultats fournis par les différents modèles de classification construits, l'ensemble des paramètres (n-grammes et nombre d'agents) sera fixé pour le modèle adopté.
- Le modèle final accepté va servir à classer de nouveaux textes de catégorie inconnue de la même manière de classification des documents test.

6.3- Motivations

6.3.1- Codage en n-grammes

Comme on a vu dans les chapitres précédents, la première phase dans le processus de classification de textes, est de subdiviser le texte à traiter en plusieurs unités d'information qu'on peut appeler termes ou descripteurs qui sont, habituellement, des mots simples. La question principale qui se pose dans cette première phase de préparation des documents du corpus est : Sur le plan informatique, comment repérer un mot ? D'une autre manière, quels sont les bornes formelles pour délimiter un mot ? Si pour les langues comme le français et l'anglais la réponse est presque évidente – à savoir que toute chaine de caractères précédée et suivie d'un espace ou un signe de ponctuation est considérée comme un mot simple- Cette règle n'est pas valable pour les autres langues.

Si le mot simple ne convient pas à toutes les langues, quelle est donc l'unité d'information élémentaire la plus appropriée pour fractionner un document ?

Contrairement à la recherche d'information, dans un contexte de classification de textes, le fait que les unités d'information élémentaires n'ont pas vraiment un sens, n'est pas une contrainte lors de la classification, néanmoins ces unités atomiques doivent être facilement identifiées sur le plan informatique, et peuvent être comparées statistiquement.

Si on considère le mot comme unité d'information de base, une classification mutilingue est impossible. En traitant les mots comme des termes, la représentation des textes s'avère relativement simple pour le français ou l'anglais, mais très difficile pour des langues comme

l'allemand ou l'arabe. D'autre part, le stemming et la lemmatisation utilisés comme moyen de normalisation et de réduction du lexique constitue une contrainte non moins négligeable. La notion de n-grammes, qui depuis une quinzaine d'années donne de bons résultats, est devenue, par des récentes recherches, un axe privilégié, dans le domaine de classification de textes.

Toutes ces raisons ont appuyé notre choix, pour la suite du travail qu'on va accomplir pour le codage et la représentation des documents, sur les techniques basées sur les n-grammes qui garantissent plusieurs avantages, confirmés par plusieurs auteurs, dont le principaux sont les suivants :

- La représentation en n-grammes s'attaque aux documents presque dans leur état brut, contrairement aux autres représentations qui nécessitent des traitements purement linguistiques, comme les traitements d'élimination des mots vides, de stemming et de lemmatisation (Jalam, 2003). Ces traitements améliorent la performance des systèmes à base sur les mots mais en contrepartie la mise en ouvre informatique de ces procédures est relativement lourde. D'autre part, l'étude de (Sahami, 1999) a montré que la performance des systèmes à base des n-grammes ne progresse pas même après ces traitements linguistiques, Et ce ci peut être confirmé par le fait que si un texte contient plusieurs mots de même racine, le nombre des n-grammes correspondants augmentera sans le moindre traitement linguistique préalable (Jalam, 2003).
- Un autre point à souligner en faveur des n-grammes : quelques algorithmes de lemmatisation ne semble pas être en mesure de regrouper des termes comme *automatisation***,** *automatiser* et *automatique* dans la même classe, par contre, le découpage en n-grammes est suffisant pour classer les trois termes dans la même classe, les tri-grams : *aut, uto, tom, oma, mat, ati,* permettent par une mesure de similarité d'affirmer que c'est l'*automatique* dont il est question.
- Le codage en n-grammes n'a pas besoin de segmenter le texte avant d'extraire les termes, ce qui offre à cette technique une caractéristique très intéressante, capable de représenter et traiter les langues pour lesquelles les frontières entre termes ne sont pas bien marquées comme l'allemand, le chinois, ou la langue arabe, dans laquelle les pronoms, sujets et compléments sont liés dé fois aux verbes, une seule chaîne de caractères unie représentant une phrase comme, par exemple, « kalamtouhou » ("je lui ai parlé") (Jalam, 2003). Notons bien, qu'une segmentation en n-grammes de caractères satisfait toutes les langues qui utilisent un alphabet, pour reconstruire le texte on procède à la concaténation, (Biskri & Delisle, 2001).
- Le choix des n-grams permet aussi de contrôler la taille du lexique et de le conserver à un seuil raisonnable. Parmi les grandes difficultés, qui s'opposent aux algorithmes d'analyse des grands corpus, c'est la taille du lexique. En effet, un découpage en mots fait que la taille du lexique, est d'autant plus importante, que le corpus est important. Cette contrainte persiste, malgré les aménagements appliqués sur les textes durant la phase de prétraitement (Lemmatisation, suppression des mots-vides, etc..). Le nombre de n-grammes d'un corpus, ne peut dépasser la taille de l'alphabet à la puissance n. Un découpage en quadri-grams pour la langue anglaise (Alphabet de 26 caractères) nous donne une taille maximale de 26^4 entrées, soit un vocabulaire d e 456 976 qaudri-grammes possibles. Si on élimine les combinaisons comme FFFF, RRRM, KPPP, etc.., qu'on ne trouvera jamais, ce nombre diminue d'une façon considérable. (Lelu & Hallab ,2000) estime ce nombre à quelques 13 087 quadri-grams pour un texte de 173 000 caractères.

- Ce type de codage est multilingues : La langue du document ne pose aucune contrainte particulière à sa représentation en n-grammes. En conséquence, aucune connaissance linguistique préalable n'est requise, contrairement aux systèmes basés sur les mots qui sont dépendants des langues dans lesquels il faut utiliser des dictionnaires spécifiques à chaque langue (féminin-masculin ; singulier-pluriel ; conjugaisons ; etc.) (Jalam, 2003),(Clech, 2004).
- Comparativement à d'autres techniques, les n-grammes extraient automatiquement les racines des mots les plus fréquents (Jalam, 2003),(Clech, 2004)
- Enfin, les erreurs d'orthographe et les éventuelles déformations d'un texte relatives à l'utilisation des systèmes de reconnaissance optique de caractères (OCR) n'ont pas d'incidence grave sur le profil d'un document. La reconnaissance optique d'un texte scanné est en général approximative. Par exemple, le mot "chapitre" peut être lu comme "clapitre". Une représentation à base de mots aura du mal à reconnaître qu'il s'agit du mot "chapitre" puisque le mot est mal orthographié, tandis qu'une représentation à base des n-grammes capture normalement les autres n-grammes significatives comme "apit", "pitr" "itre", etc... (Jalam & Teytaud, 2001). Des études ont montré que des systèmes de recherches d'information à base des n-grammes ont préservé leurs performances malgré une déformation de 30%, contrairement à un système à base de mots qui commence à se dégrader à partir d'un taux de 10% de déformations (Miller & all, 1999).

6.3.2- Pondération des termes

Plusieurs options s'offrent à ce stade du processus, certainement la plus utilisée c'est TF-IDF, mais puisque notre objectif c'est chercher plus d'efficacité avec des résultats acceptables, et ne pas alourdir l'algorithme avec des calculs supplémentaires, nous avons choisi d'utiliser, dans l'ensemble de nos expérimentations, la façon la plus simple pour calculer cette pondération à savoir la fréquence du terme dans le document ou dans la catégorie.

6.3.3- Naïve Bayes

Naïve Bayes classifier est le représentant le plus populaire des classifieurs probabilistes et son théorème est au cœur de la problématique de la classification. C'est l'une des méthodes les plus pratiques d'apprentissage, avec les kPPV, Rocchio, les SVM, les arbres de décision, les réseaux de neurones. En revanche si les modèles vectoriels fonctionnent bien, leur fondement est entièrement empirique. Ils sont le résultat de nombreuses années de test. Le modèle probabiliste, au contraire, s'appuie sur une base théorique précise.

Le classifieur bayésien naïf reste un des outils de catégorisation de documents les plus pratiques en raison de ses performances reconnues dans ce domaine, et est aujourd'hui intégré à de nombreux produits commerciaux. Il s'appuie sur un modèle de génération d'un document à partir duquel on peut déduire la ou les classes les plus probables d'appartenance du document. Les paramètres du modèle sont estimés à partir d'un corpus d'apprentissage. Plusieurs expériences ont démontré les bonnes performances de l'algorithme. L'équipe Microsoft a développé son propre algorithme NB connu par MNB (**M**icrosoft **N**aïve **B**ayes) qui est un algorithme de classification fourni par Microsoft **S**QL **S**erver 2005 **A**nalysis **S**ervices (SSAS), conçu pour la modélisation prédictive. (Présenté dans l'annexe MNB).
Une autre application réussie à base du classifieur NB, celle utilisée pour enseigner l'âne Ditto les bases de la langue anglaise. (Présentée dans l'annexe Ditto-The donkey)

Historiquement plusieurs travaux de classification à base de l'approche Naive Bayes ont été développés, citons :

Maron (1961) – Indexation automatique
Mosteller and Wallace (1964) – Identification des auteurs
Van Rijsbergen, Robertson, Sparck Jones, Croft, Harper (1970) – moteurs de recherche
Sahami, Dumais, Heckerman, Horvitz (1998) – Filtres anti-spams.
Adventure Works Cycle (2009) – Marketing (Ciblage de clientèle)

Le principe qui régit et donne son nom à l'algorithme est très simple. Il indique simplement que les différents attributs (dans le cas du texte, les différents termes présents dans le document) sont considérés comme indépendants. C'est la même hypothèse que pour le modèle vectoriel mais cette fois exprimée explicitement dans le cadre de la théorie probabiliste.

Avant de justifier les raisons qui ont motivé l'adoption de cette méthode dans notre approche proposée, nous avons préférer rappeler quelques définitions nécessaires pour une bonne maîtrise du classifieur Naïve Bayes.

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

6.3.3.1- Probabilité conditionnelle

La notion de probabilité conditionnelle permet de tenir compte dans une estimation d'une information complémentaire. Par exemple, si je tire au hasard une carte d'un jeu, j'estime naturellement à une chance sur quatre la probabilité d'obtenir un cœur ; mais si j'aperçois un reflet rouge sur la table, je corrige mon estimation à une chance sur deux. Cette seconde estimation correspond à la probabilité d'obtenir un cœur sachant que la carte est rouge. Elle est conditionnée par la couleur de la carte ; donc, conditionnelle.
En théorie des probabilités, la probabilité conditionnelle d'un événement ***A***, sachant qu'un autre événement **B** de probabilité non nulle s'est réalisé est noté $P(A/B)$ défini par :
Le réel $\mathbb{P}(A/B)$ se lit « probabilité de ***A***, sachant ***B*** ». $\mathbb{P}(A/B)$ se note aussi parfois $\mathbb{P}_B(A)$
Mathématiquement, soient (Ω , $\mathcal{B}$, $\mathbb{P}$), un espace probabilisé et ***B*** un événement de probabilité non nulle. À tout événement ***A*** de $\mathcal{B}$, nous associons le nombre noté $\mathbb{P}(A/B)$ ou $\mathbb{P}_B(A)$ défini par:

$$\mathbb{P}_B(A) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)}$$

Nous pourrions vérifier que l'application $\mathbb{P}_B$ définie par $A \rightarrow \mathbb{P}_B(A)$ est une probabilité.

Et $P(A/B)+P(\neg A/B)=1$

6.3.3.2- Théorème de Bayes

Le théorème de Bayes ou le modèle d'indépendance conditionnelle (Naïve Bayes classifier), qui a été proposé par le mathématicien anglais Thomas Bayes (1702-1761), fournit un cadre théorique pour la problématique de la classification. Dans cette approche, tous les paramètres, sont considérés comme des variables aléatoires issues d'une distribution de probabilité.

Le théorème de Bayes est utilisé dans l'inférence statistique pour mettre à jour ou réviser les estimations d'une probabilité ou d'un paramètre quelconque, à partir des observations et des

lois de probabilité de ces observations. Il y a une version discrète et une version continue du théorème.
En théorie des probabilités, le théorème de Bayes énonce des probabilités conditionnelles : étant donné deux évènements ***A*** et ***B***, le théorème de Bayes permet de déterminer la probabilité de ***A*** *sachant* ***B***, si l'on connaît les probabilités :

- de ***A*** ;
- de ***B*** ;
- de ***B*** *sachant* ***A***.

Ce théorème élémentaire (originellement nommé « de probabilité des causes ») a des applications considérables.
Pour aboutir au théorème de Bayes, on part d'une des définitions de la probabilité conditionnelle :

$$P(A|B)P(B) = P(A \cap B) = P(B|A)P(A)$$

en notant $P(A \cap B)$ la probabilité que ***A*** et ***B*** aient tous les deux lieu. En divisant de part et d'autre par *P*(***B***), on obtient :

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Qui donne bien le théorème de Bayes.

Chaque probabilité du théorème de Bayes a une dénomination usuelle.

- $\mathbb{P}$ (***A***) est la *probabilité a priori* de ***A,*** elle est « antérieure » au sens qu'elle précède toute information sur ***B***, *P*(***A***) est aussi appelée la *probabilité marginale* de ***A***.
- De même, $\mathbb{P}$ (***B***) est appelé la *probabilité marginale* ou *a priori* des donnés d'apprentissage ***B***.
- $\mathbb{P}$ (***A/B***) est appelée la *probabilité a posteriori* de ***A*** sachant ***B*** (ou encore de ***A*** sous condition ***B***), elle est « postérieure », au sens qu'elle dépend directement de ***B***.
- $\mathbb{P}$ (***B/A***), pour un ***B*** connu, est appelé la *fonction de vraisemblance* de ***A*** (ou encore de ***B*** sous condition ***A***)

6.3.3.3- Inférence bayésienne

On nomme inférence bayésienne la démarche logique permettant de calculer ou actualiser la probabilité d'une hypothèse. Le raisonnement bayésien est appliqué à la prise de décision, on utilise la connaissance des événements pour prédire des événements futurs. Cette démarche est régie par l'utilisation de règles strictes de combinaison des probabilités, desquelles dérive le théorème de Bayes.

Exemple d'inférence bayésienne :

D'où vient ce biscuit ?

(Cet exemple est extrait de l'article anglophone www.wikipedia.en)

Imaginons deux boîtes de biscuits.

L'une, ***A***, comporte 30 biscuits au chocolat et 10 ordinaires.
L'autre, ***B***, en comporte 20 de chaque sorte.
On choisit les yeux fermés une boîte au hasard, puis dans cette boîte un biscuit au hasard. Il se trouve être au chocolat. De quelle boîte a-t-il le plus de chances d'être issu, et avec quelle

probabilité ? Intuitivement, on se doute que la boîte ***A*** a plus de chances d'être la bonne, mais de combien ?

Notons H_A la proposition « le gâteau vient de la boîte ***A*** » et H_B la proposition « le gâteau vient de la boîte ***B*** ». Dans un contexte de classification ***A*** et ***B*** c'est des classes et D est un document.

Si lorsqu'on a les yeux bandés les boîtes ne se distinguent que par leur nom, nous avons $\mathbb{P}(H_A) = \mathbb{P}(H_B)$, et la somme fait 1, puisque nous avons bien choisi une boîte, soit une probabilité de 0,5 pour chaque proposition.

Notons ***D*** l'événement désigné par la phrase « le gâteau est au chocolat ». Connaissant le contenu des boîtes, nous savons que :

- $\mathbb{P}(D / H_A) = 30/40 = 0{,}75$
- $\mathbb{P}(D / H_B) = 20/40 = 0{,}5$

La formule de Bayes nous donne donc :

$$P(H_A|D) = \frac{P(H_A)\cdot P(D|H_A)}{P(H_A)\cdot P(D|H_A)+P(H_B)\cdot P(D|H_B)}$$

$$= \frac{0{,}5\times 0{,}75}{0{,}5\times 0{,}75+0{,}5\times 0{,}5}$$

$$= 0{,}6$$

$\mathbb{P}(H_A / D)$ représente la probabilité d'avoir choisi la boîte ***A*** sachant que le gâteau est au chocolat.

Avant de regarder le gâteau, notre probabilité d'avoir choisi la boîte ***A*** était $\mathbb{P}(H_A)$, soit 0,5.

Après l'avoir regardé, nous révisons cette probabilité à $\mathbb{P}(H_A /D)$, qui sera 0,6. L'observation ***D*** « le gâteau est au chocolat » nous a donc apporté une augmentation de 10% sur la probabilité initiale de H_A « le gâteau vient de la boîte ***A*** ».

Et puisque $\mathbb{P}(H_A /D) + \mathbb{P}(H_B /D) = 1$ (pas d'autre possibilité que d'avoir choisi la boîte ***A*** ou la boîte ***B*** sachant que le gâteau est au chocolat), la probabilité d'avoir choisi la boîte ***B*** sachant que le gâteau est au chocolat est donc de 1 - 0,6 = 0,4.

6.3.3.4- La classification naïve bayesienne

La classification naïve bayesienne est un type de classification bayesienne probabiliste simple basée sur le théorème de Bayes avec une forte indépendance (dite naïve) des hypothèses.

Définition de l'indépendance :

« Two events ***A*** and ***B*** are statistically independent if the probability of ***A*** is the same value when ***B*** occurs, when ***B*** does not occur or when nothing is known about the occurrence of ***B*** »

«Deux événements ***A*** et ***B*** sont statistiquement indépendants si la probabilité de ***A*** est la même lorsque ***B*** se réalise, ou lorsque ***B*** ne se réalise pas, ou encore quand on ne sait rien sur ***B*** »

A et ***B*** sont indépendants alors : $\mathbb{P}(A|B) = \mathbb{P}(A)$

Exemple :

Supposons qu'il ya deux événements:

A : Amine enseigne la classe sinon c'est Abderahim
B : Il pleut

Nous remarquons que la météo ne dépend pas et n'as pas d'influence sur lequel des professeurs qui va enseigner la classe.

Cela peut être spécifié très simplement par :

$$\mathbb{P}(A|B) = \mathbb{P}(A)$$

Cette propriété implique les règles suivantes :

- $\mathbb{P}(\neg A / B) = \mathbb{P}(\neg A)$
- $\mathbb{P}(B / A) = \mathbb{P}(B)$
- $\mathbb{P}(B \cap A) = \mathbb{P}(B)\ \mathbb{P}(A)$
- $\mathbb{P}(\neg B \cap A) = \mathbb{P}(\neg B)\ \mathbb{P}(A)$
- $\mathbb{P}(B \cap \neg A) = \mathbb{P}(B)\ \mathbb{P}(\neg A)$
- $\mathbb{P}(\neg B \cap \neg A) = \mathbb{P}(\neg B)\ \mathbb{P}(\neg A)$

6.3.3.5- Maximum A Posteriori (MAP) et Maximum de vraisemblance (ML)

En général, nous cherchons l'hypothèse la plus probable compte tenu des données d'apprentissage :

Maximum A Posteriori (MAP)

- $\mathbf{h}_{MAP} = \arg\max \mathbb{P}(h/d)$ où h appartient à H l'espace des hypothèses et d à l'espace des événements
- $\mathbf{h}_{MAP} = \arg\max \dfrac{P(d/h)\ P(h)}{P(d)}$

P (d) peut être ignoré car il est le même pour toutes les probabilités

- $\mathbf{h}_{MAP} = \arg\max \mathbb{P}(d/h)\ \mathbb{P}(h)$

Exemple :

Pour une classification bi-classe comme par exemple les filtres anti-spam :
L'espace des hypothèses correspond à l'appartenance aux classes, on notera :

$P(c_i/d_i) = \arg\max \mathbb{P}(d_i/c_i)\ \mathbb{P}(c_i)$
$\mathbf{d}_i$ est un document
$\mathbf{c}_1$ correspond à la classe spam
$\mathbf{c}_2$ correspond à la classe non spam
$P(c_i/d_i) = \arg\max\{ P(d_i/c_1) P(c_1) , P(d_i/c_2) P(c_2)\}$

Maximum de vraisemblance (ML)

Si on suppose que les probabilités à priori sont les mêmes pour toutes les classes (exemple des deux boîtes de biscuits) alors on aura :

$P(c_i) = P(c_j)$

Une simplification plus poussée nous conduit à :

- $\mathbf{h}_{ML} = \arg\max P(d_i/c_i)$

L'estimation des paramètres pour les modèles de Bayes naïf utilise la méthode du maximum de vraisemblance.

Le calcul de $P(d_i/c_j)$ dépend du modèle de génération des exemples. Dans le domaine de classification de textes, les plus populaires sont le modèle multivarié de Bernoulli et le modèle multinomial.

6.3.3.6- Le modèle multivarié de Bernoulli

Dans le modèle le plus simple, un document est un vecteur binaire de la taille du vocabulaire. Il est généré par tirage aléatoire : chaque terme du vocabulaire peut être présent ou absent avec une certaine probabilité. Seule la présence/absence des termes est utilisée. Leur nombre d'occurrences dans le document n'a pas d'incidence. Le document se représente comme le résultat du tirage de T variables aléatoires indépendantes : $t_1,\ldots,t_v$. Pour rendre les formules exploitables, on fait de plus l'hypothèse d'indépendance des termes (hypothèse naïve Bayes). $P(d_i/c_j)$ se simplifie alors en un produit de probabilités d'occurrences de chaque terme :

$$P(d_i|c_j) = \prod_{t_k \in d_i} P(t_k|c_j)$$

$$= P(t_1/c_j) * P(t_2/c_j) * \ldots\ldots * P(t_n/c_j)$$

Il est possible d'estimer $P(t_k/c_j)$ à partir des exemples du corpus d'apprentissage. On utilise généralement l'estimateur du maximum de vraisemblance.

Que faire si on retrouve une probabilité nulle $P(t_j / c_i) = 0$? (Un terme absent de tous les documents d'une classe)

Comme les probabilités de chaque terme sont multipliées, il suffit en effet qu'un seul terme dans un document à classer ne soit présent dans aucun document d'une classe (ce qui arrive souvent dans le domaine du texte où beaucoup de termes apparaissent très rarement) pour que la probabilité que le document appartienne à cette classe soit nulle.
La correction Laplacienne ou le lissage de laplace (laplace smoothing) intervient pour éviter ces probabilités nulles. Le lissage effectué pendant l'estimation, qui consiste à ajouter 1 à chaque terme, est indispensable.
Puisque le nombre de termes de la base d'apprentissage est important, l'ajout de 1 sera négligeable. On parvient alors, dans le cas du lissage de laplace, à :

$$P(t_k|c_j) = \frac{1 + \sum_{d_i \in c_j} 1_{ik}}{2 + |c_j|}$$

Avec 1_{ik} =1 si t_k appartient à d_i ,
= 0 sinon.
Pour illustrer le lissage de Laplace : dans une base composée de 750 termes, la probabilité d'un terme absent sera (1+0) / (2+1350) = 0,0007396 au lieu de 0 / 1352 = 0

6.3.3.7- Le modèle multinomial

Dans le modèle précédent, il n'est pas possible d'utiliser les fréquences des termes dans les documents. Pour prendre en compte cette information supplémentaire, un autre modèle plus

complexe a été proposé, et qui a été adopté dans notre approche proposée. Un document est une séquence de mots, chacun étant tiré aléatoirement parmi l'ensemble des mots du vocabulaire. Un document est donc généré par une distribution multinomiale des mots avec autant de tirages que de mots dans le document. Il est ainsi possible de prendre en compte la longueur des documents bien que cela soit rarement fait en pratique. L'hypothèse d'indépendance des termes reste nécessaire.

$$P(d_i|c_j) = \prod_{t_k \in d_i} P(t_k|c_j)$$
$$= P(t_1/c_j) * P(t_2/c_j) *\ldots\ldots* P(t_n/ c_j)$$

Les probabilités sont une nouvelle fois estimées à partir des occurrences des exemples du corpus avec l'estimateur du maximum de vraisemblance avec un lissage de laplace :

$$P(t|c) = \frac{1 + T_{ct}}{\sum_{t' \in V}(1 + T_{ct'})}$$

Ou V c'est le vocabulaire du corpus,
T_{ct} le nombre d'occurrences du terme t dans tous les documents de la classe c,
$T_{ct'}$ le nombre d'occurrences de tous les termes y compris t des documents de la classe c.
Quant à la classification, l'estimation $P(d_i/c_j)$ est calculée suivant le modèle utilisé en remplaçant les probabilités par leur estimateur et ***la classe c_j ayant la probabilité la plus élevée sera attribuée au document d_i.***

6.3.3.8- Description de l'algorithme

Il y a plusieurs variantes de l'algorithme NB, voici une description d'une d'entre elles :

1- Préparer les associations (texte, classe d'appartenance) pour tous les documents d'apprentissage.
2- Préparer le document à classifier sous la forme (termes, nombre d'occurrences).
3- Compter pour chaque classe le nombre de documents appartenant a cette classe.
4- Calculer pour chaque classe la probabilité à priori.
5- Calculer de nombre de termes contenus dans une classe et le nombre d'occurrences de chaque terme dans toutes les classes, ce traitement à effectuer sur tous les termes des documents de training.
6- Calculer pour chaque classe la probabilité à posteriori.
7- La classe ayant la probabilité la plus élevée sera attribuée au document.

6.3.3.8- Avantages de la méthode adoptée (Naïve Bayes Classifier)

Cet algorithme dont le modèle d'apprentissage est très général est utilisé dans de nombreux autres domaines que le texte.
David.D Lewis dans (Lewis, 2004) et Hassane Hilali dans (Hilali, 2009) listent un ensemble d'avantages du classifieur bayésien naïf, parmi lesquelles :

- Algorithme facile et simple à implémenter
- Basée sur une théorie mathématique précise
- Efficacité et rapidité dans l'apprentissage et la classification
- Facile à mettre à jour avec de nouveaux exemples d'apprentissage
- Equivalent à un classifieur linéaire, dans sa rapidité d'application

- L'hypothèse d'indépendance des paramètres assouplit l'algorithme pour qu'il soit favorable pour différents types de données
- Très efficace avec des petits corpus d'apprentissage
- Résiste au bruit existant dans les données d'entrée
- Utile pour la classification déterministe comme pour le Ranking puisque il ordonne les classes par degré d'appartenance pour un texte donné
- Requiert une petite quantité de données d'apprentissage pour estimer les paramètres
- Enfin, le plus important c'est que les méthodes Naïve Bayes donnent de bons résultats

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

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

6.3.4- Mesures de performances utilisées pour l'évaluation

Puisque nous disposons de plusieurs modèles de catégorisation, nous devons mesurer la qualité des réponses données par le classifieur.
En principe, choisir *le Rappel* ***(R),****la Précision* ***(P)*** et *la F-measure* ($\boldsymbol{F_1}$) pour évaluer et comparer les différents modèles de catégorisation construits n'a pas besoin d'être justifié, et ce choix est presque automatique puisque ces trois indicateurs sont utilisés régulièrement dans le domaine de classification de textes et la recherche d'information avec succès depuis une trentaine d'années.
$\boldsymbol{F_1}$ 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 :

$$F_1 = \frac{2 * P * R}{P + R}$$

Nous tenons aussi à rappeler que pour une catégorie C_i, *la précision* évalue la qualité du classifieur à ne pas introduire de documents d'une autre catégorie dans C_i. Il s'agit du nombre de documents bien classés sur le nombre de documents classés dans C_i.

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

L*e rappel*, quant à lui, évalue le degré de complétude, c'est-à-dire le nombre de documents bien classés sur le nombre total de documents de la classe C_i.

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

Notre objectif est converger ces deux grandeurs vers la valeur un (1) pour que $\boldsymbol{F_1}$ convergera aussi vers 1, un objectif difficile à atteindre puisque un rappel intéressant ne peut être acquis qu'au prix d'une faible précision et vice-versa.

Ces mesures précédentes vont servir à évaluer le système par rapport à une seule classe. Pour une évaluation globale du classifieur par rapport à toutes les classes du corpus, nous avons choisit d'utiliser les mesures *micro moyenne* qui correspondent à une moyenne qui pondère les classes par le nombre de documents qu'elles contiennent. La micro-moyenne accorde donc des poids importants aux catégories ayant beaucoup d'exemples.
La micro-moyenne (traduction de micro-averaging) calcule les mesures rappel et précision de façon globale, cela revient à sommer les cases VP et FP de chaque catégorie pour obtenir la table de contingence globale.
La performance globale du classifieur est indiquée en calculant les différentes moyennes (***P, R, F_1***) qui sont calculées à partir des valeurs cumulées.

La précision, le rappel et la F_1 micro-moyenne sont calculés comme suit :

$$P = \frac{VP}{VP + FP} = \frac{\sum_{i=1}^{|C|} VP_i}{\sum_{i=1}^{|C|} (VP_i + FP_i)}$$

$$R = \frac{VP}{VP + FN} = \frac{\sum_{i=1}^{|C|} VP_i}{\sum_{i=1}^{|C|} (VP_i + FN_i)}$$

$$F_1 = \frac{2 * P * R}{P + R}$$

6.3.5- Les Systèmes Multi-Agents

Contrairement aux différentes approches décrites dans l'état de l'art basé sur un seul point de vue (Intelligence Artificielle classique : le penseur isolé) , et afin d'améliorer les performances du processus de classification basé sur un seul module logiciel on est ainsi naturellement conduit à chercher à donner plus d'autonomie et d'initiative aux différents modules logiciels en optant pour la distribution de la tâche de classification à un Système-Multi-Agents autonome collaboratif (Intelligence Artificielle distribuée : la communauté de penseurs).
Comme les premiers logiciels étaient construits à l'image que l'on se fait du raisonnement humain, les logiciels d'aujourd'hui (logiciel multi-agents) sont bâtis à l'image que l'on se fait du fonctionnement d'une société d'humains : plusieurs composants (appelés agents) réalisent chacun une tâche spécifique, interagissent et communiquent entre eux pour assurer la cohérence, la complétude et la correction d'une activité globale. Comme toute société d'humains, les agents pourront se réorganiser entre eux et adapter leur comportement à l'évolution de l'environnement (concept d'apprentissage). Mais, il va sans dire que tout ce qui se passe dans la tête des agents est entièrement défini par le concepteur.
L'utilisation d'une architecture Multi-Agents adoptant un comportement social de type « *Fourmis* » peut se présenter alors, comme un vrai remède pour améliorer les performances de notre modèle de classification. Sur des corpus de taille faible, la différence avec un système classique (un seul agent) n'est pas vraiment remarquée mais dés qu'on passe au

traitement des corpus de grande taille ou même infinie comme le Web une telle architecture pourrait être une très bonne solution pour notre problème. Peu de travaux ont déjà plus ou moins traité la problématique de classification de textes à base des SMA (Lumer, 1994), (Monmar, 1999).

On est ainsi naturellement conduit à chercher à donner plus d'autonomie et d'initiative aux différents modules logiciels. Le concept de système multi-agents propose un cadre de réponse à ces deux enjeux complémentaires (et à première vue contradictoires) : **autonomie** et **organisation**.

Dans un contexte de catégorisation automatique de textes, l'autonomie des agents est exprimée dans la première phase du processus par l'attribution des documents aux classes d'une façon indépendante des autres agents, toutefois l'organisation du SMA et la collaboration des agents du système entre eux peut être expliquée par la décision finale de classification du document à une classe choisie après un vote majoritaire des agents.

6.4- Base de texte utilisée pour l'évaluation

Afin de pouvoir comparer les performances obtenues par divers algorithmes, il est nécessaire de les tester sur les mêmes corpus.
Le terme « corpus » désignait à l'origine des sources documentaires sous forme de recueil de textes rassemblant exhaustivement tous les documents pour certains champs d'étude. (Benveniste, 2000). Néanmoins cette notion d'exhaustivité n'est pas toujours possible dans tous les domaines puisque les corpus de langue vivante, par exemple, sont ouverts et des mises à jour sont proposées en permanence
Plus récemment, une définition plus vague du terme « corpus électronique » est apparue, il s'agit d'une collection de textes sous un format compréhensible par l'ordinateur.
De nombreux types de corpus ont vu le jour ces dernières années qui s'organisent en différentes typologies. M.Antoniotti, préfère classifier les corpus en fonction des caractéristiques qui les opposent. (Antoniotti, 2002).

Le tableau suivant résume les types de corpus :

Critère d'opposition	Types de corpus
Langues	Corpus monolingues / Corpus multilingues
Taille	Echantillons / Textes entiers
Evolutivité	Corpus fermés / Corpus ouverts
Thèmes traités	Corpus généralistes / Corpus spécialisés
Pré-traitements des textes	Corpus bruts / Corpus préparés

Tableau 6.1 : Types de corpus

Quelques bases de textes sont donc émergées comme « corpus de référence » pour la catégorisation de textes. Ils doivent regrouper un certain nombre de documents qui sont tenus d'être de diverses utilisations. Ainsi une dimension suffisante et la diversité des documents sont deux caractéristiques principales d'un corpus pour qu'il soit qualifié de « référence ».
L'utilisation de ces corpus standards permet ainsi une comparaison plus aisée des performances des différentes techniques de classification.
On trouve principalement des comparaisons sur la base ***Reuters***, qui est une classification de dépêches de presse. Dans le domaine médical, on se réfère également à la base ***Ohsumed***.
Y.yang a mis cet aspect en évidence dans son étude qui synthétise les performances sur le corpus Reuters en évaluant les résultats obtenus de plusieurs algorithmes d'apprentissages sur divers versions de Reuters. (Yang, 1999).

6.4.1- Présentation générale du corpus Reuters

Reuters est un corpus de dépêches en langue anglaise qui a été proposé par l'agence de presse Reuters en 1987. Il correspond à une problématique de classification en plusieurs classes (un document appartient à une ou plusieurs classes).
Deux qualités principales caractérisent les documents de Reuters c'est qu'ils sont courts et plutôt homogènes avec un vocabulaire riche (environ 17000 mots), et la disponibilité gratuitement de la base dans le Web (http://www.research.att.com/~lewis/reuters21578.html), pour la version Lewis et sur (http://www-2.cs.cmu.edu/~yiming/) pour les versions Yang, Apte et PARC.
Les diverses expérimentations des chercheurs sur la base ont fait de ce corpus comme corpus de référence dans le domaine de la classification supervisée de textes.
Citons à titre d'exemple les auteurs : (Yang & Liu, 1999] avec 13 algorithmes (SVM, RN,AD,NB, etc..), (Schapire & all, 1998) avec Rocchio, (Joachims, 1998) et (Dumais & all, 1998) et (Jalam, 2003) avec (SVM), qui ont utilisé Reuters comme corpus pour apprendre, tester et évaluer les performances de leurs classifieurs.

Le tableau 6.2 illustre un exemple de texte du corpus :

```
<TITLE>AMERICUS TRUST &lt;HPU> EXTENDS DEADLINE</TITLE>
<TEXT>
Americus Trust for American Home
Products Shares said it extended its deadline for accepting
tendered shares until November 26, an extension of nine months.
    The trust, which will accept up to 7.5 mln shares of
American Home Products &lt;AHP>, said it has already received
tenders for about four mln shares.
    The trust is managed by Alex. Brown and Sons Inc &lt;ABSB> and
was formed November 26, 1986.
 </TEXT>
```

Tableau 6.2 : Exemple de texte du corpus Reuters-21578.

(On peut noter l'utilisation du sigle <> pour signaler le nom d'entreprise)

6.4.2- Historique

Ce corpus initialement nommé Reuters-22173 comportait 22173 dépêches de presse qui ont été publiés par en 1987. Tous les articles ont été rassemblés et indexés à des catégories par le personnel de l'agence et le groupe Carnegie (Carnegie Group Inc - CGI).
En 1990, les documents ont été mis à la disposition du laboratoire de recherche d'information (Université de Massachusetts à Amherst), par Reuters et CGI, pour des fins de recherche. Le formatage des documents et la génération de fichiers associés a été réalisée en 1990 par David D. Lewis et Stephen Harding au même laboratoire.
Un autre formatage et génération des fichiers associés a été fait été fait en 1991 et 1992 par David D. Lewis et Peter Shoemaker au Centre d'information et études du langage (Université de Chicago). Cette base de fichiers a donné naissance au 01/01/1993, à la première version dénommée "Reuters-22173"

6.4.3- Evolution du corpus

Lors de la conférence ACM SIGIR 96 en août 1996, un groupe de chercheurs dans le domaine de catégorisation de texte ont débattu une démarche qui pourrait faire de Reuters-22173 corpus de référence dans toutes les études. Il a été a décidé qu'une nouvelle version de la base

doit être produite avec moins d'ambiguïtés, avec une orthographe soignée. L'occasion était également parfaite, pour corriger un certain nombre d'erreurs typographiques, des erreurs dans la catégorisation et dans le formatage des documents. Steve Finch et David D. Lewis ont réalisé ce travail de Septembre à Novembre 1996, se basant essentiellement sur le SGML. Le résultat était la suppression de 595 documents qui étaient des répliques exactes d'autres documents du corpus. Le nouveau corpus épuré contenait donc 21 578 documents, et est donc appelé Reuters-21578 collection.
Initialement, Reuters-21578 est disponible dans 22 fichiers. Chacun des 21 premiers fichiers (reut2-000.sgm jusqu'à reut2-020.sgm) contient 1000 documents, tandis que le dernier (reut2-021.sgm) contient 578 documents. Les fichiers sont sous format SGML.
Une autre mise à jour faite par Lewis en écartant 1765 textes sans catégories prédefinies, qui sont inutiles dans un contexte d'apprentissage supervisé. Cette version est connue par Reuters "ModLewis" composée de 13265 documents pour l'ensemble d'apprentissage, et 6188 documents pour l'ensemble de test.
Depuis, plusieurs versions ont été diffusées, les différences entre ces versions concernent essentiellement le nombre des catégories du corpus, ainsi que la manière de définir le découpage des corpus d'apprentissage et de test.
Ainsi pour évaluer les méthodes de catégorisation, la collection de textes Reuters est généralement répartie en deux ensembles : l'ensemble d'apprentissage (textes pré-catégorisés) et l'ensemble de test (textes à catégoriser).
Le découpage le plus souvent rencontré se nomme découpage *Apté* du nom des premiers auteurs à l'avoir proposé (Apté & all, 1994). La base d'apprentissage initiale est constituée des documents antérieurs au 8 avril 1987, soit 9603 documents, et la base de test de tous les documents ultérieurs, soit 3299 documents, soit 8676 documents écartés.
Malheureusement, il existe de légères modifications à ce découpage qui rendent certaines comparaisons difficiles. Ainsi (Yang & Liu, 1999) ont supprimé de la base de test tous les documents qui n'appartiennent à aucune catégorie : ils n'utilisent que 3019 documents sur la base de test. La suppression de ces documents ne peut qu'améliorer les résultats par rapport au découpage traditionnel, puisque les risques de mauvais classement sont réduits. (Dumais & all, 1998) considèrent 118 catégories : certaines catégories n'ont donc pas de documents étiquetés sur la base de test, et la façon dont ces catégories sont prises en considération dans leur évaluation n'est pas évidente.

Le tableau 6.3 montre 5 versions proposées parmi d'autres, avec les statistiques concernant chacune d'elles :

Nombre de catégories	Nombre des documents d'apprentissage	Nombre des documents de test	Nombre global des documents	Corpus
182	21450	723	22173	Reuters- 22173
182	20856	722	21578	Reuters-21578a
135	14704	6746	21578	Reuters-21578b
113	13625	6188	19813	Reuters-ModLewis
118	9603	3299	12902	Reuters-ApteMod94
90	9586	3745	13331	Reuters-ApteModb
10	7194	2788	9982	Reuters-Top10

Tableau 6.3 : Principales versions de la collection Reuters

En conséquent, il reste délicat de comparer les performances obtenues sur les différentes versions du corpus. Cependant, les caractéristiques globales du corpus sont restées identiques, et les remarques sur le comportement général des systèmes étudiés sont toujours valables.
N'empêche que les réelles comparaisons restent l'évaluation des différents classifieurs sur les mêmes versions du corpus. Comme c'est le cas dans nos expérimentations qui vont être effectuées sur la même version du corpus à savoir ***Reuters21578-Top10*** qui sera tout simplement nommé corpus **Reuters** dans la suite de ce mémoire.

6.4.4- Définition des catégories du corpus Reuters-21578-ApteMod

Les mises à jour Reuters-21578 ApteMod ou ModeApté, ont été obtenues par la suppression des documents non étiquetés que comportait la version précédente (textes ambigus) , elles ont permis aussi de supprimer les documents présents deux fois, de corriger des erreurs typographiques, et d'autre part par la conservation des catégories ayant au moins un document dans la base d'apprentissage et un dans la base de test. Pour se ramener à moins d'une centaine de catégories. Il en résulte 90 catégories avec 9586 documents pour l'ensemble d'apprentissage et 3745 pour l'ensemble de test.

Les 90 catégories issues du découpage des ensembles d'apprentissages et de tests sont présentées dans le tableau 6.4, ainsi que le nombre de documents associés disponibles sur chaque base. Elles sont classées par ordre décroissant du nombre de documents associés sur la base d'apprentissage. Le nombre de documents disponibles pour effectuer l'apprentissage décroît rapidement ; dès la vingt-sixième catégorie, ce nombre est inférieur à cinquante. Il faut noter que les documents sont à peu près également répartis sur les deux bases, c'est-à-dire que les catégories ayant beaucoup (respectivement peu) de documents associés à la base d'apprentissage ont également beaucoup (respectivement peu) de documents associés à la base de test.

	Catégorie	Apprentissage	Test
1	Earn	2877	1087
2	Acquisition	1650	719
3	Money-fx	538	179
4	Grain	433	149
5	Crude	389	189
6	Trade	369	118
7	Interest	347	131
8	Wheat	212	71
9	Ship	197	89
10	Corn	182	56
11	Money-supply	140	34
12	Dlr	131	44
13	Sugar	126	36
14	Oilseed	124	47
15	Coffee	111	28
16	Gnp	101	35
17	Gold	94	30
18	Veg-oil	87	37
19	Soybean	78	33
20	Nat-gas	75	30
21	Bop	75	30
22	Livestock	75	24

	Catégorie	Apprentissage	Test
46	Tin	18	12
47	Rapeseed	18	9
48	Orange	16	11
49	Housing	16	4
50	Strategic-metal	16	11
51	Hog	16	6
52	Lead	15	14
53	Soy-oil	14	11
54	Heat	14	5
55	Soy-meal	13	13
56	Fuel	13	10
57	Lei	12	3
58	Sunseed	11	5
59	Dmk	10	4
60	Lumber	10	6
61	Tea	9	4
62	Income	9	7
63	Oat	8	6
64	Nickel	8	1
65	L-cattle	6	2
66	Groundnut	5	4
67	Instal-debt	5	1

23	Cpi	69	28
24	Reserves	55	18
25	Cocoa	55	18
26	Carcass	50	18
27	Copper	47	18
28	Jobs	46	21
29	Yen	45	14
30	Ipi	41	12
31	Iron-steel	40	14
32	Cotton	39	20
33	Gas	37	17
34	Barley	37	14
35	Rubber	37	12
36	Alum	35	23
37	Rice	35	24
38	Palm-oil	30	10
39	Meal-feed	30	19
40	Sorghum	24	10
41	Retail	23	2
42	Zinc	21	13
43	Silver	21	8
44	Pet-chem	20	12
45	Wpi	19	10

68	Rape-oil	5	3
69	Platinum	5	7
70	Sun-oil	5	2
71	Jet	4	1
72	Coconut	4	2
73	Coconut-oil	4	3
74	Potato	3	3
75	Propane	3	3
76	Cpu	3	1
77	Copra-cake	2	1
78	Palmkernel	2	1
79	Naphtha	2	4
80	Palladium	2	1
81	Rand	2	1
82	Dfl	2	1
83	Nzdlr	2	2
84	Rye	1	1
85	Cotton-oil	1	2
86	Lin-oil	1	1
87	Castor-oil	1	1
88	Sun-meal	1	1
88	Groundnut-oil	1	1
90	Nkr	1	2

Tableau 6.4 : Répartition des documents par catégorie, avec le nombre de documents associés pour l'apprentissage et le test

Ce corpus souffre d'une mauvaise définition de ces catégories, et d'un grand déséquilibre dans la répartition des documents entre ces catégories. En effet, il existe des catégories qui sont favorisées par rapport aux autres en termes de nombre des documents présents dans les jeux de tests et apprentissage. La figure 6.1 montre que seules les 20 premières catégories contiennent plus de 100 textes.

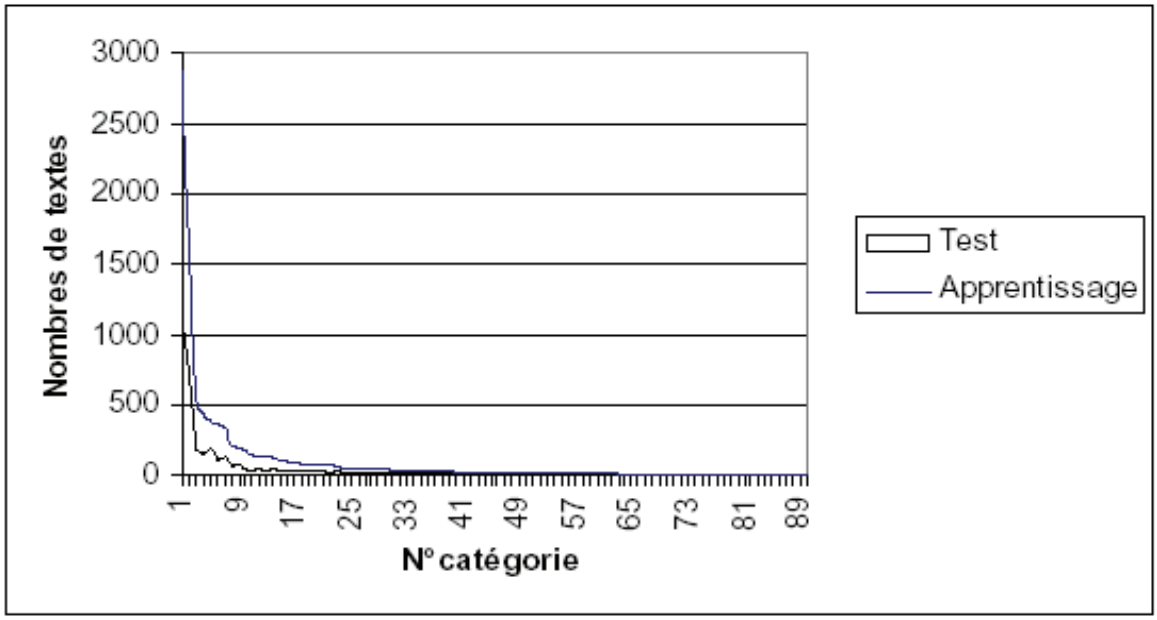


Figure 6.1 : Nombre de textes par catégories de la collection Reuters - (Jalam, 2003)

6.4.5- Reuters21578-ModeApté[10]

Plusieurs expériences ont été réalisées sur la base de textes Reuters-21578. Pour nos travaux, l'ensemble de nos évaluations a été réalisé sur une sous-collection de cette base connue par Reuters21578-Top10 que nous avons utilisé pour entraîner et tester notre classifieur.
Puisque notre objectif est d'associer un texte à une catégorie exclusive, nous avons opté pour une version du corpus qui ne contient ni les textes non étiquetés, ni les textes de multiples étiquettes. En outre, toutes les classes mal représentées avec moins de 150 dépêches d'apprentissage et moins de 50 dépêches de test ont été éliminés. L'ensemble des dépêches qui en résulte est 9982 dont 7194 textes d'apprentissage et 2788 de test. Les 9982 documents, correspondent aux 10 classes les plus représentées dans le corpus. Le tableau 6.5 illustre la répartition de ces 9 982 documents sur les 10 classes :

	Catégorie	Apprentissage	Test	Total catégorie
1	Earn	2877	1087	3964
2	Acquisition	1650	719	2369
3	Money-fx	538	179	717
4	Grain	433	149	582
5	Crude	389	189	578
6	Trade	369	118	487
7	Interest	347	131	478
8	Wheat	212	71	283
9	Ship	197	89	286
10	Corn	182	56	238
Total Général		**7194**	**2788**	**9982**

Tableau 6.5 : Reuters21578-Top10

Reuters-Top10 est donc, une version abrégée du corpus Reuters-21578, qui conserve seulement les 10 premières catégories ayant le plus d'effectifs en nombre de documents associés, toutefois elle compte presque 50% des documents du corpus Reuters21578 1^{ére} version et presque 80% de la version "ModApté".
Certainement, manipuler 80% du corpus avec 10 catégories seulement assouplit le traitement et allège considérablement le processus. Donc de toute évidence et d'une manière générale traiter, comparer et présenter les résultats de 10 catégories est plus pratique que 90. Enfin, nous tenons à préciser que nous ne sommes pas les seuls à choisir et opter pour le Top10 de Reuters mais plusieurs auteurs ont confirmé dans leurs travaux qu'un classifieur qui réussit à classer dans les 10 catégories les plus représentées de Reuters, ne va pas échouer dans les autres. En revanche le seul inconvénient qui s'oppose c'est bien les possibilités de comparaisons qui se rétrécirent seulement aux expériences qui ont testé leurs classifieurs avec Reuters-Top10.

6.5- Applications opérationnelles

Dans cette partie on expose comment les techniques décrites dans ce mémoire ont été intégrées dans une application opérationnelle de catégorisation des dépêches de Reuters.
Dans une première application, ces modèles sont utilisés pour classer les différents documents du corpus avec une approche classique non distribuée basée sur une seule entité logicielle avec une variation dans la représentation des documents (2, 3, 4, 5, 6, et 7-grammes).

Contrairement à la première application, la deuxième application est basée sur une approche distribuée dans laquelle on va déléguer la tâche de classification à un Système Multi-Agents (SMA) autonome collaboratif.
Nous présentons dans cette section une description de l'ensemble de nos expérimentations avec les deux approches distribuées et non distribuées suivie d'une comparaison des différents résultats qui sera développé en fin de cette section.

6.5.1- Environnement de développement

Java est le nom d'une technologie mise au point par Sun Microsystems qui permet de produire des logiciels indépendants de toute architecture matérielle. Cette technologie s'appuie sur différents éléments qui, par abus de langage, sont souvent tous appelés Java :

- le **langage Java** est un langage de programmation orienté objet ;
- un programme java s'exécute dans une machine virtuelle, dite machine virtuelle Java ;
- le bytecode Java est le résultat de la compilation d'un programme écrit en Java par le compilateur Java ;
- la plate-forme Java correspond à la machine virtuelle Java plus des spécifications d'API :
    - Java Platform, Standard Edition (Java SE) contient les API de base et est destiné aux ordinateurs de bureau ;
    - Java Platform, Enterprise Edition (Java EE) contient, en plus du précédent, les API orientées entreprise et est destiné aux serveurs ;
    - Java Platform, Micro Edition (Java ME) est destiné aux appareils mobiles tels que assistants personnels ou smartphones ;

Le **langage Java** est un langage de programmation informatique orienté objet créé par James Gosling et Patrick Naughton employés de Sun Microsystems avec le soutien de Bill Joy (cofondateur de Sun Microsystems en 1982), présenté officiellement le 23 mai 1995 au *SunWorld*.

Le langage Java a la particularité principale que les logiciels écrits avec ce dernier sont très facilement portables sur plusieurs systèmes d'exploitation tels que UNIX, Windows, Mac OS ou GNU/Linux avec peu ou pas de modifications. C'est la plate-forme qui garantit la portabilité des applications développées en Java.

Le langage reprend en grande partie la syntaxe du langage C++, très utilisé par les informaticiens. Néanmoins, Java a été épuré des concepts les plus subtils du C++ et à la fois les plus déroutants, tels que les pointeurs et références, et l'héritage multiple remplacé par l'implémentation des interfaces. Les concepteurs ont privilégié l'approche orientée objet de sorte qu'en Java, tout est objet à l'exception des types primitifs (nombres entiers, nombres à virgule flottante, etc.)

L'utilisation native du langage Java pour des applications sur un poste de travail restait jusqu'à présent relativement rare à cause de leur manque de rapidité. Cependant, avec l'accroissement rapide de la puissance des ordinateurs, les améliorations au cours de la dernière décennie de la machine virtuelle Java et de la qualité des compilateurs, plusieurs technologies ont gagné du terrain comme par exemple Netbeans et l'environnement Eclipse, les technologies de fichiers partagés Limewire et Azureus. Java est aussi utilisé dans le programme de mathématiques Matlab, au niveau de l'interface homme machine et pour le calcul formel. Les applications Swing apparaissent également comme une alternative à la technologie .NET. (www.fr.wikipedia.org)

Ainsi notre choix a été justifié par ces avantages qui s'ajoutent au fait que la plateforme Java avec tous ses éléments est téléchargeable gratuitement.

6.5.2- Approche non distribuée

6.5.2.1- Démarche à suivre

- **Prétraitements**
  - Conversion des majuscules en minuscule, éliminer le signes de ponctuations ()[]{}=:?!;-,_"+*/.",<>≥≤%«»&, de même pour les chiffres qui ne sont pas pris en compte.
  - Segmentation des textes en n-grammes.
  - Construire la liste du vocabulaire de tous les termes distincts qui apparaissent dans tous le corpus d'apprentissage.
  - Calcul des fréquences des termes (attributs) dans les documents d'apprentissage.
  - Tous les documents du jeu d'apprentissage sont transformés en une matrice des fréquences des termes dont les colonnes sont les dix classes du corpus et les lignes correspondent à tous les termes du vocabulaire.
- **Apprentissage**
  - Entrainer le classifieur sur le corpus d'apprentissage, en calculant les probabilités à priori des dix classes et les vraisemblances de tous les termes du vocabulaire relatives à ces classes.
- **Test**
  - Tester le classifieur en utilisant les documents du corpus 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.
  - Générer les matrices de contingence correspondantes : Vrai Positif (**VP**), Faux Positif (**FP**), Faux Négatif (**FN**), Vrai Négatif(**VN**).
  - Calculer les mesures de performances Précision/Rappel/F-measure.
- **Choix du meilleur classifieur**
  - Répéter le même processus pour 2, 3, 4, 5, 6 et 7-grammes.
  - Evaluer et comparer les résultats des différents classifieurs construits, par les mesures de performances calculées précédemment.
  - Le modèle de classification qui fournit les meilleures F-measure ($\boldsymbol{F_1}$) sera adopté pour l'approche distribuée.
- **Classification de nouveaux textes**
  - Conversion des majuscules en minuscule, éliminer les mêmes signes de ponctuations, de même pour les chiffres.
  - Segmentation du texte en 4-grammes.
  - Calculer les probabilités à posteriori d'appartenance du document aux dix classes.
  - Le document est assigné à la catégorie ayant la probabilité à postériori la plus grande.

6.5.2.2- Résultats expérimentaux

- **Matrices de contingence**

  Les documents correctement classés sont des VP ;

Les documents correctement non classés sont des VN ;
Les documents incorrectement classés sont des FP ;
Les documents incorrectement non classés sont des VN.

Catégorie Ci		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	**VP_i**	**FP_i**
	Non	**FN_i**	**VN_i**

■ **Algorithme de construction des matrices**

Début
Répéter pour tous les documents du corpus de test :
Si C_i_Doc est correctement attribué à C_i
Alors
$VP_i=VP_i+1$;
$VN_j=VN_j+1$; pour toutes les classes autre que C_i

Sinon C_i_Doc est incorrectement attribué à C_j
$FN_i=FN_i+1$; $FP_j=FP_j+1$
Fin de la boucle Répéter
Ecrire VP, FP, FN, VN des 10 classes
Fin de l'algorithme

■ **Calcul de précision, rappel et F-measure**

Pour chaque classe : $R_i = \frac{VP_i}{VP_i + FN_i}$, $P_i = \frac{VP_i}{VP_i + FP_i}$, $F1_i = \frac{2 * P_i * R_i}{P_i + R_i}$

Les mesures MicroMoyennes :

$$P = \frac{VP}{VP + FP} = \frac{\sum_{i=1}^{|C|} VP_i}{\sum_{i=1}^{|C|} (VP_i + FP_i)}, \quad R = \frac{VP}{VP + FN} = \frac{\sum_{i=1}^{|C|} VP_i}{\sum_{i=1}^{|C|} (VP_i + FN_i)}, \quad F_1 = \frac{2 * P * R}{P + R}$$

- Dérouler la démarche avec des textes représentés en 2-grammes

R=90.3%
P=97.0%
F_1=93.6%

Catégorie Earn C1		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	982	30
	Non	105	707

R=53.4%
P=90.6%
F_1=67.2

Catégorie Acquisition C2		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	384	40
	Non	335	1305

R=57.0%
P=21.5%
F_1=31.2%

Catégorie Money-fx C3		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	102	372
	Non	77	1587

R=4.0%
P=50.0%
F_1=7.5%

Catégorie Grain C4		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	6	6
	Non	143	1683

R=3.2%
P=7.5%
F_1=6.1%

Catégorie Crude C5		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	6	2
	Non	183	1683

R=18.8%
P=56.4%
F_1=28.2%

Catégorie Trade C6		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	22	17
	Non	95	1667

R=86.3%
P=19.2%
F_1=31.3%

Catégorie Interest C7		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	113	477
	Non	18	1576

P=53.5%
R=35.2%
F_1=42.5%

Catégorie Wheat C8		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	38	70
	Non	33	1651

R=16.9%
P=55.6%
F_1=25.9%

Catégorie Ship C9		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	15	12
	Non	74	1674

R=42.1%
P=52.3%
F_1=36.2%

Catégorie Corn C10		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	21	72
	Non	35	1668

Matrice de contingence globale du corpus		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	**1689**	**1098**
	Non	**1098**	**15201**

Tableaux 6.6 : Matrices de contingence 2-grammes

La F-measure MicroAveraged du classifieur à base des documents codés en 2-grammes est

$$F_1=60.6\%$$

- Répéter la même démarche avec des textes représentés en 3-grammes

R=96.0%
P=91.4%
$F_1=93.6\%$

Catégorie Earn C1		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	1043	98
	Non	44	1117

R=80.9%
P=94.6%
$F_1=87.3\%$

Catégorie Acquisition C2		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	582	33
	Non	137	1578

P=71.5%
R=41.2%
$F_1=52.2\%$

Catégorie Money-fx C3		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	128	183
	Non	51	2032

R=26.2%
P=53.4%
$F_1=35.1\%$

Catégorie Grain C4		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	39	34
	Non	110	2121

R=40.2%
P=93.8%
$F_1=56.3\%$

Catégorie Crude C5		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	76	5
	Non	113	2084

R=76.9%
P=61.2%
$F_1=68.2\%$

Catégorie Trade C6		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	90	57
	Non	27	2070

R=77.9%
P=55.7%
F_1=65.0%

Catégorie Interest C7		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	102	81
	Non	29	2058

R=63.4%
P=39.5%
F_1=48.6%

Catégorie Wheat C8		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	45	69
	Non	26	2115

R=37.1%
P=63.5%
F_1=46.8%

Catégorie Ship C9		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	33	19
	Non	56	2127

R=39.3%
P=31.4%
F_1=34.9%

Catégorie Corn C10		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	22	48
	Non	34	2138

Matrice de contingence globale du corpus		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	**2160**	**627**
	Non	**627**	**19440**

Tableaux 6.7 : Matrices de contingence 3-grammes

La F-measure MicroAveraged du classifieur à base des documents codés en 3-grammes est

F_1=77.5%

- Le même processus avec des textes représentés en 4-grammes

R=93.5%
P=92.0%
F_1=92.7%

Catégorie Earn C1		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	1016	88
	Non	71	1229

R=90.5%
P=90.2%
F_1=90.4%

Catégorie Acquisition C2		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	651	71
	Non	68	1594

R=76.0%
P=74.3%
F_1=75.1%

Catégorie Money-fx C3		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	136	47
	Non	43	2109

R=40.9%
P=50.4%
F_1=45.2%

Catégorie Grain C4		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	61	60
	Non	88	2184

R=52.4%
P=86.6%
F_1=65.3%

Catégorie Crude C5		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	99	15
	Non	90	2146

R=88.0%
P=46.6%
F_1=60.9%

Catégorie Trade C6		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	103	118
	Non	14	2142

R=71.8%
P=66.2%
F_1=68.9%

Catégorie Interest C7		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	94	48
	Non	37	2151

R=56.3%
P=42.6%
F_1=48.5%

Catégorie Wheat C8		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	40	54
	Non	31	2205

R=30.3%
P=64.3%
F_1=41.2%

Catégorie Ship C9		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	27	15
	Non	62	2218

R=32.1%
P=40.9%
F_1=36.0%

Catégorie Corn C10		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	18	26
	Non	38	2227

Matrice de contingence globale du corpus		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	**2245**	**542**
	Non	**542**	**20205**

Tableaux 6.8 : Matrices de contingence 4-grammes

La F-measure MicroAveraged du classifieur à base des documents codés en 4-grammes est

$$F_1=80.6\%$$

- Ensuite avec des textes représentés en 5-grammes

R=80.1%
P=91.5%
F_1=85.4%

Catégorie Earn C1		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	871	81
	Non	216	1232

R=89.0%
P=84.8%
F_1=86.6%

Catégorie Acquisition C2		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	640	115
	Non	79	1463

R=84.4%
P=65.1%
F_1=73.5%

Catégorie Money-fx C3		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	151	81
	Non	28	1952

R=66.4%
P=48.1%
F_1=55.8%

Catégorie Grain C4		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	99	107
	Non	50	2004

R=60.3%
P=77.0%
F_1=67.7%

Catégorie Crude C5		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	114	34
	Non	75	1989

R=91.5%
P=37.4%
F_1=53.1%

Catégorie Trade C6		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	107	179
	Non	10	1996

R=52.7%
P=65.7%
F_1=58.5%

Catégorie Interest C7		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	69	36
	Non	62	2034

R=35.2%
P=44.6%
F_1=39.4%

Catégorie Wheat C8		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	25	31
	Non	46	2078

R=%
P=%
F_1=%

Catégorie Ship C9		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	26	19
	Non	63	2077

R=1.8%
P=50.0%
F_1=3.5%

Catégorie Corn C10		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	1	1
	Non	55	2102

Matrice de contingence globale du corpus		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	**2103**	**684**
	Non	**684**	**18927**

Tableaux 6.9 : Matrices de contingence 5-grammes

La F-measure MicroAveraged du classifieur à base des documents codés en 5-grammes est

$$F_1=75.5\%$$

- Répéter la démarche une autre fois avec des textes représentés en 6-grammes

R=77.2%
P=89.5%
F_1=82.9%

Catégorie Earn C1		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	839	98
	Non	248	1196

R=87.2%
P=81.9%
F_1=84.4%

Catégorie Acquisition C2		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	627	139
	Non	92	1408

R=87.2%
P=60.5%
F_1=71.4%

Catégorie Money-fx C3		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	156	102
	Non	23	1879

R=79.2%
P=48.0%
F_1=59.7%

Catégorie Grain C4		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	118	128
	Non	31	1917

R=54.5%
P=75.2%
F_1=63.2%

Catégorie Crude C5		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	103	34
	Non	86	1932

R=84.6%
P=35.7%
F_1=50.3%

Catégorie Trade C6		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	99	178
	Non	18	1936

R=47.3%
P=59.0%
F_1=52.5%

Catégorie Interest C7		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	62	43
	Non	69	1973

R=1.4%
P=50%
F_1=2.7%

Catégorie Wheat C8		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	1	1
	Non	71	2035

R=30.3%
P=55.1%
F_1=39.1%

Catégorie Ship C9		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	27	22
	Non	62	2008

R=7.1%
P=33.1%
F_1=11.8%

Catégorie Corn C10		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	4	8
	Non	52	2031

Matrice de contingence globale du corpus		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	**2035**	**752**
	Non	**752**	**18315**

Tableaux 6.10 : Matrices de contingence 6-grammes

La F-measure MicroAveraged du classifieur à base des documents codés en 6-grammes est

F_1=73,0%

- Et enfin calculer les mêmes mesures avec des textes représentés en 7-grammes

R=74.0%
P=86.8%
F_1=79.9%

Catégorie Earn C1		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	804	122
	Non	283	1138

R=86.9%
P=78.7%
F_1=82.6%

Catégorie Acquisition C2		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	625	169
	Non	94	1317

R=77.1%
P=59.5%
F_1=67.2%

Catégorie Money-fx C3		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	138	94
	Non	41	1804

R=65.8%
P=47.3%
F_1=55.1%

Catégorie Grain C4		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	98	109
	Non	51	1844

R=45.0%
P=62.0%
F_1=52.1%

Catégorie Crude C5		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	85	52
	Non	104	1857

R=81.2%
P=32.6%
F_1=46.6%

Catégorie Trade C6		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	95	196
	Non	22	1847

R=48.9%
P=53.3%
F_1=51.0%

Catégorie Interest C7		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	64	56
	Non	67	1878

R=1.4%
P=33.3%
F_1=2.6%

Catégorie Wheat C8		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	1	2
	Non	71	1942

R=30.3%
P=50.9%
F_1=38.0%

Catégorie Ship C9		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	27	26
	Non	62	1915

R=10.7%
P=24.0%
F_1=14.8%

Catégorie Corn C10		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	6	19
	Non	50	1936

Matrice de contingence globale du corpus		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	**1942**	**845**
	Non	**845**	**17478**

Tableaux 6.11 : Matrices de contingence 7-grammes

La F-measure MicroAveraged du classifieur à base des documents codés en 7-grammes est

$$F_1=70.3\%$$

- **Choix du N (du N-Grammes)**

  Le modèle de classification de textes représentés en 4-grammes qui donne les meilleurs résultats à savoir les meilleures F_1, sera adopté pour l'approche distribuée.

6.5.3- Approche distribuée

6.5.3.1- Démarche à suivre

- **Prétraitements**
  - Conversion des majuscules en minuscule, éliminer le signes de ponctuations ()[]{}=:?!;-,_"+*/.",<>≥≤%«»&, de même pour les chiffres qui ne sont pas pris en compte.
  - Segmentation des textes en 4-grammes.
  - Construire la liste du vocabulaire de tous les termes distincts qui apparaissent dans des sous-ensembles du corpus d'apprentissage, obtenus en partageant la base de textes sur le nombre d'agents.
  - Calcul des fréquences des termes dans les documents d'apprentissage correspondants.
  - Les documents du mini-corpus d'apprentissage sont transformés en une matrice des fréquences des termes dont les colonnes sont les dix classes du corpus et les lignes correspondent à tous les termes du vocabulaire du mini-corpus.
- **Apprentissage**
  - Entrainer le classifieur multi-agents sur les sous-ensembles du corpus d'apprentissage : Chaque agent exercera son apprentissage sur un échantillon du corpus. Les résultats de l'apprentissage sont les probabilités à priori des dix classes et les vraisemblances de tous les termes du vocabulaire relatives à ces classes, pour chaque agent.
- **Test**
  - Tester le classifieur sur tout le corpus test : chaque document va être traité par l'ensemble de tous les agents, chaque agent dans le système fera sa propre classification pour le même texte.

- Le document sera catégorisé par un agent dans la classe qui possède la plus grande probabilité à posteriori.
- La catégorisation finale du texte sera faite dans la classe qui a été nommé par le plus grand nombre d'agents.
- Générer les matrices de contingence correspondantes.
- Calculer les mesures de performances Précision/Rappel/F-measure.

■ **Choix du meilleur classifieur**

- Répéter le même processus en augmentant le nombre d'agents 3, 9, 21, 33, 61, 99 et 181 agents (Le nombre est choisi impair pour éviter une égalité complète dans les votes et une catégorie l'emportera toujours)
- Evaluer et comparer les résultats des différents classifieurs multi-agents construits, par les mesures de performances calculées précédemment ajouté à un facteur très important à savoir le temps d'exécution.
- Le classifieur multi-agents qui procure les résultats les plus stables sera adopté.

■ **Classification de nouveaux textes**

- Conversion des majuscules en minuscule, éliminer les mêmes signes de ponctuations, de même pour les chiffres.
- Segmentation du texte en 4-grammes.
- Chaque agent du SMA va calculer indépendamment les probabilités à posteriori d'appartenance du document aux dix classes.
- Le document est assigné à la catégorie élue par le plus grand nombre d'agents.

6.5.3.2- Résultats expérimentaux

■ **Construction des matrices de contingence pour 3, 9, 21, 33, 61, 99 et 181 agents**

■ **Calcul de précision, rappel et F-measure**

- Exécuter le processus par un SMA composé de 3 agents

R=93.7%
P=92.6%
F_1=93.2%

Catégorie Earn C1		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	1019	81
	Non	68	1225

R=91.2%
P=90.6%
F_1=90.9%

Catégorie Acquisition C2		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	656	68
	Non	63	1594

R=73.7%
P=73.3%
F_1=73.5%

Catégorie Money-fx C3		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	132	48
	Non	47	2109

R=40.9%
P=51.7%
F_1=45.7%

Catégorie Grain C4		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	61	57
	Non	88	2174

R=54.0%
P=86.4%
F_1=66.4%

Catégorie Crude C5		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	102	16
	Non	87	2136

R=93.2%
P=48.7%
F_1=63.9%

Catégorie Trade C6		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	109	115
	Non	8	2141

R=69.5%
P=66.4%
F_1=67.9%

Catégorie Interest C7		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	91	46
	Non	40	2150

R=57.7%
P=42.7%
F_1=49.1%

Catégorie Wheat C8		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	41	55
	Non	30	2200

R=34.8%
P=70.5%
F_1=46.6%

Catégorie Ship C9		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	31	13
	Non	58	2210

R=37.5%
P=45.7%
F_1=41.2%

Catégorie Corn C10		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	21	25
	Non	35	2225

Matrice de contingence globale du corpus		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	2263	524
	Non	524	20164

Tableaux 6.12 : Matrices de contingence 3 agents

La F-measure MicroAveraged du classifieur à base d'un SMA de 3 agents est

$$F_1=81.2\%$$

- Répéter le même processus par 9 agents

R=94.1%
P=93.0%
$F_1=93.6\%$

Catégorie Earn C1		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	1023	77
	Non	64	1215

R=92.4%
P=92.0%
$F_1=92.2\%$

Catégorie Acquisition C2		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	664	58
	Non	55	1587

R=74.9%
P=73.6%
$F_1=74.2\%$

Catégorie Money-fx C3		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	134	48
	Non	45	2105

R=43.0%
P=50.8%
$F_1=46.5\%$

Catégorie Grain C4		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	64	62
	Non	85	2159

R=56.6%
P=87.0%
$F_1=68.6\%$

Catégorie Crude C5		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	107	16
	Non	82	2130

R=95.7%
P=51.1%
$F_1=66.7\%$

Catégorie Trade C6		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	112	107
	Non	5	2141

R=71.0%
P=69.4%
$F_1=70.2\%$

Catégorie Interest C7		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	93	41
	Non	38	2136

R=63.4%
P=47.9%
$F_1=54.5\%$

Catégorie Wheat C8		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	45	49
	Non	26	2111

R=36.0%
P=76.2%
F_1=48.9%

Catégorie Ship C9		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	32	10
	Non	57	2208

R=39.3%
P=48.9%
F_1=43.6%

Catégorie Corn C10		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	22	23
	Non	34	2225

Matrice de contingence globale du corpus		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	2296	491
	Non	491	20017

Tableaux 6.13 : Matrices de contingence 9 agents

La F-measure MicroAveraged du classifieur à base d'un SMA de 9 agents est

F_1=82.4%

- Dérouler la même démarche par 21 agents

R=95.1%
P=94.0%
F_1=94.6%

Catégorie Earn C1		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	1034	66
	Non	53	1214

R=93.2%
P=92.8%
F_1=93.0%

Catégorie Acquisition C2		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	670	52
	Non	49	1579

R=78.8%
P=77.5%
F_1=78.1%

Catégorie Money-fx C3		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	141	41
	Non	38	2104

R=48.3%
P=57.1%
F_1=52.4%

Catégorie Grain C4		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	72	54
	Non	77	2154

R=56.6%
P=82.9%
F_1=67.3%

Catégorie Crude C5		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	107	22
	Non	82	2132

R=95.7%
P=53.8%
F_1=68.9%

Catégorie Trade C6		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	112	96
	Non	5	2137

R=71.8%
P=67.6%
F_1=69.6%

Catégorie Interest C7		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	94	45
	Non	37	2128

R=66.2%
P=51.1%
F_1=57.7%

Catégorie Wheat C8		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	47	45
	Non	24	2100

R=39.3%
P=79.5%
F_1=52.6%

Catégorie Ship C9		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	35	9
	Non	54	2201

R=44.6%
P=55.6%
F_1=49.5%

Catégorie Corn C10		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	25	20
	Non	31	2117

Matrice de contingence globale du corpus		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	2337	450
	Non	450	19866

Tableaux 6.14 : Matrices de contingence 21 agents

La F-measure MicroAveraged du classifieur à base d'un SMA de 21 agents est

F_1=83.9%

- Dérouler une autre fois la même démarche par un SMA de 33 agents

R=95.8%
P=94.6%
F_1=95.2%

Catégorie Earn C1		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	1041	59
	Non	46	1220

R=92.8%
P=92.4%
F_1=92.6%

Catégorie Acquisition C2		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	667	55
	Non	52	1584

R=80.4%
P=77.4%
F_1=78.9%

Catégorie Money-fx C3		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	144	42
	Non	35	2112

R=53.0%
P=64.8%
F_1=58.3%

Catégorie Grain C4		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	79	43
	Non	70	2142

R=59.8%
P=87.6%
F_1=71.1%

Catégorie Crude C5		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	113	16
	Non	76	2128

R=97.4%
P=54.8%
F_1=70.2%

Catégorie Trade C6		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	114	94
	Non	3	2141

R=74.8%
P=70.5%
F_1=72.6%

Catégorie Interest C7		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	98	41
	Non	33	2136

R=73.2%
P=56.5%
F_1=63.8%

Catégorie Wheat C8		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	52	40
	Non	19	2096

R=42.7%
P=86.4%
F_1=57.1%

Catégorie Ship C9		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	38	6
	Non	51	2196

R=50.0%
P=62.2%
F_1=55.4%

Catégorie Corn C10		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	28	17
	Non	28	2108

Matrice de contingence globale du corpus		***Jugement Expert***	
		Oui	Non
Jugement classifieur	Oui	2374	413
	Non	413	19863

Tableaux 6.15 : Matrices de contingence 33 agents

La F-measure MicroAveraged du classifieur à base d'un SMA de 33 agents est

$$F_1=85.2\%$$

- Recommencer les mêmes traitements avec un SMA de 61 agents

R=96.0%
P=64.9%
$F_1=95.5\%$

Catégorie Earn C1		***Jugement Expert***	
		Oui	Non
Jugement classifieur	Oui	1044	56
	Non	43	1218

R=92.9%
P=92.5%
$F_1=92.7\%$

Catégorie Acquisition C2		***Jugement Expert***	
		Oui	Non
Jugement classifieur	Oui	668	54
	Non	41	1590

R=81.6%
P=78.5%
$F_1=80.0\%$

Catégorie Money-fx C3		***Jugement Expert***	
		Oui	Non
Jugement classifieur	Oui	146	40
	Non	33	2111

R=55.7%
P=68.0%
$F_1=61.3\%$

Catégorie Grain C4		***Jugement Expert***	
		Oui	Non
Jugement classifieur	Oui	83	39
	Non	66	2139

R=60.3%
P=88.4%
$F_1=71.7\%$

Catégorie Crude C5		***Jugement Expert***	
		Oui	Non
Jugement classifieur	Oui	114	15
	Non	75	2122

R=99.1%
P=55.8%
$F_1=71.4\%$

Catégorie Trade C6		***Jugement Expert***	
		Oui	Non
Jugement classifieur	Oui	116	92
	Non	1	214

R=77.1%
P=72.7%
F_1=74.8%

Catégorie Interest C7		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	101	38
	Non	30	2125

R=77.5%
P=59.8%
F_1=67.5%

Catégorie Wheat C8		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	55	37
	Non	16	2084

R=46.1%
P=93.2%
F_1=61.7%

Catégorie Ship C9		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	41	3
	Non	48	2184

R=55.4%
P=68.9%
F_1=61.4%

Catégorie Corn C10		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	31	14
	Non	25	2102

Matrice de contingence globale du corpus		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	2399	388
	Non	388	17889

Tableaux 6.16 : Matrices de contingence 61 agents

La F-measure MicroAveraged du classifieur à base d'un SMA de 61 agents est

F_1=86.1%

- Encore une fois avec 99 agents

R=94.3%
P=93.0%
F_1=93.7%

Catégorie Earn C1		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	1025	77
	Non	62	1218

R=91.7%
P=91.4%
F_1=91.5%

Catégorie Acquisition C2		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	659	62
	Non	60	1595

R=78.2%
P=78.2%
F_1=78.2%

Catégorie Money-fx C3		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	140	39
	Non	39	2114

R=52.3%
P=60.9%
F_1=56.3%

Catégorie Grain C4		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	78	50
	Non	71	2142

R=56.6%
P=82.9%
F_1=67.3%

Catégorie Crude C5		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	107	22
	Non	82	2129

R=93.2%
P=52.4%
F_1=67.1%

Catégorie Trade C6		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	109	99
	Non	8	227

R=74.8%
P=70.5%
F_1=72.6%

Catégorie Interest C7		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	98	41
	Non	33	2127

R=73.2%
P=56.5%
F_1=63.8%

Catégorie Wheat C8		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	52	40
	Non	19	2088

R=43.8%
P=88.6%
F_1=58.6%

Catégorie Ship C9		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	39	5
	Non	50	2200

R=51.8%
P=64.4%
F_1=57.4%

Catégorie Corn C10		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	29	16
	Non	27	2118

Matrice de contingence globale du corpus		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	2336	451
	Non	451	17958

Tableaux 6.17 : Matrices de contingence 99 agents

La F-measure MicroAveraged du classifieur à base d'un SMA de 99 agents est

$$F_1=83.8\%$$

- Et enfin répéter le processus par un SMA de 181 agents

R=92.5%
P=91.3%
$F_1=91.9\%$

Catégorie Earn C1		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	1006	96
	Non	81	1233

R=88.6%
P=88.6%
$F_1=88.6\%$

Catégorie Acquisition C2		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	637	82
	Non	82	1603

R=70.9%
P=70.6%
$F_1=70.8\%$

Catégorie Money-fx C3		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	127	53
	Non	52	2138

R=49.7%
P=57.4%
$F_1=53.2\%$

Catégorie Grain C4		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	74	55
	Non	75	2145

R=51.3%
P=75.2%
$F_1=61.0\%$

Catégorie Crude C5		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	97	32
	Non	92	2141

R=86.3%
P=48.6%
$F_1=62.2\%$

Catégorie Trade C6		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	101	107
	Non	16	248

R=69.5%
P=65.5%
$F_1=67.4\%$

Catégorie Interest C7		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	91	48
	Non	40	2138

R=66.2%
P=51.1%
F_1=57.7%

Catégorie Wheat C8		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	47	45
	Non	24	2095

R=37.1%
P=75.0%
F_1=49.6%

Catégorie Ship C9		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	33	11
	Non	56	2214

R=41.1%
P=51.1%
F_1=45.5%

Catégorie Corn C10		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	23	22
	Non	33	2127

Matrice de contingence globale du corpus		*Jugement Expert*	
		Oui	Non
Jugement classifieur	Oui	2236	551
	Non	551	18082

Tableaux 6.18 : Matrices de contingence 181 agents

La F-measure MicroAveraged du classifieur à base d'un SMA de 181 agents est

F_1=80.2%

- **Fixer le nombre d'agents composant notre SMA**

  Le modèle de classification de textes construit à base d'un SMA constitué de 61 agents, offrant les résultats optimaux, sera adopté pour la classification finale.

6.5.4- Comparaison des résultats

Nous proposons une série de comparaisons en quatre étapes :

1- En mettant en compétitions en premier temps nos résultats obtenus par les 6 variantes de Naïve Bayes (Documents représentés en 2, 3, 4, 5, 6 et 7-grammes).

2- Dans la deuxième comparaison nous allons confronter le meilleur résultat fourni par nos 6 modèles précédents avec des résultats références de 6 méthodes de catégorisation dans le domaine, à savoir les machines à vecteurs supports, Rocchio, les plus proches voisins, les arbres de décision, Naïve Bayes et les réseaux de neurones, appuyés sur les résultats obtenus par (Dumais & all, 1998), (Joachims, 1998), (Yang & Liu, 1999) et (Li & Yang, 2003), confrontés dans plusieurs études comme dans (Sebastiani, 2002), (Nakache, 2007) et (Manning & all, 2008), pour se situer dans une échelle des spécialistes du domaine et donner à nos performances une certaine crédibilité.

3- La troisième consiste à comparer les résultats obtenus en Mono-Agent avec ceux des SMA en renforçant chaque fois le nombre d'agents (3, 9, 21, 33, 61, 99 jusqu'à 181 agents).

4- La dernière comparaison va mettre en oppositions tous les six classifieurs ajouté à notre classifieur Naïve Bayes (Approche non distribuée) avec notre nouveau modèle Naïve Bayes basée sur une approche distribuée (SMA) composé de 61 agents.

6.5.4.1- Comparaison des résultats obtenus aves différentes valeurs de N (N-grammes)

Ce tableau reflète les résultats obtenus par l'algorithme Naïve Bayes avec des textes codés en 2, 3, 4, 5, 6 et 7-grammes :

F_1	N=2	N=3	N=4	N=5	N=6	N=7
Earn	93.5%	**93.6%**	92.7%	85.4%	82.9%	79.9%
Acq	67.2%	87.3%	**90.3%**	86.8%	84.4%	82.6%
money-fx	31.2%	52.2%	**75.1%**	73.5%	71.4%	67.1%
Grain	7.5%	35.1%	45.2%	55.8%	**59.7%**	55.1%
Crude	6.1%	56.3%	65.3%	**67.7%**	63.2%	52.1%
trade	28.2%	**68.2%**	60.9%	53.1%	50.3%	46.6%
Interest	31.3%	65.0%	**68.9%**	58.5%	52.5%	51.0%
ship	42.5%	**48.7%**	48.5%	39.4%	2.7%	2.6%
wheat	25.9%	**46.8%**	41.2%	38.8%	39.1%	38.0%
corn	28.2%	34.9%	**36.0%**	3.5%	11.8%	14.8%
Micro-Avg	**60.6%**	**77.5%**	**80.6%**	**75.5%**	**73.0%**	**70.3%**

Tableau 6.19 : Comparaison des résultats obtenus avec les différents N (N-grammes)

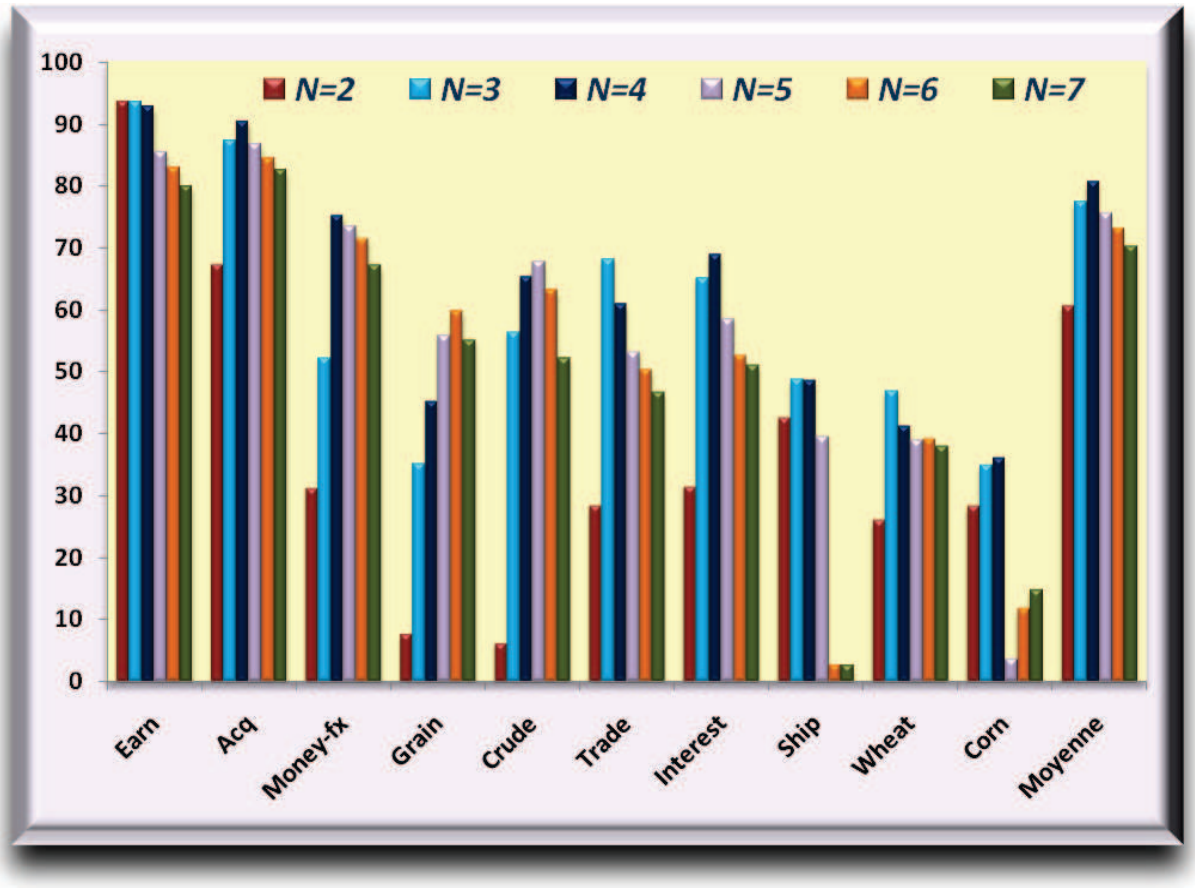


Figure 6.2: Comparaison des résultats obtenus avec les différents N (N-grammes)

6.5.4.2- Comparaison des résultats d'autres algorithmes

Le résultat de comparaison des méthodes SVM, Rocchio, kNN, les arbres de décision, Naïve Bayes et les réseaux de neurones avec notre algorithme Naïve Bayes (4-grammes) est le suivant :

F_1	SVM	Rocchio	kNN	Arb.Déc	Rés.Neur	N.Bayes	Notre NB
Earn	98.0%	92.9%	96.7%	97.8%	94.1%	95.9%	**92.7%**
Acq	93.6%	64.7%	91.6%	89.7%	88.8%	87.8%	**90.3%**
Money-fx	74.5%	46.7%	78.0%	66.2%	74.2%	56.6%	**75.1%**
Grain	94.6%	67.5%	86.4%	85.0%	73.8%	78.8%	**45.2%**
Crude	88.9%	70.1%	87.4%	85.0%	86.5%	79.5%	**65.3%**
Trade	75.9%	65.1%	77.3%	72.5%	79.5%	63.9%	**60.9%**
Interest	77.7%	63.4%	73.7%	67.1%	83.9%	64.9%	**68.9%**
Ship	85.6%	49.2%	49.4%	74.2%	89.9%	85.4%	**48.5%**
Wheat	91.8%	68.9%	69.1%	92.5%	79.7%	69.7%	**41.2%**
Corn	90.3%	48.2%	48.5%	91.8%	77.2%	65.3%	**36.0%**
Micro-Avg	**92.0%**	**64.6%**	**81.8%**	**88.4%**	**82.8%**	**81.5%**	**80.6%**

Tableau 6.20 : Comparaison des résultats obtenus avec ceux des autres algorithmes

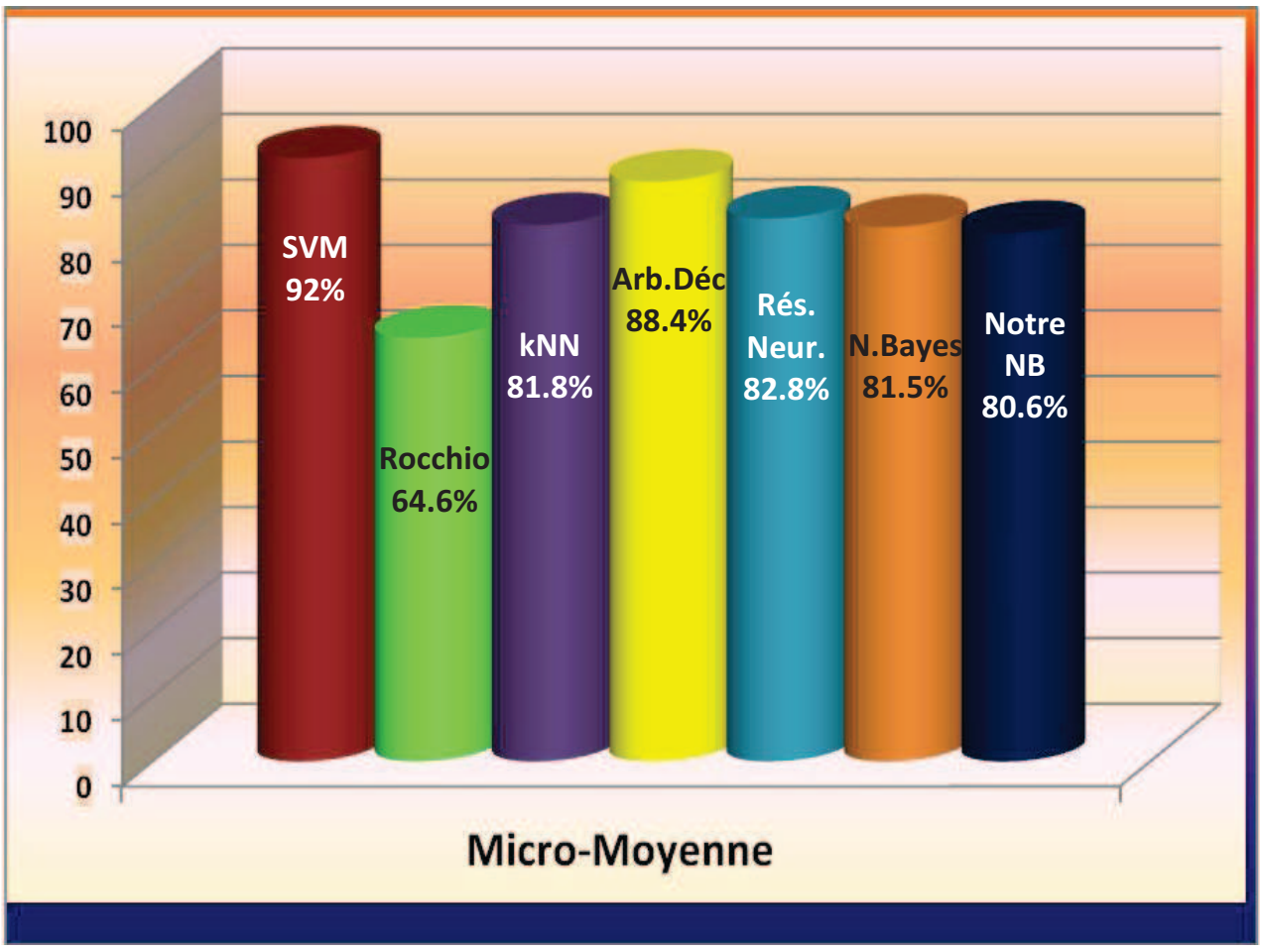


Figure 6.3 : Comparaison des résultats obtenus avec ceux des différents algorithmes

6.5.4.3- Comparaison des approches Mono et Multi-Agents en variant le nombre d'agents

Les deux tableaux suivants opposent tous les résultats obtenus avec le même algorithme Naïve Bayes avec des textes codés en 4-grammes, en commençant par le Mono-Agent et en augmentant au fur et à mesure le nombre d'agents. Les comparaisons vont être appuyées sur deux critères principaux à savoir les performances du classifieur en qualité des ses résultats et son efficacité en temps d'exécution du processus (prétraitement, apprentissage et test) sur tout le corpus.

F_1	MonoAgent	3Agents	9Agents	21Agents	33Agents	61Agents	99Agents	181Agents
Earn	92,7%	93,2%	93,6%	94,6%	95,2%	**95,5%**	93,7%	91,9%
Acq	90,3%	90,9%	92,2%	93,0%	92,6%	**92,7%**	91,5%	88,6%
Money-fx	75,1%	73,5%	74,2%	78,1%	78,9%	**80,0%**	78,2%	70,8%
Grain	45,2%	45,7%	46,5%	52,4%	58,3%	**61,3%**	56,3%	53,2%
Crude	65,3%	66,4%	68,6%	67,3%	71,1%	**71,7%**	67,3%	61,0%
Trade	60,9%	63,9%	66,7%	68,9%	70,2%	**71,4%**	67,1%	62,2%
Interest	68,9%	67,9%	70,2%	69,6%	72,6%	**74,8%**	72,6%	67,4%
Ship	48,5%	49,1%	54,5%	57,7%	63,8%	**67,5%**	63,8%	57,7%
Wheat	41,2%	46,6%	48,9%	52,6%	57,1%	**61,7%**	58,6%	49,6%
Corn	36,0%	41,2%	43,6%	49,5%	55,4%	**61,4%**	57,4%	45,5%
Micro-Avg	**80,6%**	**81,2%**	**82,4%**	**83,9%**	**85,2%**	**86,1%**	**83,8%**	**80,2%**

Tableau 6.21 : Comparaison des résultats obtenus avec les différents nombres d'agents

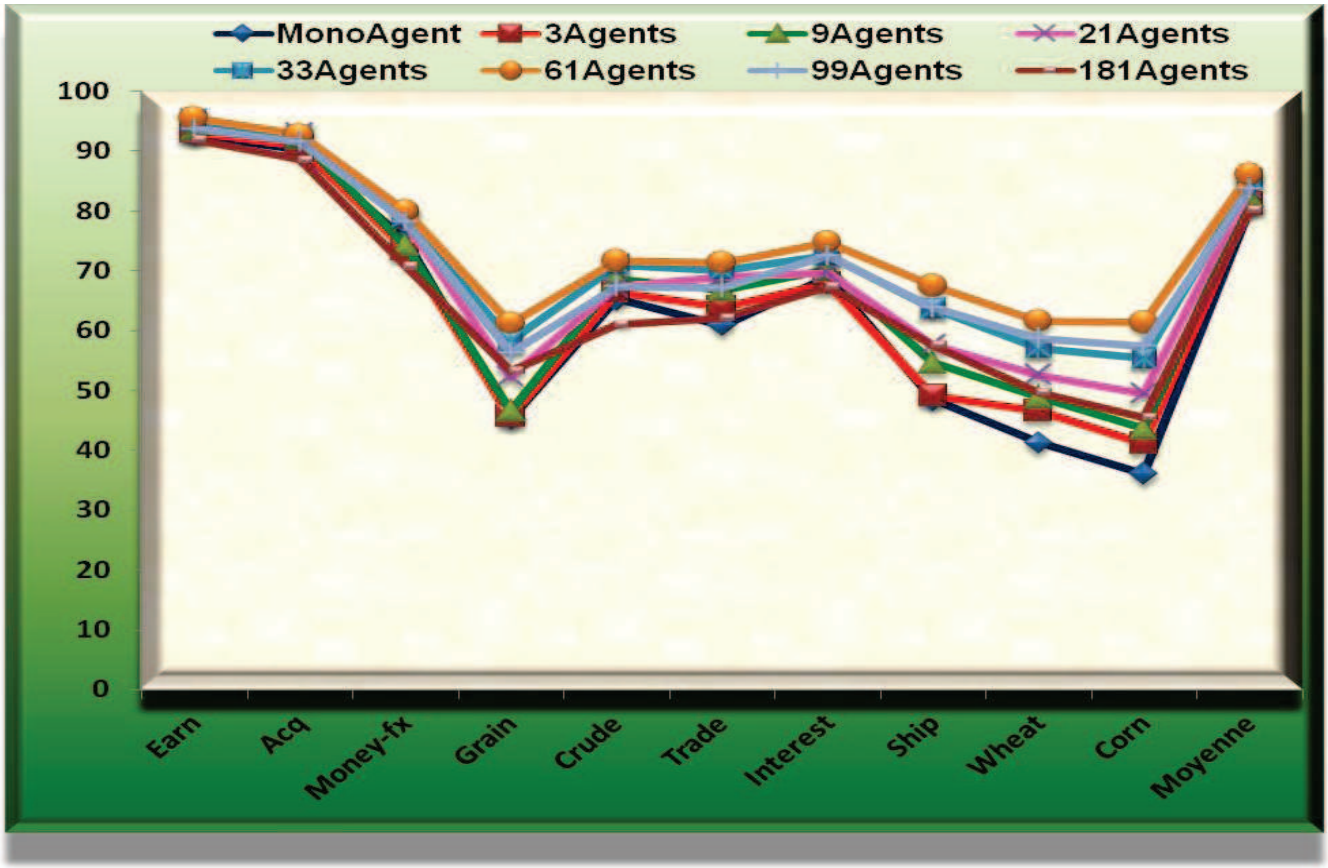


Figure 6.4 : Comparaison des résultats obtenus avec les différents nombres d'agents

Corpus	MonoAgent	3Agents	9Agents	21Agents	33Agents	61Agents	99Agents	181Agents
Prétrait.	334 Min	169 Min	75 Min	36 Min	30 Min	24 Min	27 Min	29 Min
App+Test	3 Min	8 Min	19 Min	37Min	53 Min	70 Min	105 Min	149 Min
Temps Exéc	**337 Min**	**177 Min**	**94 Min**	**73 Min**	**83 Min**	**94 Min**	**132 Min**	**178 Min**

Tableau 6.22 : Evaluation des temps d'exécution des systèmes mono et multi-agents

N.B : Les temps d'exécution sont évalués sur un HP Compaq DX7500, Pentium(R) Dual-Core CPU 2.5Ghz, 3 Go de mémoire vive.

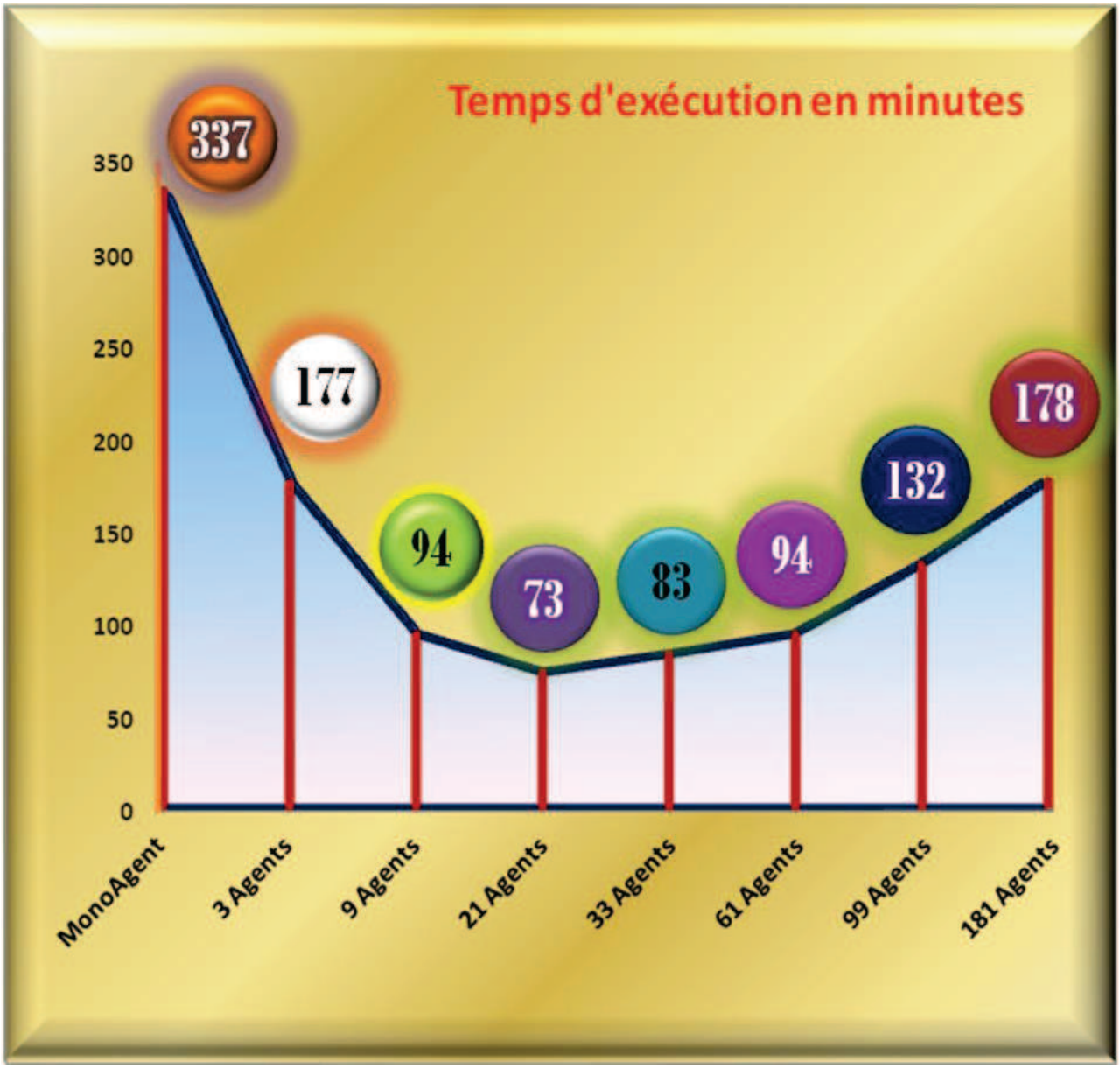


Figure 6.5 : Evaluation des temps d'exécution des systèmes mono et multi-agents

6.5.4.4- Comparaison des approches non distribuées avec notre approche SMA

La dernière confrontation va mettre en compétition les six classifieurs ajouté à notre classifieur Naïve Bayes Mono-Agent avec notre nouveau modèle construit Naïve Bayes basée sur une approche distribuée (SMA) composé de 61 agents.

F_1	SVM	Rocchio	kNN	Arb.Déc	Rés.Neur	N.Bayes	Notre NB	NB(SMA)
Earn	98.0%	92.9%	96.7%	97.8%	94.1%	95.9%	92.7%	95,5%
Acq	93.6%	64.7%	91.6%	89.7%	88.8%	87.8%	90.3%	92,7%
Money-fx	74.5%	46.7%	78.0%	66.2%	74.2%	56.6%	75.1%	80,0%
Grain	94.6%	67.5%	86.4%	85.0%	73.8%	78.8%	45.2%	61,3%
Crude	88.9%	70.1%	87.4%	85.0%	86.5%	79.5%	65.3%	71,7%
Trade	75.9%	65.1%	77.3%	72.5%	79.5%	63.9%	60.9%	71,4%
Interest	77.7%	63.4%	73.7%	67.1%	83.9%	64.9%	68.9%	74,8%
Ship	85.6%	49.2%	49.4%	74.2%	89.9%	85.4%	48.5%	67,5%
Wheat	91.8%	68.9%	69.1%	92.5%	79.7%	69.7%	41.2%	61,7%
Corn	90.3%	48.2%	48.5%	91.8%	77.2%	65.3%	36.0%	61,4%
Micro-Avg	**92.0%**	**64.6%**	**81.8%**	**88.4%**	**82.8%**	**81.5%**	**80.6%**	**86,1%**

Tableau 6.23 : Comparaison des différents résultats avec l'approche distribuée

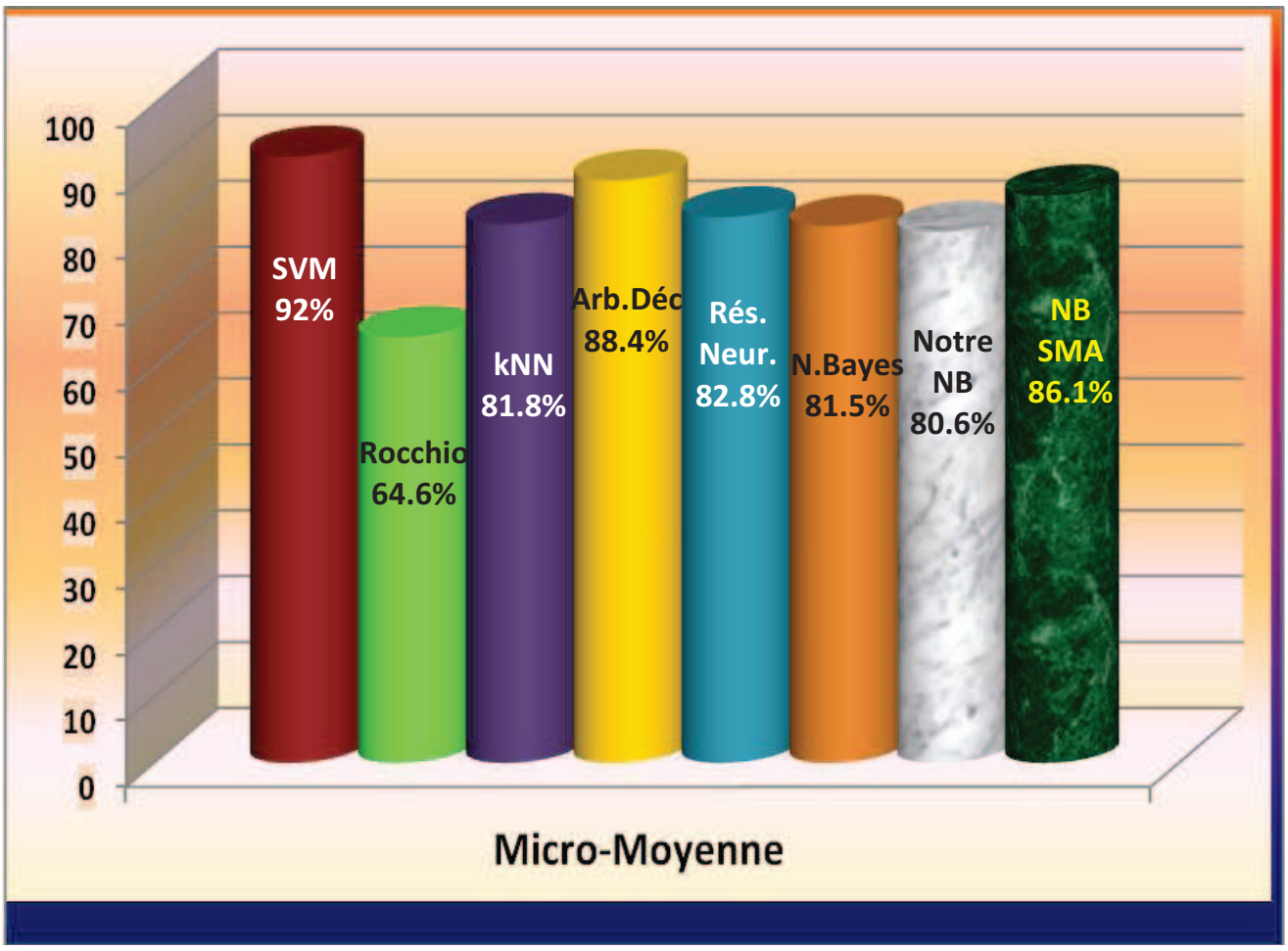


Figure 6.6 : Comparaison des différents résultats avec l'approche distribuée

6.6- Discussion

6.6.1- L'influence du N dans les résultats de l'approche

La représentation basée sur les N-grammes est dépendante d'un paramètre essentiel : la valeur de N, c.-à-d. le nombre de caractères que contiendra chaque N-grammes.
Qu'elle est la valeur de N qui donne les meilleurs résultats ?
Pour répondre à cette question, nous avons appliqué l'approche NB sur le corpus Reuters pour des valeurs de N comprise entre 2 et 7.
Le tableau 6.19 et la figure 6.2 Présentent les résultats obtenus en utilisant la mesure de performance F_1.
En analysant les résultats du tableau, on remarque que les performances s'améliorent en accroissant la valeur de N jusqu'à N=4 qui présente la valeur optimale. Ces performances commencent à rechuter à partir de N=7.

6.6.2- L'influence du nombre d'agents dans les résultats de classification

La variation du nombre d'agents était une expérience très intéressante. Après les différentes expérimentations sur 3, 9, 21, 33, 61, 99 et 181 agents, les résultats commencent à se stabiliser à partir de 61 agents. Ainsi une forte distribution sur un grand nombre d'agents n'est pas nécessaire, puisque que les résultats à partir de 61 agents commencent à rechuter légèrement. Ce qui nous amène à conclure qu'une Soixantaine d'agents est très satisfaisante pour un système de catégorisation automatique de textes.
Les différentes mesures F_1 obtenus en variant le nombre d'agents sont exposées dans le tableau 6.21 et la figure 6.4.

6.6.3- L'apport de la distribution de classification

Le développement d'un modèle fondé sur une architecture multi-agents a porté ses fruits puisque les résultats obtenus par notre modèle SMA sont nettement meilleurs du modèle mono-agent, mais sans autant abuser dans la distribution car une forte distribution va générer des vocabulaires assez pauvres en pouvoir informatif qui va certainement engendrer une dégradation considérable dans la qualité des résultats, d'une part (tableau 6.23 et figure 6.6).
D'autre part, l'amélioration en matière d'efficacité du classifieur à savoir le temps d'exécution du processus pour accomplir les différentes fonctions de prétraitement, apprentissage et test, est remarquable puisque les résultats figurés dans le tableau 6.22 et la figure 6.5 sont considérables.
Toutefois nous tenons à signaler que la distribution a influencé sur le temps d'exécution du prétraitement et de l'apprentissage très favorablement contrairement au test ou la distribution a augmenté légèrement le temps d'exécution puisque tous les agents vont tester les mêmes documents du corpus (2788 documents) chacun à son tour sachant en fin que l'opération de prétraitement est plus exigeante en temps d'exécution que le test.
Pour que nos comparaisons, des différents classifieurs Mono et Multi-Agents, en temps d'exécution soient effectives, nous avons comptabilisé le temps global d'exécution du processus (prétraitement + apprentissage + test).
Si on prend en considération le critère vitesse d'exécution, évidemment le modèle à 21 agents est le plus rapide mais avec un taux de classification pas aussi performant que le modèle à 33 ou 61 agents, sachant que ce dernier critère, à savoir la qualité de résultats de notre classifieur, est supposé la première priorité de nos études. Sans oublier à rappeler également, que les opérations de prétraitements, apprentissage et test se font en offline, donc le temps d'exécution ne sera d'une importance que si on veut refaire ces opérations ou traiter un autre

corpus. D'ailleurs, l'objectif majeur ciblé dans cette étude était la recherche du meilleur compromis qualité/efficacité.
Finalement après une lecture et une synthèse faite des résultats, le modèle SMA à 61 agents s'illustre comme le modèle optimal pour une classification automatique distribuée de Naïve Bayes de documents représentés en 4-Grammes.

6.7- Conclusion

Au cours de ce chapitre, nous avons présenté l'approche proposée avec toutes ces étapes, une approche qui tire son profit de l'utilisation des n-grammes comme méthode pour représenter les textes, et de l'algorithme Naïve Bayes comme algorithme d'apprentissage mais surtout l'apport considérable de notre approche c'est la distribution du processus de classification basée sur le paradigme agent.

Nous avons analysé les résultats de cette approche sur le corpus Reuters21578-Top10.

Les expérimentations réalisées ont mené aux constatations suivantes :

1- Le choix de la valeur N, influence sur les résultats de l'approche. En effet, les expérimentations ont montré que les quint-grams sont idéales pour le corpus. Cette valeur peut changer pour d'autres corpus.

2- En général, Les modèles Naïve Bayes classent bien dans le domaine textuel, et en particulier notre classifieur à base des N-Grammes a amené à des résultats très encourageants (80.6 %), mais insuffisants à l'égard des méthodes connues dans la littérature par la qualité de leurs résultats.

3- La nouvelle utilisation du classifieur Naïve Bayes, basée sur une architecture multi-agents introduite dans notre approche, a atteint l'objectif tracé puisque nous avons amélioré considérablement les performances du modèle basé sur un seul module logiciel (+ 5.5%), en s'approchant nettement des meilleurs résultats obtenus dans la littérature sur Reuters Top10 à savoir les SVM et Arbres de Décision.

4- Un autre atout dans la distribution de classification est en matière d'efficacité du classifieur qui s'améliore très nettement, bien sûr sans exagérer dans la distribution.

Conclusion générale

Table des matières

1- Conclusion générale

La classification de textes s'est avérée au cours des dernières années comme un domaine majeur de recherche pour les entreprises comme pour les particuliers.
Ce dynamisme est en partie dû à la demande importante des utilisateurs pour cette technologie. Elle devient de plus en plus indispensable dans de nombreuses situations où la quantité de documents textuels électroniques rend impossible tout traitement manuel.
La catégorisation de textes a essentiellement progressé ces dix dernières années grâce à l'introduction des techniques héritées de l'apprentissage automatique qui ont amélioré très significativement les taux de bonne classification.
Il reste néanmoins difficile de fournir des valeurs chiffrées sur les performances qu'un système de classification peut actuellement atteindre.
Les travaux de recherche dans le domaine se focalisent surtout sur deux aspects : l'efficacité et l'amélioration de performances.
Dans ces deux optiques nous avons entamé notre projet de recherche en proposant une approche dans le domaine de classification supervisée intitulée « **Classification automatique de textes – Approche orientée Agent** »,
Cette application propose un couplage original des méthodes issues du Traitement Automatique du Langage Naturel (TALN), des méthodes d'Apprentissage Automatique (AA) et de l'Intelligence Artificielle Distribuée (IAD). Grâce à cette utilisation associée, il est possible de disposer d'une base d'apprentissage de grande taille et très représentative du problème que l'on cherche à apprendre.

Nous avons décrit dans ce mémoire une nouvelle méthode de classification automatique de textes, dont voici les marques principales :

- La transformation ou le codage des documents est la préparation à « l'informatisation » de ces derniers, qui va se faire par la technique des N-Grammes connue pour son indépendance des différentes langues et son non exigence des traitements linguistiques préalables.
- Pour l'algorithme d'apprentissage et classification, le modèle d'indépendance conditionnelle (Naïve Bayes classifier) a été utilisé pour sa simplicité d'une part, et d'autre part, comme tous les modèles probabilistes, il s'appuie sur une base théorique précise.
- Le corpus Reuters dans sa version Reuters21578-Top10 avec ses deux mini-corpus d'apprentissage et de test, nous a été très utile pour les différentes expérimentations.
- Pour pouvoir comparer les résultats obtenus dans les différentes expérimentations, on a utilisé les mesures de performance Rappel, Précison et F-measure ($\boldsymbol{F}_1$).
- Mais sans doute la marque principale de notre travail, c'était l'intégration du paradigme agent dans un processus de classification et tenter d'améliorer les performances du classifieur mono-agent en distribuant la tâche de classification à plusieurs agents autonomes dans leurs apprentissage et classification, et collaboratifs dans la décision finale de catégorisation prise après un vote majoritaire.

Enfin de ce mémoire nous pouvons conclure que notre satisfaction est pleine puisque le double objectif de nos travaux fixé dés le début a été atteint, à savoir :

- D'une part, comment pouvoir palier l'inconvénient majeur de la méthode de classification Naïve Bayes basée sur l'hypothèse d'indépendance qui peut diminuer les performances du classifieur surtout 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.
- Et d'autre part comment exploiter l'efficacité et la simplicité de ce modèle pour qu'il soit compétitif face aux méthodes connues par des meilleures performances.

La distribution du processus de classification basée sur le paradigme agent était un choix très favorable puisque distribuer la tâche de classification en 1ére manche pour bénéficier de l'efficacité et la simplicité du modèle, et une collaboration et concertation entre les différents agents pour conforter une prise de décision finale de classification issue d'un votre majoritaire pour la 2éme manche, étaient les deux atouts principales de notre approche. La conception globale du système est fondée sur plusieurs modèles Naïve Bayes pour en constituer un seul.

Si on veut dresser un bilan, les résultats obtenus sont là et la différence peut s'apercevoir facilement. La rentabilité et les performances de notre nouveau modèle de classification est nettement supérieur à celles du modèle mono-agent et les résultats de notre classifieur peuvent être compétitifs avec les meilleures méthodes de classification dans le domaine.

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

2- Perspectives

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

- Une première perspective se présente en diminuant les dimensions des profils des documents et catégories soit avec sélection des n-grammes les plus importants soit avec élimination des termes très fréquents (mots outils) et mots très rares sachant que la sélection de descripteurs est un des principaux enjeux du système, puisque du choix des descripteurs et de la connaissance précise de la population va dépendre la mise au point du classifieur. Ces entrées non discriminantes doivent être supprimées pour deux raisons différentes : réduire le temps de calcul et diminuer le sur-apprentissage.
- La deuxième perspective s'annonce dans une variation dans la représentation des documents d'entrée en utilisant d'autres techniques de représentation de textes (sac de mots, lemmes, etc..). Chaque agent s'entraînera sur des textes codés par une méthode différemment des autres. Cette diversité dans l'apprentissage ne peut qu'enrichir le processus.
- Une autre alternative se présente en changeant les pondérations des termes et en s'appuyant sur des représentations plus riches en informations que la représentation fréquentielle basique, à savoir le codage TF-IDF. Le nombre d'occurrences du terme dans la catégorie est la façon la plus simple de calculer cette pondération mais elle n'est pas très satisfaisante au sens où elle ne prend pas en compte les autres catégories, or on désire pouvoir faire une

comparaison. Ainsi une autre expérience très intéressante à réaliser, en employant Naïve Bayes avec la pondération la plus largement utilisée à savoir TF-IDF basée sur une architecture multiagents.

- Une autre expérience consiste à expérimenter notre approche distribuée sur des mini-corpus d'apprentissage générés aléatoirement au lieu d'une distribution préparée.
- Une nouvelle perspective très passionnante dans l'utilisation de différentes méthodes de classification. Chaque agent ou groupe d'agents va s'entraîner de sa manière et se spécialiser selon son propre modèle de classification. On aura des agents SVM, agents arbres de décision, agents kNN, agents réseaux de neurones ou même des agents mixtes pour accomplir les mêmes tâches de classification. Ainsi on bénéficiera des performances de plusieurs classifieurs dans un système unique de classification.
- Parmi les futurs travaux est l'application de l'approche distribuée sur un corpus focalisé sur un domaine bien spécifique en utilisant une ontologie de domaine. Cela va nous permettre de construire des modèles spécialisés et aussi de minimiser les problèmes d'ambiguïté.
- Enfin, il faut étendre nos réflexions aux catégorisations des documents manuscrits et ne pas se limiter à la version électronique du corpus (Il existe des versions reconnues du corpus Reuters manuscrit) Sachant que l'information textuelle contenue dans ces documents n'est accessible que grâce à un processus de reconnaissance, qui en toute évidence, induit des erreurs dans le texte résultant.

Annexes

Annexe1 : La conférence TREC

La plus ancienne campagne annuelle d'évaluation, celle qui a le plus de participants et dont les données sont les plus volumineuses est la conférence **T**ext **RE**trieval **C**onference. Qui a été lancée en 1992 par Donna HARMAN et Charles WAYNE. Elle est organisée chaque année sous l'égide du NIST (National Institute of Standards and Technology), ITL (Technology Laboratory's) et DARPA (Defense Advanced Research Project Agency); et les groupes de recherche de l'(IAD) Information Access Division et de l'IARPA (Intelligence Advanced Research Projects Activity.
Cette campagne de grande envergure permet de comparer les systèmes concurrents dans différentes tâches de recherche documentaire et classification. La campagne francophone Amaryllis (Coret & all, 1997), de taille beaucoup plus modeste que TREC en constitue l'équivalent pour la recherche documentaire en français.
TREC est ouverte à toutes les équipes ayant préalablement participé à la compétition.
Elle offre un forum d'évaluation et de discussions pour la communauté scientifique qui se consacre au traitement automatique des textes en général, et au filtrage en particulier.
Un ensemble de tâches différentes est proposé aux différents participants qui soumettent des résultats à autant de tâches qu'ils le souhaitent. Certaines tâches font uniquement appel à des approches issues du traitement automatique du langage naturel et d'autres, comme la tâche de filtrage, nécessitent l'utilisation de méthodes à base de statistiques.

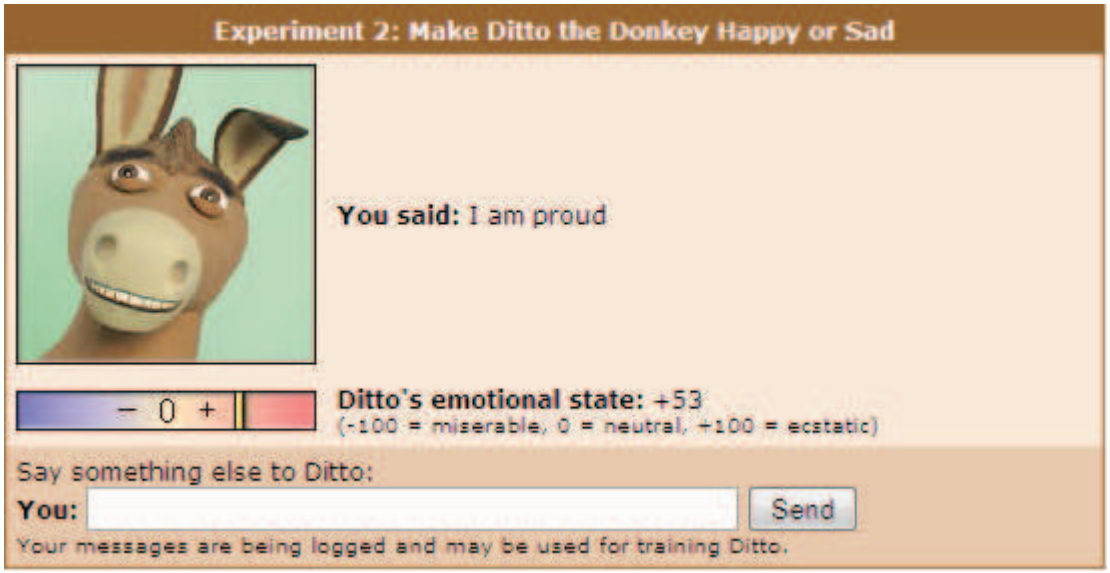
Une description générale de la huitième édition de cette conférence (TREC8) peut-être trouvée dans (Voorhees & Harman, 2000) ; cette huitième édition de la conférence a regroupé soixante six équipes différentes venant de seize pays différents. Parmi les multiples tâches proposées dans cette compétition, la recherche *ad hoc* (La recherche d'informations était la tâche principale jusqu'à l'édition TREC9). Cette tâche a notamment permis d'étiqueter une très grande quantité de textes pour un grand nombre de thèmes différents.
On peut trouver un aperçu historique sur les évolutions de la conférence avec un portrait de la 16éme édition de cette conférence (TREC-2007), dirigé par Ellen VOORHEES et David LEWIS, sur le site officiel de la conférence : http://trec.nist.gov ; cette seizième édition de la conférence a regroupé pas plus de 90 équipes différentes venant de plus de 20 pays.
La conférence est actuellement à sa 18 édition TREC-2009.
Cet ensemble constitue un corpus de référence, qui peut être utilisé par la communauté scientifique pour comparer des méthodes d'apprentissage et les faire progresser.
(Si on regarde les rapports TREC4, on remarque que la taille des corpus tests utilisés est passée de quelques Méga-octets durant les premières TREC à plusieurs Giga-octets pour TREC8.)

Annexe 2 : Algorithme MNB (Microsoft Naïve Bayes)

L'algorithme MNB (**M**icrosoft **N**aïve **B**ayes) est un algorithme de classification fourni par Microsoft **S**QL **S**erver 2005 **A**nalysis **S**ervices (SSAS) qui est conçu pour la modélisation prédictive. Cet algorithme calcule la probabilité conditionnelle entre les colonnes d'entrée et les colonnes prévisibles, et suppose que les colonnes sont indépendantes. Cet algorithme est informatiquement moins lourd que d'autres algorithmes Microsoft et est, par conséquent, utile pour générer rapidement des modèles d'exploration de données permettant de découvrir les relations entre les colonnes d'entrée et les colonnes prévisibles. Vous pouvez utiliser cet algorithme pour effectuer des explorations initiales de données et appliquer ensuite les résultats pour créer des modèles d'exploration de données supplémentaires avec d'autres algorithmes qui sont informatiquement plus lourds et plus précis.
L'algorithme MNB (Microsoft Naïve Bayes) calcule la probabilité de tous les états de chaque colonne d'entrée, en fonction de chaque état possible de la colonne prévisible. Vous pouvez utiliser la Visionneuse de l'algorithme MNB dans Business Intelligence Development Studio pour voir comment l'algorithme distribue les états.

Annexe 3 : Ditto-The donkey

Est une application en ligne basée sur Naïve Bayes Classifier, utilisée pour enseigner l'âne Ditto les bases de la langue anglaise. Lorsque Ditto reçoit un message en ligne, il l'évalue du point de vue gentillesse ou méchanceté, il répond alors émotionnellement sur une échelle de -100 à 100.
Ditto a été formé à l'aide d'exemples 5525, il connaît actuellement environ 1.000 mots, et son vocabulaire s'améliore de plus en plus que les gens interagissent avec lui. Toutefois, il ne peut pas vous répondre, car il n'a pas encore commencé à apprendre à parler. Cette expérience a été mise en place comme un exemple simple d'apprentissage supervisé. En utilisant un ensemble de d'exemples d'entrainement qui reflètent les sentiments gentil, méchant ou neutre, nous l'apprenons pour les distinguer. (http://www.convo.co.uk/x02/)



Bibliographie

(Antoniotti, 2002) M.Antoniotti « Receuil d'un corpus électronique à partir du Web »
(Armstrong & all, 1995) R.Armstrong, D.Freitag, T.Joachims, T.Mitchell « WebWatcher : a Learning apprentice for the World Wide Web »
(Amini, 2001) M.R.Amini « Apprentissage automatique et recherche d'information: application à l'extraction d'information de surface et au résumé de texte »
(Apté & all, 1994) C.Apté, F.J.Damerau, S.M.Weiss « Automated learning of decision rules for text categorization »
(Benveniste, 2000) B.Benveniste « Corpus de français parlé. Méthodologies et applications linguistiques »
(Beaune, 1999) P.Beaune « Apprentissage automatique dans les SMA »
(Bellot, 2002) P.Bellot, M.El-Béze « Classification locale non supervisée pour la recherche documentaire »
(Bigi & all, 2000) B.Bigi, R.De-Mori, M.El-Bèze, T.Spriet, « A fuzzy decision strategy for topic identification and dynamic selection of language models »
(Biskri & Delisle, 2001) I.Biskri, S.Delisle « Les n-grams de caractères pour l'extraction de connaissances dans des bases de données textuelles multilingues »
(Boissier, 2001) O.Boissier « Modèles et architectures d'agents, Principes et architectures des systèmes »
(Boissier & all, 1999) O.Boissier, Z.Guessoum, M.Ocello « Plates-Formes de développement de systèmes multi-agents »
(Bond & Gasser, 1988) A.Bond, L.Gasser « Readings in Distributed Artificial Intelligence »
(Boudaoud, 2002) K.Boudaoud « Un système multi-agents pour la détection d'intrusions »
(Bourron, 1992) T.Bouron « Structures de communication et d'organisation pour la coopération dans un univers multi-agents »
(Breiman & all, 1984) L.Breiman, J.Friedman, R.A.Olshen, C.J.Stone « Classification and regression trees »
(Briot & Demazeau, 2001) J.P.Briot, Y.Demazeau « Agent et systèmes multiagents »
(Briot & Demazeau, 2002) J.P.Briot, Y.Demazeau « Principes et architectures des systèmes multi-agents »
(Briot & Demazeau, 2002) .P.Briot, Y.Demazeau « Sciences Principes et architectures des systèmes multi-agents »
(Briot & all, 2006) J-P.Briot, Z.Guessoum, S.Aknine, A.L-Almeida, N.Faci « Experience and Prospects for Various Control Strategies for Self-Replicating Multi-Agent Systems »
(Brooks, 1991) R.Brooks « Intelligence without Reason »
(Brown & Chong, 1998) G.Brown, H.A.Chong « The Guru System in TREC-6 »
(Brown & all, 1992) P.Brown, V.J.D.Pietra, P.V.de Souza, J.C.Lai, R.L.Mercer
« Class-bassed n-gram models of natural language »
(Buckley & all, 1994) C.Buckley, G.Salton, J.Allann, A.Singhal « Automatic query expansion using SMART »
(Carpinto & all, 2001) C.Carpinto, R.De-Mori, G.Romano, B.Bigi « An information theoritic approch to automatic query expansion »

(Caropreso & all, 2001) M.F.Caropreso, S.Matwin, F.Sebastiani « A learner-independent evaluation of the usefulness of statistical phrases for automated text categorization »
(Castelfranchi, 1995) C.Castelfranchi « Commitments: From Individual Intentions to Groups and Organizations»
(Chaib-draa, 2010) B.Chaib-draa « Agents Intelligents »
(Chaib-draa & all, 1992) B.Chaib-Draa, M.Moulin, R.Mandiau, P.Millot « Trends in Distributed Artificial Intelligence »
(Chaib-Draa & all, 2001) B.Chaib-draa, I.Jarras, B.Moulin « Systèmes multiagents : Principes généraux et applications »
(Chaib-Draa & Gageut, 2002) B.Chaib-draa, L.Gageut « Aspects formels des Systèmes Multi-Agents »
(Chandra, 1998) B.Chandrasekaran, J.R.Josephson, V.R.Benjamins « The Ontology of Tasks and Methods »
(Cheikhrouhou & all, 1998) M.Cheikhrouhou, P.Conti, R.T.Oliveira, J.Labetoulle «Intelligent Agents in Network Management, a state of the art »
(Chethan & all, 2007) J.S.Chethan, G.R.Geeta, J.Pereira, K.Jakkula «Bayesian Classification»
(Clech, 2004) J.Clech « Contribution méthodologique à la fouille de données complexes »
(Clech & Zighed, 2004) J.Clech, D.A.Zighed « Une technique de réétiquetage dans un contexte de catégorisation de textes »
(Clergue, 1997) G.Clergue « L'apprentissage de la complexité »
(Cohen & Levesque, 1995) P.R.Cohen, H.J.Levesque « Communicative Actions for Artificial Intelligence »
(Corbara & all, 1993) B.Corbara, A.Drogoul, D.Fresneau, S.Lalande « Simulating the Sociogenesis Process in Ant Colonies with MANTA »
(Dagan, 1999) I.Dagan, L.Lee, F.Pereira « Similarity based models of word co-occurrence probabilities »
(Dagan & all, 2005) I.Dagan, O.Glickman, B.Magnini « The Pascal recognising textual entailment challenge »
(De Loupy, 2000) C.D.Loupy « Évaluation de l'Apport de Connaissances Linguistiques en Désambiguïsation Sémantique et Recherche Documentaire »
(Demazeau & Müller, 1991) Y.Demazeau, J.P.Müller « Decentralized AI »
(Denoyer, 2004) L.Denoyer « Apprentissage et inférence statistique dans les bases de documents structurés : Application aux corpus de documents textuels »
(Deerwester & all, 1990)S.Deerwester,S.Dumais,T.Landauer,G.Furnas, R.Harshman « Indexing by latent semantic analysis »
(Deneubourg & all, 1991) J-L.Deneubourg, S.Goss, A.Sendova-Franks, C.Detrain, L.Chretien « The Dynamics of Collective Sorting Robot-like Ants and Ant-like Robots »
(Drogoul, 1993) A.Drogoul « De la simulation multi-agent à la résolution collective de problèmes. Une étude de l'émergence de structures d'organisation dans les systèmes multi-agents »
(Dumais, 1991) S.Dumais « Improving the retrieval of information fromexternal sources »
(Dumais & all, 1998) S.Dumais, J.Platt, D.Heckerman, M.Sahami « Inductive learning algorithms and representations for text categorization »
(Dumais & Chen, 2000) S.T.Dumais, H.Chen « Hierarchical classification of Web content »
(Dutech & all, 2003) A.Dutech, B.Olivier, F.Charpillet « Apprentissage par renforcement pour la conception de systèmes multi-agents réactifs »
(Erceau & Ferber, 1991) J.Erceau, J.Ferber « L'intelligence Artificielle Distribuée »

(Fayech, 2003) B. Fayech « Régulation des réseaux de transport multimodal : systèmes multi-agent et algorithmes évolutionnistes »
(Fayet-Scribe, 1997) S.Fayet-Scribe « Chronologie des supports, des dispositifs et des outils de repérage de l'information »
(Ferber 1995) J.Ferber « Les Systèmes Multi-Agents, Vers une intelligence collective »
(Généreux, 2010) M.Généreux « Classification de textes en comparant les fréquences lexicales »
(Ferber, 1999) J.Ferber « Multi-Agent Systems. An Introduction to Distributed Artificial Intelligence »
(Ferber & Drogoul, 1992) J.Ferber, A.Drogoul « Using Reactive Multi-Agent Systems in Simulation and Problem Solving »
(Ferber & all, 2009) J.Ferber, T.Stratulat, J.Tranier « Towards an Integral Approach of Organizations: the MASQ approach »
(Ferber & all, 2010) J.Ferber, R.Dinu, T.Stratulat « A formal approach to MASQ »
(Finin & Fritzson, 1994) T.Finin, R.Fritzson « KQML - A Language and Protocol for Knowledge and Information Exchange »
(Fipa, 2000) « FIPA ACL Message Structure Specification »
(Gates, 1999) B.Gates « The road ahead »
(Gilli, 1988) Y.Gilli « Texte et fréquence »
(Gotab, 2009) P.Gotab « Apprentissage automatique et Co-training »
(Guessoum, 1996) Z.Guessoum « Un environnement opérationnel de conception des SMA »
(Guessoum & Occello, 2001) Z.Guessoum & O.Occello«Environnements de développement»
(Guessoum & all, 2006) Z.Guessoum, N.Faci, J-P.Briot « Adaptive Replication of Large-Scale Multi-Agent Systems - Towards a Fault-Tolerant Multi-Agent Platform »
(Gurvitch, 1963) G.Gurvitch « La vocation actuelle de la sociologie »
(Gutknecht & Ferber 1999) O.Gutjnecht, J.Ferber « Vers une Méthodologie Organisationnelle de Conception de Systèmes Multi-Agent »
(Haddad, 2002) M.H.Haddad « Extraction et Impact des connaissances sur les performances des Systèmes de Recherche d'Information »
(Hayes & Weinstein, 1990) P.Hayes, S.P.Weinstein « Construe/Tis : A system for content-based indexing of a database of news stories »
(Hertzmann, 2004) A.Hertzmann « Introduction to Bayesian Learning »
(Hilali, 2009) H.Hilali « Application de la classification textuelle pour l'extraction des règles d'association maximales »
(Jaillet & all, 2003) S.Jaillet, M.Teisseire, J.Chauche, V.Prince. « Classification automatique de documents : Le coefficient des deux écarts »
(Jalam, 2003) R. Jalam « Apprentissage automatique et catégorisation de textes multilingues »
(Jalam & Teytaud, 2001) R.Jalam, O.Teytaud « Identification de la langue et catégorisation de textes basées sur les n-grammes »
(Jégou & all, 2010) H.Jégou, M.Douze, C.Schmid « Représentation compacte des sacs de mots pour l'indexation d'images »
(Jelinek & Mercer, 1980) F.Jelinek, R.L.Mercer « Interpolated Estimation of Markov Source Parameters from Sparse Data »
(Jennings & all, 1998) N.Jennings, K.Sycara, M.Wooldridge « A Roadmap of Agent Research and Development »
(Joachims, 1998) T.Joachims « Text categorization with support vector machines: learning with many relevant features »

(Joachims, 1999) T.Joachims « Transductive inference for text classification using support vector machines »
(Jones & Furnas, 1987) W.Jones, G.Furnas « Pictures of relevance : A geometric analysis of similarity measures »
(Kinny & Georgeff, 1997) D.Kinny, M.Georgeff « Modelling and Design of Multi – Agents Systemd »
(Koller & Sahami, 1997) D.Koller, M.Sahami « Hierarchically Classifying Documents using very Few Words »
(Kohonen & all, 2000) T.Kohonen, S.Kaski, K.Lagus, J.Salojärvi, J.Honkela, V.Paatero, A.Saarela « Self organization of a massive document collection »
(Labidi & Lejouad 1993) S.Labidi, W.Lejouad « De l'Intelligence Artificielle Distribuée aux Systèmes Multi-Agents »
(Laichour, 2002) H.Laichour « Modélisation Multi-agent et aide à la décision : Application à la régulation des correspondances dans les réseaux de transport urbain »
(Latour, 1989) B.Latour « La Science en action. La Découverte »
(Latour, 2006) B.Latour « Efficacité ou instauration ? »
(Latour & Lemonnier , 1994) B.Latour, P.Lemonnier « De la préhistoire aux missiles balistiques - l'intelligence sociale des techniques »
(Lang, 1995) K. Lang « NewsWeeder : Learning to Filter Netnews »
(Lavalley & all, 2009) R. Lavalley, P. Bellot, M. El-Bèze « Interactions entre le calcul de collocations et la catégorisation automatique de textes »
(Lefévre, 2000) P. Lefévre « La recherche d'information - du texte intégral au thésaurus »
(Lelu & Hallab ,2000) A.Lelu, M.Hallab « Consultation "floue" de grandes listes de formes lexicales simples et composées : un outil préparatoire pour l'analyse de grands corpus textuels »
(Leray, 2006) P.Leray « Quelques Types de Réseaux de Neurones - La rétropropagation »
(Lestel, 1986) D.Lestel « Contribution à l'étude du raisonnement expérimental dans un domaine sémantiquement riche »
(Lestel & all, 1994) D.Lestel, B.Grison, A.Drogoul « Les agents réactifs et le vivant dans une perspective d'évolution coopérative »
(Lewis, 1992) D.D.Lewis « An evaluation of phrasal and clustered representations on a text categorization task »
(Lewis, 2004) D.D.Lewis « Bayesian Text Classification for Spam Filtering »
(Loupy, 2000) C.Loupy « Évaluation de l'Apport de Connaissances Linguistiques en Désambiguïsation Sémantique et Recherche Documentaire »
(Loupy & El-Bèze, 2000) C.de Loupy, M.El-Bèze « Using few cues can compensate the small amount of resources available for WSD »
(Lumer & Faieta, 1994) E.D. Lumer & B.Faieta « Diversity and Adaptation in Populations of Clustering Ants »
(Manning & all, 2008) C.D.Manning, P.Raghavan & H.Schütze « Introduction to Information Retrieval » Cambridge University Press 2008.
(Mari & Napoli, 1996) J.F.Mari et A.Napoli « Aspects de la classification »
(Magendaz, 1995) T.Magendaz « On the Impacts of Intelligent Agents Concepts on Future Telecommunication Environments »
(McCallum & all, 1998) A.McCallum, R.Rosenfeld, T.Mitchell, A.Y.Nigam « Improving Text Classification by Shrinkage in a Hierarchy of Classes »
(Miller & all, 1999) E.Miller, D.Shen, J.Liu, C.Nicholas « Performance and Scalability of a Large-Scale N-gram Based Information Retrieval System »

(Monmar & all, 1999) « AntClass : Discovery of clusters in numeric data by an hybridization of an ant colony with Kmeans algorithm »
(Moulinier, 1996) I.Moulinier « Une approche de la catégorisation de textes par l'apprentissage symbolique »
(Moutarde, 2008) F.Moutarde « Brève introduction aux arbres de décision »
(Müller, 1996) J.P.Müller. « The Design of Intelligent Agents - A layered Approach »
(Nakache, 2007) D.Nakache, « Extraction automatique des diagnostics à partir des comptes rendus médicaux textuels »
(Nakache & Metais, 2005) D.Nakache, E.Metais « Evaluation : Nouvelle Approche avec juges »
(Nwana, 1996) H.S.Nwana « Software Agents: An Overview »
(Nwana & Ndumu, 2000) H.S.Nwana, D.T.Ndumu « A Perspective on Software Agents Research »
(Oliveira, 1998) R.T.Oliveira « Gestion des Réseaux avec Connaissance des Besoins: Utilisation des Agents Logiciel »
(M.M.Ould Sidi & all, 2005) M.M.Ould Sidi, S.Hammadi, S.Hayat, P.Borne «Urban transport disrupted networks regulation strategies making and evaluation: A new approach»
(Pessiot & all, 2004) J.F.Pessiot, M.Caillet, M.R.Amini, P.Gallinari **«** Apprentissage non-supervisé pour la segmentation automatique de textes »
(Pesty & all, 2001) S.Pesty, C.Webber, N. Balacheff: « Baghera : une architecture multi-agents pour l'apprentissage humain »
(Pisetta & all, 2007) V.Pisetta, G.Ritschard, D.A.Zighed « Choix des conclusions et validation des règles issues d'arbres de classification »
(Porter, 1980) M.F.Porter « An algorithm for suffix stripping »
(Pothin & Richard, 2007) J.B.Pothin, C.Richard « Apprentissage de métrique appliqué à la classification de textes par méthodes à noyaux »
(Quinlan, 1986) J.R.Quinlan « Induction of decision trees »
(Quinlan, 1993) J.R.Quinlan « Programs for Machine Learning »
(Ralaivola, 2006) L.Ralaivola « Modèles de représentation, sélection d'attributs, classification, catégorisation »
(Raza, 2009) M.Raza « Command agents with human-like decision making strategies »
(Robertson & Sparck-Jones, 1976) S.Robertson, K.Sparck-Jones « Relevance weighting of search terms »
(Ritschard & all, 2009) G.Ritschard, S.Marcellin, D.A.Zighed « Arbre de décision pour données déséquilibrées : sur la complémentarité de l'intensité d'implication et de l'entropie Décentrée »
(Rocchio, 1971) J.Rocchio « Relevance feedback in information retrieval »
(Rocher, 1968) G.Rocher « Introduction à la sociologie générale »
(Russell & Norvig, 1995) S.J.Russell, P.Norvig «Artificial Intelligence. A Modern Approach»
(Sahami, 1999) M.Sahami « Using Machine Learning to Improve Information Access »
(Salton, 1968) G.Salton « Automatic information organization and retrieval »
(Salton & McGill, 1983) G.Salton & M.McGill « Introduction to Modern Information Retrieval»
(Salton & Buckley, 1988) G.Salton, C.Buckley « Term-weighting approaches in automatic text retrieval »
(Sansonnet, 2002) J-P.Sansonnet « Concepts d'agents : Introduction aux concepts et aux architectures des systèmes multi-agents »
(Saporta, 1990) G.Saporta « Probabilités, Analyse des données et Statistique »
(Shannon, 1948) C.Shannon « The Mathematical Theory of Communication »

(Schapire & all, 1998) R.E. Schapire, Y.Singer, A.Singhal « Boosting and Rocchio applied to text filtering »
(Sebastiani, 1999) F.Sebastiani « A tutorial on automated text categorisation »
(Sebastiani, 2002) F.Sebastiani « Machine learning in automated text categorization »
(Schmid, 1994) H.Schmid « Probabilistic part-of-speech tagging using decision trees »
(Scott & Matwin, 1999) S.Scott, S.Matwin « Feature Engineering for Text Classification »
(Sen & Weiss, 1999) S.Sen, G.Weiss « Learning in Multiagent Systems »
(Shah & all, 2002) C. Shah, B. Chowdhary, P. Bhattacharyya « Constructing better document vectors universal networking language (unl) »
(Skarmeas, 1998) N.Skarmeas « Agents as Objects with Knowledge Base State »
(Shoham, 1993) Y.Shoham « Agent Oriented Programming. Artificial Intelligence »
(Smith, 1980) R.G.Smith « The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver »
(Stricker, 2000) M.Stricker « Réseaux de neurones pour le traitement automatique du langage: conception et réalisation de filtres d'information »
(Trinh, 2008) A.P.Trinh « La classification des textes d'opinion par les Séparateurs à Vaste Marge (SVM) avec sorties probabilistes »
(Turenne, 2000) N.Turenne «Apprentissage statistique pour l'extraction de concepts à partir de textes. Application au filtrage d'informations textuelles »
(Uchida & Zhu, 1999) T.D.S. H.Uchida, M.Zhu « The UNL, A Gift for a Millennium »
(Uschold & King, 1995) M.Uschold et M.King «Towards a Methodology for Building Ontologies »
(Usunier & all, 2005) N.Usunier, M.R.Amini, P. Gallinari « Résumé automatique de texte avec un algorithme d'ordonnancement »
(Van Rijsbergen, 1979) C.J.Van Rijsbergen, « Information Retrieval »
(Vapnik, 1995) V.Vapnik « The Nature of Statistical Learning »
(Vinot & all, 2003) R.Vinot, N.Grabar, M.Valette « Application d'algorithmes de classification automatique pour la détection des contenus racistes sur l'Internet »
(Vinot & Yvon, 2002) R.Vinot, F.Yvon « Quand simplicité rime avec efficacité : analyse d'un catégoriseur de textes »
(Wei, 2009) Z.WEI « Avancée en classification multi-labels de textes en langue chinoise»
(Wiener, 1995) E.D.Wiener « A neural network approach to topic spotting in text »
(Wies, 1995) R.Wies « Policies in Integrated Network and Systems Management »
(Wilkinson & all, 1996) R.Wilkinson, J.Zobel, R.Sacks-Davis « Similarity measures for short queries »
(Wooldridge, 1999) M.Wooldridge « Intelligent Agents »
(Wooldridge, 2002) M.Wooldridge «Multi-agent Systems »
(Wooldridge & Jennings, 1994) M.Wooldridge, N.Jennings « Towards a Theory of Cooperative Problem Solving »
(Yang, 1999) Y.Yang « An evaluation of statistical approch to text categorization »
(Yang, 2001) Y.Yang « Problem-based Learning on the World Wide Web in an Undergraduate Kinesiology Class: an Integrative Approach to Education »
(Yang & Liu, 1999) Y.Yang, X.Liu « A re-examination of text categorization methods »
(Yvon, 2006) F.Yvon « Des apprentis pour le traitement automatique des langues »
(Zighed & Rakotomalala, 2000) D.A.Zighed, R.Rakotomalala « Graphes d'induction. Apprentissage et Data Mining ».
(Zipf, 1949) G.K.Zipf « Human Behavior and the Principle of Least Effort »