N° d'ordre : 01/2004-E/IN

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

Université des Sciences et de la Technologie Houari Boumediène (U.S.T.H.B.) Alger

Faculté d'Electronique et d'Informatique
Département Informatique

THÈSE

Présentée pour l'obtention du diplôme de

DOCTORAT D'ÉTAT en **INFORMATIQUE**
Spécialité : **Informatique**

par

Omar NOUALI

Thème

Filtrage d'Information Textuelle sur les réseaux une Approche Hybride

soutenue le 20 Novembre 2004, devant la commission d'examen :

Mr. N. BADACHE,	Professeur,	USTHB.	Président
Mr. Ph. BLACHE,	Dteur de Recherche,	CNRS/FRANCE	Dteur de thèse
Mme. A. AISSANI,	Professeur,	USTHB.	Co-Dteur de thèse
Mr. M. AHMED-NACER,	Professeur,	USTHB.	Examinateur
Mr. M. BOUFAIDA,	Professeur,	Univ. CONSTANTINE	Examinateur
Mr. M. BOUALEM ,	C. S,	FRANCE/TELECOM	Invité

Remerciements

De près ou de loin, de nombreuses personnes m'ont aidé et soutenu pour la realisation de cette thèse. Qu'ils trouvent ici l'expression de ma sincère reconnaissance et l'assurance de mes remerciements sincères et amicaux.

Je voudrais exprimer ma grattitude au Professeur Nadjib Badache de l'université des Sciences et de la Technologies Houari Boumédiène d'Alger, qui m'a encouragé à aller toujours plus loin et est toujours prêt à me donner les bons conseils, et de me faire l'honneur aujourd'hui de présider le jury de cette thèse.

Je suis très honoré par l'intérêt porté à ce travail par Monsieur Mohamed Ahmed-Nacer, professeur à l'université des Sciences et de la Technologies Houari Boumédiène d'Alger, Monsieur Mahmoud Boufaida, Professeur à l'université de Constantine. Je les remercie pour avoir bien voulu évaluer ce travail et participer au jury de cette thèse.

Je suis très redevable à mon directeur de thèse Monsieur Philippe Blache, Directeur de recherche au CNRS et Directeur de Laboratoire de Recherche Parole & Langage. Je le remercie pour toute la confiance qu'il m'a accordée et pour m'avoir souvent accueilli dans son équipe et particulièrement pour son encadrement et son soutien pendant toutes ces années lors de la préparation de cette thèse. J'ai eu un grand plaisir à travailler avec lui. Sa bonne humeur et sa gentillesse ont également contribué au bon déroulement de mes recherches au sein de son équipe.

Je remercie Aicha Aissani, professeur à l'université des Sciences et de la Technologies Houari Boumédiène d'Alger, d'avoir accepté la charge de co-rapporteur.

J'exprime ma reconnaissance à Malek Boualem, Chercheur Sénior à France Télécom, R&D-DMI/GRI pour l'honneur qu'il me fait en participant au jury en tant qu'invité.

Plus particulièrement, j'exprime ma profonde gratitude à Monsieur Abdelkader Khelladi, professeur à l'université des Sciences et de la Technologies Houari Boumédiène d'Alger et Directeur du CERIST, et à Monsieur Moussa Benhamadi, ex directeur du CERIST, pour les moyens qu'ils ont mis à ma disposition pour la réalisation de cette thèse.

Je souhaite également adresser particulièrement mes sincères remerciements à Monsieur Halim Khelalfa, professeur à l'université de Wollogong, Dubai, UAE, et ex chef de laboratoire des logiciels de base, dont les encouragements et l'amitié m'ont soutenu dans les moments difficiles.

Je souhaite exprimer ma sympathie à tout le personnel du CERIST, à tous les membres du Laboratoire des Logiciels de Base du CERIST, en particulier madame Hassina Aliane, responsable du laboratoire, et à tous les membres du laboratoire Parole et Langage d'Aix en Provence, France.

Enfin merci à ma famille qui m'a épaulé toutes ces années de thèse, tout particulièrement à ma chère femme Nadia pour son aide continue, ses conseils, ses encouragements et sa patience, à mes chères filles Maya et Sarah d'avoir supporté mon absence, et à mes beaux parents d'avoir pris en charge mes filles, surtout pendant mon absence.

Je ne pourrais terminer cette série de remerciements sans avoir une pensée pour ma chère mère et mes tantes qui ne sont plus de ce monde.

Résumé

Le sujet de la thèse se situe dans la problématique globale du traitement de l'information dynamique et de l'analyse de contenu. Elle est motivée par le souci de faciliter à l'utilisateur, submergé d'informations diverses, l'accès à l'information pertinente. Plus précisément, l'objet des travaux de recherche présentés, concerne l'automatisation du processus de filtrage de l'information pertinente et personnalisée. Il s'agit d'offrir une assistance à l'utilisateur, visant à optimiser le temps consacré à la recherche et à la consultation de l'information, en prenant en compte l'importance relative de l'information et les besoins en ressources pour son traitement.

Les premières investigations dans ce travail ont été d'explorer le potentiel des techniques de plusieurs domaines de recherche liés au traitement de l'information textuelle. L'un de ces domaines concerne l'apprentissage automatique, qui constitue une phase incontournable dans la conception d'un système de filtrage automatique de l'information. Nous proposons une solution évolutive qui offre au système de filtrage la possibilité d'apprendre à partir de données ciblées (profils des utilisateurs), d'exploiter ces connaissances apprises (pour filtrer l'information) et de s'adapter à la nature de l'application (textes traités) dans le temps.

Un autre domaine concerne le traitement automatique du langage naturel. Il intervient par la nécessité d'utiliser des ressources et des traitements linguistiques dans le processus de filtrage. Sur ce volet, notre objectif est de (dé)montrer que l'intervention de connaissances et de traitements linguistiques peut considérablement améliorer les performances d'un système de filtrage de l'information. En effet, le couplage entre méthodes statistiques et symboliques (quantitatives et linguistiques) donne plus d'efficacité au filtrage. Ce constat est d'ailleurs souvent évoqué pour un grand nombre d'applications liées au traitement de l'information textuelle. Ainsi, l'apport du domaine linguistique dans notre travail se concrétise sous plusieurs aspects. D'une part, nous proposons un ensemble de connaissances linguistiques sous forme de modèles réduits (issues de modèles linguistiques de textes). Il s'agit d'un ensemble d'indicateurs sur le texte, portant sur la structure et sur le contenu. Un texte est soumis à un processus d'analyse automatique qui permet de lui associer un ensemble de termes et de propriétés linguistiques, qui servent à le caractériser et permettent de le situer par rapport à d'autres textes. Ces connaissances, classées sous plusieurs niveaux (matériel, énonciatif, structurel et syntaxique), sont indépendantes du domaine d'application. Par ailleurs, la fiabilité des traitements repose sur l'opération d'apprentissage. Dans le cadre de ce travail, l'objectif n'est pas d'effectuer une analyse complète et profonde du contenu des textes. Il s'agit d'effectuer une analyse dite partielle, s'échelonnant sur plusieurs niveaux, pour identifier certaines propriétés linguistiques. Celles-ci permettent de distinguer les différents types de textes et de classer ensuite les nouveaux textes. D'autre part, pour l'aspect sémantique, nous proposons d'utiliser un ensemble de connaissances linguistiques (réseau lexical et cooccurrence de critères) permettant d'améliorer la représentation du texte. Des termes complémentaires sont ainsi impliqués dans le processus de décision, même s'ils n'apparaissent pas explicitement dans le texte (par exemple, la substitution de certains termes par d'autres termes proches sémantiquement).

Pour la validation de notre approche, un outil d'aide à la génération d'interfaces de filtrage (baptisé GIFI) a été développé. Il est destiné à faciliter la tâche des utilisateurs développeurs dans l'élaboration de systèmes de filtrage de l'information. Il permet d'assister l'utilisateur dans le processus d'acquisition de l'application (corpus de textes) et de génération de ressources (vocabulaire lexical, propriétés linguistiques, modèle de filtrage). Il repose sur une

conception modulaire, lui permettant de s'adapter à des extensions ou à des mises à jour éventuelles. Cet outil est basé sur une architecture ouverte permettant l'ajout de composants et offrant à l'utilisateur la possibilité de choisir, à chaque étape du processus de génération, les outils à utiliser. Ainsi, cette "boite à outils" matérialise l'implémentation d'une approche hybride de filtrage de l'information. Elle repose sur le principe d'une analyse partielle utilisant un ensemble de connaissances, où le repérage de propriétés linguistiques permet, d'une part, d'améliorer la représentation des textes, et d'autre part un filtrage de meilleure qualité.

Pour l'évaluation de notre approche et afin de statuer sur sa faisabilité et sur son apport en terme d'efficacité, nous l'avons expérimentée sur une application pratique de filtrage de l'information : *filtrage du courrier électronique*. La période actuelle voit une prolifération colossale et démesurée des courriers électroniques non sollicités et indésirables (appelés *Spams)*. Paradoxalement, au moment où le courrier électronique s'impose comme le moyen de communication incontournable pour les entreprises, les institutions académiques et même pour les particuliers, le problème des courriers indésirables atteint des proportions intolérables. Ce problème devient très sérieux pour les utilisateurs du courrier électroniques et engendre des pertes considérables, en temps et en argent, pour les entreprises. A travers les différentes expériences réalisées, nous avons montré l'applicabilité et l'adaptabilité d'une approche hybride au processus de filtrage de l'information. En effet, les résultats obtenus sur le corpus de messages utilisé, nous ont permis de valider l'intérêt des connaissances linguistiques et de l'apprentissage automatique pour l'amélioration des performances d'un système de filtrage de l'information.

Mots clés :
Filtrage de l'information, apprentissage automatique, propriétés linguistiques, modèles linguistiques réduits, spam.

Table des Matières

Chapitre III
Apprentissage et Classification Automatique de textes…… 95

Partie 2 : Implémentation & Evaluation

Chapitre IV
Architecture de filtrage et Générateur d'interfaces…… 133

Chapitre V
Filtrage, Typologie et Caractéristiques des messages électroniques 171

Introduction

Les nouvelles technologies de l'information et de la communication (NTIC), matérialisées notamment à travers Internet, ont considérablement facilité la communication et l'accès à l'information. Cependant, la quantité de données produites et échangées ne cessant de croître, les utilisateurs se retrouvent souvent submergés d'informations qu'ils n'arrivent plus à gérer. L'augmentation permanente de la quantité d'information disponible au format électronique, induit de nouveaux besoins d'accès à l'information de la part des utilisateurs. Afin d'économiser un temps précieux pour la recherche d'information utile, le recours à de nouveaux outils s'avère inévitable pour les utilisateurs. Ce besoin a engendré des investigations et des recherches pour la mise en place de nouveaux médiateurs, entre les sources d'information et les utilisateurs, parmi lesquels les systèmes de filtrage de l'information. Par ce rôle de médiateurs, ces systèmes doivent posséder les méthodes et les connaissances nécessaires pour traiter, évaluer, filtrer et extraire l'information pertinente pour l'utilisateur.

Le sujet des travaux de recherche présentés se situe dans la problématique globale du traitement de l'information dynamique et de l'analyse de contenu. Elle est motivée par le souci de faciliter à l'utilisateur, submergé d'informations diverses, l'accès à l'information pertinente. Plus précisément, l'objet des travaux de recherche présentés, concerne l'automatisation du processus de filtrage de l'information pertinente et personnalisée. Il s'agit d'offrir une assistance à l'utilisateur, visant à optimiser le temps consacré à la recherche et à la consultation de l'information, en prenant en compte l'importance relative de l'information et les besoins en ressources pour son traitement. Outre l'analyse du contenu des documents (ou des flux d'information), il est nécessaire de prendre en compte également des événements (ou méta-données) tels que la date de réception de l'information, son expéditeur, etc. D'autres facteurs non négligeables sont également à prendre en compte comme le comportement de l'utilisateur vis-à-vis de l'information (temps de lecture, destruction, routage vers un autre utilisateur, mise en priorité, etc.). Il est à noter que dans le présent travail de thèse, le comportement de l'utilisateur n'est pas pris en compte, sauf dans le cas de feedback (donner un avis sur le comportement du système, déplacer un document d'un répertoire à un autre, etc.).

Du fait que le domaine du filtrage de l'information soit étroitement lié au domaine de la recherche d'information, les principales techniques employées actuellement dans le domaine du filtrage sont basées, d'une façon directe ou indirecte, sur les techniques et méthodes traditionnelles de recherche d'information (modèle booléen, modèle vectoriel, etc.).

La plupart des systèmes de filtrage actuels enregistrent des lacunes ou des faiblesses sur l'efficacité du filtrage de l'information. En effet, certains systèmes sont basés sur un

traitement partiel du contenu (par exemple, dans le cas des courriers électroniques, le filtrage opère seulement sur la partie structurée : adresse émettrice, objet, etc.). D'autres systèmes sont basés sur un balayage superficiel du contenu permettant aux utilisateurs d'écrire manuellement des règles logiques de filtrage à base de mots clés : ils se basent généralement sur une propriété lexicale, la présence ou l'absence de mots-clés que l'utilisateur doit indiquer au logiciel. Parmi les inconvénients de ce type de systèmes, on note le manque de précision du à la non prise en compte du niveau sémantique et la nécessité de procéder à des mises à jour fréquentes des règles compte tenu de la nature dynamique de l'information qui varie au cours du temps.

Néanmoins, le domaine du filtrage de l'information reste très ouvert vers diverses autres tendances. Ainsi, certaines approches utilisent des techniques traditionnelles en essayant de les améliorer en captant plus d'information d'ordre sémantique. De ce fait et vu la diversification des tendances classiques actuelles, il n'y a pas encore de conclusion concrète. L'indexation des phrases ou des expressions au lieu des mots clés apporte des améliorations certaines à l'efficacité du filtrage, mais requiert un pré-traitement élaboré (analyse partielle ou totale et analyse syntaxique de la phrase). L'état actuel des connaissances ne permet pas de concevoir un système capable de comprendre toute phrase écrite en langage naturel et de fonctionner dans tous les contextes d'utilisation. L'espoir de mettre en place des analyseurs robustes capables de traiter des textes libres en profondeur, a conduit de nombreux chercheurs à mettre en œuvre des analyseurs variés ces dernières années.

Ces analyseurs varient en terme de stratégie (déterminisme vs non-déterminisme, analyse partielle vs analyse complète) et en terme de bases théoriques (analyseurs statistiques vs analyseurs symboliques, etc.). Bien entendu, chacun des analyseurs est plus ou moins dédié à certaines tâches spécifiques et aucun analyseur ne peut, aujourd'hui, prétendre effectuer une analyse complète de toutes les phrases dans un corpus de textes libres.

Les méthodes statistiques, bien que prometteuses, ne suffisent pas, à elles seules, pour traiter tous les aspects du traitement automatique de la langue. Les grammaires symboliques sont également nécessaires afin d'obtenir une représentation précise et fiable de la sémantique; ce qui est crucial pour nombreuses applications du traitement automatique des langues.

Le couplage entre méthodes statistiques et symboliques (quantitative/linguistique) est, selon beaucoup de chercheurs, garant pour une analyse plus efficace des textes et, par conséquent, pour un filtrage plus précis. En effet, comme les documents transitant sur Internet (ex : courriers électroniques) sont souvent peu linguistiquement corrects, il est donc intéressant de combiner les deux types de méthodes pour conserver l'avantage des deux : des traitements simples et purement statistiques et des traitements fondés sur une connaissance linguistique forte. Dans notre travail, en se basant sur un corpus de textes donné, nous avons mis en évidence l'intérêt de la combinaison entre traitements statistiques et connaissances linguistiques sur les contextes morpho-syntaxiques et sémantiques.

Problématiques traitées :

- La première problématique concerne le domaine de l'apprentissage automatique, qui constitue une phase incontournable dans la conception d'un système de filtrage automatique de l'information. En effet, l'apprentissage automatique constitue la propriété « intelligente » d'un système automatique. Les tâches d'apprentissage visées sont d'acquérir de meilleures ou

de nouvelles connaissances et d'offrir un mécanisme ou une procédure (ex: classification) permettant d'améliorer les performances du système de filtrage (adaptation, optimisation, etc.). En outre, la quantité grandissante d'informations électroniques permet de constituer des échantillons de données variés et significatifs. Un système qui permet d'apprendre automatiquement des profils d'utilisateur et d'exploiter ces connaissances pour filtrer l'information semble incontournable.

- La deuxième problématique concerne la nécessité d'utiliser des ressources et des traitements linguistiques. La plupart des systèmes actuels de filtrage de l'information exploitent peu d'informations de nature linguistique. Dans notre approche, nous avons montré que l'utilisation de connaissances et de traitements linguistiques peut améliorer les performances d'un système de filtrage de l'information.

Approche proposée

Dans le but d'améliorer l'efficacité du filtrage de l'information, nous avons choisi la démarche suivante :

- Nous utilisons un ensemble de connaissances linguistiques sous forme de modèles réduits. Nous avons donc défini et identifié un ensemble de propriétés automatisables, qui servent à caractériser les textes. Il s'agit d'un ensemble d'indicateurs sur le texte (portant sur la structure et le contenu) qui permettent de situer un texte par rapport aux autres textes. Ces connaissances sont indépendantes du domaine d'application. Nous les avons classées en plusieurs niveaux (matériel, énonciatif, structurel et syntaxique). Dans le cadre de ce travail, nous ne cherchons pas à faire une analyse complète et profonde du contenu des textes, mais plutôt une analyse partielle sur plusieurs niveaux. Cette analyse permet d'identifier des propriétés linguistiques qui devraient permettre de distinguer les différents types de textes et de classer ensuite les nouveaux textes.

- Nous proposons une solution évolutive permettant au système de filtrage d'apprendre à partir de données (apprendre automatiquement des profils d'utilisateur) et de s'adapter à la nature de l'application (textes traités) dans le temps.

- Nous introduisons une analyse sémantique qui associe à un texte un ensemble de mots même s'ils n'apparaissent pas explicitement dans le texte. L'idée de base est que les concepts définis pour représenter un profil ne sont pas forcément les mêmes que ceux extraits à partir des textes. Pour cela, nous proposons d'utiliser un réseau lexical permettant d'améliorer la représentation du texte en prenant en considération les termes qui existent dans le texte et qui n'existent pas dans le profil. Il s'agit de les remplacer par des termes du profil sémantiquement proches.

- Nous proposons un outil d'aide baptisé GIFI (assistant à la génération d'interfaces de filtrage). Il est destiné à faciliter la tâche des utilisateurs développeurs dans l'élaboration de systèmes de filtrage. Il permet d'assister l'utilisateur dans le processus d'acquisition de l'application (corpus de textes) et de la génération de ressources (vocabulaire lexical, propriétés linguistiques, modèle de filtrage). Il repose sur une conception modulaire, lui permettant éventuellement de s'adapter à des extensions ou à des mises à jours éventuelles. Il a une architecture ouverte permettant d'ajouter des composants et d'offrir ainsi, à l'utilisateur, la possibilité de choisir, à chaque étape du processus de génération, les outils à utiliser. Cette

boite à outils constitue une implémentation d'une approche hybride du filtrage de l'information. Elle repose sur le principe d'une analyse partielle utilisant un ensemble de connaissances, où le repérage de propriétés linguistiques permet, d'une part, d'améliorer la représentation de textes, et d'autre part un filtrage de meilleure qualité. Chaque corpus (textes consultés et validés par l'utilisateur) est soumis à un processus d'analyse automatique qui permet de lui associer un ensemble de termes et de propriétés linguistiques qui servent à le caractériser. Partant de propriétés issues de modèles linguistiques de textes (sous forme de modèles réduits), la fiabilité repose sur l'opération d'apprentissage.

Organisation du document

Le présent document est organisé en deux grande parties, composée chacune d'un ensemble de chapitres.

La première partie du document concerne la description de la problématique de recherche traitée avec, en premier lieu, une présentation de l'état de l'art du domaine. La deuxième partie concerne la mise en œuvre de notre approche. La faisabilité de l'approche proposée est discutée dans cette partie, ainsi que l'évaluation de son apport en matière d'efficacité du filtrage. L'évaluation est basée aussi bien sur des données chiffrées que sur une réflexion plus qualitative. Enfin, dans une dernière partie, nous mettons en perspective les problèmes abordés au cours de notre travail, notamment au sujet du statut des études sur corpus, des rapports entre linguistique et filtrage d'information, ainsi que des relations entre linguistique et apprentissage.

Chaque chapitre comporte une introduction qui expose le fil conducteur du thème traité et se termine par une conclusion. Ceci permet au lecteur d'avoir une vue globale du contenu de chaque chapitre.

Chapitre 1

Dans ce chapitre, nous présentons le problème général d'analyse automatique de textes : situation de l'analyse automatique par rapport au courant théorique (Chomsky, etc.) et aussi par rapport au courant de TALN (Traitement Automatique des Langues Naturelles). Aussi, nous présentons l'analyse formelle et l'analyse appliquée du langage naturel en décrivant les différentes étapes nécessaires à cette analyse, les différents outils existants pour chaque niveau d'analyse, ainsi que les différentes approches d'analyse automatique de textes. Nous terminons le chapitre par une présentation des différents problèmes liés au langage naturel, son intérêt et les outils appropriés pour le domaine de filtrage d'information.

Chapitre 2

Dans ce chapitre, nous décrivons notre problématique de recherche avec une présentation du contexte. Nous nous intéressons au domaine de filtrage de l'information électronique, plus précisément l'information textuelle. Nous présentons un bref historique sur le domaine du filtrage de l'information, les domaines applicatifs ainsi que quelques travaux précédents, puis nous aborderons les questions fondamentales auxquelles doit répondre un système de filtrage, tout en décrivant ses caractéristiques et les problèmes liés à son évaluation. Nous terminons ce

chapitre en présentant les principales approches possibles de filtrage qui vont nous permettre de dégager et de proposer une solution à notre problématique. Nous décrivons le principe général, ainsi que les avantages et les inconvénients de chacune d'elles.

Chapitre 3

Dans ce chapitre nous présentons le domaine de l'apprentissage automatique, étape incontournable et nécessaire dans la conception d'un système de filtrage automatique de l'information. Nous présentons le principe général du processus d'apprentissage, les tâches, les applications pratiques, ainsi que les différentes techniques utilisées par les systèmes d'apprentissage automatique. Nous citons les approches les plus utilisées dans le contexte de la classification automatique de textes. Nous présentons les modèles et les méthodes de représentation des textes.

Chapitre 4

Ce chapitre est consacré à la présentation de l'architecture générale de notre système de filtrage. Il s'agit d'une architecture évolutive qui s'adapte à l'information dynamique et qui permet un filtrage de meilleure qualité en utilisant différentes connaissances et une méthode d'apprentissage automatique permettant au système d'apprendre et de s'adapter à l'évolution du contexte. Nous décrivons le fonctionnement général ainsi que les principaux modules du système (analyseur linguistique, générateur de critères et du modèle filtrage, module d'apprentissage), les différentes connaissances et les outils utilisés.

En matière d'innovation, nous proposons deux améliorations des systèmes de filtrage existants :

- D'une part, il s'agit d'élargir l'éventail des propriétés qui serviront à améliorer la représentation classique de textes : nous proposons, en plus des caractéristiques lexicales, un ensemble de critères automatisables, susceptibles d'influer sur le processus de filtrage, permettant ainsi de situer un texte par rapport aux autres textes. Ces propriétés sont basées sur des modèles linguistiques réduits. Ces critères sont des indices qui portent généralement sur la structure et sur le contenu des textes. Ces connaissances sont indépendantes du domaine d'application. Nous les avons classées en plusieurs niveaux linguistiques: matériel, énonciatif, structurel et syntaxique.

- D'autre part, nous décrivons notre proposition pour la prise en compte de l'aspect sémantique dans le processus de filtrage : nous exposons les besoins que vise à satisfaire cet aspect sémantique dans le processus de filtrage, puis nous décrivons les connaissances (réseau lexical et cooccurrence de critères) et les traitements sémantiques.

Nous terminons par la présentation de GIFI, un assistant à la génération d'interfaces de filtrage.

Chapitre 5

Dans ce chapitre nous décrivons une application pratique du domaine du filtrage de l'information : filtrage du courrier électronique. Nous présentons quelques systèmes commerciaux et prototypes de filtrage de messages. Nous décrivons quelques stratégies et approches de traitement et de filtrage de messages. Ensuite, nous décrivons le corpus de messages électroniques (emails) utilisé, ainsi que la typologie et les caractéristiques associées et nous présentons notre système de courrier électronique paramétrable avec plusieurs niveaux de filtrage.

La fin du chapitre est consacrée à l'évaluation des performances du système GIFI (un générateur d'interfaces de filtrage d'information), sur un corpus de messages électroniques. Nous donnons quelques mesures chiffrées de performance de notre approche de filtrage de l'information. En effet, nous avons mené des tests pour :

i) mesurer l'importance et le rôle de l'information linguistique dans la représentation des messages,

ii) mesurer les performances du système de classification du point de vue précision et rappel,

ii) montrer comment l'opération d'apprentissage agit sur l'efficacité du filtrage.

Enfin, nous complétons l'évaluation quantitative du système GIFI par des éléments qualitatifs, complémentaires des aspects quantitatifs. De manière générale, l'évaluation qualitative de systèmes automatiques destinés à une application précise, représente un domaine de recherche à part entière. Ainsi, nous nous limiterons, dans notre travail, à une expérience visant à évaluer l'utilisabilité du système par des utilisateurs « non spécialistes ».

Chapitre I

Linguistique informatique et analyse automatique de textes

Dans ce chapitre, nous nous pencherons sur le problème général d'analyse automatique de textes : situation de l'analyse automatique par rapport au courant théorique et aussi par rapport au courant de TAL (Traitement Automatique des Langues). Nous présentons l'analyse formelle et l'analyse appliquée du langage naturel en décrivant les différentes étapes nécessaires à cette analyse, les différents outils existants pour chaque niveau d'analyse, ainsi que les différentes approches d'analyse automatique de textes. Nous terminons le chapitre par une présentation des différents problèmes liés au langage naturel, son intérêt et ses outils pour le domaine de filtrage d'information.

1 La linguistique formelle et le langage

1.1 Etude du langage

La linguistique est une discipline qui cherche à caractériser et à expliquer le langage humain. Elle porte à la fois sur l'acquisition, la production et la compréhension du langage.
Le langage est flexible, ambiguë et complexe, ce qui rend difficile sa formalisation.
Nous distinguons deux grandes approches en linguistique : d'un côté une linguistique théorique, rationaliste, de l'autre une linguistique empirique.
La première suppose une connaissance à priori[1] (règles linguistiques de compétences : règles phonologiques, lexicales et syntaxiques) alors que la seconde s'appuie sur une capacité d'association, de reconnaissance et de généralisation à partir d'exemples[2] c'est-à-dire les règles linguistiques sont identifiées et conceptualisées par une démarche *d'abstraction-génération* à partir des propriétés observables des divers textes en usage dans une communauté. L'approche empirique a débuté dans les années vingt pour faire place, dans les années soixante, à l'approche rationaliste promue entre autre par Chomsky. Cette dernière s'est fondée en même temps que l'appareil formel sur lequel elle repose. Elle a le plus souvent critiqué et caractérisé l'approche empirique comme une simple méthode de

[1] Idée maîtresse: l'être humain naît avec une compétence linguistique "hard codée". Phrase clé: "Colorless green ideas sleep furiously" (Chomsky).

[2] Idée maîtresse: l'être humain est effectivement doté de compétences, mais d'une nature différente: reconnaissance de formes, déduction, généralisation, etc.

description, arguant du fait qu'elle ne pouvait ni prédire (induire des règles), ni expliquer la grammaticalité.
Depuis les années 80, l'approche empirique revient en force. En effet, la grande quantité de textes devenue disponible sous format électronique ainsi que des moyens de stockage et de traitement de plus en plus performants ont permis d'accélérer son développement. Cette approche suppose qu'on peut apprendre (voire expliquer) la structure du langage en spécifiant un modèle de langue dont les paramètres peuvent être déterminés à l'aide de statistiques et d'inférences sur des corpus de textes.
La linguistique regroupe un certain nombre d'écoles qui ont toutes en commun d'avoir le langage comme objet d'étude mais qui n'abordent pas forcément les problèmes du même point de vue.

1.1.1 Approche structuraliste de Saussure

La linguistique structurale est un courant qui réunit un groupe d'écoles dans lesquelles la langue est étudiée comme un système doté d'une structure décomposable. Saussure est le fondateur de la linguistique comme étude des structures [BAL 02]. Ce courant apparu dans les années 60 consiste à privilégier les structures par rapport aux éléments qui leur appartiennent. Il considère ainsi que tout élément du langage fait partie d'une structure qui le détermine dans sa nature par son opposition aux autres éléments. Cette approche est caractérisée par le recours aux productions linguistiques effectives, c'est-à-dire une linguistique attachée aux phénomènes langagiers et donc aux observables linguistiques.

Dans cette approche, la démarche, pour étudier les faits langagiers, est caractérisée par deux principes :

- Abstraction des unités linguistiques : organiser le réel en un ensemble d'éléments contenus dans des classes, c'est-à-dire l'objectif est de classer les observables (catégorisation).

- Relations d'opposition : une démarché basée sur le principe d'opposition : les éléments d'une langue donnée ne sont conçus qu'en ce qu'ils s'opposent à d'autres éléments. C'est-à-dire, une démarche uniquement centrée sur les observables (la forme signifiante) des éléments linguistiques, dans laquelle le sens de ces éléments n'intervient pas. L'exclusion des phénomènes sémantiques constitue pour Saussure le fondement d'une étude scientifique des faits langagiers.

1.1.2 Approche distributionnaliste de Harris [HAB 02] [BRI 92b] [HAR 91] [HAR 68] [HAR 51]

L'approche distributionnelle a été développée aux États-Unis à partir de 1930 par Leonard Bloomfield (1887-1949) et par Zellig Harris dans les années 40. L'objectif de l'analyse distributionnelle est de décrire un état de la langue. Pour y parvenir, elle utilise des procédés fondés sur la segmentation et la classification. C'est une approche descriptive et taxinomique. Elle consiste à observer et à classer des faits (juste collectionner, pas de construction). Les deux principes qui régissent l'analyse distributionnelle sont la commutation et la combinaison. Par exemple, classer deux mots (*fille* et *garçon*) dans un même ensemble (même catégorie), revient à voir s'ils ont la même distribution : c'est à dire s'ils peuvent commuter (occuper la même place) et ils entrent dans les mêmes combinaisons (*devant un*

verbe, et *adjoints à un article*, etc.). Cette approche a pour objet de décrire les unités d'une langue en fonction de la possibilité qu'elles ont ou non de s'associer entre elles. Elle est vue comme un ensemble de méthodes formelles (statistiques et contextuelles) permettant d'étudier des langues « de l'extérieur », en éliminant le recours au sens. Il s'agit d'une méthode d'analyse empirique et rigoureuse qui permet à un linguiste de construire la grammaire d'une langue (qu'il peut très bien ne pas comprendre du tout) à partir de l'ensemble des positions et contextes (la distribution) qu'un mot ou qu'un groupe de mot peut occuper dans cette langue. C'est-à-dire qu'il est possible de faire de la syntaxe et de la sémantique de façon empirique et objective, i.e. en ne comptant que sur les *jugements de grammaticalité*, à condition d'avoir un corpus important.
L'idée de base est donc de partir d'un corpus pour construire, à l'aide de méthodes formelles, une représentation compacte (structure) des langues étudiées (une approche empirique). Pour Harris, ces méthodes permettent d'obtenir de l'information sur le langage (des *procédures de découvertes*). Il n' y a pas de construction de modèle quelconque (informatique, logique, et encore moins psychologique).
L'approche consiste à segmenter l'énoncé linguistique en unités (la phrase est segmentée en constituants immédiats, puis en morphèmes), à étudier la distribution et classer les variantes. Elle est fondée sur l'inventaire de la distribution des unités (phonèmes, morphèmes, etc.).
Le critère de classement grammatical est la position (ou distribution) par opposition au sens ou à la fonction.

Exemples:
Ce qui apparaît devant un verbe (V) est un nom (N).
Ce qui apparaît devant N est un article (Art).
Ce qui apparaît après Art + N est un V.
Ce qui apparaît entre V et Art + N est une Préposition (Pré).
Etc.

L'approche distributionnelle a introduit la notion de structure en constituants (groupes ou syntagmes) et la notion de transformation qui seront reprises et systématisées par l'approche générative de Chomsky. En effet, le critère de distribution est appliqué non plus simplement à des mots isolés mais à des suites de catégories. Par exemple, la séquence Art + N se comporte comme un bloc (apparaît en effet dans les mêmes contextes que N seul) et appartient à une méta-catégorie SN (syntagme nominal), etc.
Cette approche est considérée comme une classification descriptive des types d'occurrences observés dans le corpus à l'étude (aucune prédiction quant à la forme des phrases qu'on peut trouver à l'extérieur du corpus). De plus, son incapacité de caractériser les relations entre types de phrases (phrase déclarative, phrase interrogative, etc.).

1.1.3 Approche Générative de Chomsky

Élève de Harris, Chomsky a initié et développé la théorie logique (mathématique) des langages, applicable aux langages informatiques et aux *langues naturelles* (*Structures syntaxiques*).
L'approche générative découle de l'approche distributionnelle. Au cours des années 55 à 77, Chomsky a révolutionné la linguistique en inversant le problème qui préoccupait les distributionnalistes (***révolution méthodologique***) et en introduisant un formalisme inégalé (***révolution formelle***). En effet, il ne s'agit plus, comme pour Bloomfield ou Harris, de percevoir la langue comme un gros corpus. Pour Chomsky la langue a une capacité créatrice.

Il existe une possibilité de partir d'un ensemble fini de règles pour engendrer un nombre infini de phrases par un système de combinaisons. C'est-à-dire, quel est le système de règles qui permet de produire un corpus donné ?
L'objectif de Chomsky est de construire un modèle logique (formel, mathématique) permettant de produire et analyser des phrases.
Les linguistes Chomskyens postulent l'existence de deux niveaux de représentation : *une structure profonde* et *une structure de surface*. Chaque langue est censée avoir la même structure profonde mais sa structure de surface peut varier. Ils expliquent ces différences par des règles de transformations et des choix de paramétrage qui eux sont spécifiques à chaque langue. Par exemple, les adjectifs précèdent le nom qu'ils modifient en anglais et le suivent généralement en français.
Le problème d'incapacité de rendre compte des relations entre types de phrases, est contourné par le recours aux règles de transformations (relative, négative, interrogative, etc.). Par exemple, une phrase interrogative est une phrase déclarative sur laquelle des transformations ont été effectuées.
Cette approche est passée par différentes étapes : le modèle de structures syntaxiques (1957), le modèle standard (1965), le modèle standard étendu (1970), le modèle gouvernement et liage (1980) et le modèle minimaliste (1990). La composante syntagmatique est devenue progressivement complexe et la composante transformationnelle plus réduite.

1.1.4 Les approches basées contraintes

Dès les années 80, d'autres modèles importants ont vu le jour tels que les grammaires d'arbres adjoints TAG (Tree Adjoining Grammars), les grammaires syntagmatiques généralisées GPSG (Generalized Phrase Structure Grammars), les grammaires syntagmatiques guidées par la tête HPSG (Head-driven Phrase Structure Grammars) et les grammaires lexicales fonctionnelles LFG (Functional Lexical Grammar). Ces modèles reposent sur l'unification fonctionnelle qui permet d'alléger la syntaxe, d'où le nom global de grammaires d'unification. Ces mécanismes se sont développés en réaction aux limites formelles des grammaires transformationnelles de Chomsky. Ils sont caractérisés par les propriétés suivantes :

- La phrase est analysée telle qu'elle s'énonce : pas de représentation profonde (approche de surface).

- Un seul format pour représenter toutes les informations linguistiques : *la structure de traits*. Elle permet de décrire les unités linguistiques (mot, syntagme, etc.) sous la forme d'une liste structurée et non ordonnée, contenant leurs caractéristiques (ensemble de paires attribut-valeur). Les structures de traits peuvent être simples ou complexes (hiérarchiques ou récursives). Ce mode de représentation permet de simplifier les règles de grammaire et d'affiner les contraintes au niveau du lexique.

- *L'unification* : est un mécanisme qui permet de fusionner des informations linguistiques contenues dans des structures de traits. L'opération d'unification est proche de l'opération *union* de la théorie des ensembles, à la différence que l'unification échoue lorsque les structures de traits contiennent des incompatibles.
Par exemple, pour modéliser le phénomène de l'accord (genre, nombre) dans un syntagme nominal, il suffit d'associer à chaque nœud de l'arbre syntaxique non plus une étiquette

simple (GN, GV, etc.) mais une structure de traits contenant la catégorie et les informations nécessaires à l'accord (genre et nombre) (figure I.1).
L'unification présente des propriétés intéressantes pour le traitement du langage naturel. En effet, l'unification permet de combiner les informations associées aux mots et aux syntagmes pour construire la représentation de la phrase, et de vérifier leur compatibilité.

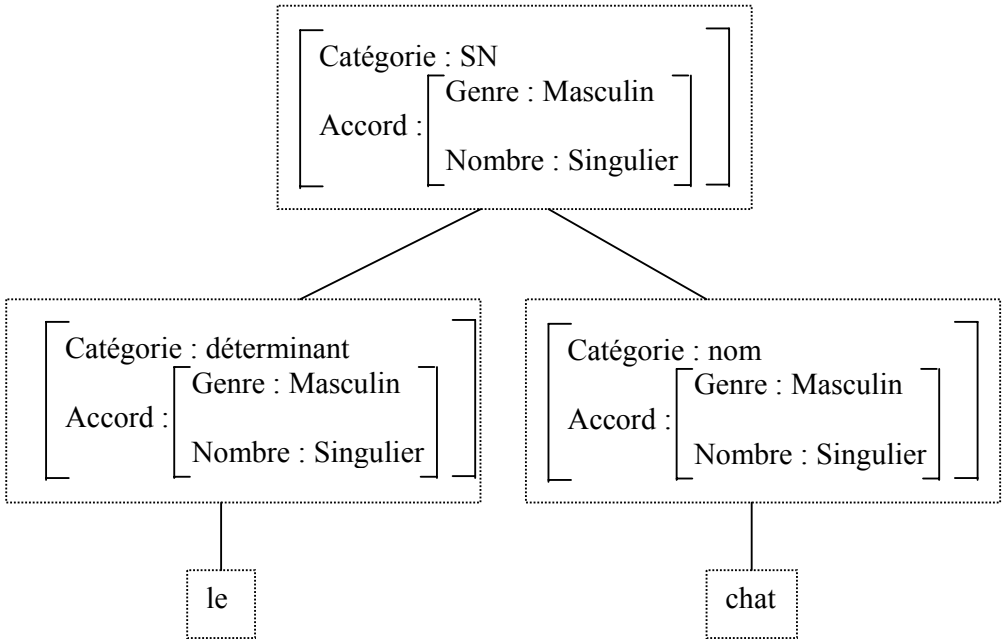


Figure I.1 : L'accord

- Une démarche déclarative, non procédurale : description des structures grammaticales plutôt que des procédures qui permettent de les obtenir.

- La grammaire doit spécifier la *décomposition* de syntagmes en constituants et les *contraintes* qui pèsent sur les différentes structures de traits. Le principe de la grammaire consiste à confronter et fusionner les informations lexicales de manière à construire progressivement une représentation globale et cohérente de la phrase. En effet, l'analyse échoue lorsqu'il y a une incompatibilité entre les traits.

- Le processus d'analyse est conçu comme un problème de satisfaction de contraintes. Tous les formalismes linguistiques font usage systématique de la notion de contrainte qui, dans son sens le plus large, indique une propriété devant être satisfaite. Cependant, l'usage qui est fait des contraintes peut être très différent d'une approche à l'autre : dans certains cas, il s'agit simplement d'un mécanisme d'appoint, dans d'autres, les contraintes sont au cœur de la théorie [BLA 00a].

a)- HPSG (grammaires syntagmatiques guidées par la tête ou Head-driven Phrase Structure Grammars) [BLA 95] [BLA 00b]:

HPSG est une théorie linguistique se proposant de fournir un cadre de modélisation (un formalisme, une méthode) de principes grammaticaux universels : c'est-à-dire un cadre

permettant de représenter de manière explicite et la plus complète possible les connaissances linguistiques.
HPSG est une théorie non dérivationnelle : elle analyse les relations entre les éléments d'une structure linguistique non pas à l'aide de transformations mais en termes de partages d'information. Par exemple, le SN *la grosse artillerie de Françoise* devra partager les traits de tête du lexème *artillerie* qui est la tête de la construction.

HPSG regroupe dans une même représentation des informations linguistiques variées (phonologiques, lexicales, syntaxiques, sémantiques et pragmatiques) : un seul format, qui est la structure de traits.
HPSG est basée sur le principe de la logique attribut/valeur. Elle est régie par le mécanisme d'**unification.**
En HPSG, les structures de traits sont typées : tous les traits appartenant à une structure de traits doivent être appropriés au type de la structure. Les types sont hiérarchisés (syntagme, tête, contenu, index, etc.). Exemples de traits : SYNSEM (traits syntaxiques et sémantiques), TETE, LOC (information intrinsèques), CONTEXTE (information pragmatique), INDEX (traits d'accord), CONTENU (sémantique), FILS-F (relation hiérarchique), REL (les relatives), etc.

b)- LFG (grammaires lexicales fonctionnelles ou Functional Lexical Grammar) [BRE 82] :

Les grammaires lexicales fonctionnelles ne font pas de distinction entre structure profonde et structure de surface (comme Chomsky) : elles permettent d'attribuer à tout énoncé de la langue:
(a) une structure de constituants (c-structure) qui est un arbre étiqueté engendré à partir de règles de réécriture et qui décrit directement l'agencement superficiel des éléments de cet énoncé.
(b) une structure fonctionnelle (f-structure) : Les relations fonctionnelles (relations prédicats/arguments) sont représentées par des structures de traits codant les différentes fonctions grammaticales : "SUJET", "OBJET", "ADJOINT", "GENRE", "NOMBRE", etc.
Il n'y a pas de règles transformationnelles. Le lexique contient des informations catégorielles et fonctionnelles.

c)- Les grammaires de propriétés [BLA 00a] [BLA 00b]

Les grammaires de propriétés constituent un nouveau formalisme qui s'appuie exclusivement sur le principe de contraintes (une contrainte indique une propriété devant être satisfaite), aussi bien du point de vue de la représentation des informations que de l'implantation.
Elles constituent une alternative pour l'utilisation systématique de contraintes. Elles proposent en effet une solution aux restrictions des autres formalismes linguistiques, utilisant la notion de contrainte et qui présentent un certain nombre de restrictions à leur utilisation, en particulier pour ce qui concerne leur implantation.
Parmi les intérêts d'une telle approche, on peut citer la simplicité et la souplesse de la représentation, la robustesse, mais également la possibilité d'intégrer des informations venant de différents niveaux de l'analyse linguistique. De plus, la satisfaction de contraintes étant effectivement le seul mécanisme utilisé : aucun processus de dérivation n'est utilisé, en d'autres termes, *aucune règle syntagmatique ni schéma de règles* n'est nécessaire pour calculer la structure syntaxique d'un énoncé.

Les applications sont nombreuses, en particulier pour ce qui concerne le traitement de la langue parlée.
De plus, les grammaires de propriétés constituent un cadre permettant à la fois de tirer parti des avantages des grammaires syntagmatiques tout en s'affranchissant des problèmes liés aux approches génératives pour ce qui concerne l'usage des contraintes en tant qu'unique composant grammatical.
Les g*rammaires de propriétés* se situent donc dans le paradigme de grammaires syntagmatiques et préservent la notion de structure syntaxique hiérarchisée.
Dans ce formalisme, les contraintes (appelées propriétés) ne portent pas sur des informations structurées (par exemple, la structure d'arbre dans HPSG), mais plutôt directement sur les catégories. Les catégories sont formées par une structure de traits comportant les informations (lexicales, syntaxiques, sémantiques, etc.) susceptibles d'intervenir dans la spécification de contraintes. Chaque catégorie (verbe, nom, etc.) est associée à un sous ensemble de propriétés telles que : *constituance* (catégories pouvant apparaître dans une unité syntaxique), *obligation* (la catégorie qui doit être présente), *unicité* (catégories uniques dans un syntagme), *exigence* (cooccurrence entre des ensembles de catégories), *exclusion* (restriction de cooccurrence entre des ensembles de catégories), *Linéarité* (contraintes de précédence linéaire), *dépendance* (relations de dépendance entre les catégories), etc.
Une grammaire de propriétés est formée donc par un ensemble de propriétés exprimant différentes relations entre les catégories formant la structure syntaxique. Ces propriétés peuvent être très spécifiques et concerner un ensemble limité de catégories ou au contraire très générales.

Le mécanisme d'analyse consiste à vérifier pour une suite de catégories (suite possible pour un énoncé donné) la satisfaisabilité de l'ensemble des contraintes associées aux différentes catégories.

L'analyse d'un énoncé passe par les étapes suivantes :

- *Phase de catégorisation* : c'est-à-dire la détermination des catégories nécessaires à l'analyse. En effet, il s'agit, en premier, simplement d'énumérer les catégories lexicales possibles. Ensuite, pour chacune des catégories lexicales ainsi spécifiées, rechercher la catégorie de niveau supérieur à laquelle elle peut appartenir (propriété de constituance).

- *Génération de suites de catégories* : par simple énumération de l'ensemble des suites de catégories possibles.

- *Caractérisation des suites* : en calculant, pour chacune des suites, l'ensemble des contraintes satisfaites et l'ensemble des contraintes non satisfaites.

Une caractérisation dont toutes les contraintes sont satisfaites correspondra à une suite grammaticale.

Cette approche est robuste et donc peut bien être utilisée dans le domaine de filtrage d'information dynamique où les énoncés ne respectent pas généralement une syntaxe stricte. En effet, il est bien entendu possible de contrôler le processus d'analyse en spécifiant un seuil maximum de contraintes non satisfaites. Ainsi, on pourra caractériser tout type d'énoncé, y compris ceux correspondant à des structures ne satisfaisant pas toutes les propriétés de la grammaire.

1.2 La linguistique formelle

Une structure communément admise pour l'analyse du langage naturel se compose classiquement des modules suivants : Les dictionnaires, les règles de grammaire de la langue étudiée, analyse morphologique, analyse lexicale, analyse syntaxique, analyse sémantique et enfin une analyse pragmatique. Ces différents modules fonctionnent en coopération et/ou de manière séquentielle [KAY 01] [BOU 98].

1.2.1 Analyse morphologique

L'analyse morphologique est l'étape préliminaire à tous les types de traitement d'un texte (étiquetage grammatical, analyse syntaxique, etc.). En effet, un texte écrit en langue naturelle, peut être considéré comme une suite d'entités (mots, signes de ponctuations et séparateurs). L'analyse morphologique permet la reconnaissance de la forme des mots, c'est-à-dire reconnaître les mots sous les différentes formes (conjugaison, déclinaisons, etc.) que leur rôle dans la phrase leur affecte.
Le problème principal de la morphologie est la variation orthographique et morphologique des mots : phénomènes d'inflexion (tout ce qui a trait à la conjugaison, au genre ou au nombre des mots), de composition et de dérivation (la façon dont les mots sont formés à partir d'autres mots).
L'analyse morphologique consiste donc à reconnaître la forme de base des mots et des différents affixes qui leur sont associés.
Cette forme est :

- Soit le *lemme :* infinitif pour un verbe, la forme masculin singulier pour un nom ou un adjectif, etc.

- Soit la racine ou radical : mange pour manger, mangeons, etc.

Cette analyse peut être effectuée soit à l'aide d'un dictionnaire de formes fléchies, soit à l'aide d'un analyseur morphologique ou lemmatiseur (stemmer), par exemple l'analyseur FLEMM [NAM 00] : FLEMM calcule le lemme de chaque mot fléchi (en fonction de son étiquette) et fournit également ses principaux traits morphologiques, avec un taux de réussite avoisinant les 99% pour un mot correctement étiqueté catégoriellement.
La plupart des analyseurs morphologiques développés se basent sur les automates d'états finis. Par exemple, l'analyseur morphologique développé à XRCE-MLTT utilisant des transducteurs (automates à états finis à deux niveaux).

Dans le cadre du filtrage d'information, l'analyseur morphologique sera utilisé pour reconnaître certaines formulations proches de concepts, et donc pour regrouper des termes appartenant à une même famille morphologique. Ce qui permet de représenter les mots de la même famille par un même terme et d'améliorer les performances des systèmes de filtrage, du moins pour les langues à caractère morphologique plus fortement marquées, comme le français.

1.2.2 Analyse lexicale

L'analyse lexicale est l'opération qui consiste à établir l'existence des entités reconnues par l'analyse morphologique, c'est-à-dire à vérifier qu'elles appartiennent bien au vocabulaire de la langue. Elle permet d'attribuer, à chaque entité lexicale reconnue (le mot), un ensemble d'informations linguistiques, pertinentes pour la suite du traitement d'un texte. Ces différentes connaissances nécessaires à l'analyse sont stockées dans une composante appelée *lexique* ou *dictionnaire*. Il est généralement accepté d'en distinguer quatre types principaux d'informations linguistiques : *phonétique*[3], *morphologique, syntaxique* et *sémantique* [BOU 98].

Les informations morphologiques concernent les propriétés grammaticales des mots : *nombre, personne*, *temps* et *mode*. Par exemple, les informations morphologiques des mots *amies* et *mangera*, sont données dans la table I.1.

amies	*Nombre= pluriel* *Personne= féminin*
mangera	*Nombre= singulier* *Personne= 3ème* *Temps= futur* *Mode=indicatif*

Table I.1 : Informations morphologiques

Les informations syntaxiques concernent les informations *catégorielles* (les différentes catégories syntaxiques : *nom, verbe, adjectif, pronom,* etc.) et *sous catégorielles* (les différentes sous classes à l'intérieur des catégories syntaxiques : *verbe transitif, intransitif*, etc.). Par exemple, les informations syntaxiques des entités *je, chien, donne* et *pars*, sont données dans la table I.2.

Je	pronom
Chien	nom
[Il] *donne* [un jouet]	Verbe transitif *(verbe à deux arguments : sujet + complément d'objet)*
[Je] *pars*	*Verbe intransitif* (verbe à un argument : sujet)

Table I.2 : Informations syntaxiques

Les informations sémantiques concernant certains traits sémantiques avec la précision du concept auquel le mot est associé en plus de son domaine d'utilisation (objet, substance, animal, être humain, etc.).

Exemple: "écrire" est une action ne pouvant être effectuée que par un humain.

[3] Ne traitant pas de la parole, nous négligerons ce type d'information.

Ces informations sont utiles au bon déroulement des phases ultérieures (analyse syntaxique, sémantique, etc.).

Généralement, l'analyse morphologique et l'analyse lexicale sont combinées en une seule analyse appelée analyse morpho-lexicale : elle comporte donc une grammaire et un dictionnaire. La grammaire contient des règles qui contrôlent la reconnaissance et la composition des formes à partir des éléments contenus dans le dictionnaire. Ces éléments sont des préfixes, des racines (radicaux ou bases), des suffixes et des désinences (terminaisons).

Les performances d'un travail portant sur l'analyse automatique d'un segment linguistique (phrase, paragraphe, texte, etc.) dépendent en partie de l'organisation du dictionnaire vu le rôle qu'elle joue dans l'identification des mots. En effet, le problème fondamental est de trouver une organisation du dictionnaire qui garantisse:

- un coût d'accès minimal,

- et un gain d'espace.

Il existe deux méthodes classiques d'organisation du dictionnaire [PIT 85]:

La première, et la plus évidente, est celle qui consiste à représenter toutes les formes des mots dans le dictionnaire (formes de conjugaisons et d'accords). L'inconvénient majeur de cette méthode est la taille volumineuse du dictionnaire. En revanche, cette méthode ne nécessite pas de traitement morphologique au niveau du mot lui même. Elle consiste donc à rechercher le mot dans le dictionnaire tel qu'il apparaît dans le texte.
La deuxième méthode est celle qui consiste à ne représenter que la racine du mot dans le dictionnaire. Cette méthode nécessite donc un traitement supplémentaire pour la reconnaissance des désinences. En effet, dès qu'une terminaison est reconnue, on vérifie que la racine (le mot privé de cette terminaison) est bien répertoriée dans le lexique. Le nombre d'accès au dictionnaire pour déterminer la bonne définition du mot augmente proportionnellement avec le nombre de couples (racine, terminaison) possible. En revanche, cette méthode présente l'avantage d'un gain d'espace.

1.2.3 Analyse syntaxique

L'analyse syntaxique est une composante très importante dans le traitement automatique des langues. Elle s'attache à un caractère formel de la langue: l'ordre dans lequel les mots doivent être disposés pour qu'une phrase soit jugée correcte. C'est-à-dire, elle s'intéresse à la structure de la phrase, et plus exactement aux relations possibles entre les diverses catégories de mots. Elle a donc pour tâche de former des groupes syntagmatiques afin d'assigner aux phrases des représentations syntaxiques qui explicitent leur structure.
La syntaxe est décrite dans une grammaire qui définit les principes et les contraintes qui régissent la combinatoire des mots (une grammaire qui décrit l'ordre dans lequel les mots peuvent être énoncés), et qui permettent de distinguer les phrases correctes des phrases incorrectes.
Elle permet de lever la plupart des ambiguïtés rencontrées lors des phases morphologique et lexicale [BOU 98].

Exemple: Dans la phrase "*je signe une lettre*", l'analyse morpho-lexicale identifie le mot "signe" comme étant soit une forme conjuguée du verbe "*signer*", soit un *substantif*. Mais l'analyse syntaxique l'interprète comme un verbe, car, en Français, un *nom commun* ne peut pas suivre un *pronom personnel objet*.

Dans ce paragraphe, sont présentées les principales méthodes formelles plus traditionnelles d'analyse syntaxique.

a)- Les grammaires à contexte libre

L'analyse d'une phrase consiste à décrire sa structure syntaxique sous forme d'arbre où:
- Les noeuds représentent des classes lexicales (nom, verbe, article, etc.) et syntaxiques (groupe nominal, verbal, etc.).

- Les feuilles correspondent aux mots de la phrase.

La question de savoir si les grammaires à contexte libre suffisent à décrire les langues naturelles, après avoir eu une réponse négative, est de nouveau ouverte depuis les travaux de Gazdar et Salkoff, mais le nombre de règles envisagées par ces chercheurs est tellement grand (de l'ordre du milliard) que les conséquences pratiques sont presque les mêmes [COU 86].
De plus, les grammaires à contexte libre ne sont pas nécessairement déterministes. En effet, parfois, l'analyseur est conduit à faire un choix. Si ce choix débouche sur une impasse, le programme doit retourner jusqu'au point de décision précédent, et employer une autre règle en effectuant un retour en arrière.
Une grammaire à contexte libre ne tient pas compte de tous les aspects de la langue. Par exemple, elle ne tient pas compte des règles d'accord en genre et en nombre (entre un nom et les adjectifs le qualifiant, par exemple) [HAR 85].
Néanmoins, il est important de noter que les grammaires à contexte libre sont parfaitement adaptées pour des sous-ensembles très limités du langage naturel [FER 84].

b)- Les grammaires transformationnelles [CHO 57]

Chomsky a défini un type de grammaire plus complexe, ce sont les grammaires transformationnelles. Cette théorie suppose que la formulation d'une idée se déroule en deux étapes:

- La création d'une structure profonde, seule susceptible d'être interprétée sémantiquement. Cette structure profonde est une arborescence engendrée par une grammaire, appelée grammaire de base.

- La transformation de celle-ci en une structure de surface directement liée à la forme de l'énoncé.

Des difficultés théoriques, mais surtout des inconvénients pratiques (la grammaire transformationnelle suppose qu'à l'analyse, on applique les transformations à l'envers pour retrouver la structure profonde) font qu'après avoir soulevé beaucoup d'espoir, ces grammaires ne soient guère employées que pour des études sans finalités pratiques immédiates [KAY 86].

c)- Les réseaux de transition [FER 84]

Les réseaux de transition se présentent comme un ensemble de graphes identifiés par un nom.

- Les arcs désignent des mots, des classes lexicales ou des catégories syntaxiques qui correspondent à d'autres réseaux.

-Les noeuds indiquent les étapes dans l'analyse.

Les réseaux de transition sont équivalents aux grammaires à contexte libre. Ils possèdent les mêmes inconvénients pratiques: l'indéterminisme. En effet, lorsque plusieurs arcs sont disponibles et qu'il n'est pas possible de trancher immédiatement pour savoir lequel doit être traversé (cas d'arcs conduisant à plusieurs réseaux), le système doit choisir d'emprunter tel chemin plutôt que tel autre. Comme dans le cas des grammaires à contexte libre, ce choix peut conduire à une impasse, et amener le système à effectuer des retours en arrière.
En contrepartie, les réseaux de transition offrent une formulation particulièrement simple et agréable. Ils s'avèrent aisément améliorables pour prendre en compte des grammaires plus complexes, et de plus, ils ont les avantages que possède un automate sur une grammaire (rapidité).

d)- Les A.T.N (Augmented Transaction Network)

Les ATNs développés par Woods représentent une amélioration des réseaux de transition. Les arcs de transition sont associés de deux types d'éléments:

- Des conditions à satisfaire permettant d'augmenter les critères de sélection pour le choix d'un arc.

- Des actions spécifiques à exécuter si l'arc est emprunté, servant à conserver des informations et permettant de construire des structures syntaxiques (l'arbre syntaxique par exemple).

De plus, les conditions et les actions utilisent un ensemble de registres qui servent à mémoriser des informations utiles sur l'enchaînement des différents constituants syntaxiques d'une phrase.

Une caractéristique qui a fait la réputation des A.T.N est leur clarté et simplicité. En effet, il est facile de :

- se faire une idée de l'enchaînement des différents constituants syntaxiques,

- concevoir des programmes très courts qui leur sont associées: quelques tests (par exemple conformité des accords) et mémorisation des constituants pouvant resservir ultérieurement.

En revanche, le problème de l'indéterminisme de l'analyse persiste. En plus, des retours arrière, il faut annuler l'effet des actions effectuées [KAY 86].

1.2.4 Analyse Sémantique

Elle traite le sens de la phrase. Les connaissances sémantiques concernent la description de sens des mots. Le sens d'un mot est représenté par les propriétés de l'objet correspondant dans le monde réel.
L'analyse sémantique est l'une des phases les plus délicates, car il est très difficile d'énoncer des règles strictes concernant les mécanismes de compréhension [LAS 86]. Il faudrait alors donner à la machine les moyens de simuler un raisonnement déductif afin de retrouver des informations.

Exemple:
Les oiseaux ont des ailes.
Les oiseaux peuvent voler.
Si l'on sait que: le canari est un oiseau, on déduira que les canaris ont des ailes et peuvent voler.

L'analyse sémantique vise à désambiguïser le sens. Les ambiguïtés sont généralement dues à l'utilisation des anaphores, des déictiques, des ellipses, des mots polysémiques, etc.

Les méthodes utilisées sont les méthodes symboliques, les méthodes à base de connaissances (les dictionnaires électroniques, les thésaurus et les lexiques informatiques) et les méthodes statistiques et textuelles (à base de corpus).

L'analyse sémantique a pour objectif d'extraire le sens en vérifiant les relations sémantiques entre les différents mots de la phrase. Dans ce paragraphe, sont présentées seulement quelques méthodes classiques d'analyse sémantique.

a)- Les grammaires sémantiques [BON 80]

L'intérêt des grammaires sémantiques réside dans la possibilité d'incorporer les relations sémantiques au niveau des règles de production décrivant la grammaire.
Le premier inconvénient des grammaires dites sémantiques est leur non transportabilité d'un domaine d'application à un autre; les conditions portant sur les catégories sémantiques dépendent fortement du domaine. Si l'on change de domaine, la grammaire devra être changée.
Pour des raisons pratiques, les grammaires sémantiques ne sont efficaces que si le domaine est restreint. En effet, les catégories sémantiques sont bien plus nombreuses que les catégories syntaxiques, ce qui entraîne une explosion de la grammaire.
L'avantage de l'utilisation des grammaires sémantiques, est la grande économie en temps dans l'analyse car la restriction des catégories à reconnaître rend l'analyse non ambiguë.

b)- Les A.T.N sémantiques

Les ATNs sémantiques constituent une évolution des ATNs conçus par Woods. Ce sont des ATNs dont les conditions portent, non seulement sur les caractéristiques syntaxiques, mais aussi sur leurs caractéristiques sémantiques. Les actions permettront alors de créer des morceaux de représentation de sens.

Le problème de l'indéterminisme persiste, quoique réduit, du fait de l'existence des conditions supplémentaires pour le franchissement d'un arc. En plus des retours en arrière, un autre problème essentiel est celui qui consiste à défaire les actions sémantiques quand on rebrousse chemin [KAY 86].

1.2.5 Analyse pragmatique

L'analyse pragmatique sert à analyser et à comprendre plus finement un énoncé, bien qu'aucune connaissance explicite n'apparaisse dans celui ci: ce sont des connaissances préacquises ou implicites. Ces connaissances sont très importantes; en effet, il ne suffit pas de connaître le sens des mots pour comprendre une phrase. Lorsque nous nous exprimons, nous présupposons chez notre auditeur la possibilité de comprendre, ce qui implique la connaissance d'un certain nombre de faits qui n'apparaissent pas dans la phrase et qui, dés lors, ne peuvent pas être soumis à l'observation directe. Nous cherchons donc à utiliser d'une façon quasi systématique, la possibilité de rendre implicite un certain nombre de faits (par exemple, l'utilisation des pronoms).

Exemple:
"Je suis allé au restaurant. On m'a apporté le menu. Après le café, j'ai donné un billet à la serveuse et je me suis promené".

Nous utilisons certaines de nos connaissances préacquises sur ce qui se passe habituellement dans un restaurant pour comprendre effectivement ce petit texte et déduire par exemple:
On = la serveuse,
j'ai choisi des plats, je les ai mangés,
le billet = un billet de banque,
je suis sorti du restaurant avant ma promenade,
etc.

L'analyse pragmatique a deux principaux buts:

- Lever les ambiguïtés induites par une interprétation qui est illogique dans notre univers.

- Compléter la représentation du sens: ceci n'est donc pas d'un grand apport à la compréhension, mais il l'est pour les programmes ultérieurs qui utilisent cette représentation. La manière de compléter la représentation de sens va dépendre de ce que nous voulons faire de cette représentation, mais pour cela il faut faire des déductions et des inférences. Une inférence est une déduction de faits nouveaux à partir de ce que l'on sait déjà, mais elle n'est pas aussi sûre qu'une déduction car certains faits que l'on possède peuvent être peu vraisemblables [LAS 86]. Donc si une inférence est faite sur ces derniers il n'y a rien qui assure que les résultats seront vraisemblables.
Pour compléter, par exemple, la représentation de sens, l'emploi des inférences va aider à générer des valeurs par défaut si celles-ci ne sont pas explicitées dans la phrase.

Exemple: 'Nadir a frappé sa petite sœur'.
Nous pouvons ajouter comme instrument la main et nous aurons la phrase:
Nadir a frappé sa petite soeur avec sa main.

1.2.6 Représentation de sens

Une analyse sémantique a pour objectif le calcul du sens. Elle doit tenir compte de la contribution du matériau linguistique. Calculer le sens d'un énoncé revient à déterminer la signification de ses mots et les relations sémantiques existantes entre ces mots. La signification d'un mot est déterminée par un ensemble de traits sémantiques appelés sèmes.

Pour représenter le sens d'un énoncé (par exemple, une phrase), il faut tout d'abord déterminer les notions à retenir lors de l'analyse de l'énoncé. Un énoncé porte sur plusieurs éléments et notions différentes. Les éléments sont des objets, des actions, des propriétés d'objets ou des relations entre objets.
Les objets désignent des noms propres ou de choses, comme "Omar", "livre", etc. Les actions engendrent des événements qui sont représentés par le type d'action et les valeurs de paramètres de cette action.

Exemple: Salim donne un livre à Nassima.

Action: donner
Sujet : Salim
Objet : livre
Bénéficiaire : Nassima

Les propriétés d'objets sont caractérisées par leurs types et leurs valeurs.

Exemple: "le ciel est bleu".
Objet : ciel.
Propriété (couleur) : bleu.

Les relations entre objets font intervenir plusieurs objets simultanément.

Exemple: "le livre est sur la table". (Relation de position)

Il y a plusieurs formalismes de représentation que les chercheurs utilisent, chacun possédant des avantages et des inconvénients.

a)- Les réseaux sémantiques

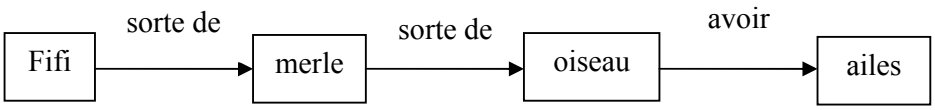
Ce sont des graphes formés de noeuds reliés par des arcs.

- Les noeuds représentent les concepts (oiseau, être humain, végétal, etc.).

- Les arcs représentent les relations entre concepts (être élément de, posséder, etc.) [BON 84].

Les réseaux sémantiques permettent de mettre en évidence les liens entre les éléments de la phrase. Il faut souligner aussi la facilité de réaliser des inférences. Ceci est à la base de la popularité des réseaux sémantiques [WOO 75].

Exemple:



L'héritage de propriétés le long des arcs "sorte de" permet de retrouver le fait que les merles en général et fifi en particulier sont des oiseaux et possédent des ailes.
Les réseaux sémantiques se prêtent bien à des études théoriques sur la représentation des connaissances. Mais, du point de vue pratique, ils sont assez lourds à mettre en oeuvre, car on aboutit, avec des phrases de difficultés moyennes, à des graphes complexes [DUD 83].

b)- Les grammaires de cas

Les grammaires de cas, introduites par Fillmore [FIL 68] permettent de représenter sémantiquement un énoncé. Pour Fillmore, la structure profonde d'un énoncé consiste en un quantificateur central appelé prédicat, qui est souvent le verbe et un ou plusieurs groupes nominaux, qui sont reliés au prédicat tout en indiquant leur rôle par rapport à ce dernier par des cas.
Une grammaire de cas est caractérisée par le nombre de ses cas sémantiques.

Les cas les plus utilisés sont:

- Agent: ce qui cause l'événement.

- Contre-agent: la force contre laquelle l'action est exécutée.

- Objet: entité qui est au centre de l'action.

- Instrument: entité qui a commis l'événement.

- Source: origine de l'événement.

- Destination: destination de l'événement.

- Lieu: lieu de l'événement.

- But: l'objectif de l'événement.

- Temps: durée de l'événement.

Exemple: 'Le nouveau chauffeur a conduit le camion, malgré le mauvais temps, d'Alger à Tlemcen'.

Prédicat: conduire,

Agent: le nouveau chauffeur,

Objet: le camion,

Contre-agent: le mauvais temps,

Source: Alger,

Destination: Tlemcen.

L'analyse d'une phrase par une grammaire de cas consiste à identifier le pivot de la phrase (prédicat) puis de reconnaître les différents cas associés au pivot. La mise en oeuvre de cette analyse, quoique délicate, est menée à bien grâce aux caractéristiques syntaxiques, ainsi qu'à la définition précise de chaque notion (contraintes sémantiques). Pour cela, il faut noter l'extrême importance du dictionnaire qui doit comporter les informations syntaxiques et sémantiques pour pouvoir faire l'association de concepts aux bons cas.
Ce qui fait le succès des grammaires de cas est le fait que les structures profondes qu'elles mettent en évidence font apparaître avec beaucoup plus de netteté le sens de la phrase [KAY 86]. Le problème essentiel reste toutefois l'association des mots aux bons cas. En effet, selon le contexte, un mot peut introduire des cas différents.

Exemples :

- "Donner à regret" introduit la manière,

- "Donner à Omar" introduit le sujet,

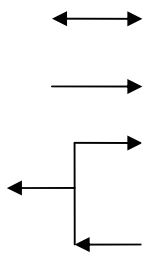
- "Donner à repasser" introduit le but.

En conclusion, les grammaires de cas nécessitent de nombreuses contraintes sémantiques, et par conséquent ne sont utilisables que sur un domaine limité.

c)- Les dépendances conceptuelles

Cette théorie développée par Schanck [SCH 70], suppose qu'il existe un ensemble restreint de primitives sémantiques permettant de modéliser la plupart des actions. Un événement est représenté par une structure comportant une action (primitive sémantique), un acteur, un objet et une direction.
Le sens d'une phrase est mis en évidence grâce à un réseau appelé: schéma de dépendance conceptuelle, qui représente des liens entre l'action, l'acteur, l'objet, etc.

Exemples de liens utilisés [PIT 85]:



lie une action à son acteur.

" " " " l'objet qui subit l'action.

déplacement d'un lieu A vers un lieu B

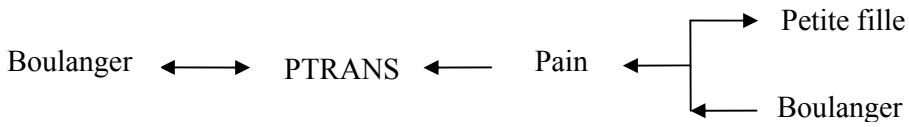
Exemples de primitives utilisées:

*MTRANS: indique le transfert d'une information.

*PTRANS: " " " changement de place d'un objet.

*MBUILD: création de nouvelles informations.

Exemple: La phrase "le boulanger a vendu du pain à la petite fille" sera représentée par le diagramme suivant:



Il faut noter deux caractéristiques de cette méthode:

- Les phrases de structures différentes mais de sens équivalent, doivent avoir la même représentation de sens, car l'intérêt du nombre réduit de primitives est d'aboutir à une normalisation des énoncés.

- Toute information implicite dans une phrase doit être rendue explicite dans la représentation. En effet, il suffit d'évoquer un concept, pour que le système de lui même évoque d'autres qui en dépendent.

La représentation de sens reste tout de même assez rigide du fait du nombre restreint des primitives. En effet, on est parfois obligé de représenter de la même manière des familles entières de mots, sans en distinguer parfois toutes les finesses (cas des mots "apprécier" et "aimer").

d)- Les Frames [LAS 86]

Les frames sont une représentation particulière des informations. Ce modèle permet une décomposition d'un univers en classes, c'est-à-dire qu'on regroupe dans un frame toutes les propriétés communes aux objets d'une même classe, d'où le nom de prototype (frame).
Un frame est constitué:

- D'un nom désignant l'objet à décrire,

- Des propriétés caractérisant l'objet auxquelles sont associées des mots clés appelés facettes et des valeurs. Ces propriétés sont nommées attributs.

Les chercheurs Winograd et Brobrow supposent qu'un certain nombre de schémas intellectuels sont préétablis dans notre esprit, et qu'extraire le sens d'une phrase revient à retrouver une situation analogue déjà existante. Comprendre une phrase reviendra donc à donner une valeur à certains attributs.

Par exemple, comprendre la phrase "mon grand père lave sa 2CV dans le jardin" revient à compléter les frames concernés.

Grand père:	**2CV:**
- une_sorte_de: humain	- <est une>: voiture
- sexe: valeur masculin	- couleur: défaut blanche
- âge: valeur vieux	- signification: valeur moyen de transport

Jardin:	**Laver:**
- une_sorte_de: espace	- sujet: valeur grand père
- composant: valeur (fleurs,..)	- objet: valeur 2CV
- signification: valeur lieu	- lieu: valeur jardin
	- instrument: défaut mains
	- temps: défaut aujourd'hui.

Une_sorte_de, est une, âge, sexe, couleur, etc.: attributs ;
Valeur, défaut: facettes ;
Masculin, vieux, etc.: valeurs.

Cette structure permet une description assez complète d'un domaine. Mais les principales difficultés proviennent du choix des frames décrivant le mieux le domaine considéré, d'une part, et des attributs associés à chaque frame d'autre part, car il est difficile de prédéfinir le nombre et le type des attributs dont on aura besoin pour décrire tous les liens qui apparaissent dans une phrase quelconque.

e)- Les graphes conceptuels [FAR 89]

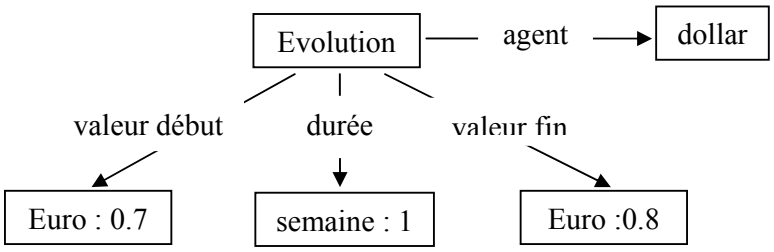
Cette nouvelle théorie, développée par John Sowa, est fondée sur la logique mathématique. Les graphes conceptuels offrent un cadre général permettant de coder les connaissances. Chaque élément de connaissance est représenté par un graphe, comportant des concepts et des relations entre ces concepts:

- Les sommets du graphe sont des concepts,

-Les arrêtes sont des relations conceptuelles.

Pour passer d'une phrase à un graphe conceptuel représentant son sens, on combine les graphes conceptuels de tous les éléments de la phrase. Pour cela, le programme d'analyse utilise un lexique sémantique. La construction du lexique sémantique est une tâche difficile, et sa qualité conditionne le résultat de l'analyse.

Exemple: La phrase "le dollar est passé de 0.7 euros à 0.8 euros en une semaine" est représentée par le graphe conceptuel suivant:



L'approche des graphes conceptuels constitue une composante importante de la représentation et du traitement de la connaissance, tant que le domaine est bien limité.

2 Linguistique de corpus

L'approche d'analyse de corpus établie en communauté scientifique est née d'un constat d'échec de la description complète de la langue par les grammaires génératives (règles) (si cela était on pourrait extraire les entités et les relations syntaxiques et sémantiques de n'importe quelle phrase). La langue évolue et transgresse les règles. Les études empiriques basées sur les corpus sont apparues au début des années 80 et sont maintenant couramment utilisées depuis 1985 en Intelligence Artificielle et dans les applications majeures du traitement du langage naturel: indexation (recherche documentaire) et traduction automatique.
Les études sur corpus, longtemps cantonnées au rang de simples outils descriptifs, connaissent un regain d'intérêt depuis quelques années, au sein de la communauté de concepteurs de systèmes automatiques.
L'ingénierie linguistique (TAL), de son coté, a toujours favorisé les études sur corpus pour l'élaboration de systèmes automatiques d'analyse linguistique. En effet, la linguistique de corpus fait appel systématiquement aux corpus électroniques pour développer, à partir des faits rassemblés, des dictionnaires et des grammaires descriptives, mais aussi pour tester des hypothèses, confronter un modèle postulé aux réalisations effectives [DAI 01] [HAB 97].
La recherche centrée sur la notion de corpus est basée sur l'idée suivante : ne pas s'intéresser à une description universelle des langues, ne pas chercher à caractériser complètement une langue, mais tirer profit des régularités/variations dans l'usage pour mettre en œuvre des traitements efficaces. Il s'agit donc d'approcher la description d'une langue sans idée préconçue, et se fonder sur l'observation des faits linguistiques rencontrés.

2.1 Corpus

Un corpus est une collection de données langagières qui sont sélectionnées et organisées selon des critères linguistiques explicites pour servir d'échantillon du langage [HAB 97]. Les corpus constituent une ressource importante, actuellement disponible sous forme électronique. Ils sont nécessaires pour des études descriptives, l'apprentissage statistique, l'extraction de terminologie, etc.
Aujourd'hui, on dispose de corpus nus (simples chaînes de caractères) et de corpus annotés d'information morphologique, syntaxique, sémantique, etc. Un corpus de textes où chaque mot est assorti d'une ou plusieurs étiquettes morphosyntaxiques est appelé corpus *étiqueté*. Par contre, un corpus muni d'arbres syntaxiques est appelé corpus *arboré* (c'est-à-dire décoré d'arbres).

Parmi les principaux corpus annotés, nous citons, par exemple, les corpus étiquetés suivants : Brown, mis au point en 1979 par W. Francis et H. Kucera, à l'université Brown, USA. Il comprend un million de mots relevant de genres de textes américains : reportages, publications scientifiques, etc.
Le corpus BNC (British National Corpus) de 100 millions de mots, mis au point en 1995. Le corpus Mittérand1, contenant 305124 occurrences et regroupant les interventions radio-télévisées du président français, F. Mitterand. Le corpus arboré Suzanne de 128000 occurrences, pris du corpus Brown, où chaque phrase est assortie d'un arbre syntaxique, mis au point en 1994.
Le corpus français partiellement arboré, Menelas de 84839 occurrences, mis au point en 1994 dans le cadre du projet européen, Menelas. D'autres corpus sont disponibles tels que les corpus *alignés*, où l'un des textes est la traduction de l'autre et où les segments de textes sont mis en correspondance.
En plus de la disponibilité de corpus annotés préalablement, on dispose aujourd'hui d'outils permettant de traiter de nouveaux textes et de constituer des corpus annotés : Etiqueteurs, analyseurs syntaxiques, etc.

2.2 Extraction de termes

L'extraction de termes consiste à obtenir automatiquement, à partir d'une collection de textes, un ensemble de termes simples ou complexes représentant ce corpus. Initialement, le corpus subit plusieurs prétraitements :

- Nettoyer le texte, qui consiste à lui ôter les parties inutiles : caractères de contrôle, figures, tableaux, équations, etc.

- Corriger les fautes éventuellement.

- Segmenter le texte, c'est-à-dire le découper en phrases et en unités lexicales. Ce traitement nécessite le repérage de séparateurs, tant pour trouver les fins de phrases que les limites des mots.

- Etiqueter le texte, c'est-à-dire associer aux mots des étiquettes (tags), plus fréquemment de type catégoriel, grammatical ou morpho-syntaxique.

Il existe trois types d'extracteurs : ceux basés sur des méthodes de surface, ceux basés sur des méthodes statistiques et ceux combinant les deux [DAI 01] [HAB 97]. Les extracteurs basés sur des méthodes de surface peuvent, par exemple, utiliser des patrons syntaxiques sur un corpus étiqueté pour repérer des termes de la forme : *Nom de Nom* (pomme de terre)*, Nom à verbe infinitif* (pince à épiler)*, Nom Adjectif, etc.*
Par contre, ceux basés sur des méthodes statistiques utilisent le repérage de co-occurrences de mots. Il s'agit de déplacer sur le corpus une fenêtre de taille fixe ou variable et de relever les fréquences de paires de mots et de calculer le lien unissant ces mots en utilisant des mesures telle que l'information mutuelle (chapitre 3).
Plusieurs systèmes combinent les deux types de méthodes : par exemple, le système Xtract [SMA 93] utilise une fenêtre de onze mots (cinq à gauche et cinq à droite du mot cible) et ne retient les termes extraits que s'ils sont dans une relation syntaxique correcte.

2.3 Etiquetage morphosyntaxique (tagger)

Etiqueter un texte, consiste à assigner à des segments de texte (souvent les mots), une ou plusieurs catégories grammaticales (tags ou étiquettes Part of Speech P0S) comme nom (NOMC, NOMP), verbe, adjectif, etc.
En général, les étiqueteurs sont aussi des lemmatiseurs : ils fournissent le lemme des mots. Les taggers sont très intéressants[4], leurs taux de bon étiquetage sont supérieurs à 95%. Ils sont utiles dans beaucoup d'applications comme l'extraction d'information, la réponse automatique à des questions, l'analyse partielle (shallow parsing), etc.
Le processus d'étiquetage se comporte comme une fonction *F* qui, à une séquence de *n* mots ($M = m_1, m_2, ..., m_n$) doit associer une suite ($C = c_1, c_2, ..., c_n$) de classes syntaxiques.

$$F : \quad M \rightarrow C = F(M)$$

Un exemple:

Phrase : « *Les difficultés financières et matérielles.* »

Etiquetage :	mot	Lemme	Catégorie morphosyntaxique
	Les	*le*	*DETDEF : déterminant défini*
	difficultés	*difficulté*	*NOMFP : nom féminin pluriel*
	financières	*financier*	*ADJFP : adjectif féminin pluriel*
	et	*et*	*CCOORD : conjonction de coordination*
	matérielles	*matériel*	*ADJFP : adjectif féminin pluriel*
	.	.	*PONCT : ponctuation*

Le processus d'étiquetage comporte trois étapes :

- La *segmentation* qui décompose un texte en unités lexicales.

- L'*analyse morphologique* qui associe chaque unité lexicale à toutes ses parties du discours.

- La *désambiguïsation* qui attribue à chaque unité lexicale une étiquette unique.

Il existe deux stratégies d'étiquetage : Un étiquetage *hors contexte*, où le codage est défini au niveau « langue », qui offre un grand nombre de possibilités (désambiguïsation se fait plus tard) et un étiquetage en *contexte* ou en *discours* qui affecte une étiquette selon l'emploi en contexte.
Le jeu d'étiquettes (le tag set) dépend de l'application et de la précision requise. Les étiqueteurs actuels fonctionnent soit avec un ensemble d'étiquettes très restreint (étiquetage minimaliste, environ 40 étiquettes), répondant aux besoins précis de l'utilisateur, soit avec un ensemble d'étiquettes très détaillé et très précis (étiquetage maximaliste, environ 400 étiquettes), voulant fournir des codes différents pour tous les mots ayant un comportement différent.
Les techniques utilisées pour réaliser un étiquetage morphosyntaxique sont l'étiquetage par consultation d'un dictionnaire, par analyse morphologique ou par combinaison des deux.

[4] : Ils sont plus facile à faire que d'analyser et comprendre une phrase. Par exemple, les étiquettes (POS) suffisent souvent à identifier des groupes syntaxiques simples comme les groupes nominaux.

2.3.1 Désambiguïsation morphosyntaxique

Un étiqueteur travaille généralement en deux étapes : une première listant pour chaque mot toutes ses étiquettes possibles (exemple : étiquetage à partir d'un lexique); une seconde consistant à désambiguïser, c'est-à-dire choisir une étiquette correcte parmi les étiquettes possibles.
L'opération de désambiguïsation se base sur les étiquettes des mots environnants, en utilisant une liste de règles (lexicales et contextuelles).
Les règles peuvent opérer sur les mots, les étiquettes ou sur les deux. Elles imposent des contraintes sur les mots qui précèdent ou qui suivent le mot à étiqueter.

Exemples :

Règle 1 : étiqueter *déterminant* le mot *de* si le mot qui le suit est un *nom*.

Règle 2 : Tout mot contigu linéairement à *une*, sur sa droite, est à étiqueter *nom* au singulier.

Règle 3 : si le mot *le* est suivi d'un verbe alors étiqueter le *pronom* et s'il est suivi d'un *nom* alors étiqueter le *déterminant*.

Règle 4 : *me* ou *te* sont suivis soit d'un *pronom* puis d'un *verbe* (*il me le garde*), soit directement d'un *verbe* (*me donner*).

Plusieurs étiqueteurs réalisent conjointement l'analyse morphologique des mots, c'est-à-dire qu'ils associent par exemple, outre l'étiquette de catégorie grammaticale, des informations flexionnelles sur le mot (genre, nombre, conjugaison, etc.) qu'ils ramènent à sa forme de base (infinitif pour un verbe, masculin singulier pour un nom ou adjectif) appelée *lemme*.

2.3.2 Quelques étiqueteurs catégoriels

De très nombreux étiqueteurs catégoriels sont développés, comme :

- Le système *CLAWS* de Church, où *l'*étiquetage est basé sur la probabilité maximale de la réalisation d'un évènement.

- Le système *Brill,* un des taggers le plus cité [BRI 92a] [BRI 95]. L'étiquetage utilise une base de connaissances (lexique, règles lexicales, bigrammes et règles contextuelles) construite par apprentissage. Il est réalisé en utilisant le lexique en choisissant l'étiquette la plus probable. Pour les mots inconnus du lexique, les règles lexicales sont utilisées pour attribuer une étiquette et les règles contextuelles pour affiner l'étiquetage. En effet, il transforme une séquence de tags (incorrecte) à l'aide d'un ensemble ordonné de règles transformationnelles qui permettent d'améliorer la séquence.

Exemple : Règle : nom (NN) ----> verbe (VB)
Contexte : le *tag* précédant est la préposition *TO*.
La règle dit: ré-étiqueter un nom en verbe (à l'infinitif) s'il est précédé de la préposition *TO*.

- L'étiqueteur statistique de Spriet et El-bèze*,* permettant d'attribuer une étiquette syntaxique à chaque élément composant une phrase. Il prend en entrée un texte et associe à chaque mot

l'étiquette la plus probable. Son fonctionnement repose sur les modèles *N-classes*[5]. Ces modèles sont basés sur l'étude de *N* mots consécutifs à l'aide de probabilité de mots sachant les classes *P(m/C)* (désigne la probabilité que le mot *m* appartienne à la classe *C*). L'étiquetage est assuré par l'algorithme de *Viterbi* qui, pour une phrase donnée, détermine la séquence d'étiquettes de probabilité maximale.

- Le système *INTEX*, utilise des dictionnaires à large couverture et des grammaires représentées par des graphes. Il permet d'étiqueter les mots, localiser des structures lexicales et syntaxiques et résoudre des ambiguïtés par automates (grammaires locales).

Les techniques statistiques et probabilistes deviennent de plus en plus utilisées dans de nombreuses applications du traitement du langage naturel [MER 95]. En effet, leur grand intérêt en traitement du langage naturel concerne la résolution des ambiguïtés. Elles fournissent une solution simple et immédiate pour résoudre les ambiguïtés (choisir l'hypothèse de plus forte probabilité). Elles ont connu un succès considérable dans le domaine des étiqueteurs automatiques [CHA 93] [MAN 99].

2.3.3 Autres types d'étiquetage

D'autres types d'étiquetage sont également possibles, comme l'étiquetage syntaxique et sémantique.
L'étiquetage syntaxique consiste à repérer les différents syntagmes dans chaque phrase. Il peut également tenter de repérer les rôles fonctionnels (sujet, complément d'objet direct, indirect, etc.) joués par les différents éléments de la phrase.
L'étiquetage sémantique consiste à associer aux mots des informations sémantiques pertinentes (exemple : associer *humain* à un nom, *verbe d'action* à un verbe, etc.). L'étiquetage sémantique est généralement basé sur les mêmes principes que les catégoriels, c'est-à-dire muni de lexiques contenant ces informations sémantiques. Il nécessite donc la détermination des classes sémantiques pertinentes pour chaque domaine visé.
Ces types d'étiquetage sont moins utilisés par leurs performances moindres et leur coût de mise en œuvre beaucoup plus élevé.

2.4 Analyse syntaxique

L'analyse syntaxique regroupe divers courants qui diffèrent par les objectifs visés et par les méthodes employées. Les méthodes couvrent par exemple les approches stochastiques, les approches par cascades (de phases d'analyse locale) et les approches plus traditionnelles d'analyse complète (ou profonde).
Les objectifs vont de la segmentation en syntagmes à l'analyse profonde avec une grammaire à large couverture, en passant par des analyseurs robustes et/ou superficiels.
Néanmoins cette diversité dans les méthodes et objectifs reflète une certaine complémentarité plutôt qu'une opposition absolue.
Il existe deux types d'analyseurs syntaxiques : des analyseurs *non-déterministes* qui génèrent plusieurs analyses par phrases et utilisent ensuite des statistiques ou des règles de grammaire

[5] Les modèles *N-classes* comptabilisent les fréquences d'apparition d'une suite de n classes syntaxiques. Les modèles *N-grammes* comptabilisent les fréquences d'apparition d'une suite de n mots. Les modèles *N-lemmes* comptabilisent les fréquences d'apparition d'une suite de n mots par l'intermédiaire de leurs racines.

basées sur des heuristiques afin de sélectionner l'analyse la plus probable parmi les réponses. Des analyseurs *déterministes* qui ne rendent qu'une seule analyse par phrase même si plusieurs sont possibles en termes réels. Les grammaires sont limitées dans ces cas-là et un certain taux d'erreur est accepté d'avance.

2.4.1 Ambiguïté syntaxique

Parmi les ambiguïtés syntaxiques, nous citons l'identification des *relations fonctionnelles* implicites entre les mots de la phrase (sujet de verbe, complément d'objet direct ou indirect de verbe, complément de nom ou d'adjectif, etc.), l'ambiguïté de *rattachement adjectival* (Nom1 Prép. Nom2 Adj. ----> rattacher l'adjectif au nom1 ou nom2 ?), l'ambiguïté de rattachement prépositionnel (Vb Dét Nom1 Adj. en Nom2 ----> rattacher la préposition *en* à *Adj*, à *Nom1* ou au *Vb.)*, etc.
Les méthodes utilisées pour résoudre les ambiguïtés de rattachement se basent éventuellement sur des ressources sémantiques externes et spécialisés (dictionnaires, thésaurus et ontologies, etc.) et généralement s'appuient sur l'analyse distributionnelle de corpus (apprentissage endogène).
Deux tendances principales se dégagent actuellement en linguistique informatique pour la résolution d'ambiguïtés syntaxiques en général: l'une basée sur l'utilisation de grammaires étendues (l'ambiguïté est résolue à l'aide d'heuristiques), l'autre sur des modèles statistiques (ambiguïté résolue par des probabilités estimées par des fréquences). Une troisième voie, hybride combine les deux approches [BER 97].

2.4.2 Les analyseurs syntaxiques

Il y a deux approches principales d'analyse syntaxique dans la recherche actuelle : statistique et symbolique.

a)- Les analyseurs statistiques

L'analyse consiste à supprimer les mots vides (et, ou, un, les, etc.) et ensuite à calculer les fréquences d'apparition de mots ou de groupes de mots. Certains mots et certaines associations de mots apparaissent plus fréquemment que d'autres. Cela indique des structures associatives privilégiées et donne des indications sur les contextes de leur apparition.
En général, les analyseurs statistiques existant génèrent toutes les analyses possibles pour chaque phrase d'un texte (analyse non déterministe) et utilisent ensuite des fréquences d'occurrences des séquences de mots afin de sélectionner l'analyse la plus probable parmi celles proposées.
Actuellement, la recherche dans les analyseurs statistiques qui déduisent automatiquement les grammaires suscite beaucoup d'intérêt. Charniak [CHA 97] décrit un analyseur statistique qui infère une grammaire à partir d'un corpus textuel syntaxiquement annoté ("Penn Treebank "). Cependant, la qualité de la grammaire extraite par ce type d'analyseur dépend énormément de la qualité et de la taille du corpus annoté utilisé. Or, il existe très peu de corpus de haute qualité et leur taille quoique importante (Le Penn Treebank contient un million de mots annotés manuellement) est très limitée en comparaison avec la quantité et la diversité des textes électroniques qui existent.

b)- Les analyseurs symboliques

Il y a deux types principaux d'analyseurs syntaxiques symboliques : ceux basés sur la grammaire syntagmatique et ceux basés sur la grammaire des dépendances.

- Dans le cas de la grammaire syntagmatique, l'ordre des mots est explicitement encodé dans les règles, et les relations fonctionnelles, telles qu'entre le sujet et l'objet du verbe, sont implicites.
Par exemple, la règle (SN ----> Det, Adj., N.) indique qu'un syntagme nominal est composé d'un déterminant suivi d'un adjectif, suivi d'un nom.
Un exemple de ce type d'analyseurs, le système Fidditch [HIN 83] : un analyseur robuste, accepte du texte non-annoté à l'entrée et sort des arbres annotés. Il contient plusieurs modules de prétraitement tel qu'un analyseur orthographique et un analyseur morphologique. L'analyseur orthographique consiste à lire le texte, et de le découper en mots. L'analyseur morphologique associe une catégorie lexicale à chaque mot ainsi que des propriétés. Par exemple, il peut associer la catégorie lexicale "nom" au mot "chien" ainsi que la propriété "singulière". Le module syntaxique dans Fidditch lit trois mots annotés à la fois et vérifie s'ils correspondent aux règles de groupes syntagmatiques qui forment la grammaire du parseur. Il identifie des groupes nominaux et verbaux, et essaie de lier les constituants entre eux.

- En ce qui concerne la grammaire de dépendance, l'accent est sur la sémantique et non pas sur la syntaxe. Elle décrit la structure syntaxique en terme de relations binaires entre des paires de mots dans une même phrase. Par exemple, dans l'arbre de dépendance ci-dessous, aucun ordre n'est imposé – elle représente aussi bien "Le grand chien chasse le chat noir " que "Le chat noir chasse le grand chien". Ce sont les relations fonctionnelles, explicitement encodées dans l'arbre, qui indiquent l'ordre des mots selon la langue en question. Donc, "chien" est le sujet du verbe "chasser" et "chat " est l'objet direct du verbe.

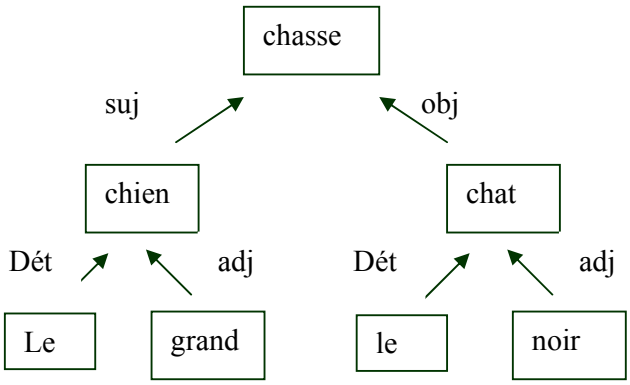


Figure I.2 : Arbre de dépendances

Un certain nombre d'analyseurs de dépendances, très performants tels que **FDG** (**F**unctional **D**ependency **G**rammar soit Grammaire de Dépendances Fonctionnelles) a été développé à l'université de Helsinki [TAP 97]. FDG associe des étiquettes aux mots dans le texte et crée des liens binaires entre la tête d'un groupe de mots et ses dépendants. Par exemple, dans le groupe nominal "Le grand chien", "chien" est la tête et deux liens binaires sont crées entre le déterminant "Le" et "chien" et entre l'adjectif "grand " et "chien". De plus, FDG extrait des relations fonctionnelles telles que le sujet et l'objet donc il identifie "chien" comme le sujet de la phrase et "chat" comme l'objet.

c)- Les analyseurs hybrides

La quête pour un analyseur syntaxique robuste capable d'analyser des textes libres en profondeur, a conduit au développement de toute une gamme d'analyseurs syntaxiques ces dernières années.
Ces analyseurs varient en termes de stratégie (le déterminisme contre le non déterminisme, l'analyse partielle contre l'analyse complète) en ce qui concerne leur base théorique (les analyseurs statistiques contre les analyseurs symboliques et les analyseurs basés sur la grammaire syntagmatique contre ceux basés sur la grammaire de dépendances, etc.). Cependant, alors que chaque analyseur a ses points forts et fonctionne bien pour certaines tâches, aucun d'entre eux n'est capable de faire une analyse complète de toutes les phrases dans un corpus de texte libre.
Les méthodes statistiques, quoique prometteuses, ne suffisent pas, à elles seules, pour régler toutes les tâches du traitement automatique de la langue. Les grammaires symboliques sont également nécessaires afin d'obtenir une représentation précise et fiable de la sémantique, ce qui est crucial pour beaucoup de tâches de traitement automatique des langues.
Récemment, il y a une tendance croissante à mélanger les approches différentes envers l'analyse syntaxique afin de profiter des points forts des différents types d'analyseur. Xerox, par exemple, a développé un analyseur syntaxique robuste hybride qui marie les structures syntagmatiques avec les structures de dépendances [AIT 01] et Collins a développé un analyseur qui améliore la qualité de l'analyse en utilisant une grammaire symbolique en conjonction avec des statistiques [COL 96].

Les systèmes hybrides de TAL développés jusqu'à présent peuvent être regroupés suivant deux stratégies principales : une hybridation faible ou une hybridation forte.
Une hybridation faible correspond à un traitement associant simplement deux modules : symbolique et statistique. En règle générale, un des modules est sensé compenser ou corriger les insuffisances de l'autre (les sorties de l'un sont les entrées de l'autre). Par exemple, le module symbolique peut corriger les erreurs ou filtrer les sorties issues du module statistique. Dans l'approche par correction, le module symbolique, à l'aide de règles hors contexte, corrige les erreurs produites par l'étiquetage probabiliste, par exemple le tagger hybride développé par le LIA [SPR 98].
Par contre, dans l'approche par filtrage, le module statistique propose ses meilleures hypothèses une à une et de manière décroissante, jusqu'à ce qu'une analyse soit acceptée par le module symbolique [CHO 89]. Cette stratégie s'avère très efficace dans certains cas, mais non optimale. En effet, elle conduit à un cumul des erreurs produites par les deux modules.
Une hybridation forte consiste à intégrer dans un seul processus des approches symbolique et statistique. On distingue deux types d'intégration:

- Intégration de probabilités dans un processus symbolique (grammaires stochastiques) : Elle se résume à l'ajout de probabilités de déclenchement associées aux règles de la grammaire. Ces probabilités sont estimées sur des corpus de phrases étiquetées.
Les grammaires probabilistes fournissent en sortie un ou plusieurs arbres de dérivation classés par probabilités décroissantes. Nous citons par exemple, les grammaires hors contexte stochastiques [BOO 73], les grammaires d'arbres adjoints [SCH 92] ainsi que les grammaires de liens [LAF 92]. D'une manière générale, ce type de grammaires permet d'améliorer la robustesse d'analyse en autorisant une dégradation douce des performances [ANT 99].

- Intégration de contraintes symboliques dans un processus stochastique: Elle consiste à intégrer dès que possible un ensemble de contraintes symboliques dans l'analyse stochastique.
Ces exemples d'intégration constituent à n'en pas douter une voie de recherche prometteuse pour la mise en œuvre d'un TAL à la fois plus précis et plus robuste.

2.5 Analyse sémantique

Dans le domaine de l'analyse et du traitement de l'information (basé contenu), l'idée de base est l'exploitation de toutes les sources de connaissance et d'information dans le processus de compréhension du langage naturel, stockées sous forme électronique. Dans ce paragraphe, sont présentées seulement les différentes ressources généralement nécessaires et utilisées dans l'analyse sémantique.

2.5.1 Ressources lexicales

L'augmentation des performances des systèmes automatiques passe par le traitement du phénomène d'équivalence sémantique. En effet, un même contenu peut être exprimé de manière différente, dans différentes configurations syntaxiques, avec différents mots. Le sens d'un mot ou d'une phrase est en général flou, complexe, ambigu, dépendant du contexte et des connaissances [SEB 99]**.** La démarche la plus souvent adoptée consiste à recourir à une base de connaissances linguistiques regroupant les mots sémantiquement proches, et structurée selon des relations hyperonymiques et/ou synonymiques.
Le coût de construction de telles ressources amène à plaider pour l'utilisation de ressources générales telle que Wordnet, Eurowordnet, EDR, MikroKosmos, Cyc, etc. Le développement de ces ressources à grande échelle prouve qu'il n'est pas indispensable de maîtriser le problème de compréhension dans sa totalité pour apporter des solutions partielles qui s'avèrent utiles.

a)- Bases de connaissances lexicales

Bases contenant les mots d'une langue donnée décrits dans leurs différents sens, leurs relations et leurs emplois. On distingue les dictionnaires, les thésaurus et les terminologies.

- WORDNET : Wordnet est une base de données lexicographique pour la langue anglaise, accessible gratuitement en ligne sur Internet, pourvue de moyens de recherche évolués (recherche par noms, verbes, adjectifs, même sens, sens opposé, etc.). C'est le réseau sémantique le plus utilisé. Il est développé par CSL (Cognitive Science Laboratory) à l'université de Princeton. Il est construit autour de la notion de synogroupe (sunset). Un synogroupe est un ensemble de sens de mots entre lesquels existent des relations sémantiques de base telles que l'hyponymie (voiture-véhicule), la métonymie (avion-aile) ou la relation de cause à effet (tuer-mourir).
Wordnet distingue quatre catégories de mots : *nom, verbe, adjectif* et *adverbe*. Il est structuré en hiérarchies de *sunsets* propres à chaque catégorie. Il est plus thésaurus que dictionnaire, car il décrit comment les sens des mots (concepts) s'organisent les uns par rapport aux autres et non les mots eux même. Un sens se définit par la place qu'il occupe dans le réseau et par les relations de proximité ou de contraste qu'il entretient avec les sens voisins.

- EUROWORDNET [VOS 98] : Eurowordnet est une base de données multilingue organisée autour de réseaux de mots pour les langues européennes. Chacun de ces réseaux est construit de la même façon que les réseaux de mots de wordnet.

Le gain apporté par le recours à de telles ressources n'a jusqu'à présent pas été démontré. L'hypothèse d'une ressource lexicale générale valable hors contexte explique essentiellement les limites de cette approche. En effet, l'utilisation systématique de ces ressources tel que Wordnet dans des domaines particuliers ne permet pas de représenter les relations de proximité sémantique.
De nombreux travaux ont montré que la définition des relations de proximité sémantique ne peut pas être menée hors domaine mais doit au contraire s'appuyer sur les caractéristiques du corpus de travail [SEB 99]. De plus, les ressources lexicales traditionnelles utilisent seulement les relations d'hyperonymie et de synonymie des groupes nominaux pour capter le contenu sémantique des textes et l'apport des autres informations linguistiques n'est pas pris en compte (l'apport sémantique du verbe, etc.).

b)- Bases de connaissances conceptuelles

Alors que les bases de connaissances lexicales structurent l'espace des mots, les bases de connaissances conceptuelles reflètent une conceptualisation du monde en représentant les concepts (classes d'objets) d'un domaine donné, leurs propriétés ainsi que les relations qu'ils entretiennent entre eux. On distingue les réseaux sémantiques et les ontologies.

- Ontologies : une ontologie est une représentation du monde en concepts (catégories ou classes d'objets) organisées en hiérarchie par des liens *IS-A (SORTE-DE)*.

Exemple : Une hiérarchie de termes d'un domaine donné.
Les méthodes de construction automatique d'ontologies se basent sur l'analyse de corpus. En effet, les concepts et les relations entre concepts sont issus d'une analyse des textes. En général, l'analyse consiste en :

- L'extraction des termes ainsi que les relations lexicales qu'ils entretiennent,

- Une analyse sémantique entre termes pour construire un réseau sémantique de concepts,

- Validation du réseau par un expert du domaine.

Cette approche basée corpus s'appuie sur des outils de traitement automatique du langage naturel et impose que le corpus soit de bonne qualité syntaxique et lexicale.

- Réseaux sémantiques : un réseau sémantique ou conceptuel est un graphe où les nœuds représentent les concepts et les arcs représentent les relations entre concepts (relations de causalité, d'appartenance, etc.).

2.5.2 Relations sémantiques

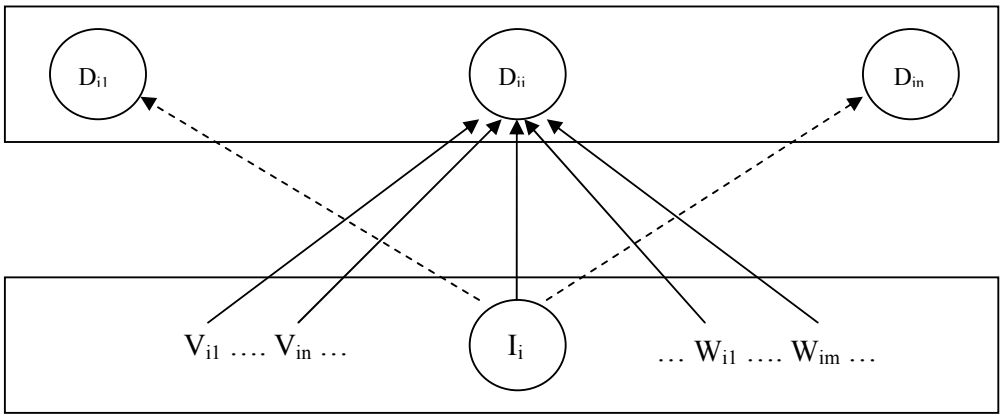
Certains travaux sur la reconnaissance de relations sémantiques entre termes dans des corpus, se basent sur une étude statistique, en étudiant leurs contextes de cooccurrence, par exemple la fréquence de cooccurrences [SMA 93]. D'autres se basent sur une étude linguistique (par exemple, une étude de contextes lexico-syntaxiques) [HAB 96].

2.6 Analyse pragmatique

L'analyse pragmatique est une analyse contextuelle permettant de calculer et d'identifier des valeurs sémantiques. Par exemple, déterminer une valeur sémantique sous forme d'une étiquette sémantique attribuée à un segment linguistique (syntagme, phrase, paragraphe, etc.). Il ne s'agit pas d'une utilisation de mots clés ou d'une simple analyse distributionnelle puisqu'elle met en jeu des processus inférentiels qui sont déclenchés par l'identification d'indicateurs linguistiques.
De plus, à l'identification d'une occurrence d'un indicateur, une exploration du contexte de cette occurrence est nécessaire pour rechercher d'autres indices linguistiques (indices complémentaires), qui viendront soit lever l'ambiguïté sémantique (invalider les hypothèses), soit lever l'indétermination sémantique (attribuer une valeur sémantique).
En effet, un indicateur linguistique est rarement un marqueur univoque d'une valeur sémantique unique. Le rapport entre signifiants (forme) et signifiés (sens) n'est pas bijectif dans les langues. La plupart des marqueurs sont multivoques et polysémiques.
D'une façon générale, un système d'analyse contextuelle va en fonction de la tâche et du problème à résoudre. Il se compose de :

- Un ensemble d'indicateurs linguistiques, jugés pertinents pour la tâche;

- Un ensemble de règles (dites contextuelles) qui, pour un indicateur donné, recherchent d'autres indices explicites dans un espace de recherche (proposition, phrase, etc.). Ces indices contextuels permettent donc de lever l'indétermination et choisir la décision qui convient, en éliminant les autres décisions (figure I.3).



I_i : Indicateur linguistique i.
V_{i1}…V_{in} : Indices linguistiques avant I_i.
W_{i1}…W_{im} : Indices linguistiques après I_i.
D_{i1}…D_{in} : Les différentes décisions possibles pour I_i.

Figure I.3 : Système d'analyse contextuelle

Exemple : Soit la proposition suivante : *... le lendemain, il démissionnait…*

En dehors de tout contexte, à cette proposition, identifiée comme indicateur pertinent, sont associées deux valeurs référentielles a priori possibles (deux décisions contradictoires) :

- soit [Il a effectivement démissionné]

- soit [Il n'a pas démissionné].

Selon les contextes, nous pouvons identifier des indices linguistiques complémentaires qui contribuent à lever l'indétermination.

Contexte 1 : *De nombreuse voix arrivèrent très rapidement pour soutenir sa proposition. Pourtant, le lendemain, il démissionnait…*
Le terme « Pourtant » constitue un indice pour affirmer qu'il a effectivement démissionné.

Contexte 2 : *Sans les nombreuses voix qui arrivèrent très rapidement pour soutenir sa proposition, le lendemain, il démissionnait…*
Le terme « Sans » constitue un indice pour affirmer qu'il n'a pas démissionné.

Parmi les applications qui utilisent ce type d'analyse, nous citions les systèmes de filtrage sémantique de textes (SAFIR, Coatis, SECAT, SEEK, SERAPHIN, FilText, ConTexto, etc.) (Crispino et al., 1999).
Cette méthode d'analyse contextuelle nécessite des compétences de linguistes pour l'acquisition des bases de connaissances linguistiques (les déclencheurs, indices complémentaires et règles contextuelles).

3 Analyse automatique de textes

En informatique, analyser un texte en langage naturel (*comprendre)* consiste à transformer celui-ci en une représentation formelle manipulable par programme : c'est-à-dire, passer d'une *forme* écrite à une *représentation conceptuelle.*
La construction de cette représentation passe par l'identification des parties du texte (éléments d'information) et l'étude des relations qu'elles entretiennent.

Le processus d'analyse d'un texte est définit par les étapes suivantes :

- le découpage du texte en éléments essentiels,

- la détermination des rapports entre ces éléments,

- produire un schéma (une représentation) de l'ensemble,

- et déduire de nouvelles informations des éléments identifiés.

Comprendre un texte revient donc à combiner le sens des unités plus petites qu'il comporte. Le but d'une analyse linguistique est de montrer comment des unités de sens plus larges (phrase, texte, etc.) résultent de la combinaison d'unités de sens plus petites. Ceci est modélisé par une grammaire.
Qu'est-il possible de faire en matière d'analyse automatique de texte libre ? L'objectif est en fait de savoir de quoi parle un texte. C'est à cette question que veut répondre toute analyse du contenu d'un texte, quelle soit humaine ou automatique. Le but est de dégager une forme exploitable ultérieurement. Il s'agit donc d'extraire de l'information spécifique et pertinente d'un texte afin de fournir une information élaborée et synthétique. Reste à déterminer comment et par quel moyen on pourra récupérer d'un texte l'information pertinente qui le caractérise.

3.1 Exemples d'applications

Les applications possibles de l'analyse de texte sont nombreuses:

- la recherche et l'extraction d'information dans des grosses bases de données;

- la classification automatique de documents;

- l'extraction de terminologie

- le résumé automatique de texte;

- la traduction automatique;

- les interfaces homme/machine en langage naturel.

- etc.

3.2 Approches d'analyse

Différents paradigmes de systèmes d'analyse des textes se sont succédés ces dernières années. Les approches oscillaient entre des analyses globales portant sur le texte dans son intégralité (les systèmes de compréhension des années 80) et des analyses beaucoup plus locales, répondant à des besoins particuliers (les systèmes d'extraction d'information du début des années 90) [POI 99].

3.2.1 Analyse globale

L'approche est très ambitieuse : La compréhension repose sur une analyse linguistique visant à décrire le texte dans son ensemble (comprendre l'ensemble du texte). Cette approche vise une analyse profonde et exhaustive du texte. Elle est vue comme un traducteur qui transforme le texte (structure linéaire) en une représentation interne et intermédiaire (une représentation sémantique profonde), laquelle est ensuite utilisée pour faire des inférences (ex : répondre à des questions). L'objectif de l'analyse est de donner du texte une représentation sémantique profonde. Beaucoup de systèmes ont été développés utilisant cette approche (ACORD, LILOG, KALIPSOS, TACITUS, etc.) [POI 99]. C'est des systèmes génériques, constitués d'une grammaire et d'un analyseur à large couverture. L'analyse sémantique repose sur les indices linguistiques et est guidée par le domaine et l'application.
Ces premiers systèmes génériques (adaptés à la tâche d'extraction d'information) ont rapidement montré leurs limites par rapport à des systèmes dédiés. En effet, cette approche générique est très coûteuse en temps (construction de ressources lexicales, etc.) et de plus a révélé ses limites en termes d'adaptation et de portabilité d'un domaine et d'une tâche à l'autre. En effet, l'adaptation d'un système à une nouvelle tâche et à un nouveau domaine nécessite de reconstruire une grande partie de la connaissance (lexique sémantique, etc.).

3.2.2 Analyse locale

Dans le domaine de la compréhension automatique des textes, une autre approche apparaît qui est radicalement différente de l'approche générique. Elle repose sur une analyse purement locale. L'objectif désormais n'est plus une analyse extensive, seule une partie minime du texte nécessite une analyse approfondie. Il s'agit d'analyser la totalité du texte libre dans le but d'extraire seulement les parties qui contiennent de l'information pertinente.
La pertinence est déterminée par des directives prédéfinies qui doivent préciser, avec le plus d'exactitude possible, quel type d'information le système doit trouver. C'est une analyse guidée par le but (connaissance à priori des informations recherchées). Il s'agit de repérer au travers d'indices textuels de surface (patrons ou schémas de phrases) certains schémas informationnels (passages pertinents).

L'analyse locale se déroule comme suit :

- En premier, rechercher des amorces dans le texte (définir à priori un ensemble d'amorces pour chaque patron).

- Lancer une analyse syntaxique sommaire sur le contexte des amorces trouvées. Généralement, cette analyse repose sur des automates à états finis décrivant des syntagmes nominaux, verbaux, etc.

- Ensuite, la reconnaissance des patrons est effectuée au moyen d'automates dont les transitions peuvent être des unités lexicales, des syntagmes, etc.

- Instancier les différents éléments de chaque patron reconnu.

Avec les conférences MUC[6], le domaine de recherche sur la compréhension de textes a glissé vers la compréhension de messages, puis vers l'extraction d'information. L'ambition est plus réduite :

- Les textes sont moins complexes : le texte est court et informatif.

- La tâche clairement définie : Il s'agit d'analyser le texte pour remplir un formulaire prédéfini (template).

De nombreux systèmes ont été développés utilisant des analyses linguistiques pour extraire de façon automatique de l'information. Parmi lesquels, nous pouvons citer le système FASTUS, développé suite à l'abandon du système générique TACITUS, qui utilise une technique d'analyse de surface rapide basée sur les FST (Finite State Transducer), ressemble beaucoup à la technique de « shallow parsing », qui applique des « patterns » pour détecter des scénarios. Un autre système LASIE (system Large Scale Information Extraction) qui utilise une analyse plus profonde que FASTUS.
Les méthodes locales ont été utilisées principalement sur des domaines de connaissances limités. Elles ont obtenu de bons résultats. Les dernières campagnes d'évaluation MUC ont révélé des limitations de ces méthodes. En effet, le coût de l'adaptation de l'analyse à de nouveaux domaines, à de nouveaux types de textes ou à de nouveaux besoins est prohibitif [MUC 7]. Pour chaque application et pour chaque nouveau domaine, il faut élaborer des nouveaux patrons spécifiques et construire les automates correspondants. De plus, l'identification des bons indices de surface est une tâche difficile.
Deux principales familles de méthodes sont actuellement utilisées. La première est basée sur une approche linguistique, elle est en général complexe et dédiée à des tâches très spécifiques. La deuxième est basée sur une approche statistique en combinant parfois certaines méthodes linguistiques et des approches pragmatiques.
Les méthodes locales restent encore largement le domaine de l'analyse linguistique. Ce n'est que très récemment que l'on a vu apparaître dans ce domaine une approche utilisant des méthodes issues de l'apprentissage. Par exemple, définir un modèle stochastique tel que le modèle de Markov caché (MMC) et utiliser un algorithme d'apprentissage tel que celui de Viterbi. Cette nouvelle approche vise au développement de systèmes rapidement portables, capables de traiter de gros volumes de textes et capables d'analyser des textes peu grammaticaux et bruités comme par exemple les messages provenant d'Internet [ZAR 98].

3.2.3 Une analyse de surface ou partielle (Shallow parsing)

Il existe deux grands courants dans l'analyse linguistique. Le premier effectue une analyse profonde et complète en déterminant des liens complexes entre presque toutes les entités de l'énoncé. Le deuxième courant utilise une analyse linguistique partielle (de surface, « shallow parsing ») : on ne cherche plus à produire une analyse complète de l'énoncé, mais à n'en délimiter que certains constituants [PIL 00]. En effet, pour certaines tâches, une analyse

[6] Message Understanding Conferences. http://www.muc.saic.com.

complète n'est pas nécessaire ou bien elle est parfois impossible: contrainte de temps réel, limitation du parser pour une plus grande robustesse, etc.
De plus, vu la versatilité des langues, l'état actuel des connaissances ne permet pas de développer des analyseurs robustes (écrire une grammaire complète) capables de traiter (comprendre) toutes les phrases imaginables dans une langue.
La difficulté des analyses de phrases tout venant parfois complexes, voire agrammaticales, conduit souvent à adopter une stratégie d'analyse partielle pour analyser des textes libres.
L'analyse partielle est une analyse grossière (légère) dont le rôle est de déterminer les acteurs d'une phrase (sujet, verbe, objet) avec leurs dépendances. Ce type d'analyse est très répandu pour identifier des couples sujet-verbe et verbe-objet, sans analyse sémantique profonde. Par contre, la détection des dépendances est de 80% (précision/rappel) si la distance est inférieur à six mots du morceau de phrase source (chunk : GV ou GN) [TUR 00].
La structure syntaxique est représentée par un arbre. Les arbres présentent des difficultés à représenter des structures discontinues (ne..pas). Par conséquent, il est difficile de constituer des corpus arborés. Une solution est l'analyse partielle : Contrairement à l'analyse complète, qui à une phrase correspond un arbre qui couvre tous les mots de la phrase (feuilles de l'arbre), l'analyse partielle : à une phrase donnée correspond (ent) un ou plusieurs arbres qui laissent des parties non analysées. Elle exploite les indices linguistiques : étiqueter les mots avec leur catégories morphosyntaxiques, définir la structure d'un syntagme nominal et d'un syntagme verbal et définir les marques définissant une frontière de syntagme.
L'analyse partielle a pour but de produire une version simplifiée de la phrase, en ignorant les parties secondaires [HAB 97].
Le chunking est un prétraitement qui intervient généralement en sortie de l'étiquetage grammatical (figure I.4).

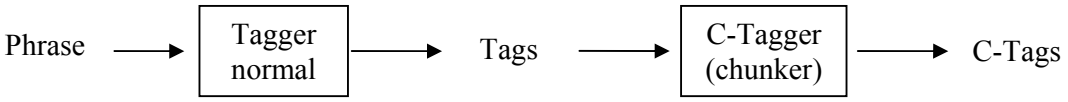


Figure I.4 : Chunker classique

Une autre façon possible, est de ne pas les séparer : le chunker est un tagger entraîné directement avec le jeu d'étiquettes des C-tags.

Suivant l'analyseur utilisé en sortie, il a pour rôle:
- soit de fournir une segmentation partielle en constituants non récursifs (les *chunks*). C'est à dire qu'on cherche à identifier l'ensemble des constituants (GN, GP, PP, etc.) de l'énoncé sans décrire leurs relations de dépendances.

- soit d'identifier les principales relations lexicales dominant/arguments de l'énoncé. Il s'agit d'une analyse de surface dont l'objectif n'est pas de construire un arbre de dérivation décrivant la structure détaillée de l'énoncé. Elle a donné lieu à des réalisations relativement robustes, à la fois par des approches symboliques (à base d'automates d'états finis essentiellement) [ANT 99] [KOS 91] [CHA 94] et des approches statistiques [CHU 88].
Il utilise un jeu réduit d'étiquettes (C-tag) permettant de caractériser les groupes (adjectivaux, adverbiaux, verbaux, nominaux, etc.). Les deux étiquettes les plus fréquentes sont I-Groupe (ex : *I-NP, I-VP, etc.*) et B-Groupe (ex : *B-NP, B-VP, etc.*), marquant respectivement le milieu d'un groupe et son début.

Exemple:

Mot	He	reckons	the	current	account	deficit	will	narrow	...
Tag	PRP	VBZ	DT	JJ	NN	NN	MD	VB	…
C-tag	B-NP	B-VP	B-NP	I-NP	I-NP	I-NP	B-VP	I-VP	…

Les analyses partielles ont fait preuve d'efficacité en traitement automatique du langage naturel : L'analyse syntaxique [ABN 96a] [ABN 96b], les études sur corpus, la terminologie, etc. Toutefois, elles sont rarement transportables d'une application à une autre.
Pour l'analyseur de Steven Abney[7] nommé CASS (Cascaded Analysis of Syntactic Structure) [ABN 90] [ABN 91], l'idée est simple : plutôt que de générer de multiples analyses pour chaque phrase qui est structurellement ambiguë, on ne génère qu'une seule analyse. Autrement dit, lorsque l'on ne peut pas déterminer avec certitude, si un constituant s'attache à un syntagme ou un autre, on ne l'attache à aucun des deux.
L'analyseur syntaxique partiel IFSP (Incremental Finite State Parsing) de XEROX permet d'annoter les groupes syntagmatiques noyau (chunks), puis d'extraire les relations fonctionnelles entre les mots (sujet, objet direct, etc.). Le texte est préalablement segmenté et étiqueté avec un étiqueteur morphosyntaxique : à chaque mot est associé une et une seule étiquette morphosyntaxique : la catégorie + traits morphologiques (le nombre, la personne, le mode de conjugaison). Ensuite, des transformations (modèles décrits sur les suites d'étiquettes morphosyntaxiques) s'opèrent pour l'annotation des groupes noyau qui sont les sous parties des syntagmes classiques (SN, SV, SP, etc.) délimités par l'élément tête du syntagme. Par exemple, la suite « le petit oiseau sur l'arbre » sera analysée comme deux groupes distincts: groupe nominal (NP) [le petit oiseau] et le groupe prépositionnel (PP) [sur l'arbre].
Enfin, l'analyseur IFSP assigne des étiquettes de fonctions syntaxiques principales (sujet, objet, etc.) et, en fonction des patterns d'extraction spécifiés sur la structure des chunks (règles de description sous forme d'expressions régulières), extrait des relations syntaxiques entre les mots (sujet passif, objet direct, objet indirect, etc.).

3.3 Typologie des textes [PAU 95]

Les multiples classements de textes (les genres de textes) existant aujourd'hui restent divergents et partiels, et aucun d'entre eux ne peut prétendre constituer un modèle de référence stabilisé et cohérent.
Cette difficulté de classement tient d'abord à la diversité des critères qui peuvent légitimement être utilisés pour définir un genre : critères ayant trait au type d'activité humaine impliquée (genre littéraire, scientifique, journalistique, etc.) ; critères centrés sur l'effet communicatif visé (genres épique, poétique, lyrique, mimétique, etc.) ; critères ayant trait à la taille et/ou la nature du support utilisé (roman, nouvelle, article de quotidien, reportage, etc.) ; critères ayant trait au contenu thématique évoqué (science fiction, roman policier, recettes de cuisine, etc.), etc.
Cette difficulté découle aussi du caractère fondamentalement historique et adaptatif des productions textuelles.
En effet, les genres sont en perpétuel mouvement : certains genres tendent à disparaître ; certains se modifient ; de nouveaux genres apparaissent ; etc.

[7] Abney est considéré comme étant le père du chunking: il recherchait des corrélations entre les tags pour identifier des groupes.

Il résulte enfin de cette mobilité que les frontières entre genres ne peuvent pas toujours être clairement établies : il existe des genres clairement définissables et étiquetables (textes plus ou moins stabilisés) et des genres pour lesquels les définitions et les critères de classement restent mobiles et/ou divergents (textes aux contours flous et en intersection partielle).
Le critère sans doute le plus objectif qui pourrait être utilisé pour identifier et classer les genres est celui des unités et des règles linguistiques spécifiques qu'ils mobilisent. L'application de ce critère a été proposée par de nombreux auteurs. En effet, le langage utilisé dans les textes a fait l'objet de nombreuses études linguistiques, en particulier les travaux de Douglas Biber [BIB 95] [BIB 93] [BIB 88], Bronckart [BRO 85], Habert [HAB 01], et de Bergounioux [HAB 00a].
Les textes sont souvent considérés en fonction de leur caractère fonctionnel ou situationnel (mode de production des textes : formel ou informel, interactif ou non interactif, littéraires ou parlés, conversation ou exposé, commun ou spécialisé, etc.). La typologie des textes a motivé de nombreuses recherches en linguistique et en analyse de discours. L'objectif est d'élaborer des modèles qui permettent de décrire la typologie des textes en se basant sur l'étude des différents traits linguistiques liés au type du texte. Le présupposé de ces modèles est que les patterns linguistiques dans les textes ne sont pas aléatoires, mais sont la trace de dimensions fonctionnelles. Chaque type de textes se caractérise donc par l'association d'un certain nombre de caractéristiques linguistiques.
La démarche typologique habituelle démarre souvent d'une classification à priori (situationnelle ou fonctionnelle), examine les textes de chaque type et leur fonctionnement linguistique, et essaie de mettre en évidence certaines corrélations entre types et traits linguistiques [HAB 97]. Elle repose sur les buts (ou fonctions) visés par les textes (informer, convaincre, distraire, etc.). Les textes se distingueraient par la domination de telle ou telle fonction, même si chacun peut faire appel à toutes les fonctions. Plusieurs chercheurs ont proposé un certain nombre de fonctions permettant de distinguer les textes. Par exemple, Condillac distinguait le didactique, le normatif et le descriptif [HAB 00a]. Werlich distinguait le descriptif, le narratif, l'expositif, l'argumentatif et l'instructif [HAB 00a]. Adam proposait 7 types textuels de base [HAB 00a] [ADA 85], Jackobson proposait 6 types [HAB 00a], etc.
Une autre démarche opposée procède à posteriori : elle consiste à développer une typologie inductive des textes : Il s'agit de faire émerger les types de textes grâce à un traitement statistique de textes (corrélations effectives entre traits linguistiques).
Cette démarche peut avoir pour objectif de confirmer ou amender une typologie préexistante (la logique de Bronckart), ou au contraire être purement exploratoire visant à révéler des types sans correspondance obligatoire avec des types répertoriés (la logique de Biber).
Quelque soit la démarche, un texte est défini par la cooccurrence d'un certain nombre de traits linguistiques et éventuellement de l'absence systématique d'autres traits. Un corpus est constitué pour examiner la répartition de traits considérés (préalablement ou à posteriori) comme discriminants et significatifs.

3.3.1 Approche empirique de D. Biber

D'après Biber [BIB 95] [BIB 93] [BIB 88], la caractérisation des textes en types est le résultat de l'analyse quantitative de la distribution d'un certain nombre de traits grammaticaux ou lexicaux associés à des paramètres discursifs :

- des marques de temps et d'aspect,

- les pronoms du discours,

- les formes interrogatives,

- les passifs,

- différents types de subordination,

- etc.

L'objectif des travaux de Biber est de dégager des corrélations de traits linguistiques, afin d'aboutir à des types de textes. L'analyse des taux de fréquence des cooccurrences des traits linguistiques permet de caractériser les textes.
Après examen des cooccurrences entre 67 traits linguistiques fixés par Biber, dans les textes d'anglais britannique contemporain écrit et oral, ils ressortissent à 16 catégories distinctes comme marqueurs de temps et d'aspect, adverbes de temps et de lieu, pronoms, questions, coordination, négation, etc. [ILL 99]. L'étiquetage mis en œuvre par Biber est donc partiel. Il ne s'intéresse qu'à des fonctionnements linguistiques spécifiques qu'il analyse. Par exemple, il privilégie certains verbes (modaux) et certains formes verbales (passif, présent, etc.), mais ne traite pas systématiquement l'ensemble des classes de verbes ni toutes les flexions verbales [HAB 97].
Les textes relèvent de genres très variés (lettres personnelles et professionnelles, conversations face à face, conversations téléphoniques, débats et interviews, etc.).
A l'aide de la statistique multidimensionnelle (comme l'analyse factorielle des correspondances), qui consiste à repérer les oppositions des traits et à rassembler les traits qui ont tendance à apparaître ensemble, un certain nombre de pôles a été dégagé, qui constituent deux à deux des dimensions. En effet, les techniques d'analyse statistique multidimensionnelle ont pour objectif de repérer les régularités et les oppositions entre des variables multiples (ex: traits linguistiques). Elle permet de certifier l'existence de corrélations, positives ou négatives entre traits.
Un ensemble de caractéristiques qui co-occurrent définissent une dimension. Chacune des dimensions obtenues oppose deux pôles de fonctionnements textuels (exp: narrative / non narrative). Dans l'approche de Biber, c'est le regroupement de caractéristiques linguistiques qui définit une fonction partagée.
La mesure quantitative est utilisée pour identifier un groupement de caractéristiques linguistiques, puis ces groupements sont interprétés en termes fonctionnels.
En fonction de chaque dimension les textes peuvent être considérés comme proches ou éloignés. Deux textes différents peuvent être proches du point de vue d'une certaine dimension et éloignés d'un point de vue d'une autre dimension. Le décompte fréquentiel fait par Biber sur un corpus de langue générale, et après l'interprétation des contrastes majeurs mis en évidence par l'analyse multidimensionnelle, ont fait émerger cinq dimensions, qui représentent les pôles de regroupement de traits linguistiques:

- informatif (utilisation de : présent, première et deuxième personne du singulier, etc.) vs non informatif (noms, mots longs, etc.),

- narratif (troisième personne, participe présent, etc.) vs non narratif (verbes au présent, etc.),

- dépendant de la situation d'énonciation (adverbes de temps et de lieu, etc.) vs référence explicite (propositions relatives, nominalisation, etc.),

- persuasif (infinitifs, subordonnées conditionnelles, etc.) vs non persuasif,

- abstrait ou style impersonnel (passifs sans agent, etc.) vs non abstrait.

Par exemple, dans la dimension « informatif vs non informatif », on trouve la cooccurrence d'infinitifs, de modaux de prédiction, de subordination conditionnelle, etc.
Chaque texte, par son emploi de traits linguistiques, est représenté par un point dans l'espace à plusieurs dimensions. En utilisant des techniques de classification automatique, Biber aboutit à huit regroupements (classes), en rapprochant et en regroupant, en fonction de leur position sur chacune de ces cinq dimensions de l'espace. Ces regroupements sont alors considérés comme des types de textes, au terme d'une nouvelle phase d'interprétation qui s'appuie en particulier sur l'examen des textes les plus proches du centre de chacune des classes définies :

- interaction interpersonnelle intime,

- interaction à but informatif,

- exposé scientifique,

- exposé savant,

- fiction narrative,

- récit,

- reportage en situation,

- persuasif (argumentation impliquée).

Ces types sont déterminés de la façon suivante :

(i) : les textes appartenant à chaque type partagent le maximum de caractéristiques linguistiques ;

(ii) : les différents types soient le plus distincts possible.

3.3.2 Approche de Bronckart [BRO 96a] [BRO 96b]

La démarche de Bronckart a pour objectif de confirmer ou amender une typologie préexistante et de caractériser les emplois correspondant à chaque type présumé. Il suppose l'existence de trois pôles de textes, appelés archétypes : le discours en situation, le discours théorique et la narration. L'examen des cooccurrences de traits est jugé valider ces trois pôles postulés et conduit également à proposer des types intermédiaires.

3.3.3 Approche de Bergounioux et d'Habert

Le but est d'essayer de dégager et d'isoler les éléments sur employés et sous-employés (significativement) dans une partie du corpus au regard de leur emploi dans le corpus entier. Une interprétation relativement immédiate de ce type de regroupements opérés à partir des

suremplois et des sous-emplois conduit à opposer à chaque fois deux types. Par exemple, l'étude faite par Bergounioux, concernant la répartition d'un certain nombre de formes (marques d'énonciation, coordination, pronoms, prépositions, etc.) dans un corpus de résolutions générales des congrès des confédérations, a conduit à opposer deux types de résolutions : analytique vs déclarative. Le premier type sur-emploie : le verbe être à la troisième personne de l'indicatif présent, pronoms à la troisième personne, etc. Le deuxième type sur-emploie les verbes déclaratifs (appelle, considère, estime, exige, etc.), etc.

Voici un tableau comparatif de quelques approches :

Travaux	**Type 1 : Démarche à posteriori**		**Type 2 : Démarche à priori**	
Approches Caractéristiques	**Biber**	**Bronckart**	**Bergounioux**	**Habert**
Principe	Explorer les régularités et Révéler des types (non prédéfinis)	Confirmer/amender une typologie	Montante (clustering)	Montante (clustering)
Genre de textes	Varié (plusieurs genres)	Varié (plusieurs genres)	Même genre (un seul)	Même genre (un seul)
Corpus	Extraits de textes	Extraits de textes	Textes complets	Textes complets
Etiquetage	- partiel - automatique	- partiel - manuel		
Grille des traits	Pré-existante	Pré-existante	Vide	Vide
Constellation des traits	Oppositions majeures entre traits		Traits sur-employés et sous-employés.	Traits sur-employés et sous-employés.
Outil utilisé	Analyse factorielle	Analyse factorielle		

Table I.3 : tableau comparatif

4 Filtrage d'information et langage naturel

4.1 Problèmes de la langue

Les problèmes rencontrés, pour le développement de systèmes de filtrage efficaces, ne sont pas seulement d'ordre technique (rapidité, taille, robustesse, fiabilité, efficacité, etc.), mais aussi liés aux propriétés même de la langue utilisée. En effet, lorsque l'analyse est automatique, la machine extrait l'information pertinente contenue dans un texte en s'appuyant sur les éléments textuels. La tâche est donc difficile car le langage naturel est complexe et ambiguë. Nous présentons certaines des difficultés les plus importantes et qui doivent être traitées par un système de filtrage.

- Niveau graphique : Un mot peut s'écrire de plusieurs façons comportant des fautes de frappe ou d'orthographe ou s'écrire avec une majuscule. Cela diminue le *rappel*. Par exemple, le système échoue si un mot est orthographié d'une certaine façon dans le profil (Khadafi) et sous une autre forme dans le texte (Kaddafi).
Une solution à ce problème, est d'utiliser des heuristiques propres aux correcteurs d'orthographes (répétition, dédoublement, etc.).

- Niveau grammatical : Un mot peut avoir plusieurs catégories grammaticales. Dans un système de filtrage, connaître la catégorie grammaticale d'un mot en contexte permet d'augmenter la précision. Par exemple, si le mot *avions* est identifié en tant que nom dans un profil, le système permet d'écarter les textes dans lesquels il apparaît en tant que verbe.

- Niveau morphologique : Les variations morphologiques (marques du nombre, de genre, la conjugaison, etc.) diminuent le *rappel*. Par exemple, un système doit pouvoir bien filtrer les textes contenant la forme *chevaux* si le profil comporte le mot *cheval*.
Une solution est d'utiliser une procédure de lemmatisation (retrouver la racine ou lemme ou forme normalisée d'un terme) après un étiquetage grammatical. Une autre solution est d'utiliser une procédure de stemming (retrouver une pseudo-racine ou *stem* d'un terme) qui prend en compte la morphologie flexionnelle et dérivationnelle. Elle permet souvent de retrouver des termes fortement liés linguistiquement (grammatical ou sémantique). Par exemple, *déménag* est le *stem* de *déménageur, déménageurs, déménagement, déménagements, déménager, déménage,* etc.
L'inconvénient de l'utilisation de cette procédure est qu'elle conduit souvent à une augmentation du bruit. De plus, elle ne pourra jamais faire correspondre *œil* avec son pluriel *yeux*.
Il est possible de profiter des avantages des deux procédures en les combinant.

- Niveau lexical : La prise en compte des expressions composées (*pomme de terre, pied de biche,* etc.) permet énormément d'augmenter la précision.

- Niveau structural : Il peut y avoir plusieurs analyses syntaxiques différentes pour une seule phrase. Par exemple, il y a deux analyses possibles pour la simple phrase suivante (figure I.5):
"Le garçon voit un homme avec un télescope."

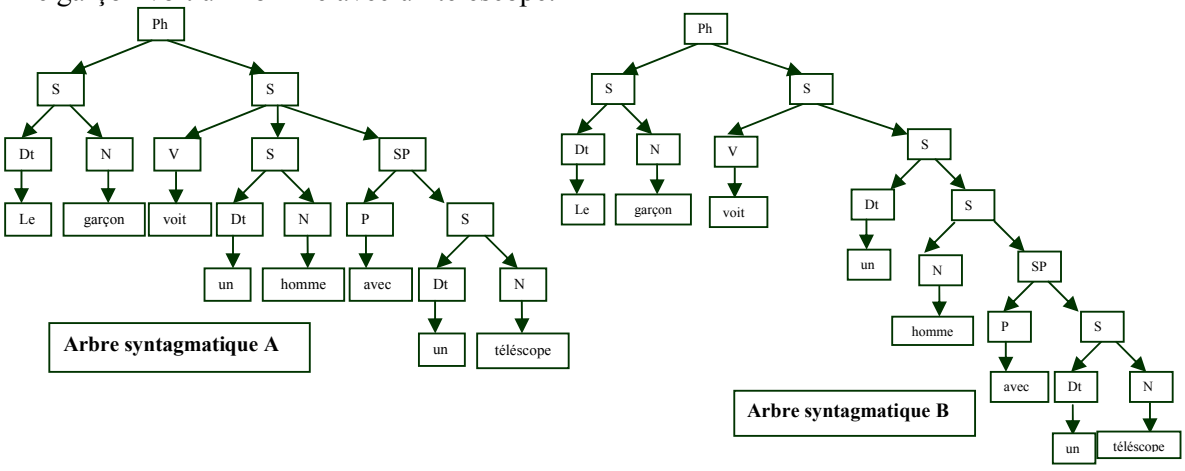


Figure I.5 : Arbres syntagmatiques

Le syntagme prépositionnel peut s'attacher soit au syntagme verbal (arbre syntagmatique "A") soit au syntagme nominal " un homme " (arbre syntagmatique "B").
Dans l'arbre A, le fait que le syntagme prépositionnel soit rattaché au syntagme verbal (SV) indique qu'il modifie le verbe et non pas le syntagme nominal qui le précède. Par conséquent, cet arbre indique que le garçon *a vu* un homme *à l'aide d'un télescope*.
Dans l'arbre B, le fait que le syntagme prépositionnel soit rattaché au syntagme nominal (SN) signifie que le syntagme prépositionnel modifie le nom *homme*. Donc, cet arbre indique que c'est l'homme qui avait le télescope et non pas le garçon.

De plus, ce type d'ambiguïté structurale a tendance à augmenter en fonction de la longueur de la phrase, surtout lorsque la phrase contient des conjonctions ou des prépositions.

On peut utiliser des heuristiques ou des statistiques afin d'éliminer les mauvaises analyses ou bien inclure des informations sémantiques dans la grammaire de l'analyseur mais ceci s'avère extrêmement difficile à gérer.

- Niveau sémantique : La synonymie : un système de filtrage doit pouvoir traiter des textes contenant des synonymes de mots constituant les profils, pour accroître le *rappel*.
L'hyponymie : relation père / fils
La polysémie : un mot avec plusieurs sens
Il est donc intéressant d'enrichir sémantiquement les profils (ajout de synonymes) et de permettre une désambiguïsation sémantique (une seule étiquette)

4.2 Approche non linguistique

Les approches les plus répandues en filtrage d'information, se basent sur une approche non linguistique de traitement de l'information. Ces approches adoptent un point de vue privilégiant les mots individuels, au détriment de la structure d'ensemble des textes. De façon plus générale, la seule hypothèse guidant ces approches non linguistiques est que le contenu informatif d'un texte donné peut-être condensé en une suite de quelques mots : des descripteurs de textes (approche 'sac de mots'). Ces approches, reposant sur des algorithmes statistiques peu dépendants des langues particulières dans lesquelles sont rédigés les textes, ont montré leurs limites :

- De nombreux éléments porteurs d'information sont éliminés au cours de l'analyse, ce qui fait baisser d'autant la qualité des résultats de filtrage;

- Les approches « sac de mots » sont complètement dépendantes des corpus sur lesquels elles opèrent, aucune généralisation n'est possible.

- Et surtout ne spécifie pas les relations entre les mots.

4.3 Approche linguistique

En linguistique informatique, une autre voie a été explorée pour tenter d'améliorer les performances des systèmes automatiques de recherche d'information (tant le rappel que la précision): l'approche linguistique, c'est-à-dire une analyse du contenu des textes guidée par des contraintes linguistiques (ex. ordre des mots, classes de termes, structuration textuelle).
Cependant, après plusieurs années d'efforts, force est de constater que la percée tant attendue des systèmes automatiques de haute qualité, grâce à des techniques linguistiques, n'a pas eu lieu.
L'échec de l'approche linguistique nous paraît majoritairement dû à l'adoption d'outils linguistiques informatiques non adaptés à la tâche, principalement dans la profondeur d'analyse mise en œuvre. Il faut adapter la profondeur d'analyse en fonction de la tâche [ABN 96a] [GRE 96] [ROC 97]. Certaines applications, dont la recherche d'information, pouvant très bien se satisfaire d'analyses partielles et locales.

4.4 Outils TAL

L'explosion de la quantité d'information électronique disponible (notamment sur Internet) a rendu le domaine du Traitement Automatique des Langues (TAL) et ses outils incontournables. Ces outils servent notamment à améliorer l'accès à l'information, comme la recherche de documents, le filtrage d'information, la traduction ou le résumé de textes, etc.
La grande majorité des approches proposées utilisent des modèles statistiques qui ont rapidement donné des résultats très prometteurs et à faible coût [ABN 96b] [CHA 97b]. Cependant, de nombreux chercheurs estiment que de telles approches sont limitées car elles ne prennent pas ou prennent peu en compte le contenu linguistique des données traitées. Ils marquent l'importance de la linguistique dans ce domaine.
Ils préconisent la construction de larges bases de données linguistiques (lexiques et grammaires), pouvant s'insérer à différents niveaux de l'analyse automatique (morphologique, syntaxique, sémantique, etc.). Cette démarche nécessite, cependant, un investissement lourd qui s'inscrit sur le long terme. Bien qu'il existe certaines données linguistiques, le plus souvent, elles ne sont pas exploitables directement et nécessitent de longues opérations de reformatage et de conversion en données applicables.
De plus, les méthodes linguistiques génèrent beaucoup de bruit du fait de l'ambiguïté naturelle de la langue qui doit être prise en compte (ambiguïtés grammaticales, syntaxiques et sémantiques). Pour pallier ces inconvénients, il est nécessaire de mettre au point des méthodes et des outils informatiques d'aide à la construction de composants linguistiques et directement applicables à des textes. En effet, de gros efforts sont faits tels que : l'élaboration de méthodes formelles et systématiques de description de faits linguistiques basées sur l'observation (par exemple les grammaires locales: la méthodologie du lexique-grammaire de M. Gross) [CON 03] ; le développement d'outils informatiques facilitant le repérage de phénomènes linguistiques complexes dans les textes (exemple : Intex, Unitex, etc.) [CON 03]; la constitution de vastes corpus annotés ou non afin d'offrir un champ d'investigation et d'évaluation plus grand: par exemple, le corpus Brown pour l'anglais et Frantext pour le français [BER 02] ; enfin, des études sur la normalisation et la diffusion des ressources construites (corpus, dictionnaires, etc.) [CON 03].

5 Conclusion

L'objectif des recherches en TALN et en IA était avant tout de modéliser, de formaliser le savoir humain, de dégager les règles sous-jacentes. C'est pourquoi les méthodes utilisées en TALN étaient alors largement symboliques, c'est-à-dire fondées précisément sur des règles.
L'étude des différents outils existants nous a menés aux conclusions suivantes:
Deux points semblent être communs à toutes ces méthodes:

- elles ne peuvent avoir un réel intérêt que si elles s'appliquent à un domaine limité,

- l'organisation et le contenu du dictionnaire jouent un rôle primordial pour leur bon déroulement.

Vu la richesse du langage naturel qui est liée aux multiples sens des mots, aux ambiguïtés, il est très difficile, voire même impossible, de développer un modèle sémantique qui pourrait le recouvrir entièrement [BON 84].
La sophistication des formalismes utilisés pour le TALN ne débouche pas toujours sur des systèmes de traitement fiables et efficaces.

En effet, tout d'abord un système de TALN a besoin de ressources (dictionnaires, grammaires) à la fois très vastes (en nombre d'entrées lexicales et de règles) et très détaillées (par exemple, les conditions syntaxiques d'emploi de mots).
Les ressources actuelles sont insuffisantes. Leur amélioration, n'est pas à chercher dans des nouvelles études, mais plutôt dans l'observation des larges ensembles de données textuelles qui sont maintenant disponibles.
L'observation de données langagières en très grande quantité et le traitement de flux d'information aussi important que ceux qui circulent aujourd'hui sur le réseau Internet conduisent inéluctablement à recourir à des approches quantitatives ou au moins à combiner approches symboliques et approches quantitatives.
L'analyse automatique de textes en langage naturel est en plein essor mais reste difficile. L'analyse de toutes les informations présentes dans un texte est un processus très complexe car il fait intervenir de nombreux paramètres.
L'interprétation d'un énoncé en texte libre nécessite non seulement des connaissances linguistiques, mais aussi des connaissances extralinguistiques (connaissances du monde, conventions, etc.). Toutes ces connaissances sont difficiles à encoder dans les systèmes d'analyse automatique, du fait de leur complexité et de leur quantité.
A l'heure actuelle, l'analyse automatique de textes est par conséquent restreinte soit à un domaine très limité, soit à une compréhension très basique.
Dans le domaine du filtrage, il faut considérer le problème des variations linguistiques. En effet, les mots clés (multiples) composant les profils de l'utilisateur n'apparaissent souvent pas de façon littérale dans le document. Par exemple, un mot peu se présenter sous une autre forme (un nom sous sa forme pluriel, un verbe au passé, etc.). Le document contenant le groupe nominal « polluted rivers » est en effet pertinent pour le profil contenant les mots clés « river » et « pollution » [COR 97]. Le traitement de la variation linguistique des mots permet d'introduire une flexibilité dans la procédure d'appariement entre le profil et le document à filtrer : en effet, des réalisations linguistiques différentes portant le même contenu informationnel peuvent être regroupées et considérées comme équivalentes.
De plus, la nécessité de la mise en rapport de formulations différentes mais sémantiquement proches (traitement du problème de l'ambiguïté des mots), permet d'élargir la portée de la représentation dans les textes et augmenter ainsi les chances d'apparier le profil et le texte à filtrer. Ce qui permet ainsi de considérer, par exemple, un texte contenant les éléments *cours de mathématiques, enseignement des maths, les maths sont enseignées, etc.* comme pertinent pour le profil *cours de maths*.
Ce la montre l'intérêt de l'analyse du langage naturel pour le domaine du filtrage de l'information. Il est donc nécessaire de développer et d'utiliser des outils efficaces et sophistiqués de traitement automatique des langues, capables de traiter la variabilité linguistique et de prendre en compte l'aspect sémantique.
Nous avons explicité les travaux de TAL pouvant servir à améliorer les performances des systèmes de filtrage d'information en apportant des réponses au double problème de la variation linguistique des mots et de leurs ambiguïtés.
Dans le cadre de notre travail, nous visons principalement, par le recours à des informations linguistiques dans le domaine de filtrage, les objectifs suivants :

- Définir des profils correctement discriminants et non ambigus.

- Représentation étendue des textes à filtrer.

- Mettre en rapport des formulations différentes mais sémantiquement proches afin d'augmenter les chances d'apparier un profil et un texte à filtrer.

Les termes des profils et/ou des textes à filtrer sont automatiquement propagés en suivant les liens exprimés dans la base lexicale, de manière à disposer d'une description plus étendue des profils et des textes.

Chapitre II

Filtrage d'Information Textuelle

Dans ce chapitre, nous présentons le contexte dans lequel est née la notion de filtrage d'information (essentiellement attachée au domaine de la documentation), sa situation par rapport aux domaines proches. Nous illustrons les processus essentiels pour la sélection d'information assimilés au processus de filtrage. Nous aborderons les questions fondamentales auxquelles doit répondre un système de filtrage, tout en décrivant ses caractéristiques et son évaluation. La fin de ce chapitre est consacrée à l'étude des approches dominantes en filtrage d'information.

1 Généralités

1.1 Histoire du filtrage d'information

La naissance du concept de filtrage d'information repose sur un besoin très concret : d'une part, fournir un service personnalisé aux utilisateurs en leur apportant une information ciblée en fonction de leurs besoins. D'autre part, en partant d'une infrastructure documentaire existante (bibliothèques).

1.1.1 Systèmes de veille économique

En 1958, LUHN [OAR 99] [LUH 58] pose les bases conceptuelles des systèmes d'information modernes. Il introduit l'idée de Systèmes de veille économique «Business Intelligence Systems », dans sa théorie appliquée à l'activité de gestion de l'information reposant sur des pratiques en documentation (ex. : au niveau d'une bibliothèque) : les opérateurs humains définissent un profil pour chaque requête bibliothécaire, ces profils seront ensuite comparés (comparaison exacte) aux différents documents afin de construire une liste de documents répondant à chacune des requêtes. Chaque profil est conçu pour identifier un utilisateur unique. De plus, le profil est mis à jour à l'arrivée de tout nouveau document (ex. : commande d'ouvrages).
Les travaux de LUHN ont permis de définir les principes phares de ce qu'il avait appelé « diffusion sélective des nouvelles informations» (**S**elective **D**issemination of new **I**nformation), terme qui aura caractérisé ce domaine de recherche durant un quart de siècle.

1.1.2 Diffusion sélective d'information

Dès la fin des années 1960, comme l'atteste l'étude de Housman [KIL 97a] [DEN 92], le besoin de systèmes de diffusion ciblée d'information (SDI), prenant en compte les besoins d'utilisateurs individuels, se faisait sentir. Dix ans plus tard, l'intérêt porté au **SDI** a conduit à la formation d'un groupe d'intérêt spécialisé dans l'**SDI** (**S**pecial **I**nterest **G**roup-**SDI**) de la société américaine pour les sciences de l'information « American Society for Information Science ». Cette même période a vue la mise en service de 60 systèmes de filtrage dont 9 desservant plus de 1000 utilisateurs chacun.
Ces systèmes mettaient en œuvre le modèle de LUHN, cependant seul quatre d'entre eux implémentaient une mise à jour automatique des profils, quant aux autres systèmes ils se contentaient de techniques de mise à jour manuelle des profils.

1.1.3 Naissance de la notion de filtrage d'information

C'est en 1982, que l'expression « Information Filtering » fut introduite par DENNING dans sa lettre du président de l'ACM parue dans la revue « Communication of the ACM ». En Mars 1982, DENNING élargissait le champ d'utilisation du filtrage qui était jusque là orienté vers la génération d'informations pour y inclure la réception d'informations. Il est l'un des premiers à utiliser le terme filtrage pour désigner un processus visant à préserver la bande passante mentale (mental bandwidth) des utilisateurs des systèmes de courrier électronique, un nouveau moyen de communication. Il mit en évidence le besoin de filtrer les messages électroniques (E-MAIL) afin de séparer les messages urgents des autres, et cela suivant les pôles d'intérêt de l'utilisateur. Cette réduction du flux d'information avait pour particularité de se baser sur le contenu des messages, et non plus seulement sur des indices tels que l'identité du correspondant.
Durant la décennie qui suivit, quelques articles sur le filtrage d'information sont apparus et tandis que le domaine initial du filtrage était le filtrage du courrier électronique, ces travaux l'étendaient aux (NEWS) et aux réseaux de diffusion d'informations.
Le travail le plus remarquable durant cette période a été publié dans « the communication of the ACM » par MALONE et AL en 1987. Ces derniers introduisirent les notions de « cognitive » (adaptatif), « economic » (économique) et « social » (social) dans leur projet appelé « Information Lens ». Plus encore, ils émirent l'hypothèse selon laquelle le filtrage serait l'alternative à l'adressage : « Les expéditeurs de messages électroniques n'ont pas besoin d'expliciter leur destinataire, ils envoient donc leur message avec la mention « Anyone » (n'importe qui), du moment que tous les destinataires sont équipés d'un filtre pour les messages électroniques, ce filtre ne laissera passer que les messages intéressants pour un utilisateur donné » [KIL 97a]. Cette hypothèse étant intéressante d'un point de vue théorique n'est pas applicable en pratique à cause du manque de structuration de la plupart des messages (E-MAIL) et de l'absence de standard dans ce domaine d'application.

En 1989, le DARPA[1] organisa la première conférence sur la compréhension des documents, elle s'occupait des méthodes d'extraction d'informations pertinentes à partir des documents.

Au début des années 90, l'exploit enregistré dans l'architecture des ordinateurs (serveurs) a ouvert de nouveaux horizons pour le filtrage qui est passé de la gestion bibliothécaire à des usages plus vastes en une quarantaine d'années.

[1] *United States Defense Advenced Research Projects Agency*: volet recherché du ministère américain de la défense.

Deux contraintes très pragmatiques ayant influencé le développement du filtrage :

- La contrainte de maximiser l'information pertinente pour chaque utilisateur, en fonction de son profil.

- La contrainte de minimiser la perte de temps induite par l'information non pertinente et par l'augmentation du volume des échanges, due aux nouveaux moyens de communication.

En Novembre 1991, un atelier sur le FI de haute performance (High Performance Information Filtering), était organisé, au cours duquel plusieurs perspectives différentes du domaine de filtrage ont été examinées : de la sélection de l'information à la modélisation de l'utilisateur, en passant par les domaines d'applications, les techniques et outils ainsi que des considérations sur la confidentialité et des études de cas. L'ensemble des communications de l'atelier fut regroupé dans une édition spéciale des communications of the ACM, en Décembre 1992.
L'article de Belkin et Croft [BEL 92], sur le filtrage d'information, a constitué un point de départ et une référence incontournable pour toutes les recherches qui l'ont suivi. Ce n'est qu'en 1995 que le terme filtrage d'information est apparu dans les conférences TREC[2], organisées par le département de la défense américaine DARPA et le NIST[3] [BAL 01].

1.2 Définitions

a)- Notion d'information

Les théoriciens de l'information, tels que Shannon et Bar Hillel, n'ont eu de cesse de distinguer entre information (véhiculée par les suites de caractères) et contenu (représentation sémantique). Nous donnons deux définitions de la notion d'information : définition quantitative et définition fonctionnelle.

- *La définition quantitative* : repose sur l'estimation de la probabilité d'occurrence d'une classe d'éléments ou d'évènements donnés (caractère, syllabe, mot, phrases). Le dénombrement de ces différents types d'évènements permet d'associer à chaque évènement $e_1, e_2, .., e_n$ les probabilités $p_1, p_2, .., p_n$. L'incertitude liée à la survenue d'un évènement i est donnée par l'entropie H (confondue avec la notion de quantité d'information):

$$H(p_i) = - p_i \log_2(p_i).$$

Cette mesure est généralement considérée comme caractérisant l'organisation des systèmes (ensembles d'évènements).

- *La définition fonctionnelle* : Bar Hillel affirme qu'aucune adéquation entre entropie (quantité d'information) et contenu véhiculé par un texte n'est possible (Bar Hillel, 1964). Il est donc nécessaire d'envisager une définition fonctionnelle de l'information (fonction informative). Une définition de la fonction informative est : un élément de contenu répondant à un besoin en information. Cette définition implique de définir le besoin en information de l'utilisateur.

[2] Text Retrieval Conference. http://trec.nist.gov/
[3] National Institut for Standards and Technology, une agence du ministère américain du Commerce.

b)- Recherche d'information

Le but initial de la recherche d'information (RI) était d'indexer et de retrouver rapidement, parmi de larges bases textuelles, des sous ensembles (quelquefois ordonnés) de documents pertinents par rapport à une requête ou un sujet déterminé. L'application typique de ces techniques est l'accès par requêtes aux bases de données de bibliothèques.
La recherche d'informations vise donc à retrouver les documents pertinents à une requête posée de façon automatique. Traditionnellement, on y procède avec des mots: les documents sont d'abord indexés par des mots jugés importants, ainsi que la requête; un document est retrouvé s'il contient les mêmes mots que la requête.

c)- Extraction d'information

L'extraction d'informations (EI) est une tâche qui consiste à extraire de l'information structurée à partir de documents textuels [ARC 02]. Elle peut être considérée comme une sous spécialité récente du large domaine du traitement du langage naturel. Elle consiste donc à identifier de l'information bien précise d'un texte en langage naturel. Par exemple, à partir d'un rapport sur un accident automobile, un système d'extraction d'information sera capable d'identifier la date et le lieu de l'accident, le type d'incident ainsi que les victimes. L'extraction est faite en alimentant un formulaire spécifique [PIL 00]. Les applications sont très multiples : interprétation de requêtes écrites en langage naturel, construction automatique de bases de données à partir de textes écrits, systèmes de dialogue, etc.

d)- Filtrage d'information

Le Filtrage d'information (FI) est une branche relativement jeune du domaine de la recherche d'information. En fait, le FI n'est ni un nouveau concept ni un concept exclusivement limité aux documents électroniques. En effet, quand nous lisons un texte quelconque sur papier ou autre support, le processus de filtrage intervient. Ainsi, nous sommes en train de filtrer une partie de l'énorme ensemble d'informations auquel nous avons accès et ce, à chaque fois que l'on désire acquérir une certaine information [FOL 92]. Les chercheurs ne se sont intéressés à ce domaine de FI que récemment.
Les concepts du FI ne sont pas bien définis et les limites entre le FI et les domaines proches (recherche, routage et extraction d'information) ne sont pas très claires.
Toutefois une définition du FI a pu être dégagée, à savoir, le FI est le processus visant, à partir d'un large volume d'informations générées dynamiquement, d'extraire et de présenter seulement l'information susceptible de correspondre aux besoins et intérêts de l'utilisateur, après que celui-ci ait établi un profil, c'est-à-dire défini ses centres d'intérêt (figure II.1).

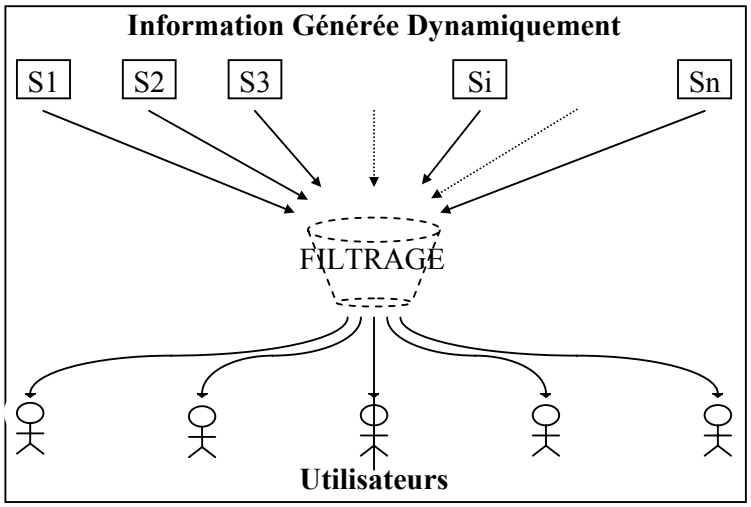


Figure II.1: Processus de Filtrage d'Information

Le filtrage est donc un mode de traitement qui assure une sélection d'information suivant un besoin exprimé [TUR 00]. Il constitue un domaine proche de la recherche d'information, dans le sens où ils ont le même but qui est de retrouver l'information pertinente pour un certain utilisateur. Si l'on considère la recherche d'information, d'un point de vue très large, comme étant un processus de sélection de l'information, alors le filtrage de l'information est simplement un cas particulier de la recherche d'information où l'information arrive d'une manière dynamique. D'un autre point de vue, si l'on considère que la recherche d'information est un processus assurant la sélection d'une information relativement statique en réponse à des requêtes relativement dynamiques, alors le filtrage de l'information est mieux vu comme étant le problème dual de la recherche d'information [KIL 97a] où on a une base de profils et les documents arrivent d'une façon dynamique (figure II.2).

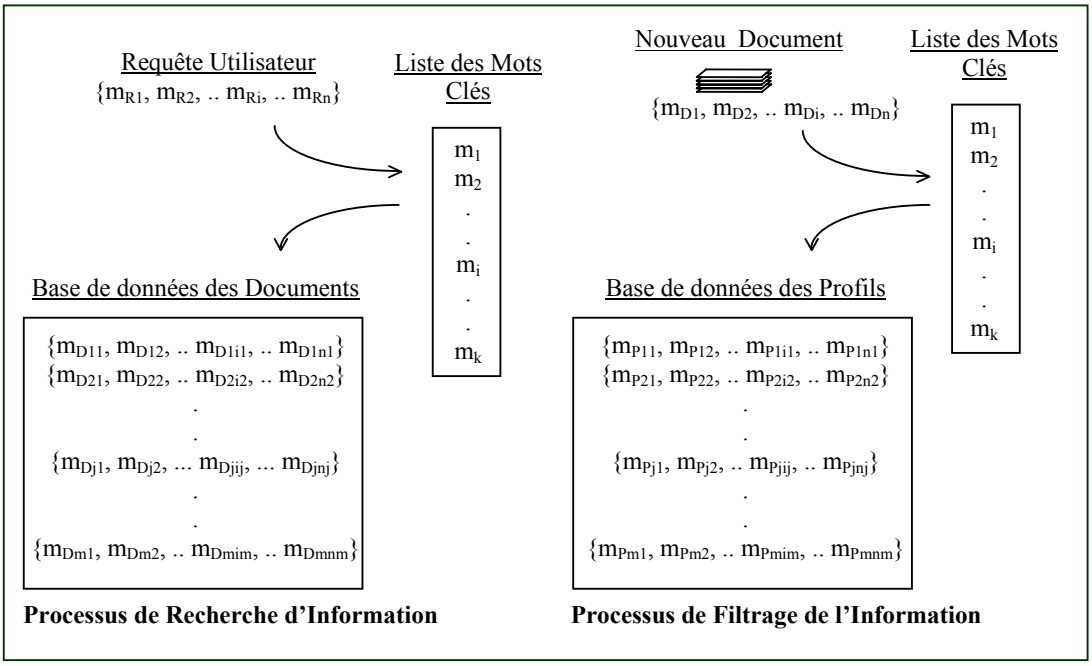


Figure II.2 : Indexation dans les processus de Recherche et de Filtrage d'Information

On peut dire qu'il existe deux activités principales (domaines) qui se distinguent par la nature du besoin en information des utilisateurs et du flux d'information : *Pull* et *Push*. Pour la première, le besoin en information est éphémère (non stable) et le fond documentaire est stable (recherche documentaire). Par contre, pour la deuxième, le besoin en information est stable et le flux de documents est dynamique (filtrage et routage d'information). Le filtrage et le routage de l'activité *push* se distinguent essentiellement par la nature de la décision de sélection opérée : Décision de sélection binaire (accepté/rejeté) pour le filtrage versus continue (classement selon un degré de pertinence) pour le routage d'information. Par ailleurs, la différence essentielle entre les deux tâches est que seuls les documents jugés pertinents sont présentés aux utilisateurs dans le cas du filtrage, alors que l'ensemble des documents, triés selon un score de pertinence, est présenté aux utilisateurs dans le cas du routage.
La plupart des systèmes destinés au filtrage d'information décrivent en réalité des heuristiques (fonction de seuil) visant à simuler une décision binaire à partir d'une décision continue.

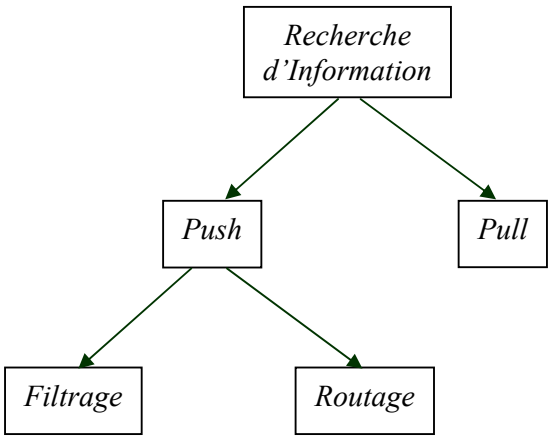


Figure II.3 : Activités de recherche d'information

L'une des toutes premières formes de filtrage de l'information se trouve être la dissémination sélective et automatique de l'information : courrier électronique, conférences électroniques, distribution d'articles, etc.

La table II.1 ci-dessous illustre quelques processus essentiels pour la sélection d'information assimilés au filtrage de l'information [OAR 96] [YAN 93]:

Processus	**Besoin d'information**	**Source d'information**
Filtrage d'Information (FI)	Stable et spécifique à long terme	Dynamique et non structurée
Recherche d'Information (RI)	Dynamique et spécifique	Stable et non structurée
Bases de Données (BD)	Dynamique et spécifique	Stable et structurée
Extraction d'Information (EI)	Spécifique	Non structurée

Table II.1 : Processus de sélection d'information

1.3 Domaines d'application du filtrage d'information

Un système de filtrage agit en tant qu'intermédiaire entre les sources de l'information et ses utilisateurs et cela dans le but de cibler l'information vraiment pertinente suivant les besoins de connaissance établis par l'utilisateur et ce, à court ou à long terme (figure II.1). Il peut être placé du côté de l'utilisateur ou bien du côté de la source d'information. Dans le premier cas, qui est le plus utilisé, le filtrage est un outil qui assiste l'utilisateur dans sa tâche de sélection de l'information pertinente. Par contre, dans le deuxième cas, il est utilisé pour cibler les utilisateurs qui ont besoin d'un certain type d'informations [LOE 92].
L'évolution explosive d'Internet et d'Intranet a motivé l'implantation de systèmes de filtrage sur les machines. En effet, plusieurs services de ces réseaux requièrent dores et déjà cet assistant personnel. Les efforts de recherche entrepris dans le domaine du filtrage font donc essentiellement ressortir deux axes principaux d'applications:

a)- Filtrage au niveau serveur (routage)

Dans ce type d'applications, la conception est centralisée sur le serveur qui dispose de profils, caractérisant les différents utilisateurs (ou groupes d'utilisateurs). Ensuite, il détermine, pour chaque information à diffuser, la liste des destinataires auxquels l'information sera effectivement transmise. Les applications sont assez variées, et d'une grande importance économique, parmi elles:

- Mailing list: L'utilisateur s'inscrit dans des listes d'intérêt et reçoit passivement des messages dans le domaine de son choix via la messagerie électronique.

- Usenet News: Similaire à Mailing list, l'utilisateur s'inscrit dans des newsgroups, et il accède périodiquement pour consulter les articles relatifs à son domaine.

- Les services de dissémination de l'information: Le système collecte de l'information à partir de différentes sources. Et périodiquement, il distribue l'information, après l'opération de filtrage basée sur un ensemble de profils, à une large population d'utilisateurs (*Clearing house Service*).

b)- Filtrage au niveau des destinataires de l'information

L'utilisateur doit définir des profils caractéristiques de ses centres d'intérêt. Une information reçue par un utilisateur (par exemple, transmise par un serveur, autre utilisateur, etc.) lui est effectivement présentée si elle sélectionne au moins un de ses profils caractéristiques. Parmi les applications :

- La messagerie électronique: L'utilisateur décrit au système un ensemble de règles et de directives qui permettent de filtrer et de classer automatiquement les messages.

- Le Web: Le filtrage de l'information pour le Web est plus difficile que les autres à cause du fait que les différents documents d'une part arrivent dynamiquement et d'autre part, constituent un hypergraphe. De ce fait, la rencontre de chaque connexion ou lien vers un autre document entraîne l'application du processus de filtrage à ce document qui constitue un sous-graphe.

1.4 Quelques travaux précédents

a)- SIFT (Stanford Information Filtering Tool)

SIFT [YAN 00] est un outil de dissémination de l'information qui permet de sélectionner, à partir de larges volumes d'informations, les informations pertinentes et de les envoyer aux personnes qui en ont besoin (figure II.4).

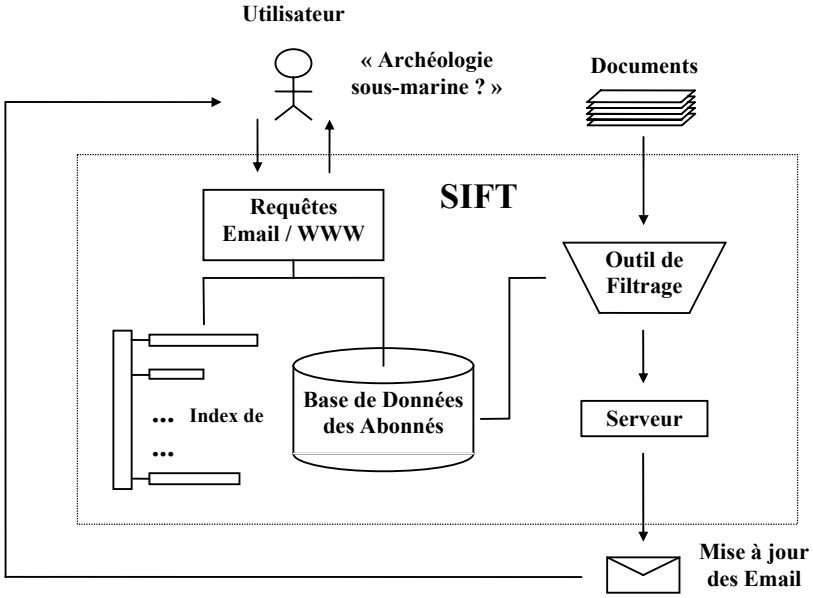


Figure II.4: Architecture et fonctionnement de SIFT

L'utilisateur intéressé par un tel service s'inscrit en soumettant les profils qui décrivent ses intérêts. Ensuite, il reçoit passivement les nouvelles et les informations filtrées qui répondent à ses besoins. SIFT supporte le filtrage **full-text** en utilisant les modèles classiques de recherche d'information (modèle booléen). L'un des premiers systèmes de filtrage d'information par reconnaissance de mots clés. Il repose sur une définition et une mise à jour complètement manuelle des profils (listes de mots). Il est capable de traiter un large volume d'information avec un grand nombre de profils en utilisant les nouvelles techniques d'indexation. Il s'exécute sous UNIX. Les capacités du système ont été testées sur différents serveurs news (ex. : Usenet). Il fournit une liste ordonnée d'articles, triés selon un taux de pertinence par rapport aux listes servant de profils.

b)- INFOSCOPE

Infoscope (Stevens, 1992) est proche de SIFT dans le sens où il est également destiné au filtrage des serveurs news. Cependant, ce système offre une fonctionnalité de modélisation automatique des profils d'utilisateurs, reposant sur un algorithme d'apprentissage. La création de profils est basée sur l'interaction entre le système, qui propose des solutions, et l'utilisateur qui valide, corrige ou refuse ces propositions. Le système induit ainsi des règles de sélection binaire à partir des réponses de l'utilisateur, et sur des paramètres simples tels que le temps dédié à la consultation d'un message donné. Il fut conçu dans le but d'éviter à l'utilisateur

d'expliciter ses intérêts et afin de fournir un outil convivial capable de s'adapter à chaque utilisateur.

c)- GROUPLENS

Un système basé sur une classification binaire (important ou non important). Un message est important s'il va être lu par l'utilisateur. L'importance du message est estimée par l'utilisateur sous forme d'une valeur prise dans l'intervalle discret [non important, important]. A base d'un ensemble d'exemples de messages évalués par l'utilisateur, le système traite chaque nouveau message pour lui attribuer une valeur d'importance. Certains systèmes utilisant la même approche, donnent la possibilité à l'utilisateur de définir un seuil permettant de décider si un message est important ou non [KIL 97a].

d)- BROWSE

Un lecteur de Usenet News, utilisant les réseaux de neurones comme modèle utilisateur. Il permet d'affecter une note aux messages et de les classer soit acceptés (lu) ou rejetés (non lu). Il se base sur les 300 premiers mots du message [KIL 97a].

e)- NEWSWEEDER

Un système utilisé à l'université de Carnegie Mellon, basé sur un filtrage collaboratif [KIL 97b]. A l'aide d'une interface Web, l'utilisateur visite des news-groups et affecte une note(allant de 1 à 5) à chaque article jugé intéressant. Le système collecte les différentes notations faites par les utilisateurs et calcule une note moyenne pour chaque article.

f)- SMART

Moteur de recherche et d'indexation originel de Salton, constitue le système duquel découle l'ensemble des systèmes commerciaux les plus répandus du marché. Par exemple, le moteur PRISE de NIST, utilisé dans les conférences TREC pour indexer des corpus textuels variés (ex. : journaux, dépêches journalistiques, etc.).

g)- INFOSCAN

Un outil qui permet de filtrer l'information suivant les intérêts spécifiques des utilisateurs. Il sert à filtrer, à repérer ou à classer n'importe quel document de format **texte**. L'utilisateur doit décrire ses intérêts au système en tapant des mots clés dans des filtres (profils). Les filtres sont des descriptions de sujets qui intéressent l'utilisateur.
Chaque mot clé d'un filtre est accompagné d'une pondération et d'une portée. La pondération pour l'aspect priorité du filtre et la portée permettent de chercher les mots clés dans les documents. Le système cherche les mots clés dans les documents et affiche les résultats sur un écran de radar qui permet à l'utilisateur de voir, d'un simple coup d'œil, les documents les plus pertinents sans même lire un seul mot (figure II.5).

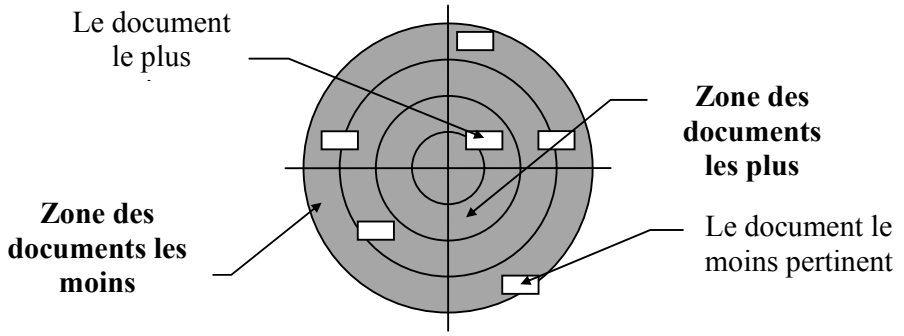


Figure II.5: Présentation des documents par le système INFOSCAN

h)- PEFNA [KIL 97b]

Un système d'évaluation d'articles News développé à l'université de Stockholm par Kilander, Fšahraeus, Lantz et Palme. Il est basé sur un filtrage par classification ou catégorisation. Il permet à l'utilisateur de définir, de créer et de maintenir une liste de classes. A chaque classe est associé un ensemble d'articles jugés intéressants par l'utilisateur. Pefna est composé donc d'un lecteur de news et d'un agent de classification.
L'utilisateur, à travers son lecteur de news, attribue à chaque article lu une catégorie, avant de l'archiver. Avec le temps, et avec l'utilisation fréquente du lecteur de News, l'utilisateur aura accumulé plusieurs catégories d'articles disposant chacune d'un ou plusieurs articles lui appartenant. Après une phase de création d'une liste de classes, un agent de filtrage est lancé pour calculer les distances entre un nouvel article et chaque classe définie par l'utilisateur. Les articles se rapprochant d'une catégorie seront automatiquement classés dans cette dernière. La fonction utilisée pour calculer la distance est la mesure du cosinus [SAL 88].

i)- CORAIL

Le système CORAIL [BAL 01] est un prototype de filtrage d'information, basé sur une approche faisant essentiellement appel à des ressources linguistiques : dictionnaires, grammaires locales, ainsi que des tables du lexique-grammaire. Les tables du lexique-grammaire sont des tableaux associant à une entrée lexicale ses contraintes syntaxico-sémantiques sous forme binaire (table II.2).

a	b	c	d	e	f	g	h	
N0	N1	V	N0 V	N0 V N1	Nominalisation	Actif	Passif	…
		faire muter	-	+	+	+	+	…
		moderniser	+	+	+	+	+	…
		modifier	-	+	+	+	+	…
		…	…	…	…	…	…	…

Table II.2 : Lexique-grammaire des verbes

N0 et N1 désignent des noms spécifiques, V verbe. La table 1 présente les contraintes syntaxico-sémantiques des verbes : les constructions syntaxiques acceptées : N0 V, N0 V N1 ;

les relations lexicales : nominalisation, synonymie, etc. ; les transformations : passive, active ; etc. CORAIL permet de repérer des séquences jugées pertinentes par l'utilisateur et de leurs variantes syntaxiques, par une analyse partielle (transducteurs élaborés). Il se présente sous la forme d'une plate forme distribuée (système multi-agents), développée sous java.

j)- ENAIM

Le système ENAIM [TUR 00] est un prototype de filtrage d'information, basé sur un modèle utilisateur. Il permet de générer automatiquement le profil de l'utilisateur. Les règles de filtrage sont de la forme « *conditions alors conclusion ».* Les prémisses portent sur le champ expéditeur, le sujet et les phrases du contenu du message. La conclusion est une action binaire, car il s'agit d'une classification : consiste à transférer ou non un message dans une classe.

Exemples de règles :
Si « le message parle de sport » alors transférer le message dans la classe *Sport* ;
Si « l'auteur est Nadir » alors transférer ;
Si « le message contient Football » alors transférer.

2 Un Système de Filtrage

Un bon outil de filtrage doit répondre à deux questions fondamentales :

(i) Quelles sont les méthodes les plus efficaces pour la correspondance (*matching*) des intérêts des utilisateurs avec l'information disponible ?

(ii) Et comment devrait-on décrire les intérêts d'un utilisateur ?

2.1 Architecture de base

Un système de filtrage peut être composé des éléments de base suivants (figure II.6):

- Analyse et représentation de l'information.

- Représentation des intérêts des utilisateurs ou profils.

- Une fonction de mesure de similarité.

- Processus de filtrage : Actions.

- Exploitation des résultats de mesure : apprentissage et adaptation.

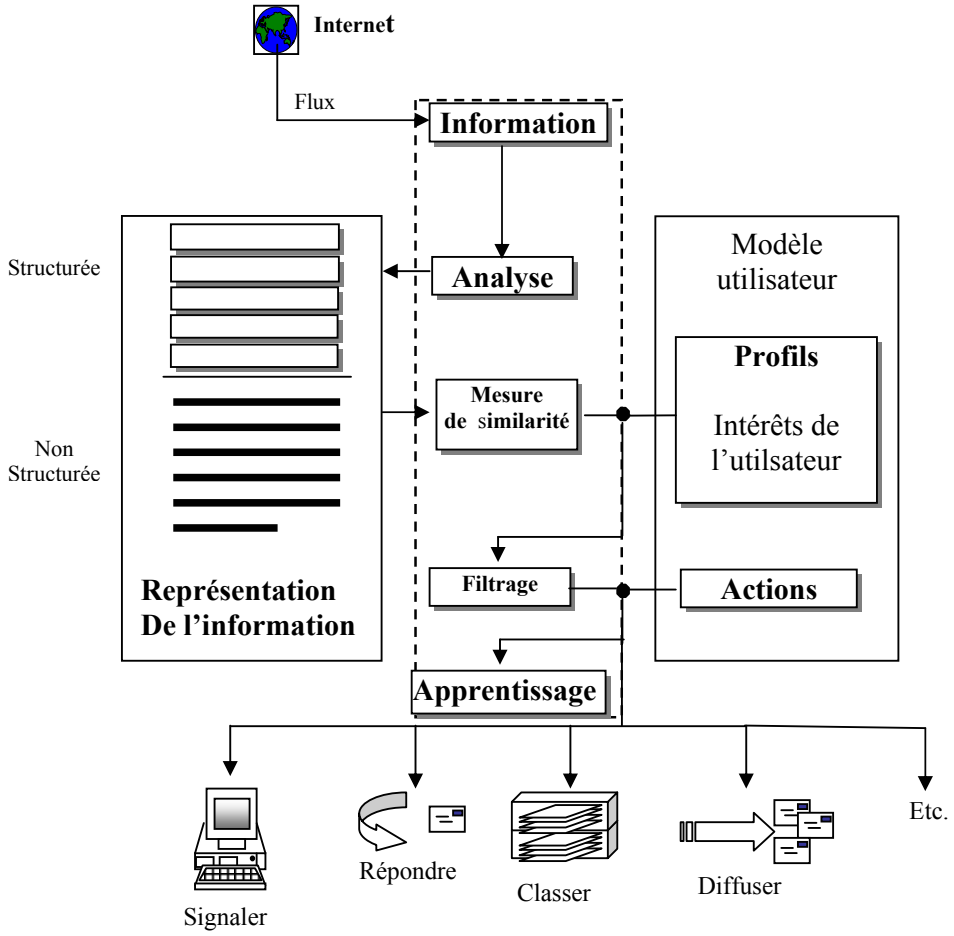


Figure II.6 : Modèle de base de filtrage

Ce modèle de filtrage, inspiré du modèle de RI, considère, d'une part, les utilisateurs du système de filtrage (caractérisés par des besoins relativement stables à moyen et long terme), d'autre part, le flux d'information provenant de différentes sources. L'évaluation de l'information se fait en la confrontant avec les besoins des utilisateurs.

2.2 Caractéristiques du Filtrage

Un système de filtrage doit pouvoir, d'une part comprendre et interpréter les souhaits de l'utilisateur, et d'autre part examiner et orienter l'information disponible vers les utilisateurs intéressés.
Un système de filtrage doit tenir compte des caractéristiques suivantes :

- *La nature spécifique de l'information* : concerne des informations peu ou pas structurées, contrairement aux bases de données conventionnelles, qui utilisent des données structurées sous forme d'enregistrements. Un exemple de données peu structurées est la messagerie électronique, où l'entête du message a des champs bien définis (*From, To, Subject, etc.)*, et donc structurés tandis que le corps est un texte libre non structuré.

- *Le multimédia* : traite en général des informations textuelles, mais peut également intégrer des informations multimédia tels que l'image ou le son. La mise au point de techniques spécifiques adaptées aux informations multimédia est un problème particulièrement complexe pour lequel les solutions réellement satisfaisantes restent encore à découvrir, malgré les progrès notables dans les domaines de reconnaissance de la parole et des formes [RAJ 97].

- Concerne un flux d'information émanant de sources extérieures diverses ou adressé à l'utilisateur (mailing list, usenet news, filtrage des mails, etc.). Le filtrage, est aussi utilisé, pour décrire le processus d'accéder et de rechercher des informations dans des ressources distantes (filtrage dans le Web, dissémination de l'information, etc.).

- Opère une décision de sélection binaire (oui / non) : toute information traitée par le système doit être attribuée à un ou plusieurs profils, ou bien rejetée. Cette décision de sélection est simulée au moyen d'une fonction de seuil.

- Une prise en compte plus fine du contenu effectif de l'information traité.

- *Une prise en compte du profil utilisateur :* Le filtrage s'opère à partir du profil défini par un usager ou un groupe d'usagers. Le profil représente les besoins à moyen et à long terme.

- Consiste aussi à éliminer certaines informations du flot d'information provenant de sources distantes (non pas l'extraction d'information appropriée).

- *L'incrémentabilité :* les performances du système de filtrage en temps de traitement.
Le FI se caractérise par des contraintes fortes portant sur le temps de traitement, ainsi que sur la qualité de la sélection automatique de l'information.

2.3 Evaluation du filtrage

L'évaluation des performances et de l'efficacité d'un système de filtrage donné permet d'une part de déterminer sa capacité à répondre aux besoins des utilisateurs et d'autre part, de le comparer à d'autres systèmes existants. Cette évaluation, est une tâche difficile. En effet, la satisfaction de l'utilisateur, pourtant fondamentale, ne peut pas être évaluée de manière certaine.
Plusieurs méthodes permettent de mesurer l'efficacité d'un système. Elle est souvent mesurée à l'aide d'indicateurs quantificateurs. Mais la mise en place d'environnements d'évaluation et même l'interprétation des résultats obtenus ne sont pas triviales [BAL 02]. Il existe de nombreux programmes d'évaluation de systèmes : nous évoquons certains programmes les plus importants.

2.3.1 Notion de pertinence (Relevance)

La notion de pertinence est très importante. On s'y réfère pour tenter de définir la relation qui relie de manière satisfaisante deux documents, dans un contexte donné. Étant donné qu'elle présuppose un mécanisme de *compréhension* du langage, cette notion ne peut pas être définie algorithmiquement dans le contexte des systèmes actuels, du moins pas de manière complètement satisfaisante [BAL 02]. La pertinence est une notion subjective. Différents utilisateurs peuvent avoir des opinions différentes à propos de la pertinence ou non-pertinence

d'un certain document [COR 97]. La pertinence joue le même rôle mystérieux que le concept de 'sens', difficilement qualifiable ou quantifiable. Par conséquent, une machine n'est pas en mesure de juger la pertinence d'un document, mais seulement de suggérer à l'utilisateur qui prendra la décision finale. Une façon d'augmenter la pertinence d'un système est de confectionner un corpus de test, constitué de documents pertinents, qui servira de référence ultime.

2.3.2 Critères d'évaluation

Il existe un grand nombre de points qu'il conviendrait d'évaluer pour comparer deux systèmes: couverture, rappel, précision, temps de réponse, effort fourni par l'utilisateur, présentation des résultats (interface), etc.
La plupart des systèmes se basent généralement sur les mesures de *rappel* et de *précision*. Pour caractériser un système de filtrage, nous présentons quelques mesures ou métriques de performance standard utilisées dans le domaine de traitement de l'information automatisée.

La figure II.7 ci-dessous présente les différents cas qui peuvent se présenter lors du processus de filtrage : exemple de filtrage de documents.

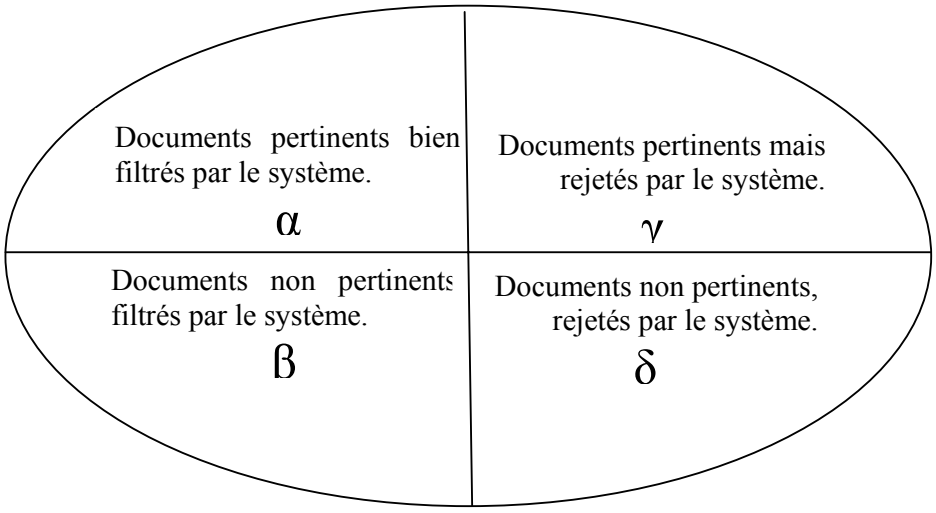


Figure II.7: les différents cas possibles lors du processus de filtrage

- Précision : c'est la proportion des documents pertinents bien filtrés par le système par rapport au nombre total de documents filtrés par ce dernier.

$$\text{Précision} = \frac{\alpha}{\alpha + \beta}$$

- Rappel (Recall) : c'est la proportion des documents pertinents bien filtrés par le système par rapport à ce qu'il aurait dû filtrer au meilleur des cas.

$$\text{Rappel} = \frac{\alpha}{\alpha + \gamma}$$

- Fallout : c'est la proportion des documents filtrés comme étant pertinents mais qui ne le sont pas, par rapport à l'ensemble des documents qui ne devaient pas être filtrés par le système.

$$\text{Fallout} = \frac{\beta}{\beta + \delta}$$

- Mesure_F : elle est définie par :

$$mesure\_F = \frac{2 * \text{Précision} * \text{Rappel}}{\text{Pr}\acute{e}cision + \text{Rappel}}$$

La précision et le rappel sont les fonctions à maximiser pour un système. Ils varient de manière inversement proportionnelle, plus le rappel est grand et plus la précision décroît [PIL 00]. En effet, en tentant par exemple d'améliorer la précision, c'est-à-dire d'augmenter la performance du système à ne trouver que des documents pertinents (autrement dit sa performance à filtrer les documents non-pertinents) on empiète nécessairement sur sa capacité à trouver le plus grand nombre possible de documents pertinents, étant donné que pour ce faire on doit réduire la fenêtre des possibilités, et non l'augmenter. Par exemple, une solution est de choisir parmi les solutions qui augmentent le rappel, celle qui diminuera la précision dans une moindre mesure [COR 97]. L'examen simultané de ces deux mesures constitue une procédure d'évaluation standard des systèmes de recherche et de filtrage d'information.

- Fonction d'Efficacité (Utilité) : elle permet de calculer la qualité du filtrage [HUL 98] en tenant compte de la différence entre les textes pertinents filtrés et non pertinents filtrés. Elle est définie comme suit :

$$F(c) = A*R+ - B*N+$$

Où:
C : corpus de textes,
A : nombre de textes pertinents,
R+ : nombre de textes pertinents filtrés,
B : nombre de textes non pertinents,
N+ : nombre de textes pertinents filtrés.

2.3.3 Programmes d'évaluation

Le département de la défense américaine (DARPA) a encouragé plusieurs programmes d'évaluation de systèmes d'analyse de textes. Ces programmes ont joué un rôle prépondérant dans le développement du domaine de traitement de l'information automatisée. En effet, ces programmes, en regroupant des équipes internationales (domaine public et privé), ont eu pour ambition de confronter des approches de techniques différentes et comparer les performances des systèmes grâce à des métriques communes sur des données normalisées. Nous présentons les principaux programmes d'évaluation au niveau anglophone et francophone.

a)- Conférences MUC (Messages Understanding Conferences)

Les conférences américaines, MUC, sont organisées par l'ARPA (Advanced Research Projects Agency) pour évaluer et suivre l'évolution des performances des systèmes d'extraction d'information. Il s'agit de déterminer le bruit (information extraite d'une manière erronée) et le silence (information non extraite). La première conférence (MUC-1) était organisée en 1987. L'évaluation dans MUC porte sur des textes provenant de différents domaines : récits d'attentats (MUC-3, MUC-4), dépêches d'agence de presse (MUC-5, MUC-6). De MUC-1 à MUC-5, la tâche consistait à remplir un formulaire (template) prédéfini unique allant d'une dizaine de champs (MUC-2) jusqu'à cent (MUC-5). Les performances furent bonnes (rappel 57 % et précision 64 %), mais le temps d'adaptation des systèmes à la tâche a posé problème (06 mois). De plus, tous les systèmes utilisaient des techniques robustes et de type (pattern matching). A partir de MUC-6 (1996), l'objectif était d'encourager le développement de systèmes génériques, portables et indépendants du domaine d'application. Le but est aussi de promouvoir une compréhension en profondeur des textes.

b)- Conférences TREC (Text REtrieval Conference)

Conférences destinées pour la recherche d'information. L'objectif est de filtrer des textes de manière fine et en prenant en compte les aspects multilingues. Chaque évaluation comporte un nombre fixe de requêtes qui doivent être appliquées à un corpus donné. Une requête comporte différents éléments (titre, description, explication, négation, etc.). L'évaluation dans TREC se fait à l'aide d'indicateurs *rappel* et *précision*. La première conférence TREC était organisée en 1992 et en est à la neuvième édition. Le filtrage d'information a connu des débuts très hésitants : il n'était considéré que comme une recherche exploratoire. Il n'apparaît qu'à partir de la 4ème édition de TREC comme une tâche de sélection binaire de documents, initialement confondu avec la tâche de routage (scores de pertinence).
Les conférences TREC s'en sont tenues aux techniques de filtrage par le contenu, indépendamment de conditions d'utilisations réelles : notamment la diversité des besoins en information, les interactions entre utilisateurs (filtrage collaboratif) et la prise en compte de l'évolution des centres d'intérêt.

c)- TIPSTER

Programme, lancé en 1990, pour évaluer des systèmes de résumé de textes. Il mettait l'accent sur le recours à des techniques statistiques pour la sélection de documents, phase considérée comme essentielle et devant précéder toute autre technique plus poussée, traitement automatique du langage naturel notamment.

Parmi les expériences francophones, nous citons les projets suivants: le projet européen, ECRAN, pour l'extraction automatique d'information, multi domaines et multilingue. Le système bilingue EXIBUM (français, anglais) de traitement de dépêches d'agences de presse sur les attentats en Algérie [KOS 98], développé à l'université de Montréal. Le projet, AMARYLLIS**,** lancé par l'INIST-CNRS et l'AUPELF-UREF pour l'évaluation des systèmes de recherche d'information.

3. Approches pour le filtrage d'information

Cette partie est consacrée à l'étude des approches dominantes en filtrage d'information. Belhin et Croft [BEL 92] définissent le filtrage comme suit : « Etant donné un objet du flux de données entrant et un ensemble de profils, le filtrage consiste à sélectionner les meilleurs couples objet-profils ». Différentes méthodes ont été proposées, utilisant le principe de mise en correspondance de la représentation du document et de celle des profils.

3.1 Méthodes classiques

Ce sont des méthodes qui mettent en œuvre des concepts statistiques, ainsi elles se basent dans leur analyse sur la fréquence (ou la présence) des mots constituant le profil utilisateur dans les documents à filtrer.

3.1.1 Filtrage par chaînes de caractères (Fulltext)

C'est une des méthodes les plus élémentaires dans le domaine du filtrage d'informations, c'est une méthode directe qui consiste, en se basant sur le parcours du texte, à sélectionner tous les documents contenant des chaînes de caractères précises (mots clés, expression booléenne, etc.) [GUT 94]. Très simple à mettre en œuvre, cette méthode donne cependant des résultats médiocres car elle ne peut détecter la ressemblance entre documents et profils s'ils ne partagent pas les mêmes mots ce qui va exclure des documents intéressants. De plus, cette approche ne prend pas en considération la fréquence d'apparition de ces termes dans les documents ce qui diminue considérablement sa précision.

3.1.2 Filtrage par langage restreint

Ce type de méthodes repose sur l'idée qu'il est beaucoup plus facile de filtrer des documents construits à partir d'un « vocabulaire » prédéfini et restreint [JIA 93]. Il s'agit d'effectuer une indexation des documents à filtrer en utilisant le langage préalablement défini. De même les profils utilisateurs ne doivent contenir que des termes tirés de ce langage. Le principal défaut de cette approche est sa lourdeur et la complexité d'élaboration à la fois du langage et des programmes de translation.

3.1.3 Filtrage par regroupements (Clustering)

Le « Clustering » est le processus de division d'un ensemble d'objets en plusieurs sous-ensembles d'objets appelés « Clusters » [TAK 97]. Cette approche consiste à comparer un ensemble de documents (selon des critères, par exemple : pertinent/non pertinent, court/long, etc.) et de le diviser en clusters. Il est à noter que le regroupement peut nécessiter une connaissance préalable des différents documents à filtrer. Comme à l'inverse, aucune connaissance précise des documents à regrouper, n'est nécessaire.
Il existe plusieurs algorithmes de regroupement dont les principales différences sont le degré de complexité, les performances et les propriétés des regroupements. Néanmoins nous pouvons distinguer deux grandes familles d'algorithmes de regroupement qui sont :

- Algorithmes de regroupement par lots (*Batch clustering algorithms*) : ils nécessitent la présence de l'ensemble d'objets au complet pour effectuer les regroupements.

- Algorithmes de regroupement progressif :(*Incremental clustering algorithms*) : ils effectuent les regroupements au fur et à mesure que les objets constituant l'ensemble de départ sont connus.

Le deuxième type d'algorithme est plus adapté au problème de filtrage d'informations, car les objets à regrouper (documents) constituent un flux continu et ne sont par conséquent pas connus préalablement.

3.1.4 Méthodes booléennes

Le modèle booléen est l'un des modèles les plus utilisés. Il se base sur la comparaison exacte entre le profil et les documents. Dans ce modèle, l'utilisateur exprime ses profils par des mots qui doivent exister ou ne doivent pas exister dans le document à recevoir. Le système sélectionne les documents qui satisfont une expression logique sur les termes du profil. Les opérations de base pour ce modèle sont les connecteurs logiques : ET (AND), OU (OR) et SAUF (NOT). Par exemple, le profil exprimé par l'expression logique P= (« intelligence » OU « raisonnement ») ET « ARTIFICIEL », permet de sélectionner tous les documents contenant le terme « artificiel » et un des termes « intelligence » ou « raisonnement ».
De nombreux systèmes de recherche d'information reposent sur ce modèle [APT 94] [COH 96a]. Parmi les premiers systèmes intégrant (basés sur) ce modèle, nous citions : SMART and SIRE (Salton & McGill, 1983) en recherche d'information. L'un des premiers systèmes de filtrage basé sur ce modèle est le système SIFT [YAN 00] destiné aux serveurs de news.
De nombreuses méthodes basées sur ce modèle ont été développées. Elles différent entre elles par les algorithmes utilisés pour comparer un document et un ensemble de profils. Nous citons les algorithmes suivants [YAN 93] :

a)- BFM (Brute Force Method)

Chaque élément du profil est recherché dans le document à filtrer. Le document correspond à un profil si tous les éléments de ce dernier apparaissent dans le document. Le processus est réitéré pour tous les profils. Cette méthode est simpliste et présente l'inconvénient de vérifier tous les profils.
L'algorithme est décrit comme suit (algorithme II.1) :

```
Structure de donnée
Tconcept : liste de concepts ;
        Profil : Tableau de TConcept ;
        Document : TConcept ;
Algorithme BFM
BFM(Document):
        Pour i :=1 à Taille(Profil) faire
            vérifiée := vrai ;
            p:= Profil[i] ;
            TantQue ((vérifiée) et (p<>nil)) faire
              Si Document.Appartient(p.concept) alors p := p.suivant
                                                Sinon   vérifiée := faux ;
            FTQ
            Si vérifiée alors « Le profil[i] correspond au document d» ;
        FPour
Fin.
```

Algorithme II.1 : Algorithme BFM

b)- CM (Counting Method)

Pour chaque élément, est associé une liste inversée constituée des profils contenant celui ci. Ce qui permet d'éviter de parcourir tous les profils. Par contre, un profil de *k* éléments apparaît dans *k* listes inversées. L'algorithme est décrit comme suit (algorithme II.2) :

```
Structure de donnée
        Total : Tableau d'entier init taille de chaque profil.
        Count : Tableau d'entier init 0 ;
            Occurrence :    Tableau de concepts ;
            Répertoire :    Tableau de   structure
                                concept ;
                                li : pointeur vers ListeInversée ;
        ListeInversée : Tableau de    structure
                            Profil : Tprofil ;
        Suivant : pointeur vers ListeInversée
Algorithme CM
    Pour i :=1 à Taille (Occurrence) faire
        p :=EntréeRépertoire(Occurrence[i]).li ;
        TantQue p<>nil faire
            Count[p.profil]=Count[p.profil]+1;
            p :=p.suivant ;
        FTQ
    FPour
    Pour i :=0 a taille(Total) faire
        Si Total[i]=Count[i] alors « Le profil i correspond au document d » ;
    FPour
Fin.
```

Algorithme II.2: Algorithme CM

c)- KM (Key Method)

Dans cette méthode, un profil de *k* éléments apparaît dans une seule liste inversée d'un élément, constituant ainsi une clé. La structure d'une liste inversée sera composée d'un champ identificateur du profil, longueur du profil ainsi que les éléments le constituant excepté la clé. L'algorithme est décrit comme suit (algorithme II.3) :

```
Structure de donnée
        Occurrence : Tableau de concepts ;
        Répertoire : Tableau  de      structure
                                      concept ;
                                      li : pointeur vers liste inversée ;
        ListeInversée : Tableau de    structure
                                      Profil : Tprofil ;
                                      Longueur : entier ;
                                      Lconcept Liste de concept ;
Algorithme KM
        Pour i :=1 a taille(Occurrence) faire
                p:= EntréeRepertoire(Occurrence[i]).li ;
                TantQue  (p<>nil) faire
                        Vérifiée := vrai ;j=1
                        TantQue (( j<=p.longeur) et( verifiée))  faire
                                Si non(appartient(p.lconcept[j])      alors vérifiée := faux
                                                                      Sinon j=j+1 ;
                        FTQ
                        Si vérifiée      alors  « Le profil p.Profil correspond au document d »
                                         Sinon p :=p.suivant ;
                FTQ
        FPour
Fin.
```

Algorithme II.3 : Algorithme KM

d)- TM (Tree Method)

Considérons un profil *P* de *k* éléments. La suite $(w_1, w_2, \ldots, w_i)$ est appelée préfixe de P et $(w_{i+1}, w_{i+2}, \ldots, w_k)$ est appelée postfixe de P, avec $0 \le i \le k$. Par exemple, *(), (a),* et *(a,b)* sont des préfixes du profil *(a,b)* et *(a,b), (b), ()* sont respectivement les postfixes correspondant. Un préfixe $(w_1, w_2, \ldots, w_i)$ identifie P si $i = k$ ou lorsqu'il n'y a pas de profils ayant $(w_1, w_2, \ldots, w_k)$ comme préfixe. Notons qu'un préfixe doit identifier un et un seul profil. Le plus petit préfixe identifiant un profil constitue son préfixe identificateur. Un identificateur est organisé en une structure d'arbre, la racine est au niveau 0. Un nœud *n* au niveau i correspond au préfixe $\sigma = (w_1, w_2, \ldots, w_i)$, ses fils sont des nœuds correspondant au préfixe $(w_1, w_2, \ldots, w_k, v)$. Il est composé des champs suivants :

- *Fils* : liste de paire $(v, P_n(v))$ où $v = w_{i+1}$ et $P_n(v)$ est un pointeur vers le nœud fils.

- *Profil* : contient un profil ayant σ comme préfixe, nil s'il en n'existe pas.

- *Longueur* : longueur du postfixe de σ.

- *Postfixe* : contient les concepts du postfixe.

L'algorithme est décrit comme suit (algorithme II.4) :

```
Structure de donnée
        Pile : pile d'élément ;
        Racine : Liste de concepts ;
Algorithme TREE
        Pile.Empiler(racine) ;
        TantQue  (non(Pile.pilevide()) faire
                p=Pile.Dépiler()
                Si (Document.Appartient(p.v)) alors
                  Si P.Profil <> NULL alors « le profil P.profil correspond au document »
                                        Sinon vérifié :=  vrai ;
                j := 1 ;
                TantQue ((vérifié) and (j < P.Longueur)) faire
                        Si Document.Appartient(P.Postfixe[j]   alors  j:= j+1
                                                               sinon  vérifié := faux ;
                Ftq
                Si vérifié alors « le document décrit le profil P.profil »
                          sinon Pile.Empiler(p.fils.P(v))
        Ftq
Fin.
```

Algorithme II.4 : Algorithme TM

Cette méthode est intéressante lorsque plusieurs profils partagent des éléments similaires. Ceci permet de réduire la taille des profils.

L'avantage du modèle booléen repose principalement sur sa simplicité, mais il souffre particulièrement de fortes limitations [HUL 94]:

- C'est difficile (voire impossible) de déterminer la différence entre les termes les plus importants et ceux qui ne le sont pas, car tous les termes sont traités de la même manière et ont le même degré d'importance.

- La définition des profils reste limitée à la constitution de listes de mots clés à reconnaître (sac de mots), sur lesquelles des opérations de logique booléenne sont effectuées.

- Les documents pertinents dont la représentation ne correspond qu'approximativement au profil ne sont pas sélectionnés. En effet, les résultats de filtrage sont peu intuitifs : par exemple, si le profil est T1 ET T2 ET T3 ET T4, seuls les documents contenant ces quatre termes sont sélectionnés, mais pas ceux contenant trois d'entre eux.

- Les documents sélectionnés sont restitués dans un ordre quelconque, et généralement il n'y a pas moyen de les classer.

3.1.5 Méthodes utilisant la logique floue [MIC 99]

Ce modèle est une extension du modèle booléen. Les profils sont exprimés par des mots combinés par des conjoncteurs de coordination dans des équations booléennes. Une fonction d'association, des termes des profils avec les documents, est définie.
La fonction peut être binaire (logique booléenne) et indiquer si le terme apparaît ou non dans le document. Elle peut aussi être calculée suivant le nombre d'occurrences du terme dans le document. A partir de cette fonction d'association, nous pouvons calculer pour chaque profil une valeur de son association avec le document. Une valeur 0 indiquera qu'il n'existe aucune association, une valeur 1 indiquera une association certaine, et une valeur entre 0 et 1 indiquera une association partielle.
L'utilisateur peut spécifier un seuil à cette valeur à partir duquel les documents sont jugés acceptables. Ces valeurs sont calculées en redéfinissant les équations booléennes en logique floue.

Exemple : soient deux termes (*Ta, Tb)* d'un profil donné et (*A, B) les* fonctions d'associations des termes *Ta* et *Tb*. Le calcul de la valeur d'association du profil avec un document est calculée selon l'équation booléenne : Si l'équation booléenne est de la forme (*Ta AND Tb*) alors la valeur de (*A AND B*) est calculée par le minimum de *A* et *B*. Si l'équation booléenne est de la forme (*Ta OR Tb*) alors la valeur de (*A OR B*) est calculée par le maximum de *A* et *B*, et si elle est de la forme (*NOT Ta*), nous calculons *1-A*.

3.1.6 Méthodes vectorielles

a)- Vector Space

Dans ce modèle, les profils et les documents sont identifiés par un ensemble de termes ou mots clés et représentés dans un espace euclidien à la dimensionnalité donnée par la taille du vocabulaire. Un document est représenté par un sous-ensemble des mots qu'il contient, considérés comme des descripteurs suffisamment fiables du contenu du document. Le vocabulaire étant l'effectif total de mots différents contenus dans l'ensemble de documents considéré (base ou corpus).
Contrairement au système booléen, les mots sont représentés avec des poids, se fondant sur des mesures de fréquence, qui permettent de distinguer leur degré d'importance.
Le modèle vectoriel est plus intéressant, il tente de quantifier et d'évaluer le niveau de pertinence reliant deux documents.
Cependant, il nécessite la définition des espaces vectoriels, en plus les notions d'ordre des constituants ainsi que la structuration textuelle (ex. : phrases, paragraphes, chapitres, etc.) ne sont pas prises en compte. Par exemple, un document comme '*A horse is better than a car*' est regardé de la même manière que '*A car is better than a horse*'. Il ne permet pas de représenter l'aspect sémantique. Les documents sont considérés comme « sacs de mots » typographiques[4].

[4] Toute séquence de caractères délimitée par deux séparateurs typographiques : espace, ponctuation.

Par ailleurs, dans ce modèle, tous les mots n'ont pas le même statut. En effet, les mots très fréquents (articles, prépositions, etc.) sont considérés comme peu porteurs d'information. De ce fait, ils sont généralement supprimés. De plus, les différences entre majuscules et minuscules ne sont généralement pas prises en compte, ce qui entraîne la considération des entités nommées (ex. : les noms propres) de la même façon que les autres mots[5].
Les documents sont classés en fonction de leur proximité avec le profil. Autrement dit, il s'agit essentiellement de mesurer la distance entre deux vecteurs : celui représentant le profil (ex. : requête) et celui d'un document à filtrer. L'ensemble des documents est ainsi trié en fonction d'une métrique de distance calculée entre le vecteur profil et chaque vecteur document. Parmi les métriques de mesure vectorielles les plus utilisées nous citons : la distance euclidienne, la mesure du cosinus, etc. (chapitre 3).
Les variantes de ce modèle reposent sur des algorithmes propriétaires destinés à optimiser les phases d'indexation et de calcul de similarité, ou en fixant, de façon plus ou moins empirique, des seuils en dessous desquels les documents ne sont plus considérés comme pertinents.
La détermination du nombre de documents qui vont être retournés à un utilisateur reste un problème crucial. Parmi les directions futures de la recherche dans le domaine du filtrage est la détermination de ce nombre.
Une solution est de fixer le nombre arbitrairement. Il y a risque d'avoir beaucoup de documents non intéressants pour une période donnée, et d'ignorer certains documents intéressants pour une autre. Une autre solution est de laisser l'utilisateur spécifier un seuil de similarité qui peut modifier ultérieurement (selon les résultats qui lui arrive).
En déça de l'aspect statistique, les principes fondateurs de ce calcul de similarité sont simplement ancrés dans le sens commun. Intuitivement, il est clair qu'une requête contenant un mot rare w_r ainsi qu'un mot commun w_c sera plus probablement reliée à un document contenant le mot rare w_r (et d'autres mots) qu'à un autre document contenant le mot commun w_c (mais pas le mot rare w_r). Le mot rare ayant une importance relative plus grande que le mot commun, il jouera simplement un plus grand rôle dans le mécanisme. Cette intuition se trouve formalisée dans l'évaluation heuristique de la similarité nommée *tfidf* (chapitre 3), qui la raffine et permet de l'intégrer dans différents algorithmes ayant prouvé leur pertinence en permettant d'obtenir des résultats raisonnables dans différents contextes d'évaluation [JAU 01].

b)- Latent Semantic Indexing

C'est une extension de la méthode vectorielle standard. Elle tente de représenter le sens d'un document par un ensemble de concepts (un concept est la forme canonique correspondant à une classe de mots ou de syntagmes) déduits à partir des mots contenus dans le document et qui serviront à la mesure de la pertinence.
Le principe de la méthode consiste à construire une matrice *A* (termes-documents), qui est ensuite décomposée et réduite en lui appliquant une méthode mathématique appelée *SVD* (singular value decomposition) en un espace des concepts [FOL 90]. La matrice *A* est décomposée donc en un produit de trois matrices (figure II.12): $A = U\Sigma V^T$ où les matrices U et V sont orthogonales, et Σ une matrice diagonale dont les éléments ont une valeur positive ou nulle.

[5] Exemple : V. Poutine, Président actuel de la Russie, est considéré de la même façon que la « poutine », spécialité québécoise.

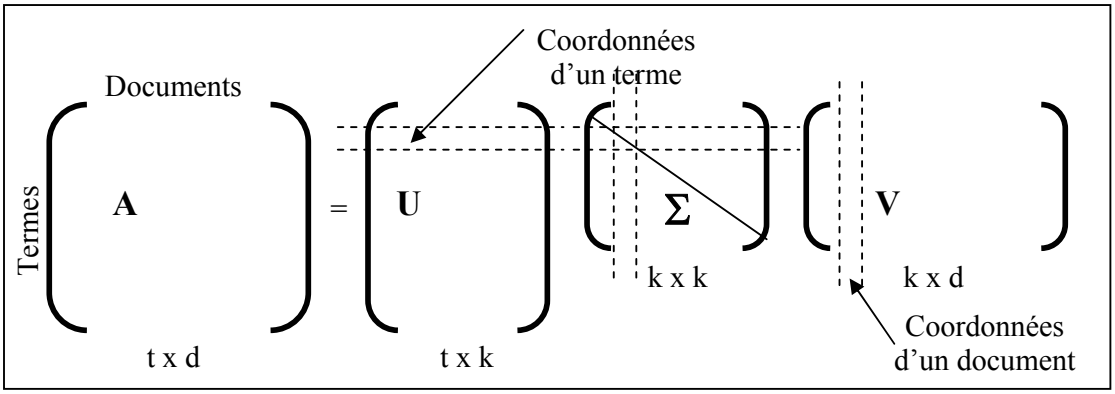


Figure II.8 : Décomposition en valeurs singulières

La matrice Σ est ensuite simplifiée en ne retenant que les axes correspondants aux *k* valeurs les plus élevées (intuitivement : on ne garde que les axes les plus importants et on néglige les autres). On peut alors interpréter la décomposition comme établissant la relation entre les documents et les termes au travers d'un espace de *k* concepts : chaque document est décrit par des concepts (dans la matrice *U*), concepts dont l'importance est indiquée par les éléments de la matriceΣ, et ses concepts décrivent dans l'espace des termes par la matrice *V*.
Voici un algorithme récapitulatif (algorithme II.5) du procédé de décomposition en valeurs singulières [BER 92] :

1. Soit *A mxn* la matrice document ayant pour terme général $A = [\, a_{ij} \,]$, où a_{ij} est le poids du terme *i* dans le document *j* .
2. Soit *r* le rang de la matrice *A* tel que $r < MIN\ (m\ ,\ n\ )$.
3. Notons par s_i les valeurs singulières de *A* et qui constitueront les éléments de la matrice diagonale S_0 .
4. Notons par u_i les vecteurs singuliers gauches de la matrice *A*, ils constitueront la matrice T_0 de sorte que $T_0 = (u_1\ u_2\ u_3\ u_4\ ...)$
5. Enfin notons ici par v_i les vecteurs singuliers droits de la matrice *A*, ils constitueront la matrice D_0 de sorte que $D_0 = (v_1\ v_2\ v_3\ v_4 ...)$
6. L'algorithme commence par la construction de la matrice *B* à partir de *A*

$$B = \begin{pmatrix} 0 & A \\ {}^{t}A & 0 \end{pmatrix}$$

7. La matrice *B* est une matrice creuse de taille $(n+m * n+m)$.
8. L'algorithme itératif de Lanczos est utilisé pour le calcul de valeurs propres : Les valeurs propres (λ_i) de *B* et les vecteurs propres (x_i) associés.
9. Les valeurs singulières de $A : s_i = | \lambda_i |$.
10. Les *m* premières composantes de chaque vecteur propre de *B* constituent un vecteur singulier gauche de *A*, et les n restants constitueront le vecteur singulier droit correspondant.

Algorithme II.5 : Algorithme de décomposition en valeurs singulières

Une fois cette décomposition effectuée il est possible de comparer soit des termes, soit des documents ou encore des termes avec des documents, et cela à travers l'espace des concepts.
La similarité entre deux documents, appelée « *similarité conceptuelle* », se calcule par le produit scalaire de leurs représentations dans l'espace des concepts.
Si un document est représenté par le vecteur X de ces termes (une matrice *t*x*1*), sa représentation dans l'espace des concepts sera le vecteur de taille *1*x*k*.
Cette méthode *LSI* est une méthode très intéressante par son aspect sémantique, car elle ne voit pas les documents comme des termes mais comme des concepts sémantiques. Elle a prouvé une meilleure performance. Elle permet de sélectionner des documents même s'ils n'ont pas de mots communs avec le modèle utilisateur.
Mais l'opération de mise à jour de l'espace des concepts est coûteuse en termes de calcul. Elle nécessite (1) la disponibilité d'un corpus pour construire la matrice termes-profils et (2) un temps d'exécution important pour la méthode *SVD* pour donner un résultat assez satisfaisant. Par conséquent, pour économiser et ne pas dégrader les performances du système, cette opération pourrait s'accomplir régulièrement pendant des périodes creuses.

3.1.7 Méthodes probabilistes

Le modèle probabiliste consiste à calculer la probabilité qu'un document soit pertinent ou non par rapport au profil (besoin en information). Cette pertinence est estimée à partir d'une collection de documents d'apprentissage [MIC 99]. Ce modèle utilise la théorie de décision de Bayes (chapitre 3).
La méthodologie pour mettre en œuvre une méthode probabiliste (un modèle probabiliste) consiste à (figure II.9) :

- Identifier le problème à résoudre (par exemple, développer un étiqueteur grammatical).

- Définir les paramètres : modéliser le problème en faisant apparaître les probabilités de certains évènements qui constituent les paramètres du modèle.

- Apprentissage à partir de données (un corpus) : estimer des valeurs de probabilités par collecte de fréquences ou un algorithme d'apprentissage tel que Baum-Welch.

- Utiliser le modèle (probabilités estimées) pour traiter de nouvelles données.

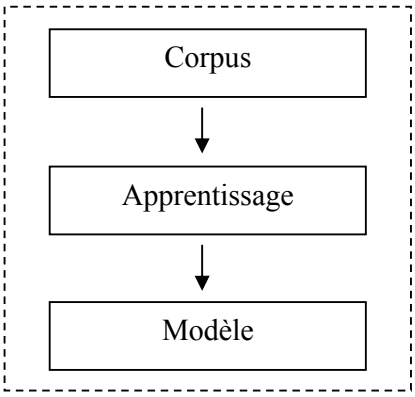


Figure II.9 : Une approche probabiliste

L'avantage d'un modèle probabiliste est sa capacité d'apprentissage (efficacité). Les inconvénients sont : vision mathématique dans la conception du modèle et difficultés de la mise en œuvre (complexité de la programmation et problème d'estimation des probabilités). En effet, la qualité des modèles probabilistes et de la performance des applications qui les utilisent dépendent de la disponibilité de vastes corpus de données et du progrès technologique (performance des ordinateurs en terme de stockage et de calcul).
Un des types de modèles probabilistes les plus utilisés est le modèle de Markov : il est simple et efficace. Les travaux fondateurs pour les modèles probabilistes sont ceux de Marron et Kuhns [MAR 60], Robertson et Sparck-Jones [ROB 76] et de Croft et Harper [CRO 79]. Par la suite, de nombreuses approches probabilistes ont été proposées dans le domaine de la recherche d'information : par exemple, le modèle proposé par Turtle et Croft [TUR 91] qui utilise un réseau d'inférence Bayésien pour inférer la probabilité que la requête soit satisfaisante par un document.
Une autre approche développée par Ponte et Croft [PON 98] qui consiste à construire un modèle par document et estime la probabilité que la requête ait pu être générée par chacun de ces modèles. Les documents sont ensuite ordonnés par rapport à leur probabilité. Plus récemment, de nombreuses techniques basées sur les modèles de Markov cachés (MMC) ont été introduites [MIL 99].
Par exemple, dans le modèle de recherche et d'extraction d'information, proposé par Hugo Zaragoza et Patrick Gallinari [ZAR 98], l'extraction d'information est basée sur un modèle stochastique, modèle de Markov cachés (MMC) : il est défini par deux processus stochastiques : une chaîne de Markov définie par un ensemble d'états (concepts) et les transitions (grammaire de concepts) entre ces états, des probabilités dites d'émission associées à chaque état, qui donneront la probabilité de générer une observation dans un état donné.
Le modèle MMC permet de trouver la séquence de concepts les plus probables pour une séquence d'observation (mots).
D'ailleurs, des versions un peu plus sophistiquées de ce modèle MMC ont obtenues des résultats satisfaisants dans TREC-6.

3.2 Méthodes symboliques

3.2.1 Filtrage par règles

C'est un filtrage très simple, basé sur une liste de termes appélés mots clés. Généralement, dans ce type de filtrage, on a recours à un système expert(SE) qui utilise un ensemble de règles et de directives introduites préalablement par l'utilisateur (figure II.10).

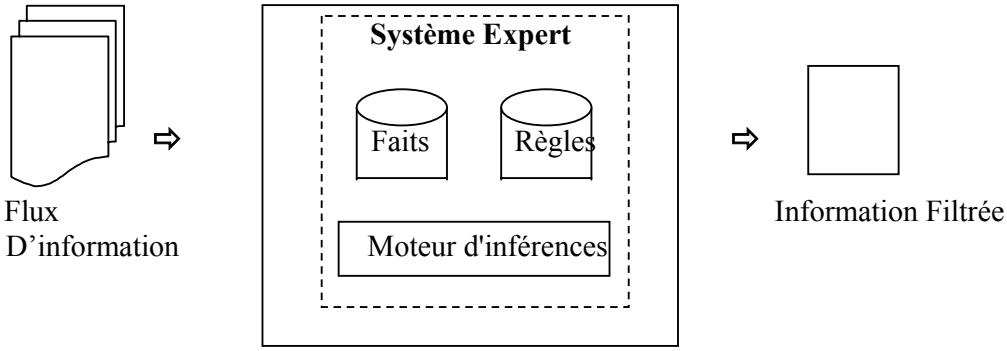


Figure II.10: Filtrage par Système Expert

Les connaissances du SE sont organisées sous forme de règles de production de la forme:
Si *(suite de conditions)* ***Alors*** *(suite d'actions)*

Par exemple, pour filtrer des E-mails, les conditions portent sur les différents champs du courrier (*From, Subject, to, date*, etc.) et les actions à entreprendre sont: sauvegarder ou classer, supprimer, générer une réponse automatique, etc.
La figure II.11 montre un exemple de règles dans le système ISCREEN [POL 88].

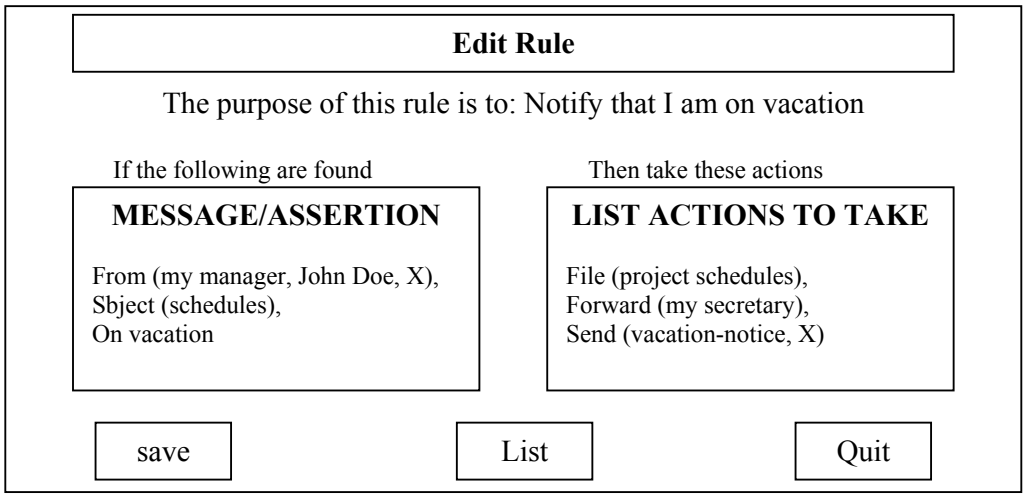


Figure II.11: Exemple de règles dans le système ISCREEN

Si l'utilisateur reçoit un message provenant de son directeur ou de John Doe et le champ *subject* contient le mot clé *« schedules »* et de plus l'utilisateur est en vacances, alors le message doit être sauvegardé dans le répertoire du projet « *schedules »* , envoyer une copie à sa secrétaire et répondre à l'expéditeur *X* que l'utilisateur est en vacances.
Un autre exemple de systèmes, utilisant les mots clés (expéditeur, sujet, etc.) pour créer des règles pour filtrer les messages électroniques, est le système « the Information Lens System » [MAC 89].
L'avantage d'utiliser un système expert est qu'il peut justifier les décisions concernant l'opération de filtrage auprès de l'utilisateur en lui présentant l'ensemble des règles ayant contribué à la prise de chaque décision. Ce qui permet à l'utilisateur, à tout moment, de mettre à jour la base de règles. Mais l'utilisation de mots clés pour décrire les règles de filtrage peuvent conduire le système à sélectionner certains messages non pertinents. En effet, les mots clés ne sont pas suffisants pour décrire le contexte. Les termes utilisés dans les messages ne sont pas forcement les même que ceux utilisés par un usager pour décrire ses règles de filtrage : problème de vocabulaire. D'ailleurs Furnas, Landauer, Gomez et Dumais ont montré qu'il y a une très faible probabilité (0.1 à 0.2) que deux personnes utilisent le même mot clé pour décrire le même objet [FUR 87].

3.2.2 Filtrage textuel linguistique

Le filtrage textuel est la mise du texte sous une forme particulière en éliminant ou en extrayant un certain nombre d'informations. Nous parlons de filtrage textuel linguistique lorsque le filtrage est basé sur des critères linguistiques tels que l'analyse morpho-lexicale, syntaxique et l'analyse sémantique.

a)- Filtrage morpho-syntaxique

Ce type de filtrage concerne, en général, des textes structurés tels que les documents SGML. Des grammaires formelles telles que les grammaires hors contexte sont très utilisées pour représenter ce type de textes. Ces grammaires permettent de représenter la structure hiérarchique, l'ordre, l'optimisation, les alternatives et les structures récursives des éléments du texte.

b)- Filtrage syntaxique

Le but est de filtrer l'information en se basant sur des techniques de l'analyse syntaxique. Chandrasekar et Srinivas [CHA 98] supposent que les textes cohérents contiennent des informations « latentes » significatives, comme la structure syntaxique, qui peut être utilisée pour améliorer la performance des systèmes de recherche et de filtrage d'information [CHA 98]. Pour eux, le filtrage consiste à éliminer certaines informations (ex : documents) qui ne répondent pas aux besoins de l'utilisateur. L'idée de base est de trouver des indicateurs syntaxiques qui permettent de réaliser cette opération de filtrage (éliminer les documents non pertinents).
Pour ce faire, une analyse syntaxique de surface (shallow parser) est utilisée pour produire les groupes syntaxiques élémentaires correspondant à la structure de la phrase [BES 02].
Un exemple de telle approche est le système GLEAN, développé par Chandrasekar et Srinivas [CHA 98]. Il consiste en deux phases (figure II.12):

- Une phase d'apprentissage qui consiste à sélectionner un échantillon d'apprentissage constitué d'un ensemble de phrases pertinentes pour le domaine d'intérêts. Ensuite, une description syntaxique est associée aux mots de ces phrases. A la sortie de cette phase, nous aurons un ensemble de descripteurs ou schèmes de pertinence du domaine d'intérêt.

- Une phase d'application (recherche d'information), qui consiste à utiliser ces descripteurs pour identifier des régularités contextuelles et éliminer (filtrer) l'information non utile.

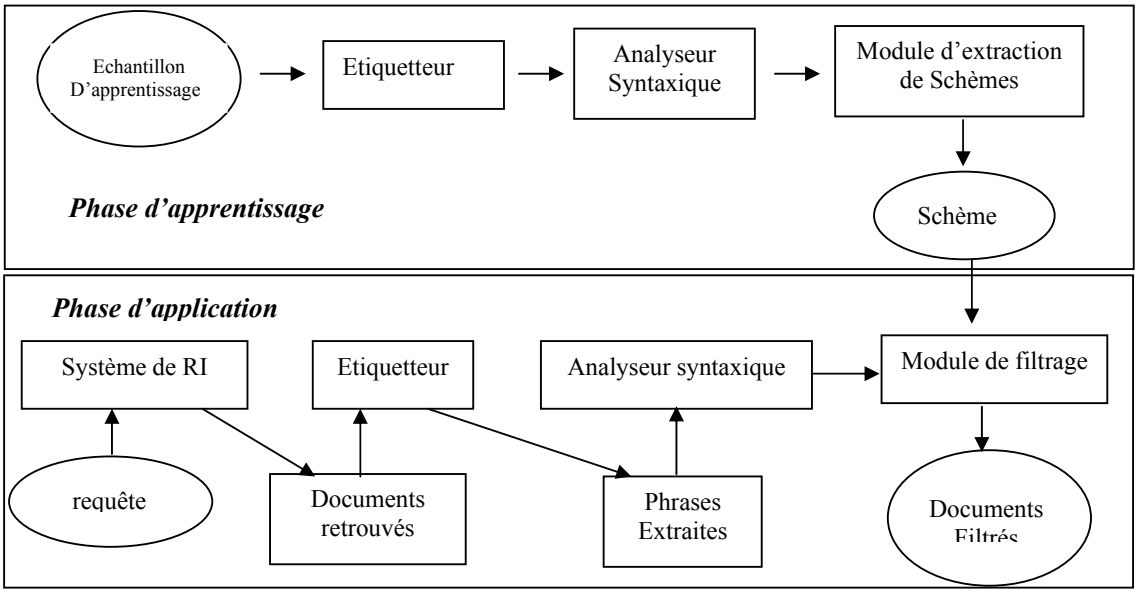


Figure II.12 : Système GLEAN

Foltz et Dumaiz précisent qu'il existe des informations latentes dans les descripteurs syntaxiques des mots de chaque document [FOL 92]. Cette structure latente peut être estimée à l'aide de certaines techniques statistiques, telles que la méthode LSI [FOL 90].
LSI permet de relier des termes même s'il n'existe pas de lien visible entre ces termes. En effet, deux termes sont utilisés dans des contextes similaires, ils vont avoir des représentations similaires dans LSI. Ce qui permet de relier un document à un profil même s'ils n'ont pas de mots clés en commun.
Une application pratique de ce type de filtrage est par exemple le calcul des fréquences de co-occurrences, on parle alors de *filtrage syntaxique de co-occurrences* [BES 02]. Il s'agit d'utiliser, dans la définition des contextes, une information supplémentaire sur la syntaxe. L'approche la plus simple est de considérer soit un contexte *positionnel* (fenêtre de taille donnée), soit un contexte *documentaire* (phrase ou paragraphe par exemple), mais s'avère insuffisante. En effet, prenons par exemple le contexte représenté par la phrase suivante :
« L'acteur porte un masque grimaçant de théâtre antique »
Après identification des unités linguistiques qui composent la phrase, on obtient un graphe des relations de co-occurrences suivant :

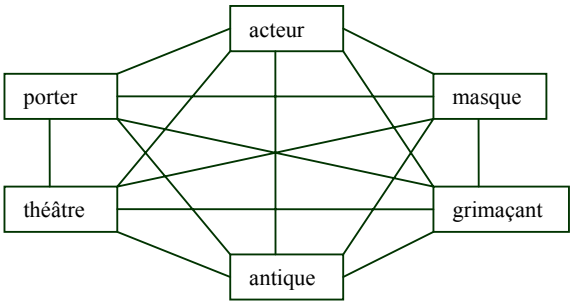


Figure II.13 : Graphe de co-occurrences possibles

Nous constatons que certaines co-occurrences semblent moins pertinentes pour être considérées (*acteur-antique* et *théâtre-grimaçant*).
Deux approches possibles pour le *filtrage syntaxique des co-occurrences* : *Filtrage par les groupes syntaxiques* et *sélection par les relations syntaxiques*.
La première consiste à éliminer certaines co-occurrences non souhaitées. Il s'agit de considérer seulement les co-occurrences entre unités linguistiques appartenant à un même groupe syntaxique ou entre têtes de différents groupes syntaxiques (en pratique, la tête d'un groupe nominal est le nom de ce groupe, et la tête d'un groupe verbal est le verbe principal de ce groupe). Pour cela, une analyse syntaxique de surface (shallow parser) est utilisée pour découper la phrase en groupes syntaxiques élémentaires. On obtient pour la phrase précédente, le découpage suivant :

(le + acteur) (porter) (un masque grimaçant) (de théâtre antique)

Le graphe des co-occurrences est donné comme suit :

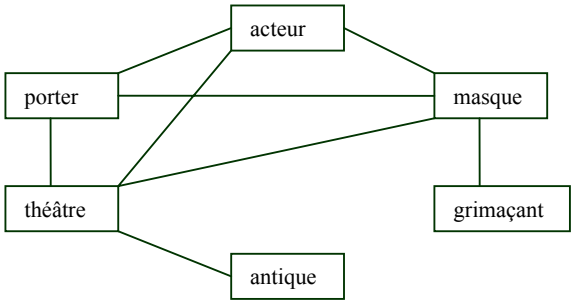


Figure II.14 : Filtrage par groupes syntaxiques

La deuxième consiste à sélectionner que les co-occurrences pertinentes (syntaxiquement fondées). Elle repose sur les résultats d'une analyse syntaxique produisant différentes relations syntaxiques entre les unités linguistiques de la phrase de type sujet-verbe (S), verbe-objet (O), complément de nom (CN), ou qualification d'un nom par un adjectif (A). Il s'agit seulement de considérer les co-occurrences entre les unités reliées par une relation syntaxique. Les relations syntaxiques de la phrase précédente sont :

S (acteur, porter) O (porter, masque) A (masque, grimaçant)
A (théâtre, antique) CN (masque, théâtre)

Le graphe de co-occurrences est comme suit :

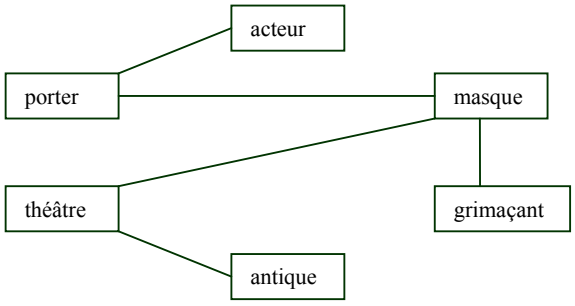


Figure II.15 : Filtrage par relations syntaxiques

c)- Filtrage sémantique

Pour passer à une analyse sémantique efficace d'un texte, il faut que tous les problèmes d'ordre syntaxique soient d'abord résolus. Pour éviter une analyse syntaxique rigoureuse, certains chercheurs préfèrent proposer des approches qui intègrent des notions sémantiques telles que la causalité ou l'expression temporelle.

- Causalité : le filtrage consiste à représenter un texte par des relations causales entre les situations exprimées dans ce texte. La notion de causalité est organisée sémantiquement par des indicateurs linguistiques (les verbes qui véhiculent cette notion) sous forme de modèle sémantique.
Un exemple de système utilisant cette notion est le système COATIS [GAR 98]. Il analyse des textes d'un domaine quelconque pour élaborer une structuration des relations causales repérées dans ces textes. Pour cela, il utilise un modèle linguistique sur l'expression de la notion de causalité en français (25 relations causales, par exemple : /créer/empêcher/faciliter/pousser à/etc.)[GAR 98]. La notion de sémantique représentée par

l'information causale est utilisée dans différents domaines d'application, tels que la modélisation d'un domaine [GAR 99], structuration d'une terminologie, filtrage automatique des textes et l'indexation automatique de documents [GAR 98] [ASS 98].

- Expression temporelle : le filtrage consiste à repérer les unités temporelles pertinentes dans un texte. Ce type de filtrage est utilisé dans les textes bien spécifiques où l'expression temporel (date, durée, etc.) est importante (par exemple en droit). Un des systèmes utilisant cette notion est celui développé par Faiz [FAI 98]. Il utilise une première analyse (filtrage 1) pour extraire les phrases contenant les indicateurs temporels (jours, mois, dates, etc.). Ensuite, une deuxième analyse (filtrage 2) pour extraire des phrases de la première analyse, celles contenant des opérateurs temporels (de, avant le, à compter du, etc.) et des fonctions temporelles (premier jour de, troisième jour du, etc.).

- Exploration contextuelle [DES 93] : par rapport aux applications classiques du traitement automatique du langage naturel, les systèmes d'exploration contextuelle sont des systèmes d'analyse de textes autonomes et économiques. Ils ne nécessitent pas une compréhension complète du texte. Ils s'appuient fortement sur des indices pertinents présents dans les textes. Ces indices sont indépendants d'un domaine de compétence particulier et correspondent à un savoir linguistique (marques linguistiques : mots vides, mots grammaticaux). Parmi les systèmes développés, nous citons le système SEEK (Système Expert d'Exploration Contextuelle) qui fonctionne à l'aide de règles d'exploration contextuelle, dont le but est de rechercher dans les textes des indices textuels qui permettent d'identifier des relations entre les objets d'un domaine de compétence.

3.3 Réseaux de neurones

Ce modèle a pour but d'imiter et de reproduire certaines fonctions « intelligentes » du cerveau humain, comme l'apprentissage, la mémorisation, la reconnaissance et l'adaptation. Il consiste à représenter les informations sous forme d'un réseau et permettre au système de faire évoluer ce réseau par la fonction d'apprentissage. Les nœuds du réseau représentent les concepts et les arcs représentent les associations entre les concepts.

3.3.1 Définition

Un réseau neuronal est une structure mathématique capable d'apprendre depuis un grand nombre d'exemples et de modéliser une série de comportements (Sorties) lorsque le réseau est soumis à des entrées. A chaque nouvelle entrée, le modèle propose un comportement résultant des exemples qu'il a appris.
Un réseau neuronal est composé d'éléments appelés neurones, connectés entre eux de manière analogue à l'architecture du cerveau humain. Chaque neurone étudie l'entrée et renvoie un comportement aux neurones auxquels il est connecté (*sortie*).

3.3.2 Modélisation Formelle

La première modélisation d'un neurone formel (artificiel) est celle présentée par Mc Culloch et Pitts dans les années quarante (figure II.16). La fonction d'un neurone consiste à effectuer une somme pondérée des entrées, puis il s'active suivant la valeur de cette sommation

pondérée. Si cette somme dépasse un certain seuil, le neurone est activé et transmet une réponse et dans le cas contraire il ne transmet rien [DAV 93].

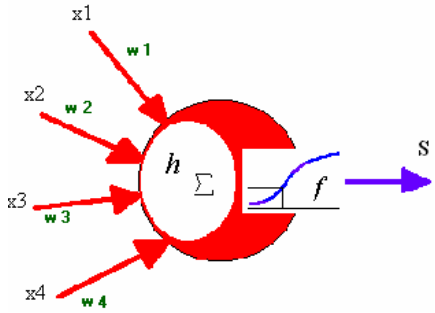


Figure II.16 : représentation formelle d'un neurone

D'une façon générale, un neurone formel est défini par les éléments suivants :

- Les entrées x_i du réseau : elles peuvent être représentées par des attributs à valeurs réelles ou symboliques, les attributs pouvant être dépendants ou non.

- Une fonction d'entrée *h*, qui fait le pré-traitement sur les entrées x_i. A chaque entrée x_i est associé un poids (ou coefficient synaptique) w_i. Il représente son degré d'importance.
La *fonction d'entrée* peut être booléenne, linéaire (somme pondérée des entrées $h(x_1,x_2,\ldots,x_n)=\sum_i W_i\, x_i$), affine ou polynomiale.

- Une fonction d'activation (ou de transfert) *f*, qui est appliquée à la fonction d'entrée *h*. Elle peut être :

* Binaire à seuil : Fonction de *Heaviside* : $H(x)=\begin{cases}1 & x > \text{seuil}\\ 0 & \text{sinon}\end{cases}$

Fonction *signe* : $Sign(x)=\begin{cases}1 & x > \text{seuil}\\ -1 & \text{sinon}\end{cases}$

* Fonction *linéaire* : $F(x)=\alpha x$ avec $\alpha>0$

* Fonction *linéaire par morceaux* : $H(x)=\begin{cases}1 & x >= 0.5\\ x+0.5 & [-0.5,+0.5]\\ 0 & \text{sinon}\end{cases}$

* Fonction *sigmoide* : $F(x)=\dfrac{e^{kx}}{e^{kx}+1}=\dfrac{1}{1+e^{-kx}}$ où *k* représente la pente.

- Une fonction de sortie *S* : la sortie d'un neurone peut être déterministe ou probabiliste. Un neurone réalise simplement une somme pondérée de ces entrées, ajoute un seuil à cette somme et fait passer le résultat par une fonction de transfert pour obtenir sa sortie (peut être réelle ou discrète).

L'architecture d'un réseau peut être: soit *sans rétroaction* (la sortie d'un neurone ne peut influencer son entrée), soit *avec rétroaction* (totale ou partielle).

La dynamique d'un réseau peut être : soit synchrone (tous les neurones calculent leurs sorties respectives simultanément), soit asynchrone (séquentiel ou aléatoire).

3.3.3 Modèles

Nous distinguons deux types de modèles de réseaux (figure II.17): récurrent et non-récurrent.

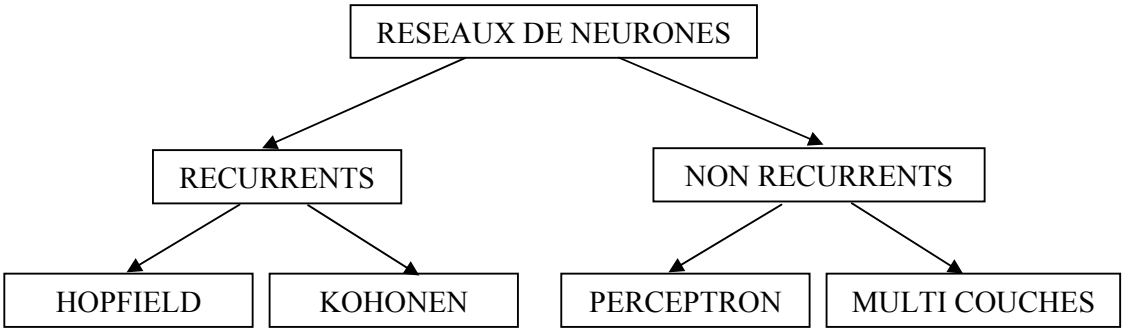


Figure II.17 : Les modèles de réseaux les plus connus

Les réseaux récurrents sont des graphes orientés qui comportent des boucles ou de chemins circulaires. Parmi les réseaux récurrents, nous citons le modèle de Hopfield [DAV 90], développé en 1982, utilise des réseaux totalement connectés basés sur la règle de Hebb et le modèle de Kohonen [DRE 02] [KOH 84], développé deux ans plus tard, utilise un algorithme non supervisé basé sur l'auto-organisation.
Les réseaux non récurrents sont des graphes orientés qui se caractérisent principalement par l'absence de boucles ou de chemins circulaires. Parmi les réseaux non récurrents, nous citons le modèle Perceptron et le modèle multi couches [DRE 02] [DAV 93]. Le Perceptron, premier modèle de base pour lequel un processus d'apprentissage a pu être défini. Il a été développé par Frank Rosenblatt en 1958. Un perceptron est constitué d'un seul neurone. Il prend en entrée n valeurs $x_1,..., x_n$ et calcule une sortie o. Un perceptron est défini par les coefficients synaptiques $w_1, ..., w_n$ et le seuil (ou le biais) θ (figure II.18).

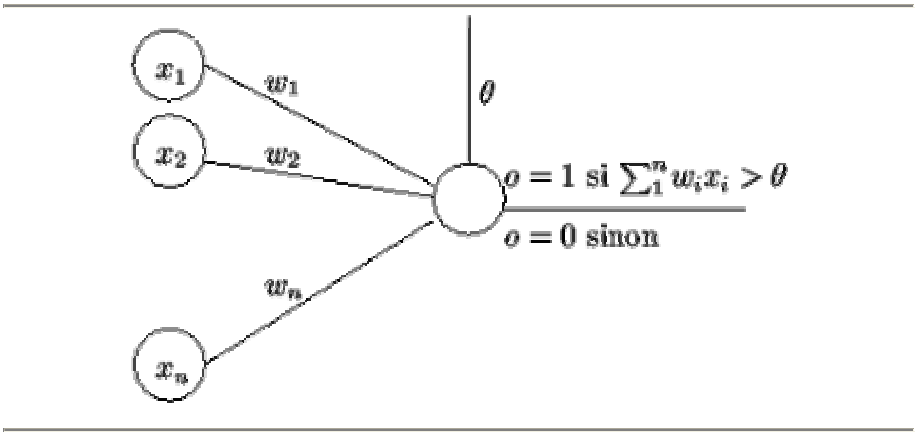


Figure II.18 : Perceptron

Les entrées $x_1,..., x_n$ peuvent être à valeurs dans {0,1} ou réelles, les poids peuvent être entiers ou réels. La sortie o est calculée par la formule :

$$o = \begin{cases} 1 & \sum_i W_i \, x_i > \theta \\ 0 & \text{sinon} \end{cases}$$

Une variante très utilisée de ce modèle est de considérer une fonction de sortie prenant ses valeurs dans {-1,1} plutôt que dans {0,1}. Il existe également des modèles pour lesquels le calcul de la sortie est probabiliste.
Du point de vue géométrique, un perceptron linéaire à seuil divise l'espace des entrées en deux sous-espaces délimités par un hyperplan. Réciproquement, tout ensemble linéairement séparable peut être discriminé par un perceptron. Un perceptron est bien adapté pour des échantillons linéairement séparables. Cependant, dans la plupart des problèmes réels, cette condition n'est pas réalisée.
Le Modèle multi-couches (à couches cachées) est défini par une architecture vérifiant les propriétés suivantes :

- Les cellules sont réparties de façon exclusive dans des couches C_0, C_1, ..., C_q,

- La première couche C_0 est la rétine composée des cellules d'entrée qui correspondent aux n variables d'entrée ; les couches C_1, ..., C_{q-1} sont les couches cachées ; la couche C_q est composée de la (ou les) cellule(s) de décision,

- Les entrées d'une cellule d'une couche C_i (avec i >= 1) sont toutes les cellules de la couche C_{i-1} et aucune autre cellule.

La dynamique du réseau à couches cachées est synchrone et son architecture est sans rétroaction. Un modèle de réseaux de neurones très utilisé est le modèle PERCEPTRON MULTI-COUCHES (PMC). Les cellules élémentaires sont des perceptrons linéaires à seuil. Il est caractérisé par :

- Les neurones sont déterministes, sortie réelle, calculée à l'aide de la fonction sigmoide,

- Une dynamique asynchrone séquentielle,

- Architecture sans rétroaction, les neurones sont organisés en 3 couches (figure II.19) : la couche des stimulus ou des entrées (E), la couche des neurones intermédiaires ou cachés (C) et la couche des comportements ou des sorties(S).

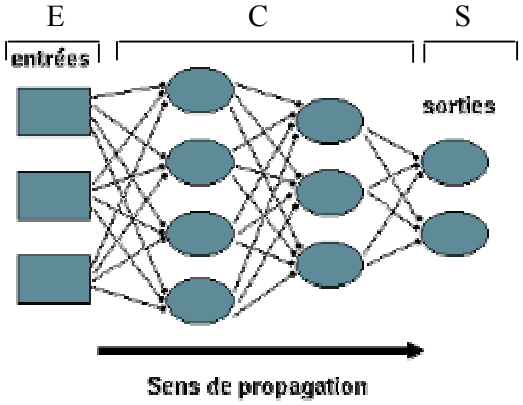


Figure II.19 : Réseau multicouches

Les neurones de la première couche (E) sont reliés au monde extérieur et reçoivent tous le même vecteur d'entrée (l'entrée du réseau). Ils calculent alors leurs sorties qui sont transmises aux neurones de la deuxième couche (N), etc. Les sorties des neurones de la dernière couche (S) forment la sortie du réseau.
L'utilisation des réseaux multi-couches, nécessite deux choses indispensables :

- Une méthode indiquant comment choisir une *architecture* de réseau pour résoudre un problème donné. C'est-à-dire, pouvoir répondre aux questions suivantes : combien de couches cachées ? Combien de neurones par couches cachées ? Ceci constitue un sujet de recherche actif. Le choix d'une architecture est très important : un réseau de neurones, ayant un bon pouvoir de généralisation, nécessite une architecture assez riche, mais en même temps, même difficulté que pour les arbres de décision, risque d'une sur-spécialisation (apprentissage "par coeur"). Le choix de l'architecture est une tâche pas facile, sans expérience approfondie. Il est fait, soit par l'expérience, soit par des essais successifs. Quelques algorithmes d'apprentissage ont été proposés dont le rôle est double : apprentissage de l'échantillon avec un réseau courant, et modification du réseau courant, en ajoutant de nouvelles cellules ou une nouvelle couche, en cas d'échec de l'apprentissage.

- Une fois l'architecture choisie, un algorithme d'apprentissage qui calcule, à partir de l'échantillon d'apprentissage, les valeurs des coefficients synaptiques pour construire un réseau adapté au problème. Pour cela, en général, on découpe l'échantillon en trois ensembles: un ensemble d'apprentissage *A* (pour choisir l'architecture, apprendre et régler les paramètres), un ensemble de test *T* (pour estimer la qualité du réseau produit, c'est-à-dire choisir l'architecture et le réglage ayant obtenu les meilleures performances sur *T*) et un ensemble de validation *V* (pour estimer l'erreur réelle en relançant l'apprentissage sur *V*).

Les réseaux de neurones possèdent plusieurs propriétés intéressantes comme celles d'interagir avec des donnés parasitées par du bruit, de s'adapter aux circonstances, de tolérer une certaine erreur. Les réseaux de neurones excellent dans les domaines où l'on ne peut pas exprimer des solutions algorithmiques. Ils sont utilisés avec succès dans beaucoup de domaines où les méthodes conventionnelles échouent comme pour les problèmes de classification et les problèmes d'approximation qui tolèrent une certaine marge d'imprécision.
Une caractéristique des réseaux de neurones est le fait qu'ils ne sont pas programmés, mais ils apprennent à l'aide d'exemples. C'est une approche radicalement différente de la méthode traditionnelle qui implique le développement de logiciels (programmation). Dans un programme informatique, chaque pas que l'ordinateur exécute est spécifié à l'avance par le programmeur. Par contre, le réseau de neurones commence avec des échantillons d'entrée et de sortie, et apprend à donner la réponse correcte pour chaque entrée.
Les réseaux de neurones constituent un modèle dynamique et évolutif, capable d'apprendre et de modifier progressivement son comportement au fur et à mesure de ses utilisations. L'apprentissage est une nécessité pour s'adapter à un environnement évolutif. Une fois l'apprentissage achevé, ce dernier se comporte comme une « boîte noire » à laquelle on peut soumettre de nouvelles données, et qui fournit les réponses appropriées.
Cependant l'un des principaux reproches fait aux réseaux de neurones tient de leur incapacité à expliquer les résultats qu'ils fournissent. Ils sont définis par une architecture et un grand ensemble de paramètres réels (les coefficients synaptiques). Les réseaux se présentent comme des boîtes noires. De plus, il semble assez facile de concevoir des algorithmes qui classent correctement l'échantillon, mais beaucoup plus difficile d'en obtenir qui aient un bon pouvoir de généralisation.

3.4 Méthodes collaboratives

Malone présente trois aspects de filtrage : cognitif, économique et social [KIL 97b]. Le filtrage cognitif concerne l'évaluation du contenu, le filtrage économique concerne les ressources consommées (par exemple, pour le filtrage de documents : la taille, le temps de lecture, etc.), et le filtrage social ou collaboratif basé sur les opinions des autres utilisateurs.

3.4.1 Filtrage collaboratif

Un système de filtrage collaboratif consiste à faire profiter une communauté d'utilisateurs des efforts d'évaluation produits de manière individuelle par ses membres, en acheminant l'information (ex: les documents) jugée intéressante par les uns, aux autres utilisateurs dont les intérêts sont proches.
Cette collaboration bouleverse l'architecture classique des systèmes de filtrage d'information. Il faut donc repenser les systèmes de filtrage d'informations afin de rompre avec leur aspect monolithique et mono utilisateur. Le filtrage d'information est alors plus orienté utilisateur et tient compte des besoins en information d'un groupe que ce soit pour un filtrage synchrone, presque synchrone ou asynchrone. En effet, une approche collaborative signifie que l'on offre la possibilité aux utilisateurs de s'entraider, chacun pouvant bénéficier des compétences des autres ainsi que de leur résultat. Par exemple, dans le filtrage collaboratif de documents, les utilisateurs insèrent leurs remarques à l'intérieur du document (par exemple, dire qu'un document est intéressant ou pas) et ces remarques dites annotations sont accessibles par les filtres des autres utilisateurs. La modification des profils se base donc sur les annotations des autres utilisateurs.
La collaboration en filtrage d'information présente à priori beaucoup d'intérêt, car elle doit permettre aux utilisateurs de:

- Gagner en qualité du résultat obtenu en bénéficiant des connaissances et compétences des autres participants.

- Gagner du temps en réutilisant les évaluations des autres participants.

- Gagner en satisfaction, le travail collectif étant souvent plus valorisant et plus agréable qu'un travail individuel. Cela permet une mise en commun des résultats, et surtout, une analyse et une comparaison de la pertinence des documents filtrés.

La particularité du filtrage collaboratif, est de ne pas tenir compte du contenu des documents et de ne se baser que sur les évaluations des documents faites par les utilisateurs. Dans un tel système, l'utilisateur est partie prenante du fonctionnement du système, et le processus de filtrage ne peut bien fonctionner que si suffisamment d'utilisateurs participent (problème de la masse critique). Un certain nombre de systèmes de filtrage collaboratif existent, mais de nombreuses questions restent sans réponse. Il faut donc évaluer la qualité du filtrage offerte par un système, inciter les utilisateurs à faire vivre le système, le problème essentiel qui se pose est d'assurer un bon niveau de filtrage tout au long de la vie du système. Ce dernier problème suppose de concevoir des outils de contrôle du processus de filtrage afin de maintenir une bonne qualité de filtrage tout au long de son utilisation.
Un exemple de système, utilisant le concept du filtrage collaboratif, est le système D-SIFTER (Distributed Smart Information Filtering Technology for Electronic Resources) [MUK 97]. Il consiste en un ensemble de sous systèmes de filtrage interconnectés qui communiquent pour

effectuer une classification collaborative de documents. Chaque sous système est doté d'un profil et d'un thésaurus spécifiques lui permettant de filtrer ses propres documents. Il est constitué de deux modules : apprentissage et classification. Chaque sous système local peut aider un autre sous système distant à filtrer les documents. De plus, il peut lui aussi demander de l'aide pour filtrer ses propres documents.
Kohrs & Mérialdo ont utilisé une approche de classification automatique (approche hiérarchique) dans le cadre du filtrage collaboratif [KOH 99]. Le filtrage collaboratif consiste à sélectionner l'opinion la plus probable d'un certain nombre d'utilisateurs pour évaluer et filtrer l'information. Cette approche de classification automatique permet d'avoir une fonction de prédiction ou fonction discriminante comparable aux résultats des méthodes classiques d'extraction de fonction discriminante telle que les moindres carrés ou le coefficient de corrélation de Pearson [CYR 02] [BOY 02]: plus il y a d'utilisateurs plus la fonction est prédictive.

3.4.2 Filtrage par agents

Une des approches possibles est l'utilisation du paradigme multi-agents [BAC 92] [MAE 94] [SHE 93]. C'est une approche excitante, car d'avant-garde. En effet, l'intelligence artificielle distribuée porte sur la distribution de l'intelligence ou l'expertise parmi plusieurs agents, non soumise à un contrôle centralisé. Elle s'intéresse aux applications complexes qui nécessitent une représentation multiple des connaissances et des tâches à entreprendre, en terme de niveaux d'abstraction mais aussi de points de vues. A cet effort de modélisation correspond une augmentation nécessaire des capacités de raisonnement et de contrôle. Il s'agit alors d'élaborer des systèmes constitués d'un groupe d'agents, chacun étant doté d'une certaine autonomie et devant être capable de planifier, d'agir et de travailler dans un environnement commun, au prix de conflits éventuels. De nouvelles notions émergent de cette distribution, telles que la coopération, la coordination d'actions, la négociation, etc.
L'émergence de cette approche a contribué grandement à l'élaboration de nouvelles méthodes capables de supporter diverses applications sur le réseau Internet, nouveau média de production et de diffusion. Parmi ces applications, nous citons les agents d'information « Internet Matching Agents » pour faire des recommandations sur les sites web, les agents de recherche d'information qui détiennent la connaissance de diverses sources d'information et permettant l'accès à leurs données, les agents de filtrage d'information qui ont pour rôle de résoudre le problème de la surcharge de l'information pour choisir les documents adéquats, et les agents de veille technologique permettant le suivi des changements de l'information sur les sites web [DRI 01].
Baclace a mis en œuvre un système de filtrage basé sur le concept d'agents [BAC 92]. Le système dispose d'une base de données d'agents. Chaque agent possède une somme d'argent et un numéro de classement dans l'intervalle des réels [-1, 1] (priorité de l'agent). Le système analyse le document et extrait un ensemble de propriétés (exp : auteur= « Descle » et mot clé= « schème »). Ensuite, il active les agents qui sont sensibles aux propriétés trouvées. Le document est ensuite estimé et classé par chaque agent activé. La priorité finale du document est donnée par la somme de tous les classements des agents activés. Chaque agent est activé par une unique propriété ou par une conjonction de propriétés.
Les agents sont en compétition à travers un modèle économique « de vente et d'achat ». A chaque fois qu'un agent est activé, il paye un coût fixe de transaction.
Après l'opération de feedback, les agents qui sont loin de la moyenne de l'évaluation de l'utilisateur sont sanctionnés. Le montant de la sanction est redistribué entre les agents qui ont une note supérieure à la moyenne.

Actuellement, certains chercheurs travaillent sur les modèles de filtrage collaboratifs. Nous citons par exemple le modèle collaboratif économique, le modèle collaboratif à une seule opinion et le modèle collaboratif à plusieurs opinions. Dans le modèle économique, chaque système (client, agent, etc.) est doté initialement d'une certaine somme d'argent. A chaque besoin d'assistance, il paie une certaine somme au système distant sollicité pour l'aider etc. Dans le modèle collaboratif à une seule opinion, un seul système distant est sollicité pour filtrer les documents. Par contre, dans le modèle collaboratif à plusieurs opinions, tous les systèmes distants sont sollicités pour l'opération de filtrage.

3.5 Modélisation des intérêts de l'utilisateur

Un système de filtrage est d'autant plus efficace qu'il fournit des moyens permettant une adaptation fine aux particularités des différents utilisateurs. En effet, quelle que soit la qualité des méthodes de filtrage proposées, leur application indifférenciée à de larges populations d'utilisateurs potentiellement hétérogènes se traduit, de la part des systèmes, par un comportement insuffisant, pénalisant pour les performances individuellement perçues par chacun des utilisateurs. La modélisation des intérêts de l'utilisateur est donc une tâche importante pour un système de filtrage de l'information. L'efficacité du filtrage est étroitement liée à cette modélisation.
La mise en pratique d'un modèle utilisateur est difficile car, s'il est relativement aisé de proposer des formalismes permettant de décrire des modèles, il est par contre particulièrement ardu de produire les modèles réels décrivant un utilisateur (ou un groupe d'utilisateurs) donné. L'utilisateur lui-même a d'ailleurs de la peine à décrire de manière formelle et explicite ses propres spécifications.
On définit un profil utilisateur par l'ensemble des données identifiant ses centres d'intérêts thématiques. Ce profil est très souvent qualifié par des suites de termes relevant d'un ou plusieurs domaines thématiques [TUR 00]. Nous présentons dans ce qui suit, quelques pistes ou approches prometteuses pour contourner cette difficulté de modélisation.

3.5.1 Étude d'observation

Pour savoir comment configurer les modèles des utilisateurs pour un système de filtrage de l'information, une étude d'observation peut être entreprise pour noter comment font les lecteurs pour décider des documents qui leur parviennent. Par exemple, l'étude menée par Lantz et Kilander, à l'université de Stokholm, a permis de donner un ordre d'importance aux différents critères susceptibles d'influer sur le processus de filtrage de messages électroniques (table II.3).

Critère	**Poids**
Expéditeur	37
Sujet	30
Date	21
Attachement	8
Destinataire	11
Langue	10
Longueur	21
Etc.	Etc.

Table II.3 : Critères de filtrage

3.5.2 Modélisation par mots clés et modélisation par documents

Les intérêts de l'utilisateur peuvent être modélisés, soit par un ensemble de mots clés (profil de mots clés), soit par un ensemble de documents (profil de document). Dans le premier cas**,** la modélisation reste extrêmement rudimentaire et prend la forme d'ensembles de mots clés, généralement, fournit par l'utilisateur ou automatiquement extraits des documents. Cette technique est ambiguë du fait qu'un mot peut avoir plus d'un sens et qu'un concept peut être décrit par plusieurs mots. Par conséquent, les mots clés ne décrivent pas correctement le contexte. Ce qui laisse le système de filtrage sélectionner certains documents non pertinents [FOL 92]. De plus, cette technique est de fait particulièrement pauvre car elle ne prend aucunement en compte la structure linguistique des textes manipulés (ordre, proximité, etc. des mots clés), elle repose uniquement sur la présence ou absence de mots clés.
Une modélisation par mots clés n'est pas suffisante, les informations contextuelles et sémantiques doivent être impliquées.
Par contre, dans le deuxième cas, l'idée de filtrage consiste généralement à créer un espace de documents jugés intéressants par un utilisateur. Et chaque nouveau document se trouvant être proche aux documents dans cet espace, est alors considéré comme pertinent. De ce fait, le profil de document fournit une représentation simple et très efficace des intérêts d'un utilisateur. De plus, l'indication d'un petit nombre de documents pertinents est aussi efficace et beaucoup plus simple que la génération d'une longue liste de mots et/ou d'expressions pour décrire, souvent difficilement, les intérêts d'une personne.

3.5.3 Relevance Feedback

Le but de la démarche est de chercher à dériver les spécificités des utilisateurs à partir de leur interaction avec le système.

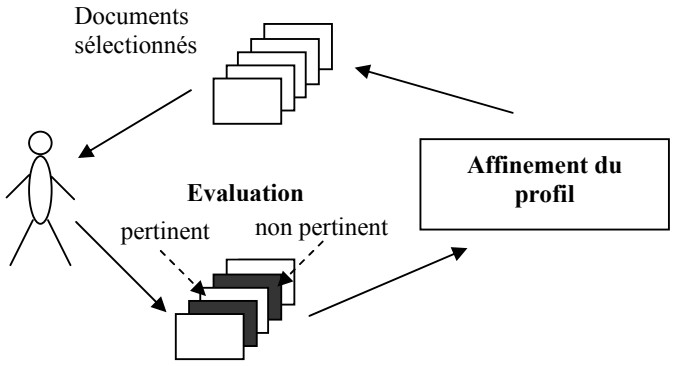


Figure II.20 : le processus de Relevance Feedback

Le processus de filtrage est décomposé en deux phases distinctes:

- Construction d'un profil initial (par exemple, une liste de mots clés). Cette première phase se traduit par la production d'une liste de documents transmise à l'utilisateur.

- Un filtrage par l'utilisateur de la liste fournie menant à l'identification d'un ensemble de documents considérés comme pertinents par l'utilisateur. Les caractéristiques de ces documents peuvent alors être utilisées pour affiner le profil initial et l'ensemble du processus peut alors être itéré jusqu'à satisfaction de l'utilisateur.

3.5.4 Annotations collaboratives

Dans certaines méthodes, par exemple le filtrage en collaboration, la construction et la modification des profils se basent sur les annotations des utilisateurs. En effet, les utilisateurs insèrent leurs remarques à l'intérieur du document (par exemple, dire qu'un document est intéressant ou pas) et ces remarques dites annotations sont accessibles par les filtres d'autres utilisateurs.

3.5.5 Anti-profil

Les systèmes actuels de filtrage se basent essentiellement sur la similarité entre profil et document. Pour améliorer les performances du filtrage, certains chercheurs utilisent une technique qui consiste à réduire et à éliminer les documents non pertinents en se basant sur des profils contenant de l'information non pertinente, appelés anti-profils [HOA 00]. Il s'agit de calculer la similarité entre l'anti-profil et les documents jugés pertinents par le profil de l'utilisateur (contenant l'information pertinente), et de rejeter tout document dont la similarité est élevée.

La figure II.21 décrit le processus de filtrage basé anti-profil.

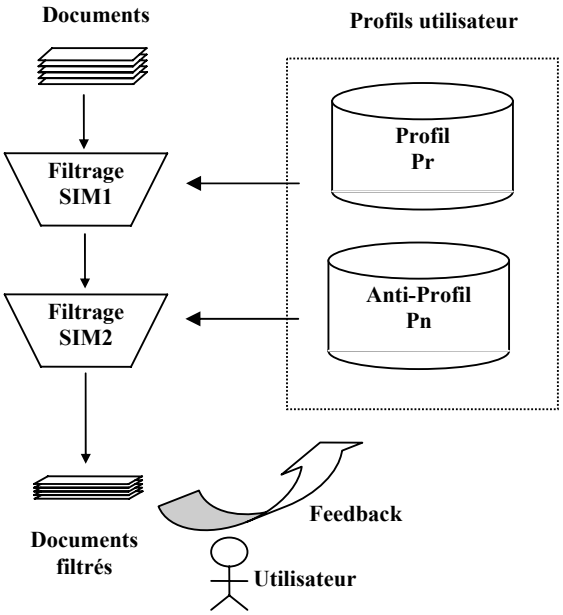


Figure II.21 : Processus de filtrage basé anti profil

Où :

Pr : profil utilisateur,

Pn : anti-profil utilisateur,

Sim1 : similarité entre un document et le profil *Pr*,

Sim2 : similarité entre l'anti-profil *Pn* et un document filtré par le profil *Pr* ,

Si la similarité entre un document et *Pn* est inférieur à un seuil alors le document est retenu et l'opération de feedback est lancée pour mettre à jour les profils *Pr* et *Pn.* Sinon, le document est rejeté.
Le problème principal est dans l'utilisation des seuils. En effet, si le seuil pour *Sim2* a une valeur très petite, il y a risque de rejeter des documents pertinents et de plus réduire le nombre d'opérations de feedback qui permet d'améliorer les profils *Pr* et *Pn*. Par contre, si le seuil a une valeur très élevée, il y a risque de retenir des documents non pertinents.
Une solution à ce problème est de lancer l'opération de feedback même pour les documents rejetés par l'anti-profil *Pn.* Cette méthode permet au seuil d'être stricte sans pour autant sacrifier l'opération de feedback [TREC-9]. D'autres approches ont été proposées pour améliorer l'opération de feedback, telles que l'approche qui consiste à attribuer un poids aux différents documents traités représentant ainsi la probabilité qu'un document soit pertinent ou non pertinent.

4 Conclusion

Les principales techniques actuelles employées dans le domaine du filtrage sont basées d'une façon directe ou indirecte sur les techniques des méthodes traditionnelles de recherche d'information: c'est seulement l'approche ou la vision qui diffère. Elles se basent sur l'occurrence d'un ensemble de mots clés pour identifier ou reconnaître l'information pertinente (modèle booléen, modèle vectoriel, etc.). L'avantage de cette approche classique (statistique) repose principalement sur sa simplicité, mais elle est basée sur une hypothèse irréaliste qui est celle que tous les mots sont complètement indépendants. En effet, les systèmes les plus répandus, ayant participer à TREC (conférence de référence) ne prennent pas en compte l'ordre des mots, et les relations de dépendances existantes entre les éléments linguistiques (mots, syntagmes, chunks, phrases, etc.). Par exemple, de tels systèmes (statistiques) ne font pas la différence entre «*le ministre de la culture*» et *«la culture du ministre».*
De plus, la mesure de la pertinence repose uniquement sur la présence ou absence de mots clés dans le texte traité.
Toute analyse (recherche de segments pertinents, etc.) effectuée sur ces bases ne peut que contenir une part d'imprécision. Par exemple, les documents pertinents dont la représentation ne correspond qu'approximativement au profil ne seront pas sélectionnés. La représentation du contenu des textes par des mots clés (automatiquement extraits ou manuellement affectés) éventuellement pondérés, est particulièrement pauvre car elle ne prend aucunement en compte la structure linguistique des textes manipulés.
Un ensemble de mots clés ne préserve qu'une faible fraction du sens du texte original. Pour toutes ces raisons, nous avançons que seul un système de filtrage d'information opérant sur des bases linguistiques est à même de garantir la qualité attendue.
Néanmoins, le domaine de filtrage de l'information reste un domaine très ouvert vers diverses autres tendances. Ainsi, certains chercheurs utilisent ces techniques traditionnelles en essayant toujours de les améliorer en proposant de nouvelles approches qui tentent d'intégrer des techniques plus sophistiquées (procédures de traitement automatique du langage naturel) permettant de conserver une part importante de la structure linguistique et de capter le plus d'information sémantique.
Les techniques utilisant l'information sémantique (méthodes de traitement du langage naturel) cherchent à améliorer les performances des systèmes de filtrage en unifiant la sémantique des textes et les profils des utilisateurs. Pour cela nous avons besoin d'un modèle sémantique qui permet de représenter les intérêts de l'utilisateur, et de faire une compréhension du texte qui

nécessite classiquement: une étape morpho-lexicale, syntaxique, sémantique et même pragmatique. Cette méthode est intéressante et donne un filtrage efficace, mais difficile à appliquer à des textes tout venant couvrant des domaines variés. Elle est avantageuse dans un domaine spécifique et se complique rapidement quand il s'agit d'une généralisation. Elle nécessite de mobiliser d'importantes ressources linguistiques (dictionnaire) et des outils de traitement automatique du langage naturel (analyseur syntaxique, grammaire, représentation textuelle).

Une autre approche intéressante pour le filtrage est l'utilisation du concept multi-agents. Il serait intéressant de montrer l'applicabilité et l'adaptabilité de l'approche intelligence artificielle distribuée, en l'occurrence les systèmes multi-agents, au filtrage du courrier électronique. Les systèmes multi-agents suscitent un intérêt dû à leur capacité d'aborder le problème de manière distribuée et d'apporter une solution réactive et robuste. En effet, les approches de conception modulaire se limitent à des applications centralisées pour la résolution de problème à flot de données statiques. Par contre, les applications telles que gérer l'ensemble des mails d'une entreprise, sont caractérisées par un flux de courrier pouvant varier de façon dynamique. Une solution pour le filtrage est la distribution du traitement. Il s'agit d'élaborer un système constitué d'un groupe d'agents, chacun étant doté d'une certaine autonomie et devant être capable de coopérer, coordonner, négocier, etc. avec les autres agents.

De ce fait et vu la diversification des tendances classiques et actuelles, il n'y a pas encore de conclusion concrète. Ainsi, la conclusion générale que l'on peut évoquer est que les méthodes les plus récentes (traitement du langage naturel, multi-agents, les réseaux neuronaux, etc.) semblent prometteuses.

Chapitre III

Apprentissage et classification automatique de textes

La propriété « intelligent » d'un système automatique (ex. un système de classification automatique) est sa faculté d'apprendre et d'améliorer son efficacité. Le but de l'apprentissage est d'acquérir de meilleures ou de nouvelles connaissances et/ou un mécanisme ou une procédure [TUR 00] [DEN 00]. Dans cette partie, nous présentons le principe général du processus d'apprentissage, les tâches, les domaines d'application, ainsi que les différentes techniques utilisées par les systèmes d'apprentissage automatique. Nous citons les approches les plus utilisées dans le contexte de la classification automatique de textes.

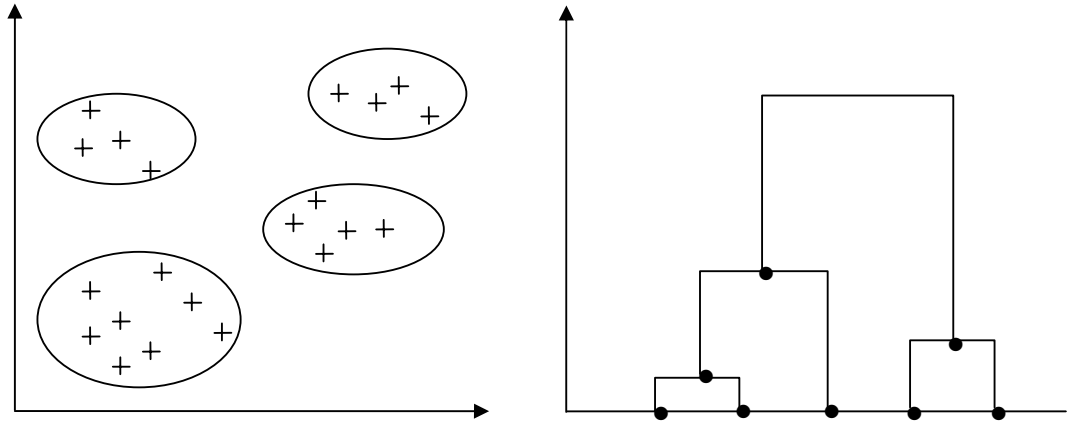
1 Système d'apprentissage

Le principe d'un système d'apprentissage consiste, à partir d'un ensemble d'exemples observés, d'induire une procédure ou une règle générale (ex. procédure de classification). La procédure générée devra traiter (ex : classifier) correctement les exemples de l'échantillon mais surtout avoir un bon pouvoir prédictif pour traiter correctement de nouvelles descriptions.
Un système d'apprentissage s'améliore avec l'expérience. Les tâches visées par l'apprentissage automatique sont : acquisition des connaissances, catégorisation (segmentation), la classification (regroupement ou clustering), amélioration de performances (devenir plus efficace), estimation, adaptation, optimisation, prédiction, etc.
La catégorisation consiste à examiner les caractéristiques d'un objet (exemple : texte) et lui attribuer une classe dans un ensemble prédéfini. Des exemples de tâche de catégorisation sont: établir un diagnostic, attribuer ou non un prêt à un client, attribuer un sujet à un article de presse, etc.
La classification consiste à former des groupes (clusters) homogènes à l'intérieur d'une population, par exemple la classification de termes (ex : extraction thématique).
L'estimation consiste à estimer une valeur d'un objet à partir de ses caractéristiques. Par exemple, attribuer une classe particulière pour un intervalle de valeurs de l'objet estimé : noter un candidat à un prêt (estimation pour attribuer ou non le prêt : classification), estimer les revenues d'un client, etc.
Les applications sont ainsi nombreuses et concernent des domaines très variés. L'apprentissage automatique est très utilisé dans le domaine de la recherche d'information et dans le domaine d'extraction d'information, et surtout avec l'avènement d'Internet qui fait

qu'une grande quantité d'information est devenue accessible. Les méthodes automatiques d'apprentissage permettent de construire des modèles complexes (beaucoup de paramètres), chose qu'il est difficile d'envisager de faire manuellement.
On évalue la performance de l'apprentissage grâce à des critères tels que : la probabilité de mauvaise classification, le risque, l'erreur quand elle est calculable, écart à un modèle attendu, avis d'utilisateurs, etc. L'évaluation et la qualité des modèles sont conditionnées par la taille de l'échantillon utilisé pour l'apprentissage.

2 Classification automatique

La classification automatique consiste à produire une procédure permettant d'associer une classe à un objet (exemple : texte). De nombreux travaux, dans le domaine d'apprentissage, ont porté sur la classification automatique de textes [BRU 02]. La classification (catégorisation) de textes est la tâche qui consiste à assigner une ou plusieurs classes à un texte donné. Les méthodes de classification utilisent l'information contenue dans le texte (mots) pour déterminer sa classe. Le texte est tout d'abord transformé en un vecteur dont les éléments représentent, par exemple, les poids des mots dans le texte.
Il existe deux grandes familles de techniques de classification : *classification supervisée* (figure III.1.a) et *classification non supervisée* (figure III.1.b).



(a) : ***Exemple de classification supervisée (Partition en 4 classes)***

(b) : ***Exemple de classification non supervisée (Hiérarchie de classes)***

Figure III.1 : Classification supervisée vs classification non supervisée

2.1 Classification supervisée

Les méthodes de *classification supervisée*, en général, posent le nombre de classes *k* comme paramètre connu et décident de l'appariement des objets en fonction des *k* classes. Les classes possibles sont connues et les objets disponibles sont déjà classés, servant d'ensemble d'apprentissage. Le problème est alors d'être capable d'associer à tout nouvel objet sa classe la plus adaptée, en se servant de ces exemples déjà classés. L'apprentissage, appelé *supervisé*, nécessite donc, d'une part, une phase « inductive » consistant à développer les règles d'identification à partir d'exemples de l'ensemble d'apprentissage et, d'autre part, une phase « prédictive » visant à utiliser ces règles pour identifier de nouvelles instances. Il existe une multitude de méthodes supervisées développées dans différents contextes, par exemple, les

méthodes probabilistes, les arbres de décision, les réseaux de neurones, etc. Ces méthodes ont le défaut, pour la plupart de fixer un nombre de classes à l'avance.

2.2 Classification non supervisée

Les méthodes de *classification non supervisée* sont caractérisées par la non disponibilité d'aucune autre information préalable que la description des exemples. Elles sont destinées à produire des groupements d'objets, selon un critère de similarité ou dissimilarité à partir d'une description sur ces objets (traits, caractéristiques, propriétés, etc.) : sont des méthodes de structuration.
L'apprentissage, appelé *non supervisé*, se base sur le principe d'observation. En effet, à partir d'un tableau de données (tableau de valeurs numériques, un tableau de contingence ou tableau de présence absence), on suppose que certains regroupements doivent exister ou au contraire, on exige que certains regroupements soient effectués sous forme de partitions ou classes.
Le processus de classification consiste à former des groupes (clusters ou classes) homogènes à l'intérieur d'une population. Pour cette tâche, nous ne cherchons pas à expliquer une classe définie à priori ou prédire une valeur (cas de la *classification supervisée*). Il s'agit de regrouper dans un même cluster les objets considérés comme similaires. Dans ce cas, le problème est alors de définir cette similarité entre objets. Typiquement, la similarité entre objets est estimée par une fonction calculant la distance entre ces objets. Une fois cette fonction distance définie, la tâche de clustering consiste à réduire au maximum la distance entre membres d'un même cluster, tout en augmentant au maximum la distance entre clusters.
Il existe trois grandes familles de ce type de techniques: classification par recherche directe, classification ascendante et classification descendante.

3 Analyse et représentation du contenu des textes

L'analyse de textes, contrairement à la génération de textes, vise à extraire leur représentation. Elle reçoit, en entrée, un texte et lui associe, en sortie, une ou plusieurs représentations. La génération de textes est considérée comme l'opération inverse de l'analyse, qui part d'une représentation de texte et fournit, en sortie, un texte en langage naturel.

3.1 Les modèles de représentation de textes

La représentation du texte est le résultat de l'analyse et le point de départ de la génération (figure III.2).

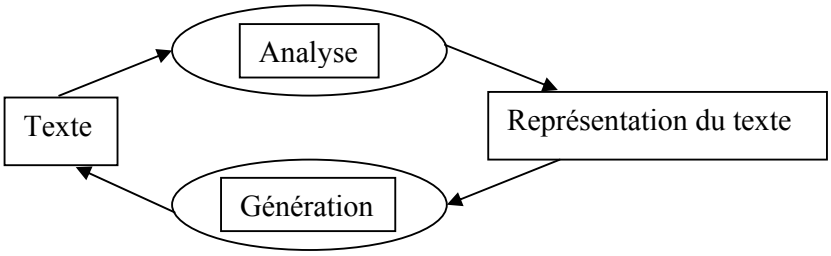


Figure III.2 : Processus d'analyse et de génération

Nous distinguons deux types de représentation : représentation linguistique et représentation non linguistique.

3.1.1 Représentation non linguistique ou « sac de mots »

La description d'un texte est représentée par un ensemble de termes (mots simples) indépendants (sans structure) : le mot est utilisé comme unité de représentation. Le texte est ainsi perçu comme un « sac de mots ou bag of words». Cette représentation porte le nom de 'sac de mots'. Elle se contente d'extraire d'un texte un ensemble de mots clés. Elle n'utilise pas d'information grammaticale ni d'analyse syntaxique des mots : seule la présence ou l'absence de certains mots est porteuse d'informations. Elle a été introduite dans le cadre du modèle vectoriel : les textes sont transformés simplement en vecteurs dont chaque composante représente un terme [SAL 75] [SAL 83] [MAN 99]. Ces vecteurs sont fournis par des prétraitements simples. On commence généralement par éliminer les mots grammaticaux (articles, prépositions, etc.) et par réduire les variantes morphologiques à une forme commune (souvent appelé terme). Puis on compte les occurrences des termes les plus importants de manière à représenter chaque texte par un vecteur dans l'espace des termes. Un corpus de textes donne donc lieu à une matrice texte-terme (table III.1).

	Terme 1	***Terme 2***	***Terme 3***	***..***	***Terme n***
Texte 1	$Freq_{11}$	$Freq_{12}$	$Freq_{13}$	..	Fre_{1n}
Texte 2	$Freq_{21}$	$Freq_{22}$	$Freq_{23}$	..	Fre_{2n}
..	..	..	..	..	..
Texte m	$Freq_{1m}$	$Freq_{1m}$	$Freq_{1m}$	..	Fre_{mn}

Table III.1: Matrice texte-terme

Dans certains systèmes, la description d'un texte considère chaque flexion d'un mot comme un descripteur différent. Par exemple, les différentes formes conjuguées d'un verbe sont considérées comme des descripteurs différents alors qu'elles ont *a priori* le même sens.
Une solution à ce problème, est de considérer uniquement la racine des mots plutôt que les mots entiers (on parle de *stem* en anglais). Plusieurs algorithmes ont été proposés pour substituer les mots par leur racine [HULL 96]; par exemple, l'algorithme de Porter [ZWE 03] [POR 80] pour l'anglais.
D'autres systèmes utilisent une analyse grammaticale afin de remplacer les verbes par leur forme infinitive et les noms par leur forme au singulier : la lemmatisation. Par exemple, l'algorithme efficace, nommé *TreeTagger* [SCH 94], 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.

La substitution des mots par leur racine ou leur lemme réduit l'espace de représentation et permet de représenter par un même descripteur des mots qui ont le même sens. Par exemple, remplacer des mots tels que *bank*, *banks*, *banking* dans un texte par l'unique racine *bank* et des formes conjuguées telles que *franchit, franchi, etc.* par le lemme *franchir*. Néanmoins ces substitutions peuvent augmenter l'ambiguïté des descripteurs en représentant par un même descripteur des mots avec des sens différents.

3.1.2 Représentation linguistique

Le texte est décrit par un ensemble d'informations implicites de différents niveaux. Le niveau est déterminé par le type d'information qui figure dans la représentation. L'approche nécessite donc des connaissances linguistiques. Nous distinguons les représentations suivantes : morphologique, syntaxique, sémantique et pragmatique.

a)- *La représentation morphologique (relative aux mots):* chaque mot du texte est représenté par : sa forme de base (forme canonique ou lemme), sa catégorie syntaxique et les informations grammaticales (nombre, genre, personne).

b)- *La représentation à base d'unités syntaxiques (relative aux phrases):* la description d'un texte est représentée par un ensemble de structures plus complexes (structures syntaxiques) associées aux termes. Contrairement à l'approche « sac de mots », les mots simples sont considérés comme termes si ils vérifient certaines contraintes linguistiques.
Cette représentation s'intéresse à la structure syntagmatique des phrases (groupe nominal, groupe verbal, groupe adjectival, groupe prépositionnel et groupe adverbial) et aux fonctions grammaticales des syntagmes (sujet, objet, tête, etc.).

c)- *La représentation à base d'unités sémantiques (relative aux sens des phrases):* La description est représentée par des relations logico-sémantiques du type *prédicat-arguments*. Ce type de représentation permet d'indiquer le lien qui existe entres des phrases de même sens, mais syntaxiquement différentes (la forme active et la forme passive).

d)- *La représentation pragmatique:* elle s'intéresse à la signification globale du texte (le but du texte, les participants, les liens entre les phrases, etc.). Elle interprète la représentation sémantique des phrases au moyen de connaissances générales sur le monde et de la situation d'énonciation.

3.2 Techniques de sélection et de réduction du vocabulaire de représentation

Les documents textuels sont souvent représentés par des vecteurs lexicaux. En pratique, il existe deux façons d'indexer un document : soit considérer le texte tout entier, soit sélectionner uniquement certaines parties (le titre, le résumé, le sommaire, etc.), dans le cas de documents structurés par exemple. La sélection de descripteurs ou vocabulaire est une étape primordiale pour la représentation des textes. Elle constitue le noyau de base sur lequel repose toute méthode d'identification et de représentation des documents textuels. En effet, si la représentation des textes n'inclut pas certains descripteurs ou si les descripteurs retenus sont trop nombreux ou mal choisis, le système pour la tâche choisie aura des performances médiocres. Par conséquent, il est indispensable de ne pas se contenter de tous les mots d'un corpus, mais d'en trouver les termes caractéristiques permettant de représenter les documents sans perte d'information. Le modèle vectoriel est probablement l'approche la plus courante : on représente un texte par un vecteur numérique obtenu en comptant les éléments lexicaux les plus pertinents [SAL 75] [SAL 83] [MAN 99].
Avant toute indexation automatique d'un texte, un ensemble de traitements préliminaires (filtrage) sont nécessaires pour réduire la taille du texte : suppression de mots vides et appliquer des règles de lemmatisation pour réduire les variantes morphologiques. Même si l'on a montré (la loi de Zipf) que les mots les plus fréquents et les plus rares pouvaient être éliminés facilement, soit parce qu'ils n'étaient pas discriminants, soit parce qu'ils n'étaient pas

exploitables statistiquement, le nombre de candidats reste très élevé. Parmi l'ensemble des descripteurs restants, on peut penser que tous ne sont pas nécessairement discriminants pour un corpus ou thème donné, et que certains peuvent être très corrélés. On cherche donc à supprimer ces mots de la représentation des textes, tout en sachant que chaque suppression de mot entraîne une perte d'information ; il faut trouver le bon compromis entre, d'une part, la nécessité de réduire l'espace des descripteurs et, d'autre part, le besoin de garder suffisamment d'information.
De nombreuses techniques de sélection de descripteurs existent. Leur but est de choisir parmi un ensemble de descripteurs possibles, les "bons" descripteurs, c'est-à-dire ceux qui vont permettre d'obtenir de bonnes descriptions (représentation interne) caractérisant un document.
Il s'agit donc de réduire la très grande dimension des vecteurs lexicaux (très grand nombre de traits, vecteurs coûteux à stocker et à traiter, vecteurs creux (90%), redondance à cause de la forte corrélation des termes, etc.) et de représenter les textes d'une façon plus compacte, significative, efficace (économique) et sans perte d'information.
Le principe de ces approches repose sur le calcul de score pour chaque descripteur, indépendamment des autres, en s'appuyant sur les statistiques d'apparition et d'absence du descripteur en fonction de la classe à laquelle appartiennent les textes. Les descripteurs sont ensuite classés selon ce score, les descripteurs en tête de liste étant les plus discriminants.
Nous présentons les techniques principales de sélection et de réduction du vocabulaire, parmi les plus étudiées et utilisées [YAN 97]. Initialement, l'ensemble des descripteurs potentiels est constitué de l'ensemble des mots du corpus (thème ou domaine).

3.2.1 Les mots les plus fréquents

Le vocabulaire est construit de façon très classique, où il contient les mots les plus fréquents (en absolu) du corpus d'apprentissage. Cette méthode suppose que les termes rares, c'est-à-dire les moins fréquents, sont non significatifs et moins utile que les termes avec une fréquence élevée. Malheureusement, en pratique, certains termes moins fréquents sont très significatifs. De plus, cette approche produit des vecteurs de très grande dimension qui sont redondants (termes fortement corrélés entre eux), très creux (90% de valeurs nulles) et coûteux (à stocker et à traiter) [DEL 02].

3.2.2 Fréquence de documents (DF)

C'est une technique très simple pour réduire le vocabulaire d'indexation [APT 94] [YAN 97] [DAG 97] [JOA 97] [JOA 98] [SEB 99].
On ne prend pas en compte la fréquence des mots, mais le nombre de documents dans lesquels chaque mot est apparu. Initialement, la méthode consiste à prendre tous les mots du corpus comme des termes candidats à l'indexation. Tous les termes dont le nombre est inférieur à un certain seuil seront exclus.

$$DF(t,C) = \#(t,C) = p(t/C)$$

Le vocabulaire résultant sera composé des mots apparus dans le plus grand nombre de documents. DF représente le nombre de documents, dans un corpus d'apprentissage donné C, dont lequel un terme t apparaît.
Cette méthode suppose que les termes rares sont non significatifs et n'influencent pas dans la représentation. Elle constitue une méthode simple mais non réaliste. En effet, en pratique,

certains termes sont moins fréquents mais très significatifs et représentent bien un texte malgré que leur DF est très petit (inférieur au seuil).

3.2.3 Information Gain (IG)

IG, également appelé information mutuelle moyenne [MIT 96], permet, tout comme la mesure d'information mutuelle, de quantifier le lien existant entre un terme et une classe (par exemple un thème ou un domaine donné) mais ne prend pas seulement en compte l'influence qu'a l'apparition d'un terme sur une classe, il considère également sa non apparition.
La mesure IG se calcule de la façon suivante [YAN 97] [LAR 98] [YAN 99] [SEB 99]:

$$IG(t,C) = P(t,C)\log\frac{p(t,C)}{p(t)p(C)} + P(\bar{t},C)\log\frac{p(\bar{t},C)}{p(\bar{t})p(C)}$$

Avec *p(t)* est la probabilité à priori du terme t (le nombre de documents du corpus qui contiennent le terme *t*), *p(C)* la probabilité a priori de la classe C, p(*t,C*) est la probabilité conjointe d'apparition de *t* et *C (*le nombre de documents de classe *C* qui contiennent le terme *t), $p(\bar{t},C)$* est le nombre de documents de classe *C* qui ne contiennent pas le terme *t* et $p(\bar{t})$ est le nombre de documents du corpus qui ne contiennent pas le terme *t.*

3.2.4 Corrélation Coefficient CHI (χ2)

Le χ2 mesure la dépendance d'un terme *t* et d'une classe *C* (i.e si la présence d'un terme *t* dans un document est liée à l'appartenance de ce document à une classe C) [MIG 02] [YAN 97] [YAN 99] [SEB 99]. Le calcul nécessite de construire pour chaque terme *t* du corpus un tableau de contingences (table III.2). La mesure χ2 d'un terme *t* du corpus est définie donc par :

$$\chi^2(t,C) = \frac{N(pq'-p'q)^2}{(p+p')(q+q')(p+q)(p'+q')}$$

	C	***¬C***
t	*p*	*q*
¬t	*p'*	*q'*

Table III.2 : Table de contingences

C est la classe des documents (par exemple, la classe des documents pertinents).
t est un terme du corpus.
p est le nombre de documents de classe *C* qui contiennent le terme *t.*
q est le nombre de documents qui ne sont pas de classe *C* (¬*C*, la classe des documents non-pertinents) mais qui contiennent le terme *t.*
p' est le nombre de documents de classe *C* qui ne contiennent pas le terme *t.*
q' est le nombre de documents qui ne sont pas de classe *C* et qui ne contiennent pas le terme *t.*
N est le nombre total de documents du corpus.

La mesure du χ2 est basée sur la fréquence d'occurrence du terme dans les documents pertinents (la classe *C)* et non-pertinents (la classe ¬*C*). Elle est nulle si *t* et *C* sont

indépendants. C'est-à-dire, *t* apparaît avec la même fréquence dans le sous-ensemble des textes pertinents et dans le sous-ensemble des textes non pertinents, ce qui se traduit par (pq'=p'q). A l'inverse, si le terme *t* apparaît systématiquement dans l'ensemble des textes pertinents et jamais dans l'ensemble des textes non pertinents, on a p' = q = 0 et χ_2 (*t*,C) vaut N, ce qui est sa valeur maximale. Cette valeur est également atteinte si un descripteur apparaît systématiquement dans l'ensemble des textes non pertinents et jamais dans l'ensemble des textes pertinents. Entre ces deux valeurs extrêmes, plus la valeur de $\chi_2(t, C)$ est grande, plus *t* et *C* sont liés. Les termes du corpus sont donc classés par ordre décroissant de $\chi_2(t, T)$, les plus discriminants figurant en tête de liste. La mesure du $\chi 2$ n'est pas pertinente pour les termes d'occurrence faible.

3.2.5 Information Mutuelle (MI)

L'information mutuelle est employée pour mesurer la quantité d'information apportée par la présence ou l'absence d'un terme dans un document. Cette mesure a été fréquemment utilisée pour la catégorisation de textes pour effectuer la sélection de descripteurs [LEW 92b] [MAN 99] [DUM 98]. En effet, pour un terme donné, on calcule sa valeur d'information mutuelle avec chacune des classes. Ensuite, retenir par exemple, pour chaque terme, la valeur d'information mutuelle maximale parmi l'ensemble des classes.
La mesure d'information mutuelle évalue, plus précisément, l'influence qu'a, sur la classe (catégorie, thème, etc.) d'un texte, la présence d'un terme dans ce texte. Pour un terme *t* et une classe *C,* elle est définie par [MIG 02] [YAN 97]:

$$MI(t,C) = \log_2 \frac{p(t,C)}{p(t)p(C)}$$

Avec p(*t,C*) est la probabilité conjointe d'apparition de *t* et *C, p(t)* est la probabilité à priori du terme t et *p(C)* la probabilité a priori de la classe C.

Cette mesure numérique permet de représenter la dépendance entre *t* et *C*. Elle permet de déceler les mots qui « s'attirent », c'est-à-dire qui tendent à apparaître en même temps.

Une information mutuelle élevée entre un terme et une classe est le signe d'un lien fort entre ces deux éléments. Par exemple, le vocabulaire d'un domaine (une classe) serait composé des mots d'information mutuelle les plus élevées.

Si *t* et *C* sont indépendants alors MI(*t, C*)=0. En effet :

$$MI(t,C)=\log_2\frac{p(t,C)}{p(t)p(C)} = \log_2 \frac{p(t)p(C)}{p(t)p(C)} = \log_2 1 = 0$$

Elle est estimée comme suit :

$$MI(t,C)\approx\log_2\frac{N*p}{(p+p')(p+q)}$$

C est la classe des documents (par exemple, la classe des documents pertinents).
t est un terme du corpus.

p est le nombre de documents de classe *C* qui contiennent le terme *t.*
p' est le nombre de documents de classe *C* qui ne contient pas le terme *t*.
q est le nombre de documents qui ne sont pas de classe *C* (¬*C*, la classe des documents non-pertinents) mais qui contiennent le terme *t.*
N est le nombre total de documents du corpus.

3.2.6 Analyse en Composantes Principales (ACP)

La méthode d'analyse factorielle est très utilisée [DEL 02]. Elle consiste à calculer un nombre réduit de nouvelles dimensions (combinaisons linéaires des anciennes) non corrélées et exprimant un maximum de variance de données. Il s'agit de calculer préalablement la matrice carrée de covariance des données et les nouvelles dimensions sont les vecteurs propres, ordonnés d'une manière décroissante. L'ACP améliore l'efficacité informatique (réduction de la dimension) et donne des résultats intéressants d'un point de vue sémantique (interprétation des nouvelles dimensions qui sont très cohérentes). Elle est ainsi un remarquable outil d'analyse thématique d'un corpus de textes.
Plusieurs variantes de l'ACP existent. Nous citons, par exemple, l'algorithme d'apprentissage non supervisé de type Hebbien : Algorithme Hebbien Généralisé (GHA) qu'est une variante neuronale de l'ACP. Il ne nécessite que les données, sans autre information. Il est utilisé, par exemple, pour faire apprendre, à partir d'un corpus de textes, à un réseau de neurones les corrélations entre traits lexicaux d'un même vecteur et d'un vecteur à l'autre en cherchant à maximiser la sortie du réseau (extraction des composantes principales). Cet algorithme permet donc de réduire la dimension des vecteurs lexicaux de très grande dimension que l'ACP classique ne pourrait pas traiter [DEL 02].

3.2.7 Latent Semantic Analysis (LSA ou LSI)

Les nouvelles dimensions sont calculées directement en effectuant une décomposition en valeurs singulières de la matrice texte-terme [DEE 90] [SCH 95]. Elle ne nécessite pas le calcul de la matrice de covariance (grande taille, très coûteuse en temps de calcul). Elle est considérée comme une généralisation de l'extraction de vecteurs propres des matrices rectangulaires. Cette méthode semble donner de bons résultats en pratique mais contrairement à l'ACP, la signification des dimensions est loin d'être claire et peu intuitive. De plus, elle demeure coûteuse en temps de calcul (matrice de grande taille).

3.3 Les techniques de pondération ou codage

Une fois les composantes des vecteurs choisies pour représenter un texte, il faut décider comment coder chaque coordonnée du vecteur. Le codage des termes consiste à assigner un poids à chaque terme dans un document. De nombreuses techniques de pondération existent, les principales sont présentées dans la table 2. TFIDF (Term Frequency, Inverse Document Frequency) est la technique de pondération la plus utilisée [SAL 88] [DRU 99]. Elle calcule le poids du terme dans un document, en se basant sur sa fréquence d'apparition dans le document tout en prenant en considération sa fréquence d'apparition globale dans l'ensemble des documents. L'idée principale est que plus un terme se retrouve dans un grand nombre de documents, moins il risque de véhiculer une valeur informative (discriminante) élevée. C'est cette notion qu'encapsule le terme *idf* en pénalisant un terme se retrouvant à plusieurs endroits

à l'échelle de la base, ce qui constitue l'inverse conceptuel de ce qu'encapsule le terme *tf*, à savoir qu'un terme véhiculera un "niveau" d'information dans un document en fonction de sa fréquence dans le contexte de ce document. Cette technique *tfidf* suppose qu'un terme est important pour un document donné s'il apparaît souvent dans ce document (*tf*) et que peu de documents le contiennent (*idf*). Ainsi, si un terme apparaît dans tous les documents ($f_t=N$), son poids a une valeur nulle puisqu'il n'apporte aucune information, si il n'apparaît que dans ce document ($f_t=1$), son poids sera fort. De très nombreuses variantes du codage *tfidf* ont été proposées, elles ont fait l'objet d'un grand nombre de comparaisons expérimentales [BUC 92], [ROB 94] et [SIN 96a]. Par exemple, le codage *Lnu* tient compte de la taille des documents. En effet, selon Singhal [Singhal, 1998], il existe deux phénomènes à considérer dans les textes longs par rapport aux textes courts : les termes présents tendent à avoir des fréquences plus élevées, et les textes longs sont plus susceptibles de contenir des termes différents. Les compétitions TREC ont permis de caractériser le potentiel de ces différentes variantes de codage sur des grands corpus. Certaines employées dans des grands systèmes de recherche d'information sont devenues des références en la matière [TREC].

Méthode	Formule	Paramètres
Poids binaire	$W_{t,d}$ ={1 si t ∈d, 0 sinon}	$W_{t,d}$: poids du terme *t* dans le document *d*.
Fréquence	$W_{t,d}= f_{t,d}$	$f_{t,d}$: la fréquence du terme *t* dans le document *d*.
Term specificity	$W_{t,d}=logN - log(f_t)+1$	*N* : le nombre total de documents. f_t : le nombre de documents contenant le terme *t*.
Inverse document Frequency	$W_{t,d}= f_{t,d} / f_t$	
TFIDF	$w_{t,d} = f_{t,d} \cdot \log \frac{N}{f_t}$	
Lnu	Lnu = $L * u$ $L = (1 + log(f_{t,d})) / (1+log(f_t\, barre))$ $u = 1/ (0.8 + 0.2U_d/Ubarre)$	$f_t\, barre$: fréquence moyenne du terme t dans le document *d*. U_d : nombre de termes uniques dans le texte *d*. *Ubarre* : nombre moyen de termes sur l'ensemble des documents.
Autre	$W_{t,d} = f_{t,d} * Term\,\mathrm{Re}l_t = f_{t,d} * \frac{r_t / R - r_t}{i_t / I - i_t}$	*R*: le nombre total de documents pertinents ; r_t : le nombre de documents pertinents contenant le terme *t* ; *I* : le nombre total de documents non pertinents ; i_t : le nombre de documents pertinents contenant le terme *t* ;

Table III.3 : Techniques de pondération

3.4 Mesure de similarité entre textes

Le calcul de la similarité tente de quantifier "à quel point un document traite de la même chose qu'un autre", ce qui est une notion se situant au niveau sémantique, au-delà des aspects morphologique et syntaxique du texte [JAU 01]. La mise au point de bonnes mesures de similarité entre textes est un sujet d'intense recherche dans plusieurs domaines (recherche d'information, analyse de données textuelles, etc.).
La notion de similarité entre textes est évidemment fortement liée au choix de la méthode de représentation des textes. La représentation la plus utilisée est la représentation vectorielle [SAL 75] [SAL 83]. Un texte est représenté par un vecteur dans un espace vectoriel dont les dimensions sont associées à des unités linguistiques spécifiques (mot, lemme, etc.). Des améliorations peuvent également être apportées dans ce modèle de représentation par l'intégration de connaissances sémantiques supplémentaires, en particulier par l'utilisation de co-occurrences [BES 02]. Un texte est représenté comme la somme pondérée des vecteurs de co-occurrences des unités linguistiques qu'il contient. Chaque composante du vecteur de co-occurrences est la fréquence de co-occurrences de chaque unité linguistique considérée avec l'ensemble des autres unités linguistiques (deux à deux).
La similarité entre textes est évaluée par une mesure définie sur cet espace. Par exemple, en recherche d'information, les documents pertinents retournés par le moteur de recherche sont basés sur le calcul de la similarité entre *requête* et *documents*, de manière géométrique [SAL 88]. L'approche s'appuie sur des formules dépendant du nombre de mots communs présents dans les textes. La notion acquiert ainsi un sens mathématique très précis qu'il est facile de formaliser.

Plusieurs méthodes mathématiques ont été utilisées pour déterminer le degré de ressemblance entre deux textes représentés vectoriellement. Parmi ces méthodes, nous citons:

- Comparaison Simple (Simple Matching)

C'est la plus simple des méthodes de calcul de degré de similarité, elle se contente de comptabiliser les mots figurant à la fois dans le vecteur texte T1 et celui de T2, plus ce nombre est grand, plus la similarité est grande [LAR 98].

$$Sim(T1,T2) = \left|T1 \cap T2\right|$$

- Dice Coefficient

Cette fonction, très simple à calculer, est surtout utilisée quand les mots ont des poids binaires. Elle est calculée par la formule suivante [TAK 97]:

$$Sim(T1,T2) = 2\frac{\left|T1 \cap T2\right|}{\left|T1\right| + \left|T2\right|}$$

- Coefficient de Jaccard (Jaccard's Coefficient)

L'indice de Jaccard définit la similarité entre deux textes T1 et T2 par [LAR 98]:

$$Sim(T1,T2) = \frac{|T1 \cap T2|}{|T1 \cup T2|} = \frac{|T1 \cap T2|}{|T1| + |T2| - |T1 \cap T2|}$$

|T1 ∩ T2| représente le nombre de mots distincts présents dans T1 et T2.
| T1 ∪ T2| représente le nombre de mots distincts présents dans T1 ou T2.

Elle exprime la proportion du nombre de mots communs entre le texte T1 et le texte T2 par rapport au nombre de mots apparaissant dans l'un ou l'autre exclusivement (pas dans les deux en même temps).

- Coefficient de l'overlap (Overlap Coefficient)

Cette mesure met en évidence la proportion de mots communs entre un texte T1 et un autre texte T2 par rapport au nombre minimal de mots constituants les deux textes. Cela se traduit par la formule suivante [TAK 97]:

$$Sim(T1,T2) = \frac{|T1 \cap T2|}{\min(|T1|,|T2|)}$$

- Le produit scalaire (la mesure du cosinus)

C'est la méthode de calcul de similarité la plus utilisée, car elle est bien adaptée à la représentation vectorielle, elle se base sur l'idée que si deux vecteurs pointent dans la même direction alors le contenu de leurs documents respectifs est sûrement proche.
Par exemple, dans le modèle d'espace vectoriel EV (quatre textes A, B, C et D), le texte B est le texte le plus proche du texte D (figure III.3).

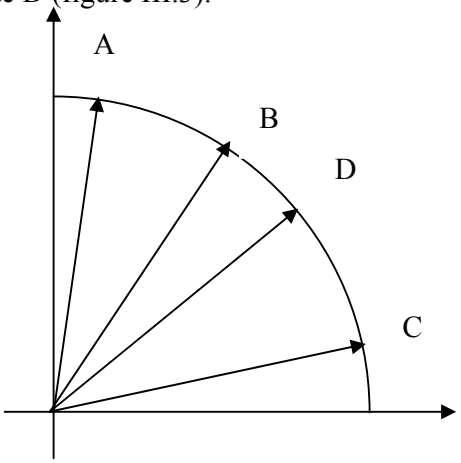


Figure III.3 : Représentation vectorielle

Nous constatons qu'au lieu de calculer l'angle entre deux vecteurs dans un espace à N dimensions il est plus simple de calculer son cosinus et cela en utilisant la formule suivante [LAR 98]:

$$Sim(X,Y) = \frac{\vec{x} * \vec{y}}{\left\|\vec{x}\right\| * \left\|\vec{y}\right\|} = \frac{\sum_i (x_i * y_i)}{\sqrt{\sum_i x_i^{\,2} * \sum_i y_i^{\,2}}}$$

Où *x* et *y* sont les vecteurs représentant deux textes, le dénominateur est un terme qui normalise le résultat par rapport à la longueur des vecteurs. On quantifie de cette manière la similitude par rapport aux directions des vecteurs. Pour des vecteurs orientés sensiblement dans la même direction, la valeur de similarité sera grande (cos (α=0) =1, si le produit est normalisé et les vecteurs parfaitement parallèles), si le cosinus de l'angle est petit alors le degré de ressemblance des documents est faible (cos (α=$\prod$/2) = 0) [SAL 88].

- La distance euclidienne

$$sim(d,q) = \sqrt{\sum_i (d_i - q_i)^2}$$

- La sous chaîne indexée [KIL 97]

$$d\left(V_q, V_d\right) = \frac{2\left|V_q \cap V_d\right|}{\left|V_q\right| + \left|V_d\right|}$$

Ces différentes mesures sont plus ou moins équivalentes. La valeur 1 est obtenue lorsque les vecteurs sont identiques et la valeur 0 est obtenue lorsqu'il n' y a aucune ressemblance entre leurs représentations.

4 Approches probabilistes

4.1 Approche Naïve Bayes

Supposons qu'un texte *X* soit représenté et décrit par un vecteur de *N* termes $\langle X_1, X_2, X_3, ..., X_N \rangle$ et considérons *K* classes de textes $\{c_1, c_2, .., c_K\}$. La classification d'un nouveau texte représenté par $x=\{x_1, x_2,.., x_N\}$ repose sur la résolution du problème probabiliste $P(c_i / x)$ qui est défini par la formule de *Bayes* [MCC 98]:

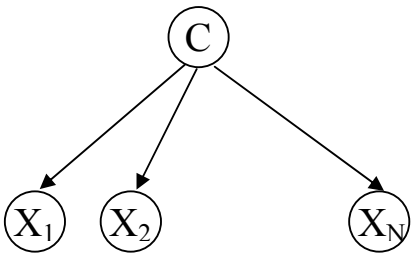
$$P(C = ci \mid X = x) = P(C = ci \mid X_1 = x_1 \,\&\, X_2 = x_2 \,\&\, ... \,\&\, X_N = x_N) = \frac{P(ci) * P(x / ci)}{P(x)}$$

Il s'agit donc d'estimer la probabilité à posteriori $P(C= ci \mid X=x)$ pour chaque classe c_i et déterminer la classe de x comme celle qui maximise $P(C \mid X=x)$.

La règle de classification de Bayes s'écrit:

$$C_{Bayes}(X = x) = ci \mid \text{argmax } P(ci) * P(x / ci)$$

$P(c_i)$ est la probabilité de la classe c_i, c-à-d la probabilité qu'un texte aléatoirement choisi appartienne à la catégorie c_i (nombre de textes de la classe c_i / nombre total de textes du corpus englobant toutes les classes). En général, les probabilités $P(x)$ et $P(x / c_i)$ ne sont pas connues et difficile à déterminer. Par définition, $P(A \& B) = P(A) \times P(B|A)$. Si A et B sont indépendants alors $P(B|A) = P(B)$ et donc $P(A \& B) = P(A) \times P(B)$.
Le modèle Bayésien naïf fait l'hypothèse qu'étant donné une classe, tous les termes (ou variables) sont indépendants les uns des autres.



Ce qui donne donc les estimations suivantes:

$$P(x) = P(X = x) = P(X_1 = x_1 \,\&\, X_2 = x_2 \,\&\, \ldots \,\&\, X_N = x_N) = \prod_i P(X_i = x_i) \ .$$

$P(x \mid ci) = P(X = x \mid C = ci) = \prod_j P(Xj = x_j \mid C = c_i)$ (pour tout j et toute classe c_i, $P(x_j/c_i)$ est estimée par la proportion d'éléments de classe c_i ayant la valeur x_j pour le jème élément du texte x).
Finalement, la règle de classification Naïve Bayes qui associe à tout texte x la classe c_i s'écrit comme suit :

$$C_{Bayes}(X = x) = ci \mid \text{argmax} \prod_j P(ci) * P(x_j / ci)$$

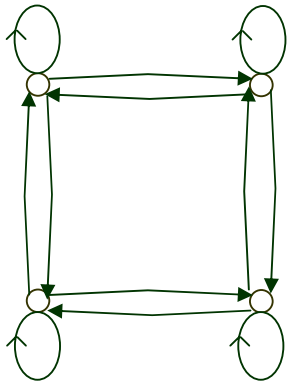
Bien que les hypothèses statistiques d'indépendance ne soient jamais vérifiées en pratique, l'utilisation de l'approche naïve Bayes est très efficace dans le domaine du texte (Dumais et al., 1998). De plus, Lewis montre en outre que ce type d'approche probabiliste simple est plus performant que les arbres de décision qui sont pourtant capables d'estimer les dépendances probabilistes entre les mots d'un texte (Lewis, 1992b).

4.2 Modèles de Markov Cachés (MMC)

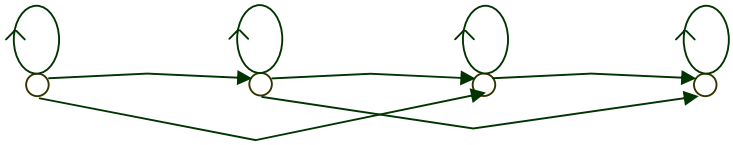
Les MMC sont des modèles stochastiques, développés vers les années soixante par Baum [RAB 89] [BAU 70]. Ils ont été étudiés pour la reconnaissance de la parole. Ce n'est que récemment qu'ils ont été employés dans d'autres domaines comme l'analyse de séquences textuelles [AMI 01]. Un MMC est défini à partir d'une chaîne de Markov et de probabilités. Une chaîne de Markov est un automate à états où chaque transition est dotée d'une probabilité. Si une probabilité d'une transition entre deux états est nulle, signifie que cette transition n'est pas autorisée.

Nous distinguons trois types de MMC les plus répandues :

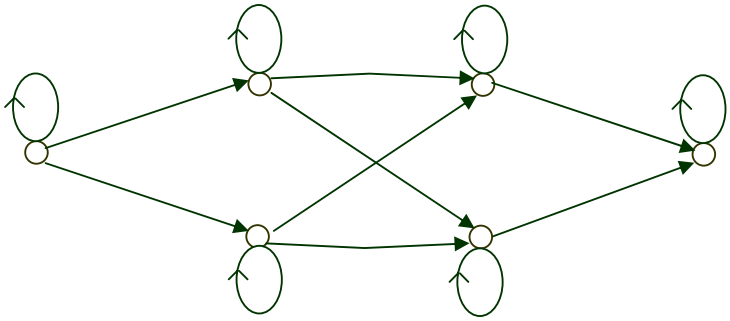
- Ergodique: Tous les états de l'automate sont reliés entre eux.



- Gauche-Droite : Aucune transition n'est autorisée sur les états inférieurs à l'état courant.



- Parallèle Gauche-Droite : Est une architecture qui combine et permet des connexions croisées entre deux MMC gauche-droite.



4.3 Machines à Vecteurs Supports (MVS)

Cette technique est initiée dès 1979 par Vapnik, mais qui connaît un essor depuis seulement quelques années [DRE 02] [JOA 98] [VAP 82] [VAP 95]. Elle consiste à trouver parmi toutes les surfaces (plans) possibles, le plan de décision ou "decision surface" qui sépare avec la plus grande marge possible entre deux classes différentes de données (par exemple, la séparation est faite entre les exemples positifs et les exemples négatifs de l'ensemble d'apprentissage pour chaque catégorie). Pour classer les nouveaux documents, on calcule dans quelle région de l'espace ils se situent et on leur attribue la classe correspondante. Intuitivement, cela

garantit un bon niveau de généralisation car de nouveaux exemples pourront ne pas être trop similaires à ceux utilisés pour trouver l'hyperplan mais être tout de même situés nettement d'un côté ou l'autre de la frontière. Un autre intérêt est la sélection de Vecteurs Supports qui représentent les vecteurs discriminant grâce auxquels est déterminé l'hyperplan : les exemples d'apprentissage (positifs et négatifs) les plus proches du plan de décision. Dans ce cas, seuls ces vecteurs supports sont utilisés pour classer un nouveau cas. Cela en fait une méthode très rapide.

L'apprentissage en *MVS* consiste à trouver deux paramètres w (vecteur) et b (constante) de l'équation du plan de décision (décrit par l'équation : $w \times x - b = 0$) qui satisfont les contraintes suivantes [DRU 99] :

$w \times x_j - b \geq 1$ *Si* $y_{ij} = 1$.

$w \times x_j - b \leq -1$ *Si* $y_{ij} = -1$.

$Min||w||^2$

Où :

x_j : est le vecteur des termes d'un texte dans l'ensemble d'apprentissage.

$y_{ij} \in \{-1, 1\}$ est la valeur de décision de son affectation à une catégorie donnée.

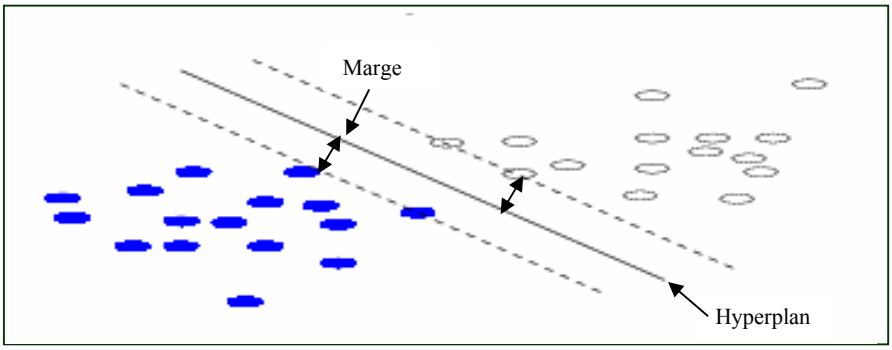


Figure III.4 : Un espace d'exemples séparés par un plan de décision (une droite)

Un nouveau texte x_j est classé dans une catégorie donnée, si $w \times x_j - b > 0$. L'efficacité des *MVS* (généralisation) est supérieure à celle de toutes les autres méthodes sur une grande variété de problèmes de la classification de textes, par exemple la reconnaissance des formes (ex : reconnaissance de caractères manuscrits, reconnaissance de visages), catégorisation des documents, etc (92% en terme d'exactitude) [DUM 98] [JOA 98].

5 Méthode des plus proches voisins

Plus connus en anglais sous le nom K-nearest neighbor (*K-NN*) [WEI 90], ou encore Memory Based Reasoning [STA 86]. Cette méthode est une technique de la reconnaissance des formes qui a prouvé son efficacité face au traitement de données textuelles [YAN 97]. Elle est didiée à la tâche de classification et qui peut être étendue à des tâches d'estimation. Elle diffère des traditionnelles méthodes d'apprentissage car aucun modèle n'est induit à partir des exemples (temps d'apprentissage inexistant). Les données restent telles quelles : elles sont simplement stockées en mémoire.

Le modèle est constitué de :

- un échantillon d'apprentissage,

- une fonction de distance (ou similarité),

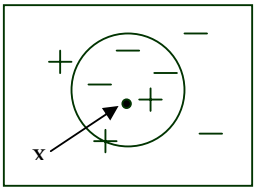
- une fonction de choix de la classe en fonction des classes des voisins les plus proches.

Pour prédire la classe d'un nouveau cas, l'algorithme cherche les *K* plus proches voisins de ce nouveau cas et prédit la réponse la plus fréquente de ces *K* plus proches voisins. La méthode utilise donc deux paramètres : le nombre *K* et la fonction de similarité pour comparer le nouveau cas aux cas déjà classés. Ces valeurs sont arbitraires mais importantes car des résultats très différents résultent de leurs choix. L'algorithme de cette méthode est très simple (algorithme III.1).

(1) Etant donné un document à classifier *d*.
(2) Trouver parmi l'ensemble des documents d'apprentissage les *K* les plus proches de *d*.
(3) Pour les catégories des *K* documents choisis, on construit une liste ordonnée décroissante en fonction des possibilités d'affectation de chaque catégorie à ce document *d,* où cette probabilité est calculée par sommation des degrés de similitude des documents qui lui appartiennent (parmi les *K* choisis) avec le document *d*.
(4) En utilisant un seuil prédéfini τ , le document est affecté aux catégories dont leurs taux d'affectation sont supérieurs à τ.

Algorithme III.1 : Algorithme K-nearest neighbor (*K-NN*)

Exemple : Etant donnés deux classes (+, -) et un nouveau document *X* à classer :



Si on choisit *K = 1*, X sera classé +. Si *K = 5*, le même *X* sera classé – (le choix de *K* est très important).

Le choix de la distance est primordial et dépendant des types de données (caractéristiques). Nous citons par exemple :

- Distance euclidienne :

$\mathrm{Sim}(x,d) = \sqrt{\sum (xi - di)^2}$ où *Sim(x,d)* est la similarité entre deux documents *x* et *d.*

Cette mesure favorise les voisins dans les caractéristiques sont assez proches.

- Distance par sommation :

$$\text{Sim(x,d)}=\sum dis\tan ce(xi,di) \text{ où } \begin{cases} numérique : dis\tan ce(xi,di) = |xi - di| \\ binaire : dis\tan ce(xi,di) = 0/1 \\ énumérative : dis\tan ce(xi,di) = valeur\_énumirée \end{cases}$$

Cette mesure tolère une distance importante sur l'une des caractéristiques.

D'autres mesures de similitude sont possibles, par exemple la mesure cosine (*Sim(x,d)=Cos(α)*, tel que *α* est l'angle séparant les deux documents).

La selection de la classe, du cas à résoudre, consiste à rechercher de cas similaires et utiliser les décisions des cas proches déjà résolus. La fonction de choix (calcul de décision ou sélection de la classe), par exemple, est calculée par la formule suivante :

$$y(\vec{x}, c_j) = \sum_{\vec{d}_i} Sim(\vec{x}, \vec{d}_i) \times y(\vec{d}_i, c_j) - b_j$$

où :

$y(d_i, c_j) \in \{0,1\}$ est la décision d'appartenance de d_i à c_j.

$Sim(x, d_i)$ est le taux de similitude entre les deux documents.

b_j est seuil prédéfini pour la catégorie c_j.

D'autres façons de décider de la classe à choisir sont possibles, par exemple le vote majoritaire (pondéré ou non), la moyenne pondérée, etc.

Les performances de la méthode *K-NN* dépendent du choix de la distance, du nombre de voisins et du mode de combinaison des réponses des voisins. Elle est très simple à mettre en œuvre. De plus, différentes expériences ont montré qu'elle est très efficace. Bien qu'elle ne produit pas de règles explicites, elle est claire et peut expliquer les résultats, en affichant les plus proches voisins qui ont amené au choix de la classe. Les expériences menées avec les *KNN* montrent qu'ils résistent bien aux données bruitées. Par contre, ils requièrent de nombreux exemples. Notons aussi que, si le temps d'apprentissage est inexistant puisque les données sont stockées telles quelles, la classification d'un nouveau cas est par contre coûteuse puisqu'il faut comparer ce cas à tous les exemples déjà classés. Pour remédier à cet inconvénient, plusieurs méthodes ont été conçues, par exemple celle appelée Category-Based Search [IWA 95] (ou encore simple nearest centroid). Elle consiste à représenter tous les documents rangés dans une catégorie par un cas unique (par exemple la moyenne des documents associés à une catégorie). Pour classer un nouveau document, on cherche le représentant le plus proche du document à classer. On gagne en rapidité puisqu'on ne compare plus tous les documents 2 à 2 avec le nouveau document à classer, mais uniquement le nouveau document avec le représentant de chaque catégorie.

6 Méthode de Rocchio

La méthode Rocchio [Roc 71] [JOA 97] [Schapire et al. 98] est une des plus vieilles méthodes de classification et l'une des plus simples. Elle consiste à générer, pour chaque classe *c* prédéfinie de documents, un vecteur principal (ou vecteur propre) *VP* en sommant les vecteurs de cette classe (exemples positifs) et en soustrayant les vecteurs des autres classes (exemples négatifs).

$$VP = \beta * \sum_{\{exemples>0\}} \frac{D}{|pos|} - \gamma * \sum_{\{exemples<0\}} \frac{D}{|nég|}$$

Avec :
D : Document.
Pos: sous-ensemble des documents d'apprentissage qui sont jugés exemples positifs pour la catégorie *c*.
Nég : sous-ensemble des documents d'apprentissage qui sont jugés exemples négatifs pour la catégorie *c*.
β, γ : paramètres de contrôle permettant d'affecter une importance relative aux exemples positifs (respectivement aux exemples négatifs).

Un document est classé dans une catégorie donnée s'il s'approche plus du centre des exemples positifs et s'éloigne plus du centre des exemples négatifs de cette catégorie (le centre est le vecteur des coordonnées moyennes des vecteurs représentant ces documents). Une autre façon de classer les nouveaux documents est de calculer le produit scalaire (mesure cosine) avec le vecteur *Vp* de chaque classe.
Plusieurs variantes ont été proposées dans le but d'améliorer cette technique, en terme d'exactitude et en temps d'apprentissage. Nous citons celle qui se base sur le choix des exemples négatifs. La méthode se base sur un apprentissage plus approfondi en considérant seulement les exemples qui sont difficiles à déceler par la méthode (appelé *near-positives*) [SCH 98].

7 Apprentissage symbolique

La procédure de classification produite peut être sous forme de règles. Parmi les méthodes symboliques les plus utilisées celles basées sur les arbres de décisions : CART (Classification And Regression Trees), ID3, MDL, etc.

7.1 Les arbres de décision

Les arbres de décision sont les plus populaires des méthodes d'apprentissage. Ils possèdent l'avantage d'être compréhensibles et aisément interprétables par l'utilisateur. En effet, ils génèrent des procédures de classification exprimables sous forme de règles. Plusieurs approches sont apparues, parmi les plus importantes on trouve : CART, ID3 [FUH 92], C4.5 (amélioration de ID3)[Coh 98a], [Coh 98b] [JOA 98] [LEW 94].
Comme toute méthode d'apprentissage supervisée, les arbres de décision utilisent un ensemble d'exemples. Un exemple est un ensemble d'attributs/valeurs (pour des documents, chaque attribut peut être un mot ou terme). Si l'on doit classer des documents dans des catégories, il faut construire un arbre de décision dont les nœuds représentent les termes et les arcs représentent les différentes valeurs de contribution d'un terme à la représentation du document testé (par exemple la fréquence ou la présence/absence). Concrètement, chaque nœud d'un arbre de décision contient un test (*IF...THEN*) et les feuilles représentent les différentes catégories (deux feuilles peuvent représenter la même catégorie). En général, chaque test examine la valeur d'un unique attribut de l'espace des descriptions. Les réponses possibles au test correspondent aux labels des arcs issus de ce noeud.

Le système affecte une catégorie à un document en testant la contribution réelle de chaque terme représenté par un nœud, en partant du sommet. Il la compare avec les différentes valeurs possibles des arcs partants de ce nœud, et prendre le chemin de la plus proche valeur, jusqu'à arriver à une feuille (la catégorie). La procédure de classification obtenue a une traduction immédiate en terme de règles de décision. Les systèmes de règles obtenus sont particuliers car l'ordre dans lequel on examine les attributs est fixé et les règles de décision sont mutuellement exclusives.
La construction d'un arbre de décision est un processus récursif. Il faut diviser récursivement et le plus efficacement possible les exemples de l'ensemble d'apprentissage par des tests définis à l'aide des attributs jusqu'à ce que l'on obtienne des sous-ensembles d'exemples ne contenant (presque) que des exemples appartenant tous à une même classe. Pour déterminer quel attribut tester à chaque noeud, on utilise un calcul statistique qui détermine dans quelle mesure cet attribut sépare bien les exemples. On crée alors un noeud contenant ce test, et on crée autant de descendants que de valeurs possibles pour ce test.
Par exemple, si on teste la présence d'un mot, les valeurs possibles sont *Présent/Absent*. A chaque fois, on aura donc deux descendants pour chaque nœud (arbre de décision binaire). On répète ce processus en associant à chaque descendant le reste des exemples qui satisfont le test du prédécesseur (figure III.5).

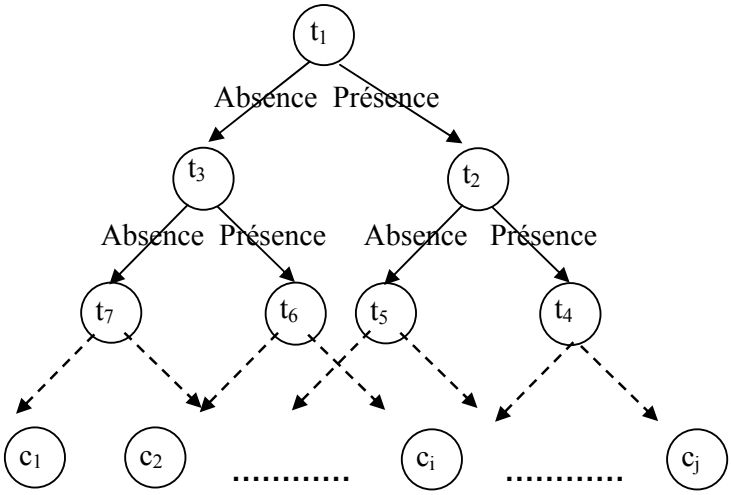


Figure III.5: Arbre de décision binaire

Il existe de nombreux algorithmes pour construire des arbres de décision. Les algorithmes construisent les arbres de façon descendante. Lorsqu'un test est choisi, on divise l'ensemble d'apprentissage pour chacune des branches et on réapplique récursivement l'algorithme. Une façon simple pour construire un arbre de décision est donnée par l'algorithme III.2.

Étant donnés un ensemble de documents *D* (ensemble d'apprentissage), un ensemble de termes *T* caractérisant ces documents, un ensemble de classes *{1,..,c}* et un arbre de décision *t*, à chaque position *p* de *t* correspond un sous-ensemble de l'échantillon qui est l'ensemble des exemples qui satisfont les tests de la racine jusqu'à cette position.

- ***Initialiser avec l'arbre vide*** (la racine de l'arbre n'est étiquetée par aucun test)

- ***Décider si un noeud est terminal***, c'est-à-dire décider si un noeud doit être étiqueté comme une feuille (par exemple, tous les exemples sont dans la même classe), c'est-à-dire affecter une classe (affecter la catégorie prévue pour ce sous-ensemble de documents).

- ***Sélectionner (sinon) un test à associer à un nœud*** (choisir un terme) qui discrimine de façon intéressante l'échantillon (par exemple, utiliser des critères statistiques qui permettent de comparer les différents choix possibles).

- ***Diviser l'ensemble des documents en sous-ensembles*** selon les valeurs possibles pour ce test, et créer autant d'arcs que de ces valeurs (créer le sous arbre).

- Répéter ces étapes récursivement pour chaque terme (passer au nœud suivant) jusqu'à ce qu'il ne reste aucun terme.

Algorithme III.2 : Algorithme de construction d'arbres de décision

Il existe plusieurs fonctions statistiques qui permettent de choisir et d'associer un test à un nœud, nous en citons deux de base: la fonction de *Gini* et la fonction *entropie* [QUI 89].

$$Entropie\ (p) = -\Sigma_{i=1}\ P\ (c_i/p) \times log\ (P\ (c_i/p))$$

$$Gini\ (p) = 1 - \Sigma_{i=1}\ P\ (c_i/p)^2$$

Avec c_i désigne la classe ou la catégorie *i*, *p* est la position dans l'arbre, à chaque position *p* correspond un sous-ensemble de l'échantillon qui est l'ensemble des exemples qui satisfont les tests de la racine jusqu'à cette position. $P(c_i/p)$ désigne la proportion d'éléments de classe c_i en position *p.*

Ces fonctions permettent de mesurer l'homogénéité des exemples pour toute position de l'arbre en construction (mesure le degré de mélange des exemples entre les différentes classes). De telles fonctions vérifient la propriété suivante : elles prennent la valeur 0 (minimum) lorsque tous les exemples sont dans une même classe (le noeud est pur) et la valeur 1 (maximum) lorsque les exemples sont équirépartis. La mesure *Gini* est utilisée par la méthode CART [BRE 84] et l'*entropie* par la méthode C4.5 [QUI 93]. Une autre mesure statistique appelée Information Gain est utilisée. Elle est définit comme suit :

$$Gain(p,t) = \mathrm{E}(p) - \sum_{j=1}^{n} P_j \times \mathrm{E}(pj)$$

Où *p* désigne une position ou le noeud, *t* un test d'arité *n* ou l'attribut, P_j est la proportion d'éléments à la position *p* qui vont en position *pj* (qui satisfont la *j*ème branche du test *t*) et *E* est la fonction entropie (ou utiliser la fonction Gini).
Le principe consiste donc à calculer cette valeur pour chaque attribut et choisir 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. En effet, d'après les propriétés de la fonction entropie, si l'entropie est faible, la plupart des éléments se trouvent dans une même classe. On cherche donc à obtenir le gain maximum.
Un arbre de décision est facile à interpréter et est la représentation graphique d'un ensemble de règles. Mais, sa taille peut être importante, ce qui rend difficile d'appréhender l'arbre dans sa globalité. En effet, l'algorithme peut donner un arbre très volumineux, à cause de la génération des branches très spécifiques (branches spécifiques à un seul document, par exemple) : problème d'apprentissage par cœur (overfitting). L'arbre de décision classe trop bien les exemples, mais est mauvais pour généraliser, c'est-à-dire qu'il prédit mal la classification de nouvelles instances (documents). L'apprentissage par cœur peut survenir lorsque les exemples sont bruités ou qu'il y a peu d'exemples. De façon générale, plus l'arbre de décision est profond, et plus le risque d'apprentissage par cœur augmente. Le problème est donc de trouver la profondeur idéale. Une solution pour ce problème est l'utilisation de techniques qui permettent de ″tailler ou élaguer″ l'arbre en éliminant ce type de branches (sous arbres). Par exemple, on divise l'ensemble des exemples en 2/3 pour construire l'arbre de décision, et 1/3 pour valider l'arbre. La procédure de validation est la suivante : après avoir construit un arbre de décision complet, qui risque donc l'overfitting, on enlève successivement des nœuds de cet arbre. A chaque fois, on valide alors le nouvel arbre amputé d'un nœud (pruned tree) grâce aux exemples de validation (1/3) : si le nouvel arbre ne classe pas plus mal les exemples de validation que l'arbre précédent l'amputation, alors on le garde. On continue à amputer des nœuds de l'arbre de décision tant que le nouvel arbre continue à classer de mieux en mieux les exemples de validation.
Dans l'algorithme CART, l'élagage, consiste à effectuer un parcours ascendant de l'arbre construit. Pour décider si un sous arbre peut être élagué, on compare l'erreur réelle estimée de l'arbre courant avec l'arbre élagué. L'estimation de l'erreur réelle est mesurée sur un ensemble test ou par validation croisée.
Dans l'algorithme C4.5, l'élagage est effectué avec l'ensemble d'apprentissage par une évaluation de l'erreur. Il propose également de générer un système de règles à partir de l'arbre de décision. Le système obtenu n'est pas une simple réécriture de l'arbre car des transformations et simplifications sont effectuées.
La caractéristique la plus importante, est certainement de pouvoir expliquer comment est classé un exemple par l'arbre, ce qui peut être fait en montrant le chemin de la racine à la feuille pour l'exemple courant. De plus, l'arbre contient les attributs utiles pour la classification. L'algorithme peut donc être utilisé comme pré-traitement qui permet de sélectionner des attributs pertinents. Cependant, les performances tendent à se dégrader lorsque le nombre de classes (feuilles) est trop important. L'algorithme n'est pas incrémental, il nécessite aussi de relancer l'apprentissage à cause de l'évolution des données dans le temps.

7.2 Les règles de décision

Le principe consiste à construire un ensemble de règles logiques pour chaque catégorie en analysant l'ensemble des documents d'apprentissage. L'apprentissage (création de nouvelles règles) s'arrête lorsque tous les exemples positifs de cette catégorie sont satisfaits, et aucun exemple négatif n'est accepté. Ensuite une passe de ″révision″ est lancée sur l'ensemble des

règles construites pour éliminer les clauses répétitives afin de réduire la taille (plus compact), diminuer le temps de test et sans diminuer l'efficacité.
Il existe plusieurs systèmes à règles de décision, qui diffèrent en terme de méthodes utilisées, heuristiques et critères de simplification. Nous citons par exemple un extrait de règles pour la catégorie *Ireland* du système *Ripper* de *W. Cohen* [COH 96b]:

Ireland ⟵ ireland ∈ document .
Ireland ⟵ ira ∈ document, killed ∈ document .
Ireland ⟵ ira ∈ document, kills ∈ document .
Ireland ⟵ ira ∈ document, belfast∈ document.
Ireland ⟵ ira ∈ document, to ∈ document, calls ∈ document.
Ireland ⟵ irish ∈ document, abortion ∈ document.
Ireland ⟵ ira ∈ document, shot ∈ document .
Ireland ⟵ ira ∈ document, out ∈ document .
Sinon le document n'appartient pas à la catégorie *Ireland*.

*D*ans cet exemple, nous décidons d'affecter à un document *d* la catégorie *Ireland* si : (ireland ∈ document) *ou* (ira ∈ document *et* killed ∈ document) *ou* ... etc.
Les règles sont faciles à comprendre et à modifier pour un utilisateur simple. Mais l'aspect sémantique n'est pas considéré ainsi que la fréquence des termes (l'apparition d'un terme une seule fois dans un document peut entraîner une mauvaise classification de ce document).

8 Apprentissage adaptatif

La procédure de classification produite est de type « boite noire ». On distingue deux grandes classes : Réseaux de neurones et algorithmes génétiques.

8.1 Réseaux de neurones

Parmi les principales applications des réseaux de neurones nous trouvons l'apprentissage et l'optimisation. L'apprentissage à l'aide des réseaux de neurones est tolérant au bruit et aux erreurs (algorithmes robustes). Le temps d'apprentissage peut être long, par contre, après apprentissage le calcul des sorties à partir d'un vecteur d'entrée est rapide.
L'apprentissage est une phase de développement d'un réseau de neurones durant laquelle le comportement d'un réseau est modifié jusqu'à l'obtention du comportement désiré. Il permet au réseau de tenir compte des nouvelles contraintes ou des nouvelles données du monde extérieur. Il permet d'assurer la stabilité du réseau en tant que système dynamique.
Le résultat de l'apprentissage est un réseau constitué de cellules organisées selon une architecture, définies par une fonction d'activation et un certain nombre de poids à valeurs réelles.
La critique principale, est que les réseaux de neurones obtenus après apprentissage se comportent comme une boite noire, non interprétable par l'utilisateur. On ne peut pas donner d'explication au calcul d'une sortie sur un vecteur d'entrée (contrairement aux arbres de décision).
L'échantillon nécessaire à l'apprentissage doit être suffisamment grand et représentatif des sorties attendues. Il faut passer un grand nombre de fois tous les exemples de l'échantillon d'apprentissage avant de converger et donc le temps d'apprentissage peut être long. De plus,

comme pour les arbres de décision, l'apprentissage n'est pas incrémental et, par conséquent, si les données évoluent avec le temps, il est nécessaire de relancer une phase d'apprentissage pour s'adapter à cette évolution.
Dans un système à base de règles, l'apprentissage consiste à ajouter de nouvelles règles dans une base de connaissances, par contre, dans un réseau de neurones, apprendre signifie modifier les poids synaptiques. En effet, ces poids synaptiques sont ajustés dans le but d'améliorer la réponse du réseau et de s'adapter au mieux aux besoins de l'application. Il est souvent impossible de décider à priori des valeurs et des poids des connexions d'un réseau pour une application donnée. A l'issu de la phase d'apprentissage les poids sont fixés. Il s'agit alors de la phase d'utilisation.

Nous distinguons trois modes d'apprentissage: mode *supervisé, semi supervisé* et *non supervisé*. Dans le mode *supervisé*, les exemples présentés au réseau sont résolus (la sortie est connue). Dans le mode *non supervisé, c*ontrairement au mode *supervisé,* il n'y a aucune information sur les données présentées au réseau. L'apprentissage consiste à classer les exemples dans des classes d'équivalence en opérant par mesure de ressemblance (ex : l'Algorithme Hebbien Généralisé, GHA) [DEL 02]. Dans le mode *semi supervisé,* la valeur exacte de la sortie n'est pas connue, mais la seule information disponible est un signal d'échec ou de succès.

Nous distinguons différents types de règles d'apprentissage : règle de Hebb [HEB 49], la correction d'erreurs, règle de *Widrow-Hoff* (*règle delta* ou *règle Adaline*) [WID 60], apprentissage compétitif [HAY 94], etc. Nous présentons, dans ce qui suit, les algorithmes d'apprentissage par correction d'erreurs les plus utilisés. Dans le modèle par correction d'erreurs, l'apprentissage tient compte de l'erreur observée en sortie : en effet, pendant le processus d'apprentissage, la sortie calculée *o* peut être différente de la sortie désirée *d*. Le principe de base consiste donc à comparer le résultat obtenu au résultat désiré (*d* - *o*), puis ajuster les poids de connexions et diminuer, petit à petit, l'erreur globale du système.

- L'algorithme d'apprentissage Perceptron

L'objectif de l'apprentissage est d'inférer à partir d'un échantillon d'apprentissage (un ensemble d'exemples) un perceptron qui classifie correctement ou au mieux les exemples (classification binaire).
Un perceptron consiste en un unique neurone, défini par des coefficients synaptiques ajustables $w_1,...,w_n$ et d'une valeur de seuil u. Il prend en entrée un vecteur x de n valeurs $x_1,..., x_n$ et calcule une sortie o, définie par :

$$o = \begin{cases} 1 & \text{si } e - u > 0 \\ 0 & \text{sinon} \end{cases} \qquad \text{et} \qquad e = \vec{x} \cdot \vec{w} = \sum_{i=1}^{n} w_i x_i$$

Lors de la phase d'apprentissage, tous les exemples sont présentés jusqu'à la convergence, c'est-à-dire jusqu'à ce qu'une présentation complète des exemples n'entraîne aucune modification de l'hypothèse en cours. Initialement, les poids du perceptron ont des valeurs quelconques. A chaque fois que l'on présente un nouvel exemple, on ajuste les poids selon que le perceptron l'a correctement classé ou non. L'algorithme s'arrête lorsque tous les exemples ont été présentés

sans modification d'aucun poids. L'algorithme d'apprentissage par correction d'erreur du perceptron est décrit comme suit (algorithme III.3) :

Entrée : un échantillon S de $R^n \times \{0,1\}$ ou $\{0,1\}^n \times \{0,1\}$
Initialisation aléatoire des poids w_i à des petites valeurs pour i entre 0 et n
Répéter
Prendre un exemple (x,d) dans S *(d : sortie désirée de l'entrée x)*
Calculer la sortie o du perceptron pour l'entrée x
- - Mise à jour des poids - -
Pour i de 0 à n
$w_i\ (t+1) \leftarrow w_i\ (t) + \eta\ (d-o)x_i$ *(η : pas d'apprentissage)*
Finpour
finRépéter
Sortie : Un perceptron P défini par $(w_0, w_1, ..., w_n)$

Algorithme III.3 : Algorithme Perceptron

Lorsque la sortie calculée o diffère de la sortie attendue d, la procédure d'apprentissage du perceptron modifie les poids comme suit ($\eta=1$):
Si o=0 et d=1, cela signifie que le perceptron n'a pas assez pris en compte les neurones actifs de l'entrée (c'est-à-dire les neurones ayant une entrée à 1) ; dans ce cas, $w_i(t+1) \leftarrow w_i\ (t) + x_i$; l'algorithme ajoute la valeur de la rétine aux poids synaptiques (*renforcement*).
Si o=1 et d=0, alors $w_i(t+1) \leftarrow w_i(t) - x_i$; l'algorithme retranche la valeur de la rétine aux poids synaptiques (*inhibition*).
A noter que le perceptron n'apprend, au sens de ``modifie ses poids'', que si $d-o \neq 0$. L'avantage de la méthode par correction d'erreur est que, si l'échantillon d'apprentissage est linéairement séparable, si tous les exemples sont présentés équitablement (c'est-à-dire que la procédure de choix des exemples n'en exclut aucun) et que le critère d'arrêt est la stabilité de l'hypothèse après une présentation complète de l'échantillon alors l'algorithme s'arrête avec un perceptron qui classifie correctement l'échantillon d'apprentissage. Cependant en pratique on ne sait pas si le problème considéré est linéairement séparable. L'inconvénient majeur donc de cet algorithme est que si l'échantillon présenté n'est pas linéairement séparable, l'algorithme ne convergera pas et l'on n'aura aucun moyen de le savoir. Même lorsque l'algorithme d'apprentissage du perceptron converge, rien ne garantit que la solution sera *robuste*, c'est-à-dire qu'elle ne sera pas remise en cause par la présentation d'un seul nouvel exemple. Cet algorithme n'a aucune tolérance au bruit : si une information mal classée, vient perturber les données d'entrée, le perceptron ne convergera jamais, c'est-à-dire des données linéairement séparables peuvent ne plus l'être à cause du bruit. De plus, les problèmes *non-déterministes*, c'est-à-dire pour lesquels une même description peut représenter des éléments de classes différentes ne peuvent pas être traités à l'aide d'un perceptron.

- Algorithme d'apprentissage par descente du gradient

Plutôt que d'obtenir un perceptron qui classifie correctement tous les exemples, il s'agira maintenant de calculer une erreur et d'essayer de minimiser cette erreur. Pour introduire cette notion d'erreur, on utilise des poids réels et on élimine la notion de seuil, ce qui signifie que la sortie sera donc réelle.
L'erreur d'un perceptron P défini par $\vec{w}=(w_1,...,w_n)$ sur un échantillon d'apprentissage S d'exemples $(\vec{x}^s, d^s)$ est définie en utilisant la fonction erreur quadratique par :

$$E(\vec{w}) = 1/2 \sum_{(\vec{x}^s, d^s) \in S} (d^s - o^s)^2$$

où o^s est la sortie calculée par P sur l'entrée $x^{\rightarrow s}$. L'erreur mesure donc l'écart entre les sorties attendues et calculées sur l'échantillon complet. On remarque que $E(w^{\rightarrow}) = 0$ si et seulement si le perceptron classifie correctement l'échantillon complet.
On suppose S fixé, le problème est donc de déterminer un vecteur $w^{\rightarrow}$ qui minimise $E(w^{\rightarrow})$. Une méthode qui permet de rechercher le minimum d'une fonction est d'utiliser la méthode du gradient. L'algorithme d'apprentissage par descente de gradient du perceptron est décrit comme suit (algorithme III.4) :

Entrée : un échantillon S de $R^n \times \{0,1\}$;
Initialisation aléatoire des poids w_i pour i entre 1 et n
Répéter
  Pour tout i $\Delta w_i \leftarrow 0$ **finPour**
  Pour tout exemple (x^s, d^s) de S
    calculer la sortie o^s
    Pour tout i $\Delta w_i \leftarrow \Delta w_i + \square (d^s - o^s) x_i^s$ **finPour**
  finPour
  Pour tout i $w_i (t+1) \leftarrow w_i (t) + \Delta w_i$ **finPour**
 finRépéter
Sortie : Un perceptron P défini par $(w_1, ..., w_n)$

Algorithme III.4 : Algorithme par descente de gradient

L'algorithme est assuré de converger, même si l'échantillon d'entrée n'est pas linéairement séparable, vers un minimum de la fonction erreur pour un ε bien choisi suffisamment petit. Si ε est trop grand, on risque d'osciller autour du minimum. Pour cette raison, une modification classique est de diminuer graduellement la valeur de ε en fonction du nombre d'itérations. Le principal défaut est que la convergence peut être très lente et que chaque étape nécessite le calcul sur tout l'ensemble d'apprentissage.

- Algorithme de Widrow-Hoff

Cet algorithme est une variante très utilisée de l'algorithme par descente de gradient. Au lieu de calculer les variations des poids en sommant sur tous les exemples de S, l'idée est de modifier les poids à chaque présentation d'exemple.
Un neurone ne modifie (augmente/diminue) l'intensité de ses synapses (n'apprend) que lorsqu'il se trompe. L'échantillon d'apprentissage est présenté au modèle dans un ordre aléatoire. La procédure est itérée jusqu'à ce que le modèle ne commet plus d'erreurs. La règle de modification des poids appelée *règle delta*, ou *règle Adaline*, ou encore *règle de Widrow-Hoff*, est :

$$\Delta w_i = \varepsilon (d^s - o^s) x_i^s$$

L'algorithme *de Widrow-Hoff* est décrit comme suit (algorithme III.5) :

Entrée : un échantillon S de $R^n \times \{0,1\}$;
Initialisation aléatoire des poids w_i pour i entre 1 et n
Répéter
Prendre un exemple (x,d) dans S
Calculer la sortie o du perceptron pour l'entrée x
- - Mise à jour des poids - -
Pour i de 1 à n
w_i (t+1)← w_i (t)+ ε (d-o)x_i
Finpour
finRépéter
Sortie : Un perceptron P défini par (w_0, w_1, ..., w_n)

Algorithme III.5 : Algorithme de Widrow-Hoff

Le critère d'arrêt généralement choisi est : pour un passage complet de l'échantillon, toutes les modifications de poids sont en dessous d'un seuil prédéfini. Il y a correction chaque fois que la sortie totale (qui est un réel) est différente de la valeur attendue (égale à 0 ou 1).
L'avantage de cet algorithme par rapport à l'algorithme par correction d'erreur est que, même si l'échantillon d'entrée n'est pas linéairement séparable, l'algorithme va converger vers une solution optimale (sous réserve du bon choix du paramètre ε). L'algorithme est, par conséquent, plus robuste au bruit. L'algorithme est très souvent utilisé en pratique et donne de bons résultats. La convergence est, en général, plus rapide que par la méthode du gradient. Il est fréquent pour cet algorithme de faire diminuer la valeur de ε en fonction du nombre d'itérations comme pour l'algorithme du gradient.

- Algorithme de rétropropagation du gradient

L'algorithme de rétropropagation du gradient est une adaptation (ou extension) de l'algorithme de Widrow-Hoff aux réseaux à couches. Il est découvert au début des années 80 par Rumelhart, McClelland, Parker, Hinton et Le Cun. Il permet à des réseaux de neurones d'apprendre des fonctions que le perceptron n'est capable d'apprendre (fonctions non linéairement séparables).
Le principe de l'algorithme est de minimiser une fonction d'erreur. Cette erreur est l'erreur quadratique mesurée sur l'ensemble des exemples d'apprentissage et peut s'écrire par la formule suivante :

$\text{Erreur} = \frac{1}{2} \sum_{Exemples} (\text{d}(x) - \text{o}(x))^2$ où x est un exemple, d est la sortie désirée et o la sortie calculée.

Il s'agit ensuite de calculer la contribution à cette erreur de chacun des poids synaptiques : chacun des poids influe sur le neurone correspondant, mais, la modification pour ce neurone va influer sur tous les neurones des couches suivantes. L'algorithme de rétroprogation du gradient consiste à mettre à jour les poids à chaque présentation d'exemple, c'est-à-dire on tend à minimiser l'erreur calculée pour chaque exemple et pas l'erreur globale. L'algorithme de rétropropagation du gradient est décrit comme suit (algorithme III.6) :

Entrée : un échantillon S de $R^n \times R^p$;
un réseau avec une couche d'entrée C_0, q-1 couches cachées C_1, ..., C_{q-1}, une couche de sortie C_q, n cellules.
Choisir une fonction d'activation (ex : sigmoïde)
Initialisation aléatoire des poids w_i pour i entre 1 et n
Répéter
  Prendre un exemple (x,d) de S et calculer o
  - - calcul des erreurs δ_i par rétropropagation
  Pour toute cellule de sortie i $\delta_i \leftarrow o_i(1-o_i)(d_i-o_i)$ **finPour**
  Pour chaque couche de q-1 à 1
    Pour chaque cellule i de la couche courante
      $\delta_i = o_i(1-o_i) \sum_{k \in Succ(i)} \delta_k w_{ki}$
    finPour
  finPour
  - - mise à jour des poids
  Pour tout poids $w_{ij} \leftarrow w_{ij} + \varepsilon \delta_i x_{ij}$ **finPour**
finRépéter
Sortie : Un réseau défini par la structure initiale choisie et les w_{ij}

Algorithme III.6 : Algorithme de rétropropagation du gradient

Après présentation de chaque entrée x et calcul de la sortie o, le calcul des erreurs δ_i sera effectué de la couche de sortie vers la couche d'entrée : pour une cellule i de sortie, la quantité δ_i correspond à l'erreur usuelle d_i-o_i. Pour une cellule i interne, le calcul de δ_i dépend de la somme pondérée des erreurs des cellules de la couche suivante. Le critère d'arrêt du processus d'apprentissage peut être, par exemple : arrêter dès que l'erreur estimée passe sous un seuil prédéfini.
Un problème est de choisir les bonnes valeurs pour les paramètres tel que ε. Pour cela, on découpe l'ensemble d'apprentissage en un ensemble d'apprentissage, un ensemble de validation et un ensemble test. Lors de la phase d'apprentissage, on arrête périodiquement l'apprentissage, on estime l'erreur réelle sur l'ensemble de validation, on met à jour les paramètres en fonction de la variation de cette erreur estimée. L'ensemble test sert à estimer l'erreur réelle à la fin de l'apprentissage.

8.2 Algorithmes génétiques

Les algorithmes génétiques (AGs) sont des algorithmes permettant d'explorer un espace de solutions en vue d'amélioration et d'optimisation. Ils s'ajoutent à la panoplie d'algorithmes d'exploration traditionnels, reposant sur les fondements de l'intelligence artificielle, palliant ainsi les limites d'explosions combinatoires. Ils tirent leur puissance de leur structure inspirée des fondements biologiques de la génétique. En effet, ils sont inspirés des mécanismes de la sélection naturelle (imitant les systèmes naturels de l'évolution des espèces) et de la génétique (schématiquement, ils copient de façon extrêmement simplifiée certains comportements des populations naturelles). Ils utilisent à la fois les principes de la survie des individus les mieux adaptés et ceux de la propagation du patrimoine génétique. Ainsi, les AGs se basent sur une population d'individus qui vont évoluer de génération en génération pour obtenir un résultat se

rapprochant de la solution optimale. Les meilleurs individus d'une génération vont créer une nouvelle génération plus adaptée au problème. De façon très intuitive, le problème est identifié à un environnement donné et les solutions à des individus évoluant dans cet environnement. A chaque génération, ne sont retenus que les individus les mieux adaptés à cet environnement. Au bout d'un certain nombre de générations, les individus restants sont particulièrement adaptés à l'environnement donné. On obtient donc des solutions très proches de la solution idéale du problème.
Les algorithmes génétiques ont été développés par John Holland, à l'université du Michigan dans les années 70. Il avait deux buts principaux : mettre en évidence et expliquer rigoureusement les processus d'adaptation des systèmes naturels, et concevoir des systèmes artificiels qui possèdent les propriétés des systèmes naturels.

8.2.1 Concepts de base

La modélisation d'un algorithme génétique repose sur les concepts de base suivants:

a)- Représentation chromosomique des solutions du problème

Un chromosome est une entité (individu) qui représente un élément de l'espace des solutions. L'ensemble d'individus (solutions) forme une population où chacun est caractérisé par un ensemble d'éléments appelés *gènes*.

b)- Fonction d'évaluation ou d'adaptation

Elle s'occupe du classement des solutions afin de choisir les meilleures[1]. Elle permet d'évaluer la probabilité qu'un individu survie dans la prochaine génération. Cette fonction d'adaptation est une fonction propre à chaque problème permettant de calculer le niveau d'adaptation d'un individu par rapport au reste de la population (une fonction sélective permettant une bonne discrimination entre les chromosomes).

c)- Les opérations génétiques (sélection ou reproduction, croisement et mutation)

Elles interviennent dans le processus de formation de la prochaine génération de solutions. C'est un processus qui simule l'hérédité de la progéniture à partir de la matière génétique des parents (notion de couplage). La création d'une nouvelle population à partir de la précédente se fait par application des opérateurs génétiques que sont :

- Reproduction (sélection)

Par analogie à la génétique en biologie, cette opération est une application d'une loi fort connue : celle du plus fort, c'est à dire que seuls les individus les plus forts sont susceptibles de donner une bonne descendance et de là, seuls les plus forts subsistent. En effet, les individus les plus forts subsistent et les autres subissent des changements pour les rendre plus

[1] Génétiquement parlant, c'est le processus de sélection naturelle.

forts ou disparaissent de la population de solutions (mauvaises solutions). La reproduction est donc une opération qui permet de choisir parmi une population les individus qui vont survivre à l'évolution ou mourir. Au cours de cette opération l'algorithme sélectionne les meilleurs chromosomes ou éléments pertinents (les plus adaptés, c'est-à-dire qui optimisent mieux la fonction d'adaptation) pour être reproduit et les moins adaptés vont mourir.

- Croisement (Crossover)

C'est l'opération d'héritage partiel des gênes du parent, elle intervient pour renforcer les individus qui ne sont pas suffisamment forts afin qu'ils puissent subsister et donner une nouvelle génération d'individus forts. Le croisement permet donc de générer deux chromosomes nouveaux "*enfants*" à partir de deux chromosomes reproduits ou sélectionnés "*parents*" (figure III.6), qui ont des chances d'être plus adaptés que leurs parents. Ce croisement s'effectue de la façon suivante :

(i)- Déterminer aléatoirement une frontière de *coupe* (point de croisement) des deux individus choisis.

(ii)- Prendre la partie avant la frontière du premier individu et combiner avec la deuxième partie du deuxième individu, afin de créer un nouvel individu contenant une part d'information de ses deux parents.

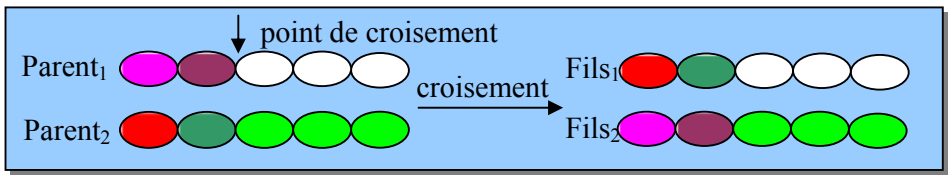


Figure III.6 : L'opérateur de croisement

- Mutation

La mutation réalise l'inversion d'un ou plusieurs gènes d'un chromosome (figure III.7). Cette mutation s'effectue de la façon suivante :

(i)- Sélectionner aléatoirement un gène à muter.

(ii)- Sélectionner aléatoirement un gène de remplacement.

(iii)- Produire un individu fils par le remplacement du gène à muter par le gène de remplacement.

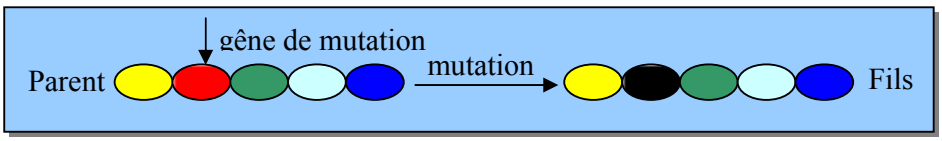


Figure III.7 : L'opérateur de mutation

Cet opérateur permet de n'oublier aucune solution possible au problème (solution oubliée par les opérateurs de *Reproduction* et de *Croisement)*.

d)- Paramètres de contrôle

Un ensemble de paramètres utiles pour le contrôle et le bon fonctionnement des algorithmes génétiques. Parmi lesquels nous citons :

- La taille de la population

Choisir un nombre initial de solutions adéquat pour converger rapidement et sûrement vers une adaptation fidèle aux exigences du problème. En effet, une population initiale trop faible entraînerait un manque de support (solutions) pour arriver à une bonne adaptation au problème, par contre une surpopulation risque d'entraîner une redondance dans la prise en charge des composants de la population initiale.

- La probabilité de mutation ou de Crossover

C'est la probabilité d'application d'une opération de mutation (respectivement de crossover) lors du processus de reproduction.

Ces deux facteurs sont très corrélés, du fait que la différence entre eux est une différence de convention (le fait que le gêne concerné par l'opération soit connu ou pas importe peu).
Cependant, tandis que le processus de mutation entraîne l'apparition de nouveaux individus par modification de gênes, le crossover effectue une diffusion des gênes existant lors du processus de reproduction : dés lors un taux de mutation trop important entraînerai la destruction de gênes sans que ces derniers n'aient la chance d'être assemblés par crossover.

8.2.2 Fonctionnement général d'un algorithme génétique

Un algorithme génétique est un algorithme itératif de recherche d'optimum, il manipule une *population* de taille constante. Son objectif consiste à générer une nouvelle population héritant les meilleures caractéristiques d'une population initiale. A chaque itération, appelée *génération*, est créée une nouvelle population avec le même nombre de chromosomes. Cette génération consiste en des chromosomes mieux "*adaptés*" à leur environnement tel qu'il est représenté par la fonction sélective. Au fur et à mesure des générations, les chromosomes vont tendre vers l'optimum de la fonction sélective (algorithme III.7).

(1) : ***Génération de la population initiale*** : génération d'un ensemble d'individus *I* constituants une population initiale G_0[1].

(2) : ***Calcul de la fonction sélective*** : Evaluation de l'adaptation *a(I)* de chaque individu *I* dans la population G_0.

(3) : **Répéter**
Sélection des individus à maintenir et ceux à éliminer en fonction de leurs résultats.
Suppression des individus jugés faibles.
Croisement des individus de la nouvelle génération.
Mutation (en respectant le taux fixé).
Calcul de la fonction sélective.
Jusqu'à satisfaction d'un critère d'arrêt

---
1 : Il est à noter que dans un cadre plus général, c'est un ensemble pouvant être généré aléatoirement.

Algorithme III.7 : les différentes étapes d'un algorithme génétique

8.2.3 Algorithmes génétiques et filtrage d'information

Les algorithmes génétiques constituent une approche de l'intelligence artificielle fondée sur l'exploration à la fois aléatoire (considération de tout l'espace de solutions), dirigée (la nouvelle génération est systématiquement meilleure que sa précédente) et automatique (les opérateurs génétiques augmentent automatiquement l'adaptation de la population). Ils constituent une solution adéquate aux problèmes d'optimisation, d'adaptation et d'apprentissage. Ces techniques se chargent de faire évoluer les solutions trouvées jusqu'à les adapter au problème avec toute la vigueur et l'efficacité souhaitée, tout cela afin de procurer des solutions précises et parfaitement adaptées. Ils sont intéressants lorsque :

- L'exploration traditionnelle de l'espace de solution risque d'engendrer une explosion combinatoire.

- Le problème présente un ensemble de caractéristiques finis.

- Possibilité d'interprétation et d'évaluation d'un ensemble de caractéristiques.

Une utilisation pratique des algorithmes génétiques dans le domaine de filtrage d'information, est l'apprentissage automatique pour améliorer le modèle utilisateur. Il s'agit d'explorer l'ensemble des résultats de filtrage puis adapter et améliorer la représentation des profils selon les structures ayant présenté la meilleure évaluation. L'exploration par les algorithmes génétiques constitue dores et déjà un moyen irréfutable.

9 Classification par recherche directe

Les groupements se font par recherche directe d'une partition, en affectant les éléments à des centres provisoires de classes, puis en recentrant ces classes, et en affectant de façon itérative ces éléments. Nous distinguons trois catégories de méthodes : la méthode des centres

mobiles (les centres sont des points de l'espace; ils sont calculés après chaque itération), la méthode *K-moyennes* (les centres sont des points ; algorithme III.8) et la méthode des nuées dynamiques ou de « ré-allocation » (les centres ne sont pas des points ; algorithme III.9) [BEL 01]. Généralement, la partition obtenue finalement, par ce type de méthodes, dépend du choix initial des centres.
Un problème est le choix initial du nombre q de classes. Ce nombre est choisi à priori et fourni à l'algorithme. Cet algorithme possède de nombreuses variantes selon la méthode utilisée pour choisir les q premiers centres, la mesure de similarité choisie, le choix de la valeur de q et le calcul des distances entre classes.

Soit un ensemble I de n éléments à partitionner, caractérisés par p variables. On suppose que l'espace $\mathbb{R}^p$ supportant les n points-éléments est muni d'une distance appropriée notée d (souvent distance euclidienne usuelle [1] ou distance du χ^2). On désire constituer au maximum q classes. Les étapes de l'algorithme sont illustrées comme suit :

Étape *0* : On détermine q centres provisoires de classes (par exemple, par tirage pseudo-aléatoire). Les q centres $\{C_1^0,\ldots,C_k^0,\ldots,C_q^0\}$ induisent une première partition P^0 de l'ensemble des éléments I en q classes $\{I_1^0,\ldots,I_k^0,\ldots,I_q^0\}$. Ainsi l'élément i appartient à la classe I_k^0 s'il est plus proche de C_k^0 que de tous les autres centres.

Étape *1*: On détermine q nouveaux centres de classes $\{C_1^1,\ldots,C_k^1,\ldots,C_q^1\}$ en prenant les centres de gravité des classes qui viennent d'être obtenues. C'est-à-dire, pour chaque classe I^0 ainsi constituée, son nouveau centre est calculée en effectuant la moyenne des éléments de la classe. Ces nouveaux centres induisent une nouvelle partition P^1 de I construite selon la même règle que pour P^0.
La partition P^1 est formée des classes notées $\{I_1^1,\ldots,I_k^1,\ldots,I_q^1\}$.

Étape *m*: On détermine q nouveaux centres de classes $\{C_1^m,\ldots,C_k^m,\ldots,C_q^m\}$ en prenant les centres de gravité des classes qui ont été obtenues lors de l'étape précédente $\{I_1^{m-1},\ldots,I_k^{m-1},\ldots,I_q^{m-1}\}$. Ces nouveaux centres induisent une nouvelle partition P^m de l'ensemble I formée des classes $\{I_1^m,\ldots,I_k^m,\ldots,I_q^m\}$.

Arrêt : l'algorithme s'arrête soit lorsque deux itérations successives conduisent à la même partition, soit lorsqu'un critère convenablement choisi (par exemple, la mesure de la variance intra-classes) cesse de décroître de façon sensible, soit encore parce qu'un nombre maximal d'itérations a été fixé *a priori.*

---

[1] La distance euclidienne est définie par : $d(X, Y) = \sqrt{\sum (Xi - Yi)^2}$

Algorithme III.8 : Algorithme k-moyennes

Etape 1 : Déterminer une partition initiale.

Etape 2 : Calculer les représentants de chacune des classes.

Etape 3 : Affecter chaque document à la classe qui lui est la plus proche.

Arrêt : L'algorithme s'arrête lorsque les documents ne migrent pas d'une classe à une autre ou que le nombre d'itérations, fixé à priori, est atteint.

Algorithme III.9 : Etapes de l'algorithme des nuées dynamiques

10 Classification ascendante

La classification ascendante procède à la construction des classes par agglomérations successives des objets deux à deux. Elle fournit une *hiérarchie de partitions* de moins en moins fines, se présentant sous la forme *d'arbres* appelés également *dendrogrammes* (figure III.8).

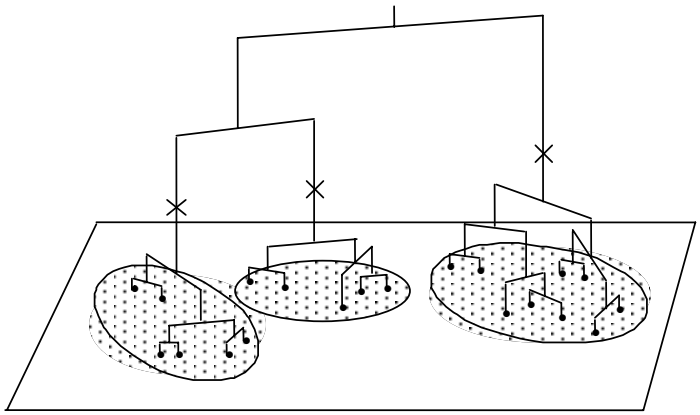


Figure III.8 : Dendrogramme ou arbre hiérarchique

L'intérêt de ces arbres est qu'ils peuvent donner une idée du nombre de classes existant effectivement dans la population. Chaque coupure d'un arbre fournit une partition, ayant d'autant moins de classes et des classes d'autant moins homogènes que l'on coupe plus haut. Le principe de la méthode est simple : à partir d'un échantillon de *m* éléments, nous commençons avec *m* groupes d'un élément chacun. Nous calculons la matrice des distances deux à deux entre éléments, appelée matrice des similarités. Nous sélectionnons la paire de groupes les plus similaires (la plus petite valeur dans la matrice) et nous les regroupons en un seul groupe. Nous réitérons le procédé jusqu'à obtenir un groupe constitué de tous les éléments (algorithme III.10).

Étape *1* : il y a *n* éléments ou objets à classer;

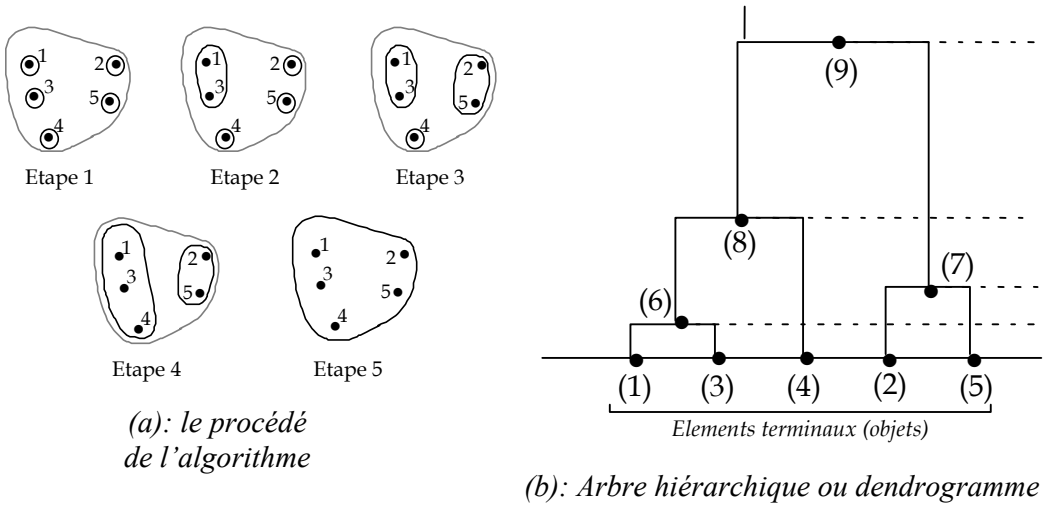
Étape *2* : on construit la matrice de distances entre les *n* éléments et l'on cherche les deux plus proches, que l'on agrège en un nouvel élément. On obtient une première partition à *n-1* classes;

Étape *3* : on construit une nouvelle matrice des distances qui résultent de l'agrégation, en calculant les distances entre le nouvel élément et les éléments restants (les autres distances sont inchangées). On se trouve dans les mêmes conditions qu'à l'étape *1*, mais avec seulement (*n-1*) éléments à classer et en ayant choisi un critère d'agrégation. On cherche de nouveau les deux éléments les plus proches, que l'on agrège. On obtient une deuxième partition avec *n-2* classes et qui englobe la première;

Étape *m* : on calcule les nouvelles distances, et l'on réitère le processus jusqu'à n'avoir plus qu'un seul élément regroupant tous les objets et qui constitue la dernière partition.

Algorithme III.10 : Classification ascendante Hiérarchique

Nous obtenons ainsi une suite de partitions de l'échantillon d'entrée ou le nombre *k* de groupes varié de *m* à *1*. A la fin, nous choisissons la valeur de *k*. La figure III.9 illustre un exemple du procédé de ce type de classification : nous considérons cinq points comme objet à classer.



(a): le procédé de l'algorithme

(b): Arbre hiérarchique ou dendrogramme

Figure III.9 : Exemple de classification ascendante

La méthode est basée sur le calcul de distances (entre éléments, entre éléments et groupes, entre groupes). Parmi lesquelles, nous citons : le *saut minimal (single link), le diamètre (complete link), la moyenne (group average), vecteurs moyens (centroids), etc* [BEL 00]. La distance la plus utilisée est le *saut minimal*. Par exemple, si *x, y, z* sont trois objets, et si les objets *x* et *y* sont regroupés en un seul élément noté *h*, on peut définir la distance de ce groupement à *z* par la plus petite distance des divers éléments de *h* à *z* :

$$d(h,z) = Min \{d(x,z), d(y,z)\}$$

11 Classification descendante

La classification descendante est dite non hiérarchique qui produit une partition des objets en un nombre fixé de classes. Elle procède par dichotomies successives (divisions) de l'ensemble des objets. En effet, la méthode démarre avec un groupe constitué de l'ensemble des objets et divise récursivement les groupes en utilisant une fonction de diversité qui minimise la distance dans un groupe (la distance moyenne) et maximise la distance entre groupes (plus petite distance entre les éléments les plus proches, les plus éloignés ou entre les centres).

12 Conclusion

L'apprentissage automatique a porté sur beaucoup de tâches et concerne des domaines très variés (recherche d'information, extraction d'information, etc.). De nombreux travaux ont porté sur la classification automatique de textes [BRU 02], d'autres sur la classification de termes (elle est connue pour servir la reformulation ou l'expansion de requêtes et pour la classification de documents en recherche d'information) [TUR 00]. Les méthodes utilisent l'information contenue dans le texte (mots). L'approche a été critiquée pour le manque de précision des résultats et par la lenteur des processus (inexploitable en temps réel).
Les derniers travaux se sont focalisés sur la sémantique. Nous citons, par exemple, le système AutoSlog développé par Riloff et Jones [RIL 99]. Le système est capable, à partir d'un corpus et d'une liste d'amorces, de construire simultanément un lexique sémantique et un dictionnaire de patrons d'extraction pour une catégorie sémantique donnée. Un autre système LIRA (Learning Information Retrieval Agents), qui aide l'internaute à trouver des pages web intéressantes, en fonction de ses intérêts. Il lui présente une sélection de documents, et après évaluation, ajuste ses paramètres en vue d'améliorer ses performances. Après 24 jours d'apprentissage, les résultats semblent beaucoup meilleurs que ceux produits par l'expert humain [BAL 95].
Les méthodes utilisées par les systèmes d'apprentissage sont très nombreuses et sont issues de domaines scientifiques variés. Nous pouvons les classer en deux grandes classes : les méthodes paramétriques et les méthodes non paramétriques. Les méthodes paramétriques se basent sur des hypothèses à priori sur les distributions des descriptions des classes et les procédures de classification sont construites à l'aide d'hypothèses probabilistes. La variété de méthodes statistiques viendra de la diversité des hypothèses possibles. Les méthodes non paramétriques, en général, sont des méthodes issues de l'intelligence artificielle qui ne se basent pas sur des hypothèses à priori (les méthodes symboliques et les méthodes adaptatives).
En tout état de cause, un fait important communément admis est : Il n'existe pas de méthode meilleure que toutes les autres. Par conséquent, à tout ensemble de données et tout problème correspond une ou plusieurs méthodes. Le choix se fera en fonction de la tâche à résoudre, de la nature et de la disponibilité des données, des connaissances et des compétences disponibles, de la finalité du modèle construit. Par conséquent, un certain ensemble de critères sont nécessaires pour le choix de la méthode. Par exemple, la complexité de la construction du modèle, la complexité de son utilisation, ses performances, etc.
Il existe une multitude d'algorithmes de classification. Le choix d'un algorithme dépend de plusieurs paramètres : le domaine d'application, la taille de l'échantillon, nature des données, etc.
Les techniques de classification ascendante et descendante présentent des avantages différents et peuvent être utilisées conjointement : les techniques mixtes. D'autres méthodes existent telles que les réseaux de neurones auto-organisés : les cartes de Kohonen.

Les techniques de classification non supervisée sont sensibles au choix de bons paramètres (en particulier, le choix du nombre q de classes à constituer). Ce nombre est généralement défini empiriquement. Ces techniques sont basées uniquement sur des calculs de similarité avec un seuil fixé arbitrairement. De plus, il est difficile d'évaluer et d'interpréter les résultats produits, c'est-à-dire d'attribuer une signification aux classes constituées. En effet, ce type de classification nécessite d'autres techniques ou une expertise pour donner une signification aux classes construites après apprentissage.

Le domaine de l'apprentissage automatique est un passage obligé dans la conception d'un système de filtrage automatique d'information. En effet, la quantité grandissante d'informations électroniques permet de constituer des échantillons de données variés et significatifs. Un système qui permet d'apprendre automatiquement des profils d'utilisateurs et d'exploiter ces connaissances pour filtrer l'information semble incontournable.

Chapitre IV

Architecture de filtrage et générateur d'interfaces

Dans cette partie, nous présentons l'architecture générale de notre système de filtrage : nous décrivons les principaux modules du système ainsi que les différentes connaissances et outils utilisés. Ensuite, nous présentons notre outil d'aide GIFI, un assistant à la génération d'interfaces de filtrage. Nous insistons sur les deux améliorations proposées des systèmes de filtrage existants. D'une part, il s'agit d'élargir l'éventail des propriétés qui serviront à améliorer la représentation classique de textes. En effet, nous proposons, en plus des caractéristiques lexicales, un ensemble de critères automatisables, susceptibles d'influer sur le processus de filtrage, et permettant ainsi de situer un texte par rapport aux autres. Ces propriétés sont basées sur des modèles linguistiques réduits. Ces critères sont des indices qui portent généralement sur la structure et le contenu des textes. Ces connaissances sont indépendantes du domaine d'application. Nous les avons classées en plusieurs niveaux linguistiques: matériel, énonciatif, structurel et syntaxique. D'autre part, nous décrivons notre proposition pour la prise en compte de l'aspect sémantique dans le processus de filtrage. Nous exposons tout d'abord les besoins que vise à satisfaire cet aspect sémantique dans le processus de filtrage, puis nous décrivons les connaissances et le traitement sémantique.

1 Architecture de base du système de filtrage

Le système est composé des principaux modules suivants :

- Un module d'identification de la langue qui détermine la langue de chaque texte et le prépare aux différentes étapes ultérieures de l'analyse en sélectionnant les connaissances nécessaires.

- Un analyseur linguistique qui analyse les textes et délivre en sortie une représentation conceptuelle associée. Il utilise un ensemble de connaissances linguistiques de base sous forme de modèles réduits.

- Un module pseudo-sémantique permettant d'améliorer la représentation du texte. Il utilise deux connaissances: un réseau lexical permettant de remplacer les termes du texte par des termes sémantiquement proches et la cooccurrence des critères de filtrage pour inférer et

ajouter, à la représentation, d'autres termes qui n'existent pas dans le texte à filtrer (connaissances implicites), mais qui corrèlent avec certains termes du profil.

- Un module de classification et de filtrage qui permet de comparer un nouveau texte avec les différents profils de l'utilisateur.

- Un module d'apprentissage qui permet d'améliorer l'efficacité et les performances du système.

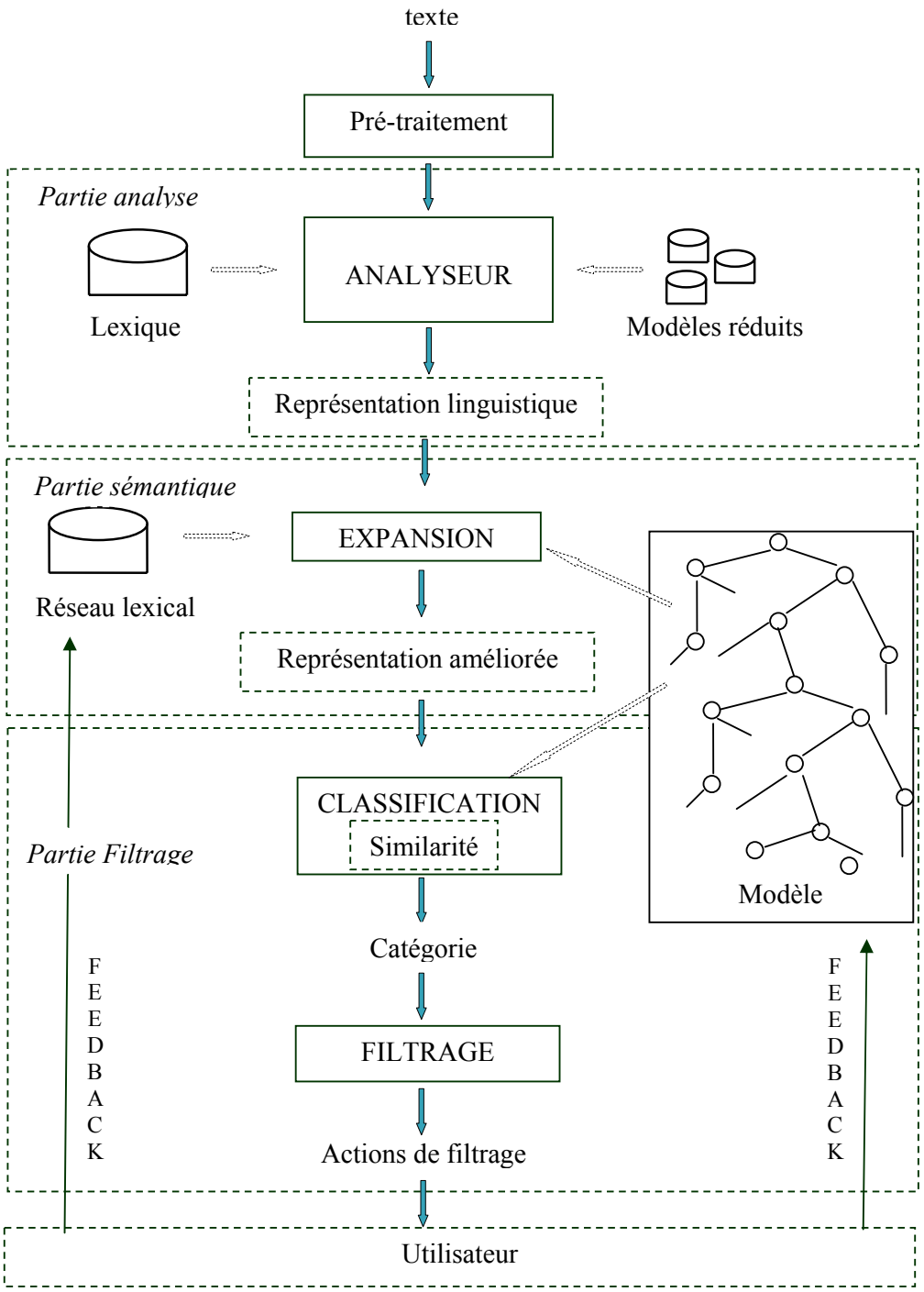


Figure IV.1 : Architecture globale du système de filtrage

2 Identification de la langue

Ce module d'*identification de la langue* consiste à identifier la langue de chaque texte (figure IV.2) et à permettre de sélectionner, selon la langue, les connaissances nécessaires au bon déroulement des différentes étapes ultérieures de l'analyse.
L'étape d'identification de la langue est une étape nécessaire. Elle permet de caractériser la langue du texte parmi deux actuellement modélisées (Français, Anglais). Elle permet aussi de signaler les textes bilingues et ceux qui ne sont pas dans l'une de ces langues. La méthode d'identification de la langue est simple : elle utilise des anti-dictionnaires (ou *stoplist*) propres à chaque langue. Il s'agit de compter, pour chaque texte, les mots communs de la *stoplist* en mettant à jour des compteurs sur la langue. Nous avons pu recenser plus de 300 termes vides ou mots communs en langue française et plus de 500 en langue anglaise que nous avons regroupés dans des *stoplist* (voir annexe B). Ces listes triées et indexées sont enregistrées dans une base de données qui sera chargée en mémoire lors du démarrage du logiciel sous forme de deux vecteurs triés pour optimiser le temps de recherche. Par ailleurs, le système est incrémental et permet facilement la prise en compte de nouvelles langues (ajouter un anti-dictionnaire propre à chaque nouvelle langue).
Cette méthode s'est révélée performante. Nous avons défini une valeur par défaut pour les textes de langue différente que les deux prises en compte par le système (*langue inconnue*).

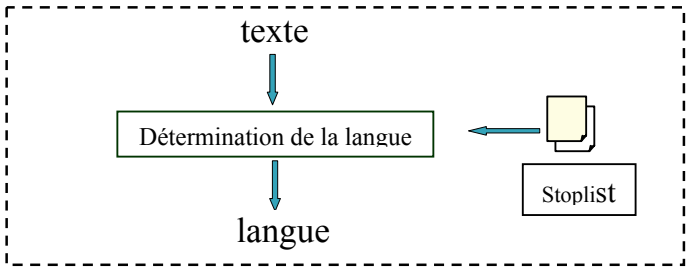


Figure IV.2: Identification de la langue

3 Etiquetage

L'extraction de certaines propriétés (d'ordre syntaxique) nécessite une phase d'étiquetage préalable. En effet, nous avons utilisé un étiqueteur morpho-syntaxique, analyseur de Brill [BRI 92a]. La Figure IV.3 présente un exemple de résultats de l'étiqueteur.

Message : *Bonjour Omar. Tout s'est très bien passé. C'est dommage que tu n'aies pas pu venir, car c'est le moment d'apprendre beaucoup de choses. C'est le moment aussi pour prendre des contacts. Bon courage. Alain.*

Message étiqueté : *Bonjour/INJ Omar/SBP:sg ./. Tout/DTN:sg s'/PRV:sg est/ECJ:sg très/ADV bien/ADV passé/ADJ2PAR:sg ./. C'/PRV:sg est/ECJ:sg dommage/SBC:sg que/SUB$ tu/PRV:sg n'/ADV aies/ACJ:sg pas/ADV pu/VPAR:sg venir/VNCFF ,/, car/COO c'/PRV:sg est/ECJ:sg le/DTN:sg moment/SBC:sg d'/PREP apprendre/VNCFF beaucoup/ADV de/PREP choses/SBC:pl ./. C'/PRV:sg est/ECJ:sg le/DTN:sg moment/SBC:sg aussi/ADV pour/PREP prendre/VNCFF des/DTC:pl contacts/SBC:pl ./. Bon/ADJ:sg courage/SBC:sg ./. Alain/SBP:sg ./.*

Jeu d'étiquettes : INJ : *Interjection.* ***SBP :*** *Substantif, nom propre ou à majuscule.* ***DTN :*** *Déterminant.* ***PRV :*** *Pronom « supporté » par le verbe (conjoint, clitique).* ***ECJ :*** *Verbe « être », conjugué.* ***ADV :*** *Adverbe.* ***ADJ2PAR :*** *Participe passé adjectival (non après auxiliaire).* ***SBC :*** *Substantif, nom commun.* ***SUB$:*** *Subordonnant possible. = Code par défaut de « que ».* ***ACJ :*** *Verbe « avoir », conjugué.* ***VPAR :*** *autre Verbe, non conjugué, participe passé après « avoir ».* ***VNCFF :*** *autre Verbe, non conjugué, infinitif.* ***COO:*** *Coordination.* ***PREP :*** *Préposition.* ***DTC :*** *Déterminant de groupe nominal, contracté.* ***Sg:*** *Singulier, etc.*

Figure IV.3 : Résultat de l'étiqueteur Brill

Nous présentons, pour le développement de notre outil de filtrage, le catégoriseur d'Eric Brill, Université de Pennsylvanie, ainsi que les connaissances générées pour le Français à l'INaLF[1] [LEC 98].

3.1 Catégoriseur Brill

Le catégoriseur de Brill est fondé sur les travaux des structuralistes américains (Bloomfiels, 1933; [HAR 51] [CHO 57]. L'idée structuraliste est la suivante : ne pas s'intéresser à une description universelle des langues, ne pas chercher à caractériser complètement une langue, mais tirer profit des régularités/variations dans l'usage pour mettre en œuvre des traitements efficaces. Il s'agit donc d'approcher la description d'une langue sans idée préconçue, et se fonder sur l'observation des faits linguistiques rencontrés.
Le catégoriseur de Brill est un outil d'étiquetage automatique de textes. Il permet d'assigner à chaque segment de textes (mot) une étiquette représentative de sa catégorie grammaticale (la plus probable). C'est un annotateur, qui ne traite que les mots, contrairement à un parseur qui traite des constituants plus larges au niveau de la phrase (syntagmes et propositions). Il est conçu initialement pour l'anglais, facilement adaptable au français et à d'autres langues. La quantité d'information à lui fournir est réduite, puisqu'il apprend lui-même et automatiquement ce qu'il estime nécessaire à la catégorisation la plus probable. Il est gratuit, existe pour des environnements Unix et Windows.
Du point de vue informatique, ce catégoriseur est un outil dont l'intérêt essentiel réside dans un « auto-apprentissage » d'une sorte de Base de Connaissances à partir de n'importe quel type de Corpus, avec n'importe quel type d'étiquettes (syntaxiques, sémantiques ou phonologiques, etc.). La connaissance apprise par le système sur le petit corpus-échantillon est projetée comme « probable », par le même système, sur de plus grand corpus.

3.1.1 Apprentissage

Le système Brill est un système probabiliste. Il utilise une base de connaissances créée par apprentissage de corpus échantillon manuellement étiqueté (codage du texte avec un ensemble d'étiquettes de « Parties du Discours ») et sélectionné au hasard dans un grand corpus à étiqueter.

[1] Institut National de la Langue Française, http://www.atilf.fr/

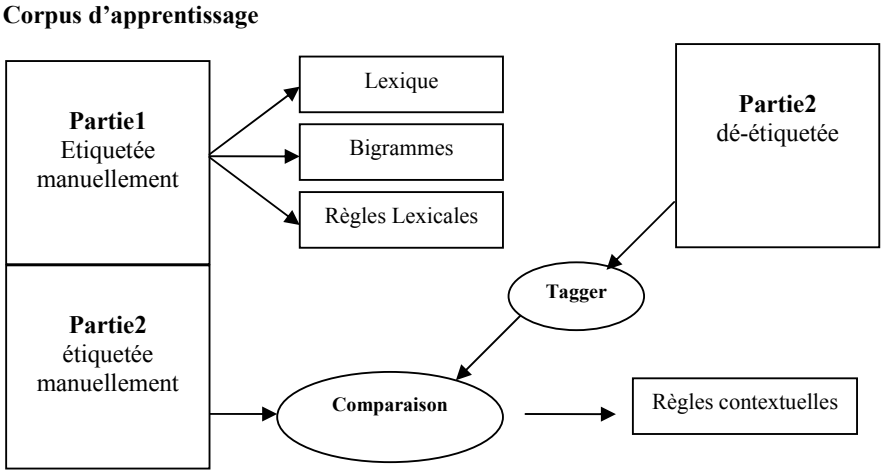


Figure IV.4: Etiqueteur de Brill

Cet apprentissage se fait en deux étapes :
La première aboutit à la création d'un fichier de règles (dites « lexicales ») destinées à l'étiquetage des mots inconnus (règles pour prédire l'étiquette la plus probable). En effet, le système utilise des méthodes de l'analyse distributionnelle pour extraire des règles lexicales. Il utilise la première moitié du corpus échantillon étiqueté manuellement ainsi que tout le corpus non codé disponible (les deux moitiés).
La seconde aboutit à la création d'un fichier de règles (dites « contextuelles ») nécessaires pour affiner l'étiquetage, c'est-à-dire tenter de revenir sur des affectations erronées. Toujours à partir du même corpus échantillon étiqueté manuellement, le système va déduire et apprendre une série de modèles de transformations qui seront déclenchées cette fois par l'environnement contextuel du code précédemment assigné.

Les quatre principales connaissances générées par apprentissage sont :

- Lexique : une liste de mots, chacun de ces mots associé à une liste d'étiquettes.

- **Règles lexicales** : spécifiant les transformations à effectuer sur la catégorie grammaticale affectée par défaut aux mots inconnus.

- **Règles contextuelles** : transformations contextuelles qui servent à affiner l'étiquetage en contexte.

- **Bigrammes** : Paires de mots adjacents

3.1.2 Codage

L'étiquetage se fait aussi en deux étapes : Dans la première, après lecture de chaque mot du texte, l'outil lui affecte son code le plus probable s'il est trouvé dans le lexique (mot connu). Pour les mots inconnus, il affecte une des deux étiquettes « par défaut » (NNP si le mot commence par une majuscule, et NN dans tous les autres cas). Cette affectation du code par défaut déclenche l'appel des règles lexicales destinées à affiner l'étiquetage des mots inconnus.

Pour chaque mot inconnu (resté NN ou NNP), chacune des règles lexicales est essayée, appliquée si les conditions sont remplies. Toutes sont essayées, successivement, et prennent en compte le résultat précédemment acquis. A la fin de cette étape, il peut rester des codes par défaut, car il se peut que le système, dans son apprentissage sur le corpus échantillon, n'ait pas rencontré un tel contexte, et n'ait donc pas pu déduire de règle de levée d'ambiguïté. Dans la seconde, le système revient sur l'étiquetage précédemment effectué, et applique systématiquement des modèles de transformations contextuelles, dans le but d'affiner l'étiquetage. A la fin de cette seconde étape, chaque mot aura reçu une étiquette correspondant à sa classe « en discours », c'est-à-dire en contexte (par exemple, un mot qui, historiquement, est un adverbe, pourra se retrouver, en contexte, étiqueté comme un nom ou un pronom).

Exemple de texte étiqueté par l'analyseur :

La/DTN:sg pédagogie/SBC:sg est/ECJ:sg une/DTN:sg oeuvre/SBC:sg de/PREP coordination/SBC:sg et/COO de/PREP rapports/SBC:pl ;/;
ne/ADV doit/VCJ:sg -/- elle/PRV:sg pas/ADV être/ENCFF considérée/ADJ1PAR:sg comme/SUB une/DTN:sg sorte/SBC:sg de/PREP philosophie/SBC:sg embrassant/VNCNT dans/PREP une/DTN:sg vue/SBC:sg d'/PREP ensemble/SBC:sg ce/PRO:sg qui/REL contribue/VCJ:sg à/PREP la/DTN:sg formation/SBC:sg de/PREP l'/DTN:sg esprit/SBC:sg ?/?
La/DTN:sg géographie/SBC:sg est/ECJ:sg tenue/ADJ1PAR:sg de/PREP puiser/VNCFF aux/DTC:pl mêmes/ADJ:pl sources/SBC:pl de/PREP faits/ADJ2PAR:pl que/SUB$ la/DTN:sg géologie/SBC:sg ,/, la/DTN:sg physique/SBC:sg ,/, les/DTN:pl sciences/SBC:pl naturelles/ADJ:pl et/COO ,/, à/PREP certains/DTN:pl égards/SBC:pl ,/, les/DTN:pl sciences/SBC:pl sociologiques/ADJ:pl ./.
Elle/PRV:sg se/PRV:sg sert/VCJ:sg de/PREP notions/SBC:pl dont/REL quelques_-_unes/PRO:pl sont/ECJ:pl l'/DTN:sg objet/SBC:sg d'/PREP études/SBC:pl approfondies/ADJ2PAR:pl dans/PREP des/DTN:pl sciences/SBC:pl voisines/ADJ:pl ./.
De/PREP là/ADV vient/VCJ:sg ,/, pour/PREP le/PRV:sg dire/VNCFF en/PREP passant/VNCNT ,/, le/DTN:sg reproche/SBC:sg qui/REL lui/PRV:sg est/ECJ:sg parfois/ADV adressé/ADJ1PAR:sg de/PREP vivre/VNCFF d'/PREP emprunts/SBC:pl ,/, d'/PREP intervenir/VNCFF indiscrètement/ADV dans/PREP le/DTN:sg champ/SBC:sg d'/PREP autrui/PRO:sg ,/, comme/SUB s'/SUB il/PRV:sg y/PRV:++ avait/ACJ:sg des/DTN:pl compartiments/SBC:pl réservés/ADJ2PAR:pl dans/PREP le/DTN:sg domaine/SBC:sg de/PREP la/DTN:sg science/SBC:sg ./.

3.2 Base de connaissances INALF (Institut National de la Langue Française)

Plusieurs versions du système Brill ont été développées. La version utilisée actuellement à l'INaLF est la version 1.14, fonctionne sous UNIX. Cette version a récemment été « portée » sous Windows95 et distribuée sous le nom de WinBrill-0.3. La version courante est WinBrill-0.4 (Windows 95/98/NT). Elle est distribuée gratuitement. La version libre de WinBrill[2] est livrée sans lexique ni règles lexicales et contextuelles. C'est à l'utilisateur de lui fournir les fichiers de données.

[2] http://atilf.atilf.fr/winbrill

En complément de WinBrill, l'InaLF fournit un lexique et les fichiers de règles issus de l'apprentissage du catégoriseur de Brill sur la base de données FRANTEXT[3], moyennant signature d'une convention avec l'INaLF.

3.2.1 Jeu d'étiquettes

D'une manière générale, il existe deux types d'approches d'étiqueteurs : une approche minimaliste où l'analyseur fonctionne avec un ensemble d'étiquettes très restreint (ex. D. LABBE, qui travaille sur le français avec 16 codes différents seulement), et une approche maximaliste avec un ensemble d'étiquettes extensif, très complet et très précis (ex. STEIN et DAMOVA à Stuttgart, qui travaillent sur le français avec 190 codes différents; ou encore pour l'anglais avec 135 étiquettes pour le corpus LOB ou 197 étiquettes pour le corpus London-Lund, etc.).
Le jeu d'étiquettes utilisé à l'INALF dans les Lexiques (TLFnome de MAUCOURT + PAPIN + REIMEN) contient environ 100 codes différents pour 5 grandes catégories du discours :

- 45 étiquettes pour les verbes conjugués,

- 3 étiquettes pour les verbes non conjugués,

- 4 étiquettes pour les adjectifs,

- 4 étiquettes pour les substantifs,

- et 47 pour le reste des catégories : pronoms, adverbes, prépositions, conjonctions, etc.

Le nombre d'étiquettes adoptées pour la version BRILL1.4-JL5 / WINBRILL-0.3 est 50 (annexe A), en plus d'une quinzaine de signes de ponctuation.

3.2.2 Corpus-échantillon

Le corpus utilisé contient **417370** occurrences et a les caractéristiques suivantes : c'est un fichier non distribuable, car il contient des morceaux de textes sous droits d'auteurs ou d'éditeurs, tirés des bases Frantext, Scitech ou autres :

Balzac, Honoré.de	César Birotteau	3008 occurrences
Lhote, Jean	La Communale	3010 occurrences
Romilly, Jacqueline de	La Montagne Sainte-Victoire	3005 occurrences
Victor, Paul-Emile	Boréal	3005 occurrences
Zola, Emile	Germinal, 1°partie (Frantext, L465)	115004 occurrences
Leroux, Gaston	Le Mystère de la Chambre	55897 occurrences
Jaune (Frantext, L782)		
Gyp	Souvenirs d'une Petite Fille	
(Frantext L269)	54099 occurrences	
Dumas, Alexandre	La Dame aux Camélias	

[3] FRANTEXT est un vaste corpus, à dominante littéraire, constitué de textes français qui s'échelonnent du XVIe au XXe siècle. http://atilf.atilf.fr/frantext.htm.

(Frantext L834)	78711 occurrences	
Sue, Eugène	Atar-Gull (Frantext, M279)	4781 occurrences
Foch, Maréchal	Mémoires (Frantext L243)	4822 occurrences
Brillat-Savarin	Physiologie du goût	
(Frantext M362)	4431 occurrences	
Karr, Alphonse	Sous les Tilleuls (Frantext M384)	4692 occurrences
Constant, Benjamin	Le Cahier Rouge (Frantext M386)	4325 occurrences
Sainte-Beuve	Volupté (Frantext M652)	4404 occurrences
Flaubert, Gustave	Smarh (Frantext M736)	4534 occurrences
Janin, J.	Ane mort et Femme guillotinée	
(Frantext M784)	4479 occurrences	
Broussais	Cours de Phrénologie (Frantext P938)	4336 occurrences
Pelt, Jean-Marie	Tour du Monde d'un écologiste	
(Scitech,T017)	16313 occurrences	
Purves ; Orian ; Heller	Biologie Animale (Scitech T022)	36120 occurrences
Jouventin, Pierre	nouv. science biol. : l'écologie	
(Scitech T025)	6197 occurrences	
ainsi qu'un fichier de définitions géologiques (systèmes d'érosion)		2445 occurrences

3.2.3 Base de connaissances

A partir du corpus-échantillon manuellement étiqueté, le système crée sa Base de Connaissances, qui servira à l'opération d'étiquetage. Cette base est constituée des connaissances suivantes :

- Le lexique : Association des mots à leurs étiquettes, de la plus vers la moins probable. Il contient actuellement 440544 entrées (lexique InaLF/TLFnome95, lexique auto-appris par le système contenant 25000 entrées). Chaque « entrée » du lexique contient l'occurrence telle qu'en contexte, dans sa forme fléchie et/ou accordée, et une seule étiquette non ambiguë, ou une suite d'étiquettes (priorité à la première, la plus fréquente).

Exemple :

réduit	VCJ:sg SBC:sg ADJ1PAR:sg VPAR:sg ADJ2PAR:sg
bon_gré_,_mal_gré	ADV
répondant	VNCNT SBC:sg
actives	ADJ:pl
indiquent	VCJ:pl
cette_fois_-_ci	ADV
énonça	VCJ:sg
Catherine	SBP:sg

Le lexique ne donne pas toutes les étiquettes possibles pour un mot donné. En effet, le lexique « auto-appris » ne recense que les emplois effectivement rencontrés en contexte. Cette philosophie permet de réduire les ambiguïtés et de plus est très rentable en terme de performances du système. Par exemple, le mot *rocher* est théoriquement possible comme verbe infinitif : il vaut mieux avoir une erreur les rares fois où il est effectivement verbe, que des erreurs fréquentes chaque fois qu'il est substantif.

- **Les règles lexicales** : Règles spécifiant les transformations à effectuer sur la catégorie grammaticale affectée par défaut aux mots inconnus. Il existe actuellement 342 règles.

Exemples de règles lexicales :
une goodright SBC:sg 59.1383656752863
SBC:sg est fgoodright ADJ:sg 71.3203621248091
SBC:pl nous fgoodright VCJ:pl 154.4
avait goodright VPAR:sg 137.659127089446
SBC:pl ais fhassuf 3 VCJ:sg 122.65
SBC:pl és fhassuf 2 ADJ2PAR:pl 119.604761904762
SBC:sg ai fhassuf 2 VCJ:sg 114
SBC:sg e fdeletesuf 1 ADJ:sg 111.458706750157
du goodright SBC:sg 106.678005154052
ées hassuf 3 ADJ2PAR:pl 103.088888888889
SBC:sg ir fhassuf 2 VNCFF 99.5673307005528
NN é fchar ADJ2PAR:sg 90.6047619047619

Règle *: une goodright SBC:sg 59.1383656752863*
signifie : Tout mot contigu linéairement à *une*, sur sa droite, est à étiqueter SBC :sg
goodright : Opérateur permettant d'atteindre le mot immédiatement à droite.
SBC:sg étiquette
*59.1383656752863 :*Score

Règle: *SBC:sg est fgoodright ADJ:sg 71.3203621248091*
signifie : Tout mot inconnu venant d'être étiqueté comme Substantif va voir son étiquette remise en question s'il est à la droite du mot *est*. Dans ce cas, il devient Adjectif (SBC :sg => ADJ :sg)
fgoodright : Opérateur permettant d'atteindre le mot immédiatement à droite, mais en posant une condition sur l'étiquette du mot en cours d'examen.

- **Les règles contextuelles** : Règles de transformations contextuelles qui servent à affiner l'étiquetage en contexte. Il existe actuellement 654 règles.

Exemples de règles contextuelles :

PREP DTN:pl WDAND2TAGAFT De SBC:pl
DTN:pl PRO:pl NEXTTAG VCJ:pl
DTN:sg PRV:sg NEXTTAG VCJ:sg
PRV:sg PRV:pl NEXT1OR2TAG VCJ:pl
ADJ2PAR:sg ADJ1PAR:sg PREV1OR2OR3TAG ECJ:sg
SBC:sg ADJ:sg PREVTAG SBC:sg
PRV:sg PRO:sg WDPREVTAG PREP elle
VPAR:sg ADJ1PAR:sg PREV1OR2OR3TAG ECJ:sg
DTN:sg PRO:sg NEXTTAG REL
SBC:sg VCJ:sg PREV1OR2TAG PRV:sg
ADJ2PAR:sg VPAR:sg PREV1OR2OR3TAG ACJ:sg
SUB$ SUB WDPREVTAG PREP que

Règle: *PREP DTN:pl WDAND2TAGAFT De SBC:pl*

Signifie : Le mot en cours d'examen est *De.* Son étiquette PREP est à transformer en DTN :pl (article pluriel) si le deuxième mot sur sa droite est étiqueté SBC :pl (Substantif pluriel).
WDAND2TAGAFT : Opérateur qui considère le mot en cours et la deuxième étiquette suivante.

Règle : *DTN:pl PRO:pl NEXTTAG VCJ:pl*
NEXTTAG : Opérateur faisant appel à l'étiquette suivante.

- **Les bigrammes** : Paires de mots adjacents

Exemples :
presque bleu
résultait qu'
constaté que
bêtes s'
aux souris
au 30
cela nous

4 Normalisation

Avant l'analyse linguistique, une phase de normalisation des mots est effectuée en réduisant les variantes morphologiques à une forme commune (rendre les verbes à l'infinitif, supprimer les formes plurielles, etc.), souvent appelée terme. Pour cela, nous utilisons un analyseur morphologique FLEMM.

FLEMM est un programme Perl[4] (v5) qui effectue l'analyse morphologique flexionnelle de textes français préalablement étiquetés par BRILL ou TreeTagger[5] [SCH 94]. Il est essentiellement basé sur l'usage de règles (i.e. il utilise un lexique de 3000 mots environ pour la gestion des exceptions). Ce programme fonctionne sur PC ou machine Unix (SE Unix, Linux, WindowsNT/9x)[6].

FLEMM calcule le lemme de chaque mot fléchi (en fonction de son étiquette) et fournit également ses principaux traits morphologiques :

- genre et nombre pour les adjectifs, déterminants, participes,

- nombre pour les noms

- genre, nombre, personne et cas pour les pronoms,

- nombre, personne, temps, mode et groupe de conjugaison pour les verbes.

Table IV.1 résume le format de sortie des mots après normalisation.

[4] http://www.activestate.com/
[5] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger.html
[6] http://atilf.atilf.fr/winbrill/

Catégorie grammaticale	Format de sortie
Verbe (conjugué)	Mot Fléchi/étiquette:personne:nombre:temps:mode/Lemme:grp/
Adjectifs, noms, déterminants, pronoms relatifs, participes.	Mot Fléchi /étiquette:genre:nombre/Lemme/
Pronoms personnels	Mot Fléchi /étiquette:personne:genre:nombre:cas/Lemme/
Autres	Mot / étiquette / Mot

Table IV.1 : Format de sortie du lemmatiseur

Figure IV.5 présente un exemple de résultat du lemmatiseur.

Message : *Bonjour Omar. Tout s'est très bien passé. C'est dommage que tu n'aies pas pu venir, car c'est le moment d'apprendre beaucoup de choses. C'est le moment aussi pour prendre des contacts. Bon courage. Alain.*

Message étiqueté : *Bonjour/INJ Omar/SBP:sg ./. Tout/DTN:sg s'/PRV:sg est/ECJ:sg très/ADV bien/ADV passé/ADJ2PAR:sg ./. C'/PRV:sg est/ECJ:sg dommage/SBC:sg que/SUB$ tu/PRV:sg n'/ADV aies/ACJ:sg pas/ADV pu/VPAR:sg venir/VNCFF ,/, car/COO c'/PRV:sg est/ECJ:sg le/DTN:sg moment/SBC:sg d'/PREP apprendre/VNCFF beaucoup/ADV de/PREP choses/SBC:pl ./. C'/PRV:sg est/ECJ:sg le/DTN:sg moment/SBC:sg aussi/ADV pour/PREP prendre/VNCFF des/DTC:pl contacts/SBC:pl ./. Bon/ADJ:sg courage/SBC:sg ./. Alain/SBP:sg ./.*

Message lemmatisé : *Bonjour/INJ/bonjour Omar/SBP/omar ./. Tout/DTN:m:s/tout s'/PRV:3p:_:s:{a|d}/lui est/ECJ:3p:s:pst:ind/être:3g très/ADV/très bien/ADV/bien passé/ADJ2PAR:m:s/passer ./. C'/PRV:3p:_:s:n/ce est/ECJ:3p:s:pst:ind/être:3g dommage/SBC:_:s/dommage que/SUB$/que tu/PRV:2p:_:s:n/lui n'/ADV/n' aies/ACJ:2p:s:pst:subj/avoir:3g pas/ADV/pas pu/VPAR:m:s/pouvoir venir/VNCFF/venir ,/, car/COO/car c'/PRV:3p:_:s:n/ce est/ECJ:3p:s:pst:ind/être:3g le/DTN:m:s/le moment/SBC:_:s/moment d'/PREP/de apprendre/VNCFF/apprendre beaucoup/ADV/beaucoup de/PREP/de choses/SBC:_:p/chose ./. C'/PRV:3p:_:s:n/ce est/ECJ:3p:s:pst:ind/être:3g le/DTN:m:s/le moment/SBC:_:s/moment aussi/ADV/aussi pour/PREP/pour prendre/VNCFF/prendre des/DTC:_:p/du contacts/SBC:_:p/contact ./.*

Bon/ADJ:m:s/bon courage/SBC:_:s/courage ./. Alain/SBP/alain ./.

Jeu d'étiquettes : INJ : *Interjection.* ***SBP :*** *Substantif, nom propre ou à majuscule.* ***DTN :*** *Déterminant.* ***PRV :*** *Pronom « supporté » par le verbe (conjoint, clitique).* ***ECJ :*** *Verbe « être », conjugué.* ***ADV :*** *Adverbe.* ***ADJ2PAR :*** *Participe passé adjectival (non après auxiliaire).* ***SBC :*** *Substantif, nom commun.* ***SUB$:*** *Subordonnant possible. = Code par défaut de « que ».* ***ACJ :*** *Verbe « avoir », conjugué.* ***VPAR :*** *autre Verbe, non conjugué, participe passé après « avoir ».* ***VNCFF :*** *autre Verbe, non conjugué, infinitif.* ***COO:*** *Coordination.* ***PREP :*** *Préposition.* ***DTC :*** *Déterminant de groupe nominal, contracté.* ***Sg:*** *Singulier, etc.*

Traits : *personne (1p, 2p, 3p), genre (m : masculin, f : féminin), nombre (s : singulier, p : pluriel), temps (pst : présent, impft : imparfait, fut : futur, ps : passé simple), mode (ind : indicatif, subj : subjonctif, cond : conditionnel, imper : imperatif), groupe (1gr, 2gr, 3gr), cas (n : nominatif, a : accusatif, d : datif et o : oblique).*

Figure IV.5 : Résultat du lemmatiseur

5 Analyseur linguistique

L'analyseur automatique a pour but d'identifier et d'extraire les différentes propriétés linguistiques permettant de caractériser le contenu de chaque texte. Il est indépendant de tout domaine d'application : il reçoit, en entrée, un texte, il délivre, en sortie, la représentation associée.

L'analyseur fait passer un texte par les différents modèles linguistiques réduits de base et délivre en sortie le vecteur texte associé (figure IV.6). L'objectif de chaque niveau est d'analyser un texte et d'extraire un ensemble de caractéristiques.
En sortie, un texte est représenté par un ensemble d'entités lexicales et d'une suite de caractéristiques (architecturales, énonciatives, structurelles et syntaxiques). Les caractéristiques sont représentées par des variables dans le vecteur texte. Ces variables dénotent la présence ou l'absence de ces caractéristiques dans le texte.

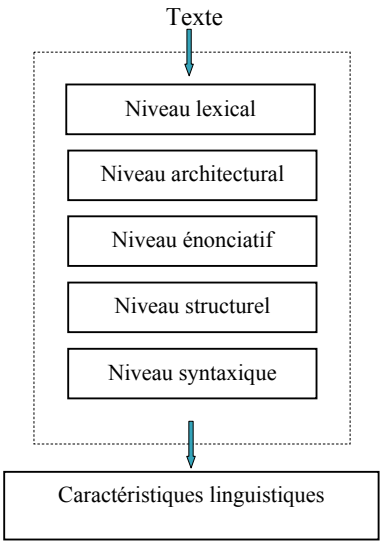


Figure IV.6: Analyse linguistique

5.1 Motivation

La représentation de textes la plus utilisée est la représentation vectorielle [SAL 83]. Un texte est représenté par un vecteur d'unités lexicales spécifiques. Cette représentation pose la question de la pertinence, qui vient de celle, plus fondamentale, de l'association d'un contenu à un ensemble de formes linguistiques. C'est-à-dire quels critères permettent de garantir que la représentation est pertinente pour un texte donné ?
Pour améliorer cette représentation classique, nous proposons d'élargir l'espace de propriétés. En effet, lorsque nous analysons un texte donné, nous déterminons également, en plus des caractéristiques lexicales, d'autres caractéristiques qui semblent être d'intérêt. Ces caractéristiques supplémentaires constituent un certain ensemble d'indices que nous ajoutons à la représentation lexicale. Ces propriétés sont basées sur des modèles linguistiques réduits. Elles constituent un ensemble d'indices qui portent généralement sur la structure et le contenu des textes. Le système de filtrage utilise donc un ensemble de connaissances linguistiques de base, sous forme de modèles réduits, pour analyser, extraire les différentes propriétés, et construire la représentation interne de chaque texte. Ces modèles sont indépendants du domaine d'application.
Dans ce qui suit, nous présentons tout d'abord les travaux antérieurs sur l'utilisation des connaissances linguistiques et nous décrivons, ensuite, les différentes ressources ou connaissances de base que nous avons construit pour le bon fonctionnement de notre système de filtrage.

5.2 Travaux antérieurs

La croissance accélérée d'Internet et la grande quantité d'information devenue disponible sur ce réseau, ont motivé les recherches dans le domaine de la linguistique informatique (computational linguistic).
Les connaissances linguistiques (données lexicales, structure du texte, etc.) sont utilisées de plus en plus dans les systèmes d'analyse de textes, par exemple pour identifier l'information pertinente [POI 99] [MAR 97] [MIN 01]. Diverses études en linguistique informatique ont proposé des méthodes de classification de textes. Il s'agit d'extraire des propriétés textuelles pour faire de la classification, par exemple, Chandrasekar & Srinivas [CHA 98] utilisent une analyse superficielle (shallow parsing).
Une littérature abondante essaie de distinguer genre et type de texte. Chez Biber, la distinction entre genre et type de texte se base sur la notion de critère interne et externe : le genre d'un texte est ce qui est déterminé par des critères externes et le type est ce qui est déterminé par des critères internes. Les critères internes sont ceux qui font appel aux seules propriétés linguistiques contenues dans les textes, (il s'agit essentiellement des propriétés syntaxiques et lexicales) et les critères externes sont ceux qui font appel à des propriétés qui déterminent la situation dans laquelle le texte intervient (par exemple sous quelle rubrique d'un journal un article est-il présenté ?). Biber identifie 67 propriétés de divers ordres pour classer les textes. Ces propriétés sont d'ordre syntaxique (temps verbaux, présence d'auxiliaires, passivation, nominalisations, etc.) mais aussi sémantiques (classes d'adverbes, types de modalités). Copeck et al 2000 ajoutent à un ensemble de propriétés syntaxiques et sémantiques, des propriétés d'ordre pragmatique telles que la présence d'une introduction ou l'utilisation de conventions.
Dans ces études les propriétés utilisées ne sont pas issues d'un modèle linguistique. Les propriétés sont collectées sur la base d'observations directes ou de travaux antérieurs. Chez Biber les propriétés linguistiques étaient sélectionnées sur la base d'études sociolinguistiques orientées pour la plupart vers la distinction entre les productions orales et écrites. Chez Copeck, les propriétés sont issues de l'introspection des analystes, de l'observation sur corpus, de la comparaison entre textes et de collectes effectuées sur des travaux antérieurs. Les groupements de ces propriétés en différents niveaux linguistiques sont postérieurs à leur collecte : on énumère dans un premiers temps des propriétés hétérogènes qui ont pour but de distinguer les différents types de textes. De plus des différences existent entre les études en fonction :

- de la nature du corpus utilisé (annoté ou non, catégorisé ou non, volumineux ou non),

- des moyens mis en oeuvre (manuel ou automatique),

- du but recherché (catégoriser, dégager un seul type pour en définir les traits essentiels, ou classifier pour observer des regroupements),

- du volume de données à traiter (ce qui permet un traitement coûteux ou non).

Dans notre cas, le point de départ est l'utilisation de modèles linguistiques réduits qui permettent de dégager des propriétés linguistiques qui devraient permettre de distinguer les différents types de textes. La fiabilité du système repose sur l'opération d'apprentissage (performances de l'apprentissage). Un texte est identifié par un ensemble de caractéristiques. Il est représenté conceptuellement par un espace vectoriel de k dimensions: $M = \{(T1,W1), (T2,W2, \ldots , (Tk,Wk)\}$ (Ti: ième caractéristique, Wi: poids et k : espace des caractéristiques).

5.3 Les modèles linguistiques réduits

Nous avons défini et identifié un ensemble de propriétés, susceptibles d'influer sur le processus de filtrage, que nous avons automatisé. Il s'agit d'un ensemble d'indicateurs sur le texte qui permettent de caractériser un texte et de le situer par rapport aux autres. C'est-à-dire, de rapprocher les textes qui appartiennent à la même classe ou éloigner ceux qui appartiennent à des classes différentes.

5.3.1 Modèle lexical

Il représente l'ensemble des entités lexicales les plus pertinentes du domaine traité. Il est généré automatiquement à partir de corpus. Il constitue le noyau de base sur lequel repose toute méthode d'identification et de représentation des documents textuels. Pour chaque application (nouveau corpus), le modèle lexical est défini et identifié d'une façon automatique. Il est classé en deux types : mots simples et mots composés.

a)- Mots simples (MS)

Ils représentent le vocabulaire de base. Il est constitué d'unités linguistiques spécifiques les plus pertinentes (mot, lemme, etc.). Initialement, chaque texte du corpus subit un prétraitement de filtrage qui permet d'éliminer les mots communs (articles, prépositions, etc.).

b)- Mots composés ou phrases très courtes (MC)

Les mots composés sont générés à partir des listes *bigrammes* et *trigrammes* apprises par le système. L'objectif de l'utilisation de mots composés dans notre cas de filtrage d'information est de représenter les textes par des éléments beaucoup moins ambigus, et de viser une précision accrue lors du processus de filtrage (éviter le bruit dû à l'ambiguité des termes simples).

Les mots simples ont pour intérêt principal de favoriser le *rappel,* mais en revanche, ils sont ambiguës (exemple : cours du dollar, cours de maths, etc.), ce qui pénalise la *précision*. Les mots composés favorisent, quant à eux, la précision, un mot simple ambigu étant fréquemment désambiguisé par son emploi au sein d'une séquence complexe, mais pénalisent le *rappel,* la séquence composée devant se retrouver exactement à l'identique dans le texte et le profil.

c)- Critère de réduction

Le vocabulaire construit lors du traitement du corpus est généralement très volumineux. Le critère utilisé, pour réduire le vocabulaire généré automatiquement, est la mesure de l'information mutuelle (Yang & Pedersen, 1997). L'information mutuelle MI (*t, C*) mesure la dépendance d'un terme *t* et d'une classe *C.* Elle est définie par :

$$MI(t,C) \approx \frac{N * p}{(p + p')(p + q)}$$

Avec :
p : est le nombre de textes de classe *C* qui contiennent le terme *t.*
q : est le nombre de textes qui ne sont pas de classe *C* mais qui contiennent le terme *t.*
p' : est le nombre de textes de classe *C* qui ne contient pas le terme *t*.
q' : est le nombre de textes qui ne sont pas de classe *C* et qui ne contient pas le terme *t.*
N : est le nombre total de textes du corpus.

Cette mesure numérique permet de déceler les mots qui « s'attirent », c'est-à-dire qui tendent à apparaître ensemble. Une information mutuelle élevée entre un terme et une classe est le signe d'un lien fort entre ces deux éléments. Le vocabulaire est réduit alors aux termes dont les informations mutuelles sont les plus élevées.

5.3.2 Modèle concernant la mise en forme matérielle (l'architecture du texte)

L'ingénierie linguistique (TAL) est confrontée à un nouveau problème : l'architecture matérielle d'un texte. En effet, il est nécessaire de travailler non plus seulement sur des phrases ou des énoncés isolés mais sur des textes entiers (pages web, emails, etc.). Effectuer un traitement automatique d'un document textuel nécessite des opérations préalables aux analyses ultérieures (syntaxiques, sémantiques et pragmatiques).
En particulier, chaque texte possède deux structures : une structure formelle et une structure discursive. La première structuration conditionne la seconde. La structure formelle est déjà porteuse d'une certaine intentionnalité signifiante; elle est le résultat d'un codage dans un système typographique et celui d'une mise en texte. Le traitement préalable d'un texte doit exploiter cette structuration formelle (repérage des titres et sous titres; découpage d'un texte en paragraphes, énoncés, phrases, propositions, mots; repérage des citations; identification des énumérations; prise en compte des ordres dispositionnels dans les textes; repérage des images, diagrammes, légendes, encadrés) avant de procéder aux opérations ultérieures et à l'exploitation de la structuration discursive (identification des cadres thématiques; des relations causales, définitoires, temporelles; des relations entre concepts, termes, événements; des liens anaphoriques; etc.).
Les signes typographiques et de ponctuation sont des « baliseurs naturels » de l'information, mais aussi, des indicateurs sur lesquels devrait désormais s'appuyer la plupart des traitements automatiques (syntaxiques, sémantiques, prosodique, extracteurs d'information, etc.). Ils jouent un rôle important dans le traitement automatique des langues (segmentation de textes, analyse syntaxique, filtrage d'informations, etc.).
L'analyse de l'architecture matérielle de textes concerne l'analyse de la structure logique de textes. Elle permet de localiser et d'identifier la nature des zones de textuelles (entête, titre, corps, paragraphe, section, listes, tableaux, etc.). Dans ce cadre, la ponctuation des textes est une extension des signes de ponctuation de la phrase (elle peut inclure le format de titres, la forme des paragraphes, les notes de bas de page). Cette ponctuation textuelle permet de percevoir des objets textuels (des chapitres, des sections, des paragraphes, etc.) et aussi des relations entre ces objets textuels (inclusion, liens sémantiques, etc.). L'ensemble des objets et des relations définit l'architecture du texte. La syntaxe de la ponctuation des textes repose sur un ordre de marques lexicales en fonction de leur contenu (comme introduction et conclusion) et sur des propriétés typo-positionnelles (par exemple qui est centré en tête de page est un titre).
Dans le cadre de ce travail, nous ne cherchons pas à retrouver précisément chaque expression sémantique qui nous permettrait de mettre à jour chaque acte de langage qui se trouverait dans toutes les configurations particulières cependant nous allons utiliser les procédés qu'utilisent

la syntaxe de la ponctuation de texte pour alimenter notre système de critères distinctifs entre les textes.
Voici quelques exemples d'indicateurs (voir annexe C): *titres, section, introduction, conclusion, images, dessins, ponctuation, type du contenu, auteur du texte, longueur moyenne des textes, longueur moyenne des phrases, caractères non alphanumérique, caractères numériques,* etc.

5.3.3 Modèle énonciatif

La prise en compte de tous les phénomènes liés aux conditions de production du discours apparaît comme pertinente pour la compréhension du fonctionnement de la langue. Lorsqu'on aborde le sens des unités linguistiques, on est inévitablement amené à les relier à des facteurs extralinguistiques, c'est-à-dire à leur référence comme à leur prise en charge par un énonciateur. C'est-à-dire le discours a ses conditions de production. Dans l'idée de BENVENISTE, le locuteur est le paramètre essentiel dans la mise en fonctionnement de la langue (il faut porter le regard sur l'acte par lequel le discours est produit). Le locuteur s'inscrit dans l'énoncé par l'intermédiaire d'indicateurs linguistiques (pronoms personnels, formes verbales, formes temporelles). Par exemple, les pronoms désignant la personne (je/tu, nous/vous ou il/ils) branchent l'énoncé à l'instance qui l'énonce.
Voici quelques exemples d'indicateurs (voir annexe C): *1ere pers du singulier, 2eme pers du singulier, 1ere pers du pluriel, 2eme pers du pluriel, 3 pers (singulier, pluriel), les déterminants (mon, ton, son, ce, etc.), discours rapporté direct, énoncer (admettre, dire, déclarer, remarquer, protester, etc.), penser, croire, révéler, supposer, estimer,* etc.

5.3.4 Modèle structurel

Des théories basées sur l'étude du discours attribuent une représentation arborescente à la structure du texte. Les feuilles de l'arbre représentent des énoncés linguistiques ou leur représentation sémantique. La structure de l'arbre est basée sur la nature de la relation qui est établie entre les empans de textes que l'on relie entre eux. Cependant ces relations qui existent entre chaque proposition sont dans la plupart des cas implicites. Par exemple une relation causale entre deux énoncés peut être établie sur une connaissance du monde partagée par les deux locuteurs. De ce fait sans une analyse sémantique ``profonde'' du texte nous n'avons accès qu'aux termes explicites de ces relations. En effet un certain nombre de marqueurs linguistiques (un ensemble de mots clé) précisent la relation ou un ensemble de relations potentielles entre les deux segments de textes reliés par ce marqueur. De nombreux travaux ont porté sur l'identification de l'information contextuelle et sémantique pour extraire l'information pertinente des textes [DES 97] [GAR 98] [BEN 02] [MIN 01].
Dans un premier temps les seuls indices que nous avons sur la structure des textes sont les cas d'explicitation sous forme de mots clés des relations 'rhétoriques' des textes. La collecte des mots clés se fait sur la base de travaux divers sur des relations [MAR 97].
Nous avons instauré une hiérarchie des termes basée sur leur position. Marcu remarque que la position de certains mots clés est liée à leur fonction qui est soit discursive, soit syntaxique. Nous avons distingué trois types de position en début de paragraphe, en début de phrase, après une virgule et les autres positions. A chacune de ces positions nous avons pondéré la relation différemment en fonction de l'importance des empans de texte qui peuvent être mis en relation. Un mot clé en début de paragraphe peut mettre en relation deux paragraphes, alors qu'un mot clé en début de phrase peut mettre en relation deux phrases : nous accordons un

poids plus important au mot clé qui porte sur une zone plus importante donc à celui qui se trouve entre deux paragraphes.
Voici quelques exemples d'indicateurs (voir annexe C): *addition (à cela s'ajoute qu, ainsi qu, aussi, d'autre part, de plus, etc.), analogie (c'est-à-dire, comme, de la même façon, de même, etc.), but (pour qu, de sorte qu, etc.), cause (afin qu, c'est pourquoi, etc.), exemple (à savoir, par exemple, etc.), focus (particulièrement, précisément, etc.), intensité (assez, au point qu, etc.),* etc.

5.3.5 Modèle syntaxique

L'analyse syntaxique est une composante très importante dans le traitement automatique des langues. L'analyse syntaxique regroupe divers courants qui diffèrent sur les objectifs visés et sur les méthodes employées. Les méthodes couvrent par exemple les approches stochastiques, les approches d'analyse locale et les approches plus traditionnelles d'analyse complète (ou profonde). Les objectifs vont de la segmentation en syntagmes à l'analyse profonde avec une grammaire à large couverture, en passant par des analyseurs robustes et/ou superficiels. Néanmoins cette diversité dans les méthodes et objectifs reflète une certaine complémentarité plutôt qu'une opposition absolue.
Dans le cadre de ce travail, nous ne cherchons pas à retrouver précisément la structure syntaxique de chaque énoncé, cependant nous cherchons à identifier un ensemble d'indices syntaxiques pour alimenter notre système de filtrage.

Voici quelques exemples d'indicateurs (voir annexe C): *taux de pronoms (à chaque personne)/nombre total de pronoms, taux de déterminants (à chaque personne)/nombre total de déterminant, taux de noms propres (substantifs)/nombre total de noms, nominalisation, nombre d'adverbes (temps/lieu)/phrase et texte, nombre d'adjectifs/phrase et texte, infinitifs, participe passé, coordination, négation, démonstratif, indéfinis (anaphorique), relatif sujet/objet, subordination, interrogation, interjection, abréviation, forme active/forme passive,* etc.

L'extraction de ces indices d'ordre syntaxique nécessite une phase d'étiquetage préalable. En effet, nous avons utilisé un étiqueteur morpho-syntaxique, analyseur de Brill (Brill, 1992).
L'analyse consiste à rechercher dans le texte étiqueté par le tagger Brill des séquences d'étiquettes correspondantes à chaque type d'indice. La table IV.2 donne quelques exemples de ce type d'analyse.

Indice linguistique	**Séquence d'étiquettes**	**Exemples**
LES INTERJECTIONS	/INJ	*hélas, oui, non, ben, hein, etc.*
LES RELATIFS	/REL	*c'est lui qui/REL sera bien attrapé.*
LES SUBSTANTIFS	/SBC, /SBP	*il roule en voiture/SBC:sg*
LES INFINITIFS	/VNCFF, /ANCFF, /ENCFF	*sans vouloir/VNCFF aller/VNCFF le dénoncer/VNCFF*
LES ABRÉVIATIONS	/ABR	p., pp., etc.
FORME PASSIVE	/ADJ1PAR	
SUBORDINATION	/SUB	
Etc.	Etc.	Etc.

Table IV.2 : Quelques séquences d'étiquettes

Par exemple, pour extraire des constructions en forme passive, l'analyseur recherche dans le texte d'étiquettes, produit par le tagger Brill, la séquence d'étiquette suivante : /ADJ1PAR.

Figure IV.7 présente un exemple simple de résultat de l'analyseur linguistique.

Message source :

. . .
> Bonjour Alain,
> J'espère que ton déplacement en suisse a été bénéfique.
> Raconte moi un peu ton aventure.
> Omar
Bonjour Omar.
Tout s'est très bien passé. C'est dommage que tu n'aies pas pu venir,
car c'est le moment d'apprendre beaucoup de choses.
C'est le moment aussi pour prendre des contacts.
Bon courage.
Alain.
p.s.: envoie moi l'article.

Message après analyse :

…
> Bonjour Alain,
> J'espère que ton déplacement en suisse a été bénéfique.
> Raconte moi un peu ton aventure.
> Omar
Bonjour Omar.
Tout s'est très bien passé. C'est dommage que tu n'aies pas pu venir,
car c'est le moment d'apprendre beaucoup de choses.
C'est le moment aussi pour prendre des contacts.
Bon courage.
Alain.
p.s.: envoie moi l'article.

Pour ne pas surcharger le texte, certaines propriétés ne sont pas représentées.

Jeu de propriétés :

- Propriétés lexicales
Bonjour, C'est, dommage, moment, beaucoup, Bon courage

- Propriétés Matérielles:
Type du message : txt, p.s., Langue: Français, Destinataires : 1, Auteur : regnier@lpl.univ-aix.fr, Taille du message (mots):38, Taille du message (phrases):7, Horaire : jour.

- Propriétés énonciatives:
Prem. Pers. Sing., moi, Deux. Pers. Sing., tu, Réponse, >.

- Propriétés Syntaxiques:
Pronoms, Substantifs, Interjections, Déterminants, Adverbes, Adjectifs, Infinitifs, Participes Passés, Coordinations, Subordinations, Abréviations.

- Propriétés structurelles:
Addition.

Représentation vectorielle :

Type	ps	Lang.	pps	dps	Aut.	rp	prn	sbc	Int	Dét	Adv	Adj	Inf	pp	coo
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Dest.	sub	Abr	taille	Hor.	addit	Autre
1	1	1	1	1	1	0

Figure IV.7 : Résultat de l'analyse linguistique

6 Expansion de la représentation

Ce module consiste à compléter la représentation d'un texte délivré par l'analyseur. L'expansion de la représentation utilise deux connaissances implicites: d'une part, un réseau lexical permettant de prendre en considération les termes qui existent dans le texte et qui n'existent pas dans le profil, en les remplaçant par des termes du profil sémantiquement proches. D'autre part, la cooccurrence des critères de filtrage en prenant en considération les termes qui n'existent pas dans le texte à filtrer, mais qui corrèlent avec certains termes du profil. Cette nouvelle représentation constitue l'entrée du module de filtrage.
Nous décrivons notre proposition pour la prise en compte de l'aspect sémantique dans le processus de filtrage. Nous exposons tout d'abord les besoins que vise à satisfaire cet aspect sémantique dans le processus de filtrage, puis nous décrivons les connaissances et le traitement sémantique.

6.1 Motivation

Après étude des principales techniques actuelles de filtrage et de classification de documents sous forme électronique, nous constatons qu'elles sont basées d'une façon directe ou indirecte sur les techniques des méthodes traditionnelles de recherche d'information. Elles se basent sur l'occurrence d'un ensemble de mots clés pour identifier ou reconnaître les documents pertinents : une approche vectorielle. L'avantage de cette approche statistique repose principalement sur sa simplicité, mais elle n'est pas très précise car l'aspect sémantique est négligé. En effet, en général, les concepts définis pour représenter un profil ne sont pas forcément les mêmes que ceux extraits à partir des textes (richesse du langage naturel), ce qui permet de dégrader les performances de filtrage : certains textes pertinents pour l'utilisateur, ne sont pas acceptés par le système de filtrage (diminution du *taux de rappel*).

6.2 Approche pseudo sémantique proposée

Nous partons du principe de base suivant : une bonne analyse du texte représente la meilleure garantie d'un filtrage précis et efficace. L'idée de base est donc d'introduire l'aspect sémantique dans le processus de filtrage, tout en gardant l'aspect automatique et général du traitement. Et ceci, en impliquant d'autres critères (exemple : mots) dans la décision de filtrage, même s'ils n'apparaissent pas explicitement dans le texte à filtrer : le but est d'étendre la représentation de textes afin de pallier la faible précision et éviter au système de filtrage à les rejeter.

Pour cela, nous proposons d'utiliser deux connaissances :

- Un réseau lexical permettant d'améliorer la représentation du texte en prenant en considération les termes qui existent dans le texte et qui n'existent pas dans le profil. Il s'agit de les remplacer par des termes du profil sémantiquement proches. Nous somme partis du principe suivant : « les connaissances disponibles dans un texte donné sont des connaissances explicites mais impliquent des connaissances implicites. En intelligence artificielle, les connaissances doivent être déclarées pour appuyer les traitements inductifs ».

- La cooccurrence des critères de filtrage en prenant en considération les termes qui n'existent pas dans le texte à filtrer, mais qui corrèlent avec certains termes du profil. En effet, certains

textes pertinents sont constitués de termes statistiquement insignifiants ce qui oblige le système de filtrage à les rejeter.

L'efficacité et la fiabilité du traitement repose entièrement sur l'apprentissage (automatique et assisté). En effet, ces connaissances sont construites initialement sur la base d'un corpus de textes qui seront enrichies et adaptées aux différents utilisateurs progressivement au fur et à mesure de l'utilisation du modèle par ces derniers.

6.3 Le réseau lexical

Un réseau lexical (ou thesaurus) est une collection de termes (vocabulaire) sélectionnés et regroupés suivant les liens qui peuvent être : synonyme, équivalent, plus général, plus étroit, etc.
La première idée qui nous est venue à l'esprit était l'utilisation d'une ressource sémantique externe: un réseau lexical existant, par exemple Wordnet. Certains travaux exploitent cette connaissance lexicale, mais l'intérêt du recourt à une telle ressource générale reste encore à démontrer, certains auteurs tendant même à invalider cette approche [VOO 94] [VOO 98]. L'objection principale opposée à cette démarche étant qu'elle fait l'hypothèse qu'une ressource lexicale non spécifique à un domaine est valable hors domaine. En effet, de nombreux travaux [BOU 97] ont montré que la définition des relations de proximité sémantique ne peut pas être menée hors contexte mais doit au contraire s'appuyer sur les caractéristiques du corpus de travail. Une solution à ce problème, c'est-à-dire l'utilisation d'une ressource générale dans un domaine particulier, est de spécialiser cette ressource à partir du corpus.
Nous avons remarqué qu'il est impossible de trouver un thesaurus universel. Plus le thesaurus est volumineux, plus il consomme de temps de recherche. De plus, il dépend de la langue utilisée. Par exemple, des ressources telles que Wordnet ou Eurowordnet consignant les contextes d'usage sont difficilement exploitables car il n'est pas rare de trouver plusieurs dizaines de sens affectés à un terme donné. Comment choisir tel ou tel sens? Certains prennent en compte la notion de distance sémantique qui est souvent le nombre de noeuds parcourus entre un terme et un autre. Il faut nécessairement pondérer les relations entre noeuds et fixer un seuil de distance, etc.
De plus, vu la richesse du langage naturel qui est liée aux multiples sens des mots, aux ambiguïtés, il est très difficile, voire même impossible, de développer un modèle sémantique qui pourrait le recouvrir entièrement [BON 84].
Dans notre cas, l'existence d'une telle ressource faisant défaut, nous nous sommes penché sur l'étude des principes sémantiques de la méthode *LSI* (Latent Semantic Indexing) dont le fondement est : « Pour toute collection de documents, la correspondance entre la similarité sémantique, et la similarité dans l'usage des termes, est suffisamment forte pour qu'on puisse extraire des informations concernant les structures sémantiques » [DEE 90] (chapitre 2)**.**
LSI est une méthode de recherche documentaire très intéressante par son aspect sémantique, car elle ne voit pas les documents comme un ensemble de termes mais comme des concepts sémantiques. Elle a prouvé une meilleure performance. Elle permet de sélectionner des documents même s'ils ne partagent pas de mots communs avec la requête de l'utilisateur. Cette méthode nécessite néanmoins (1) la disponibilité d'un corpus volumineux (une grande quantité de textes) pour construire la matrice termes-documents et (2) l'opération de mise à jour de l'espace des concepts est coûteuse en termes de calcul. En effet, elle nécessite un temps d'exécution important pour la méthode *SVD* pour donner un résultat assez satisfaisant.

Par conséquent, pour économiser et ne pas dégrader les performances du système, cette opération pourrait s'accomplir régulièrement pendant des périodes creuses.
En définitif, le point fort de cette méthode est sa capacité de détecter les structures sémantiques contenues dans les documents : aspect que nous s'efforcerons de greffer sur le processus de filtrage pour le doter d'une analyse pseudo-sémantique. En s'inspirant de ce principe, nous choisissons une approche plus simple qui consiste à extraire, à partir de corpus, des structures sémantiques qui regroupent les termes contribuant à la modélisation de la même idée (exemple : avion, aviation, pilote, hélicoptère, etc.), c'est à dire construire un réseau lexical implicite propre à l'utilisateur. De plus, à l'aide d'un module d'apprentissage, améliorer progressivement le réseau, au fur et à mesure de l'utilisation du système de filtrage.

6.3.1 Construction du réseau

Notre processus de construction et de mise à jour du réseau lexical s'inspire fortement de l'idée de base de *LSI*, qui utilise un calcul de *SVD* pour détecter et extraire les structures sémantiques à partir d'un ensemble de documents. Mais nous nous contentons d'une idée plus simple que le calcul de la *SVD*, et qui consiste a :

- Constituer un corpus de textes pour le profil considéré.

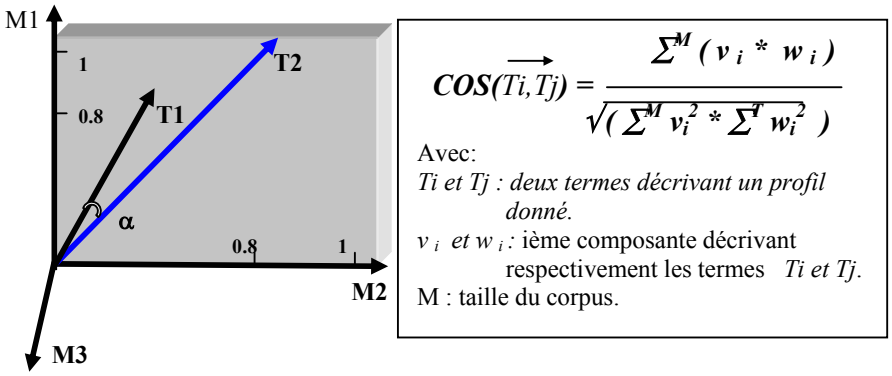
- Construire les vecteurs textes dans l'espace des termes.

- Inverser ces vecteurs pour construire d'autres vecteurs qui représentent l'ensemble de termes dans un espace, dont les axes sont les textes (vecteurs termes).

- Calculer la similarités des termes deux à deux. En effet, nous choisissons que les termes n'appartenant pas au vocabulaire de base. Ensuite, nous calculons la similarité entre ces nouveaux termes et ceux qui existent dans le vocabulaire.

- Regrouper ensemble les termes les plus proches.

Le calcul de la similarité entre termes se mesure à l'aide de la formule *Cosine* qui calcule le cosinus de l'angle entre leurs vecteurs respectifs (figure IV.8).



$$\cos(\overrightarrow{Ti,Tj}) = \frac{\sum^{M} (v_i * w_i)}{\sqrt{(\sum^{M} v_i^2 * \sum^{T} w_i^2)}}$$

Avec:
Ti et Tj : deux termes décrivant un profil donné.
v_i *et* w_i *:* ième composante décrivant respectivement les termes *Ti et Tj*.
M : taille du corpus.

Figure IV.8 : Représentation vectorielle & mesure *Cosine*

Plus le cosinus de l'angle entre les deux vecteurs est proche de 1, plus les vecteurs sont proches ce qui implique une plus grande ressemblance entre les deux termes. L'algorithme est donné comme suit (algorithme IV.1) :

```
Vecteurs : texte ( text ) et profil ( pro )
Variables : i , j , k , l :  initialisées à 0 ;
Pour i allant de 0 à la taille (pro)
    Faire
        j=j + pro[i] . poids * text[i] . poids;
        k=k+ pro[i] . poids *  pro[i] . poids;
        l=l + text[i] . poids *  doc[i] . poids;
    Fait;
Cos=Racine carrée ( j  / racine carrée (k * l ) )
Fin.
```

Algorithme IV.1 : Algorithme de calcul de cosinus

Figure IV.9 donne un aperçu du réseau généré automatiquement par le système (cas du domaine *SPAM*):

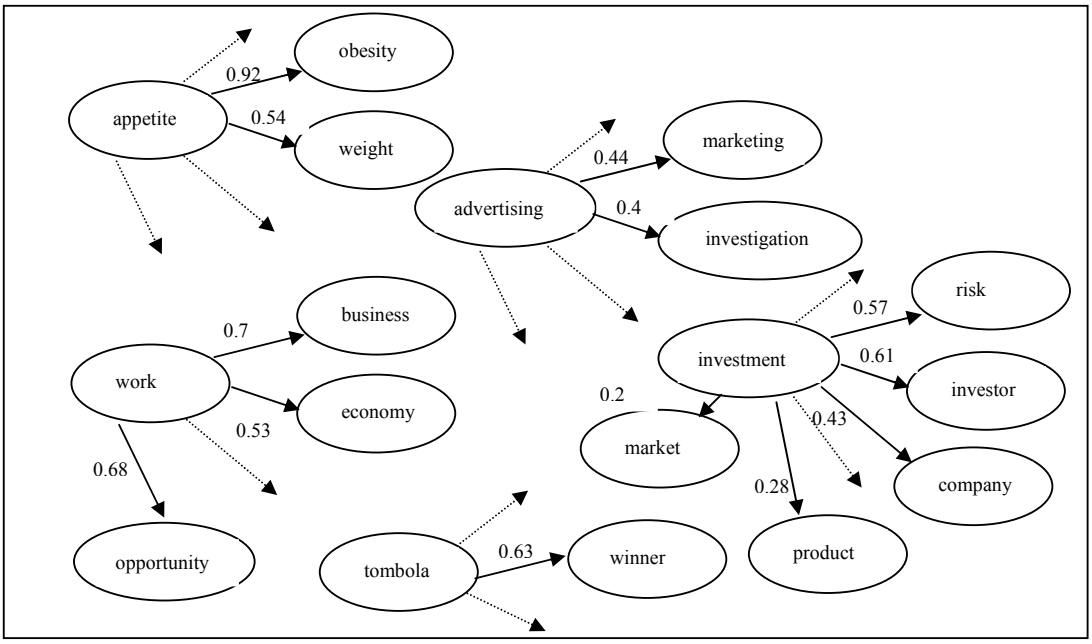


Figure IV.9 : Aperçu du réseau lexical

L'exploration du réseau lexical permet au système d'améliorer la représentation d'un texte contenant, par exemple, le terme *« obesity »* (terme non existant dans le vocabulaire lexical du système) en le remplaçant par sa classe, c'est-à-dire le terme « *appetite* ».

6.3.2 Apprentissage

L'apprentissage est un aspect intrinsèque de l'intelligence et une nécessité pour s'adapter à un environnement évolutif. Il permet d'améliorer l'efficacité et les performances d'un système.
Le système dispose d'un **apprentissage assisté** appelé ***feed-back*** où l'utilisateur est invité à donner son avis sur le comportement du système, ce qui lui permet d'approcher la pertinence de l'utilisateur et de s'adapter ainsi à ses besoins. L'apprentissage agit sur le réseau lexical, qui consiste à modifier les liens du réseau (terme-terme). L'utilisateur peut aussi ajouter et supprimer de mots à sa demande. Un agent est chargé de surveiller le comportement de toutes les structures sémantiques qui contribuent à la conception du nouveau vecteur texte dans l'espace du profil lors de la session de filtrage. Il sauvegarde une trace de chaque substitution réalisée. Ensuite, après validation des résultats de filtrage par l'utilisateur, toutes les structures sémantiques ayant contribué à la bonne décision seront renforcées (augmenter le degré de ressemblance) et celles ayant contribué à l'échec du système seront pénalisées (diminuer le degré de ressemblance). Un tel traitement ne donne de bons résultats que s'il se fait régulièrement. En effet, après une certaine période d'utilisation ou d'apprentissage, le système de filtrage sera doté d'un réseau lexical efficace et plus spécifique à l'utilisateur.
De même, nous pouvons rendre le réseau intelligent capable d'évoluer et de supprimer les structures fausses en lui appliquant les algorithmes génétiques.

6.4 Cooccurrence de critères

La cooccurrence des critères de filtrage permet l'introduction d'un deuxième type sémantique dans le processus de filtrage. Il s'agit d'impliquer des critères qui n'apparaissent pas explicitement dans le texte à filtrer, mais corrèlent avec certains critères de ce texte. Cette connaissance sémantique est représentée par des liens entre critères dans la couche cachée du réseau de neurones. En effet, les nœuds de la couche cachée *C,* représente l'ensemble des critères existants dans la base, sont reliés entre eux par des liens de cooccurrence w_{ij} (représente la cooccurrence de deux critères t_i et t_j) (figure IV.10).

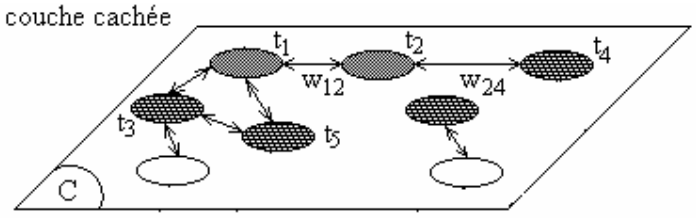


Figure IV.10: Représentation de la cooccurrence.

6.5 Processus de filtrage sémantique

Le filtrage sémantique consiste à compléter la représentation d'un texte obtenu après analyse. En effet, après l'étape de pré-traitement (identification de la langue, etc.), le texte est donné à un analyseur, qui consiste à le transformer par rapport au vocabulaire (caractéristiques linguistiques) déterminé à partir de la base d'apprentissage. La sortie de l'analyse passe ensuite par le réseau lexical qui permet de modifier la représentation initiale en remplaçant les mots inconnus (de l'analyseur) par d'autres mots plus proches, munis de poids représentant le degré de ressemblance sémantique. Ce qui constitue, en quelque sorte, un premier filtrage de type sémantique. Cette nouvelle représentation constitue l'entrée du réseau de neurones. Elle

est crée donc dynamiquement à chaque filtrage d'un nouveau texte. Ce vecteur sera propagé à travers les différentes couches du réseau pour donner, en sortie, le profil qui correspond le mieux au texte analysé. En effet, chaque neurone fait une somme pondérée des signaux qui lui parviennent, puis s'active suivant la valeur de cette somme pondérée. Si cette somme dépasse un seuil, le neurone est activé et transmet une réponse aux neurones auxquels il est connecté. Sinon il n'est pas activé et ne transmet rien (valeur nulle). A ce niveau du processus de filtrage, nous distinguons deux types de filtrage : *filtrage sans propagation* ou *vertical* et *filtrage avec propagation* ou *horizontal*. Dans le *filtrage vertical*, le vecteur texte représentant le signal d'entrée sera propagé à travers les différentes couches du réseau pour donner, en sortie, son profil représentatif. Tout au long du processus, il n'y a que les critères, seulement représentant le texte en entrée, qui sont considérés. Par contre, dans le *filtrage horizontal* qui représente le deuxième filtrage de type sémantique, la propagation se fait grâce à l'extension des signaux d'entrée à travers les liens de cooccurrence entre les critères. Dans ce mode de filtrage, après qu'un nœud évalue son activation, il va la transmettre aux proches voisins, à travers les liens de cooccurrence, ce qui implique la reformulation du vecteur d'entrée représentant le texte (figure IV.11).

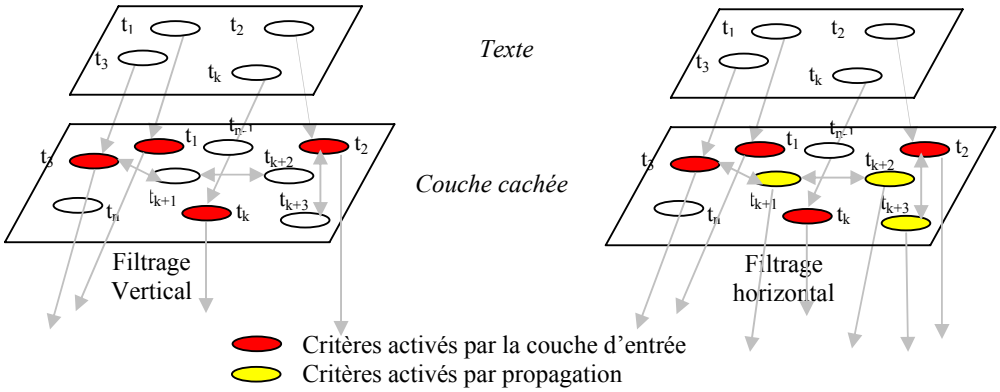


Figure IV.11: Schéma d'activation de critères

7 Processus de filtrage

Ce module est constitué de deux parties : une partie classification et une partie filtrage. La classification permet d'affecter à un texte une catégorie constituant en quelque sorte un pré-filtrage (figure IV.12). En effet, une idée pour classer les textes, est de créer des espaces de textes (un espace pour chaque type). Et chaque nouveau texte se trouvant être proche aux textes de l'un des espaces définis, est alors considéré comme pertinent pour cet espace. Donc, afin de mieux classer un texte, le processus de filtrage utilise un modèle de connaissances qui représente et modélise une typologie de textes, dont la connaissance est construite initialement sur la base d'analyse de traits linguistiques associés à chaque espace de textes.

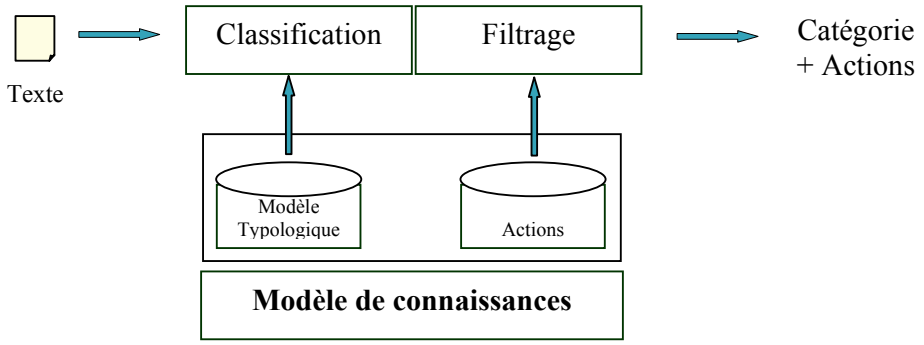


Figure IV.12 : Processus général de filtrage

Après l'étape de pré-traitement, le texte subit une analyse linguistique, indépendamment des intérêts de l'utilisateur. La sortie de l'analyse passe par un arbre de classification qui affecte automatiquement une catégorie au texte traité. Ensuite, le texte passe par un module de filtrage qui permet, selon les spécificités de l'utilisateur, de prendre, en conséquence, des actions de filtrage, tel que supprimer, sauvegarder, signaler, etc. Ce module est étroitement lié et adapté aux souhaits de l'utilisateur.

8 GIFI, un assistant à la génération d'interface de filtrage

Cette partie est consacrée à la description de GIFI, un générateur d'interfaces de filtrage d'information. Il est destiné à faciliter la tâche des utilisateurs développeurs dans l'élaboration de systèmes de filtrage. Il repose sur une conception modulaire, lui permettant éventuellement de s'adapter à toute extension et modification. En effet, dans cette conception modulaire, le but recherché est d'avoir une architecture ouverte permettant l'ajout de composants et d'offrir ainsi, à l'utilisateur, la possibilité de choisir, à chaque étape du processus de génération, les outils à utiliser. Cette plate forme (boite à outils) constitue une implémentation d'une approche hybride du filtrage d'information. Elle repose sur le principe d'une analyse partielle utilisant un ensemble de connaissances, où le repérage de propriétés linguistiques permet, d'une part, d'améliorer la représentation de textes, et d'autre part un filtrage meilleur en qualité. Nous exposons tout d'abord les besoins que vise à satisfaire cet outil, puis son architecture globale et son fonctionnement général en détaillant les services offerts ainsi que la chaîne de traitement, de l'acquisition des textes (corpus de l'application) aux ressources générées (vocabulaire lexical, propriétés linguistiques, modèle de filtrage).

8.1 Motivation

Le cadre dans lequel nous nous situons, une approche hybride (quantitative/linguistique) du filtrage d'information, présuppose un recours aux corpus pour extraire des connaissances linguistiques et estimer les paramètres du système de filtrage. Ces connaissances sont des indices linguistiques, non restreints aux mots clés (termes). Cette extraction ne peut être menée à bien que par l'étude des observables linguistiques (observer les régularités dans la distribution des formes). Dans le cadre distributionnel, le travail sur corpus demande un investissement certain de la part de l'utilisateur (pour générer son profil, dans notre cas). Par ailleurs, toute étude à forte composante manuelle, telle que l'analyse de corpus, est une tâche fastidieuse et sujette à des variations de qualité (inefficacité du profil), liée à l'opérateur

humain (néophyte, fatigue, etc.). L'outil vise donc à proposer un ensemble de traitements systématiques aidant l'utilisateur à concevoir son système de filtrage.

8.2 Acquisition de textes

Le service d'acquisition ou la collecte de textes constitue la source de l'outil de génération. L'entrée du système est constituée d'un ensemble de textes bruts, locales et/ou distribuées (disque local, serveurs news, serveurs Web, etc.). L'utilisateur introduit ses intérêts dans le système sous formes différentes : sous forme de liste de mots clés, sous forme de textes ou sous forme d'urls (ex : adresse d'un serveur). Ce module se chargera de collecter et de construire le corpus de l'application.

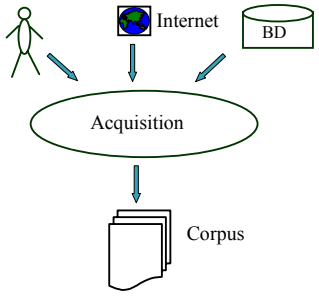


Figure IV.13 : Processus d'acquisition

8.3 Sélection des critères de filtrage

La structuration des intérêts de l'utilisateur, couvert par un ensemble de textes ou documents, est obtenue par une analyse de contenu faisant apparaître en sortie un vocabulaire lexical de base augmenté d'un ensemble de caractéristiques linguistiques. Notre approche diffère des techniques classiques de génération de profils, elle tente d'améliorer les résultats de filtrage par une adaptation de connaissances linguistiques. Nous tentons de justifier l'utilisation de critères linguistiques réduisant le biais d'une statistique descriptive des occurrences de mots clés. Les critères de filtrage sont donc générés automatiquement à partir du corpus de l'application ou du domaine traité.

8.3.1 Vocabulaire Lexical

Le vocabulaire représente l'ensemble des mots d'intérêt caractérisant les textes traités. Chaque texte subit un traitement morphologique permettant de calculer le lemme de chaque mot, et un filtrage permettant d'éliminer les mots communs.

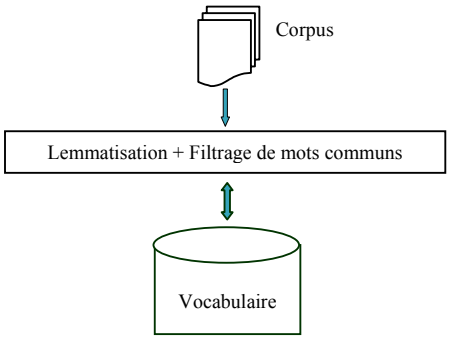


Figure IV.14 : Processus de génération du vocabulaire

Tous les mots, obtenus après traitement de tous les textes du corpus, constituent la première version du vocabulaire. L'algorithme de génération est donné comme suit (algorithme IV.2) :

Pour chaque texte du corpus
Faire
pour chaque mot du texte
Faire
Si le mot appartient déjà au vocabulaire,
l'occurrence de ce mot est incrémentée,
Si le mot n'appartient pas au vocabulaire,
le mot est ajouté au vocabulaire.
Fait
Fait

Algorithme IV.2 : Algorithme de génération du vocabulaire

Après traitement complet des textes du corpus, une mesure de pertinence (information mutuelle) peut être calculée pour chaque mot permettant de réduire éventuellement la taille du vocabulaire (chapitre 3).

8.3.2 Caractéristiques supplémentaires

En plus du vocabulaire lexical, le système génère un ensemble de caractéristiques supplémentaires caractérisant les textes traités. Le même procédé, que pour le vocabulaire lexical, est appliqué pour générer cet ensemble. Le système ne génère que les propriétés dont le poids est supérieur à un certain seuil.

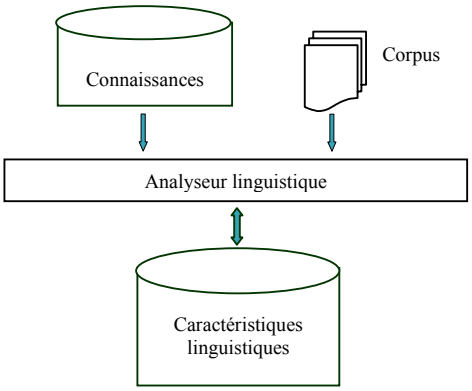


Figure IV.15 : Processus de génération de caractéristiques

8.4 Modèle de filtrage

La construction du modèle est basée sur l'utilisation de techniques d'apprentissage automatique. Il s'agit, à partir d'un ensemble d'exemples observés, d'induire une procédure ou une règle (ex. classification).

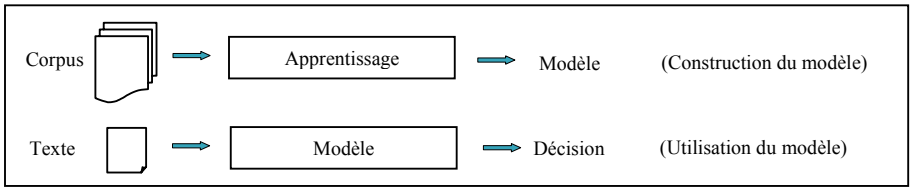


Figure IV.16 : Processus de construction et d'utilisation d'un modèle de filtrage

La procédure générée devra fournir correctement les réponses appropriées aux exemples de l'échantillon mais surtout avoir un bon pouvoir prédicatif pour répondre correctement aux nouvelles descriptions (généralisation). Nous avons implémenté les réseaux de neurones comme technique de base (par défaut). L'architecture du système est ouverte permettant ainsi l'ajout de nouveaux composants et d'offrir ainsi, au développeur, la possibilité de choisir, une méthode d'apprentissage pour générer son interface de filtrage.

8.4.1 Modèle de base adopté

Une structuration du modèle (utilisateur, domaine d'intérêt) est d'utiliser des techniques de classification automatique (clustering). L'interprétation des classes, générées automatiquement, reste difficile, due aux multiples points de vue qu'un utilisateur peut se faire des associations entre caractéristiques (exemple : termes) générées. C'est-à-dire, il est difficile d'attribuer une signification aux classes constituées. D'autres techniques ou une expertise sont nécessaires pour donner une signification aux classes construites. Généralement, ces techniques sont assistées par un thésaurus du domaine.

Pour des soucis pratiques, la démarche que nous avons choisi de suivre est de laisser, en premier lieu, à l'utilisateur le choix de proposer une catégorisation personnelle (un ensemble de catégories prédéterminées). Ce n'est que dans un deuxième temps que l'on effectue une classification (clustering) les classes ne sont pas connues à l'avance.
Le modèle de base adopté pour notre système est un réseau de neurones non récurrents (absence de boucles) à trois couches (figure IV.17). Une couche en entrée qui reçoit les entrées du réseau. Une couche cachée représentant l'ensemble des connaissances (profils). Une couche de sortie qui représente les différentes classes ou types de profils (*profil 1, .., profil N*).

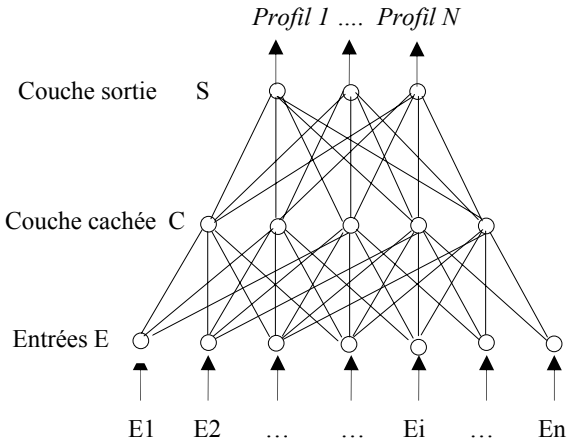


Figure IV.17 : Architecture d'un réseau à trois couches

L'avantage des réseaux de neurones est qu'ils sont bien adaptés aux applications qui font intervenir des données bruitées.

8.4.2 Apprentissage

Nous appelons apprentissage, la procédure qui consiste à estimer les paramètres d'un système, afin que celui-ci remplisse au mieux la tâche qui lui est affectée.
Dans un réseau de neurones, la connaissance est codée par la valeur des poids des différentes connexions. Ce codage est estimé par apprentissage.
Le réseau est entraîné sur une base d'apprentissage, dans le but de correctement filtrer un nouveau texte, par l'algorithme de propagation arrière ou rétro-propagation qui consiste à corriger les poids des connexions en fonction des erreurs commises. La correction se fait de la couche de sortie à la couche d'entrée. L'apprentissage utilisé est dit supervisé, c'est à dire que nous testons le réseau dans des situations connues et nous cherchons à obtenir la sortie voulue. Nous effectuons alors la modification des poids pour retrouver cette sortie imposée.

a)- Construction automatique de la base d'apprentissage

La base d'apprentissage est constituée de l'ensemble des textes du corpus, représentés par rapport au vocabulaire (caractéristiques linguistiques) déterminé à partir de l'étude du corpus.

La construction de la base d'apprentissage se fait au même temps que la sélection des critères de filtrage. En effet, les caractéristiques n'appartenant pas au vocabulaire sont ignorées. Cette représentation du corpus constitue l'entrée du processus d'apprentissage.

b)- Schéma d'algorithme d'apprentissage

L'algorithme d'apprentissage est décrit comme suit :

— Etiqueter manuellement chaque texte du corpus (le texte correspond au *profil 1, .., ou profil N*).

— Faire passer le corpus par les différents modules d'analyse pour extraire les différentes propriétés linguistiques et avoir la représentation vectorielle associée de chaque texte.

— Initialiser les paramètres du réseau : les poids, les seuils, *pas d'apprentissage* (learn rate), le nombre d'exemples (epoch length) et le nombre d'itérations.

Les poids doivent être initialisés à des petites valeurs aléatoires entre *-0.5* et *+0.5* [DRE 02]. Au départ les poids des connexions entre neurones des différentes couches sont définis par défaut à *0.5*.

Le choix du *pas d'apprentissage* est très important, car il contrôle la vitesse de modification des poids : s'il est trop petit, la convergence du réseau risque d'être très lente (entraine des modifications minimes des poids); s'il est trop grand, il y a risque d'oscillation (modifie fortement les poids à chaque mise à jour). Il est fréquent de choisir une valeur initiale qui diminue avec le nombre d'itérations. Généralement, le *pas* doit être compris entre *0.05* et *0.25* [DRE 02]. La valeur initiale est choisie "assez grande" pour une convergence rapide, puis diminue pour éviter les oscillations et converger vers un minimum. Nous avons initialisé le *pas* à *0.1*.

Le paramètre *nombre d'exemples* est nécessaire pour lancer le calcul d'erreur et effectuer la mise à jour des poids. Pour la valeur 1, il y a mise à jour des poids après chaque exemple. Dans ce cas, la convergence est rapide, mais on peut tomber dans un minimum local (on trouve un minimum de l'erreur mais ce n'est pas l'erreur minimale que l'on peut espérer). Pour la valeur maximale (le cardinal de l'ensemble d'apprentissage), on passe l'échantillon complet, on calcule l'erreur, puis on met à jour les poids. La convergence est plus lente. Nous l'avons initialisé à 1.

— Lancer l'apprentissage qui consiste à :

- Calculer la sortie du réseau pour chaque texte

$$S(T_j) = f(E(T_j))$$

$$E(T_j) = \sum_j S(C_j) * P_{js}$$

$$S(C_j) = f(E(C_j))$$

$$E(C_j) = \sum_i S(t_i) * P_{ij}$$

$$S(t_i) = q_i = \begin{cases} 1 & \text{si ti est trouvé dans le texte} \\ 0 & \text{sinon} \end{cases}$$

Où:

S(T_j) : la valeur du neurone de sortie d'indice *j*.

E(T_j) : les valeurs des entrées du neurone de sortie d'indice *j*.

S(C_j) : la sortie du neurone cachée d'indice *j*.

E(C_j) : les entrées du neurone cachée d'indice *j*.

S(t_i) : la sortie du neurone en entrée d'indice *i*.

P_{ij} : valeur du poids de la connexion du neurone d'indice *i* de la couche d'entrée vers le neurone d'indice *j* de la couche cachée.

P_{js} : valeur du poids de la connexion du neurone d'indice *j* de la couche cachée vers le neurone d'indice *s* de la couche de sortie.

f : fonction d'activation *sigmoïde* [DAV 83], choisie pour toutes les couches, définit par $f(x)=\frac{1}{1+e^{-x}}$

- Comparer et calculer l'erreur :

$$DS = S*(1-S)*(D^s - S)$$

$$DC = C_j *(1- C_j)*(P_{js} *DS)$$

Où :
DS : erreur du réseau pour la couche de sortie.
DC : erreur du réseau pour la couche cachée.
D^s : sortie désirée

- Mettre à jour les paramètres du réseau par rétro-propagation (de la couche sortie vers la couche entrée) : ajuster les poids.

$$P_{ij}(t+1) = P_{ij}(t) + r*DC*S(t_i)$$

$$P_{js}(t+1) = P_{js}(t) + r*DS*S(C_j)$$

Où :
r : le taux d'apprentissage.
t : le numéro du cycle.

— Le test d'arrêt: la convergence de l'algorithme de *rétro-propagation* est assurée soit par un test consistant à fixer le nombre d'itérations ou bien arrêter l'apprentissage dès que l'erreur devient inférieure à un certain seuil (tolérance ou l'erreur souhaitée). Si ce seuil est très proche de *0*, il y a un grand risque de *sur apprentissage*; au lieu de produire une bonne généralisation, le réseau se concentre sur les particularités des exemples d'apprentissage. En pratique, le test d'arrêt est lié aux mesures des performances du réseau. Pour mesurer les performances du réseau, il convient de constituer, outre l'ensemble d'apprentissage utilisé pour déterminer les poids, un ensemble de test, constitué d'exemples différents de ceux de l 'ensemble d'apprentissage à partir duquel nous estimons les performances du réseau après un apprentissage. Nous alternons des étapes d'apprentissage sur l'ensemble d'apprentissage et de mesures des performances sur l'ensemble de test jusqu'à atteindre des résultats satisfaisants. En effet, l'apprentissage consiste donc à trouver l'ensemble des paramètres *w* du réseau qui rendent la fonction de coût des moindres carrés *J(w)* minimum, définit par la formule suivante [DRE 02]:

$$\mathbf{J(w)}=\frac{1}{2}\sum_{k=1}^{N_a}[Y_p(x_k) - g(x_k, w)]^2$$

Où:
N_a : nombre d'exemples de l'échantillon apprentissage.
x_k : vecteur des valeurs des variables pour l'exemple k.

w : vecteur des poids du réseau.
$g(x_k, w)$ *:* valeur calculée par le réseau.
$Y_p(x_k)$ *:* valeur de la mesure correspondante.

Il s'agit d'une technique itérative, qui modifie les paramètres *w* du réseau jusqu'à ce que *J(w)* soit minimum.

Ensuite, mesurer l'indice de performance qui est l'erreur quadratique moyenne commise sur l'ensemble de test, désigné par *EQMT* [DRÉ 02]:

$EQMT = \sqrt{\frac{1}{N_t}\sum_{k=1}^{N_t}[Y_p(x_k) - g(x_k, w)]^2}$ où N_t représente le nombre d'exemples de l'ensemble *test*,

comparée à l'erreur quadratique moyenne commise sur l'ensemble d'apprentissage *EQMA* :

$EQMA = \sqrt{\frac{1}{Na}\sum_{k=1}^{Na}[Y_p(x_k) - g(x_k, w)]^2}$ où Na représente le nombre d'exemples de l'ensemble *apprentissage*.

8.5 Description de l'interface graphique

L'outil se présente sous forme d'une interface graphique : une barre de menus (figure IV.18) propose différentes fonctionnalités de traitement : *corpus, analyse linguistique, représentation des données, apprentissage, thésaurus* et *aide*.

Figure IV.18 : Interface graphique de GIFI

Le menu *corpus* permet l'acquisition des données, l'extraction et la visualisation du vocabulaire lexical de base (figure IV.19).

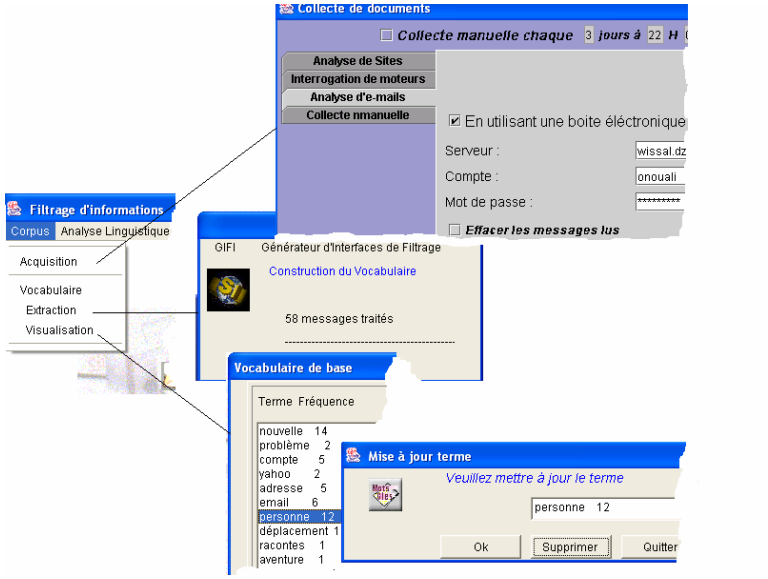


Figure IV.19 : Interface d'acquisition des données et d'extraction du vocabulaire

L'interface *acquisition des données* permet de collecter des documents et de construire le corpus de l'application, qui constitue la source de l'outil de génération, GIFI. Cette interface de collecte offre les fonctionalités suivantes :

- Analyser les documents appartenant à certains sites : l'utilisateur définit un ensemble d'adresses URL et spécifie une liste de mots clés.

- Interroger des engins de recherche : l'utilisateur choisit une liste de moteurs et soumet un ensemble de mots clés sous forme de requête.

- Collecter des messages : l'utilisateur définit le serveur et la boite de courriers.

- Collecter des documents en spécifiant directement des adresses URL correspondantes.

L'*extraction du vocabulaire* permet de traiter l'ensemble du corpus (construit à l'aide du module précédent) et de générer une liste de mots avec leur fréquence d'apparition. Enfin, la *visualisation* permet l'affichage du vocabulaire.

Le menu *analyse linguistique* permet de définir la taille du corpus, l'analyse et l'extraction des propriétés linguistique (figure IV.20).

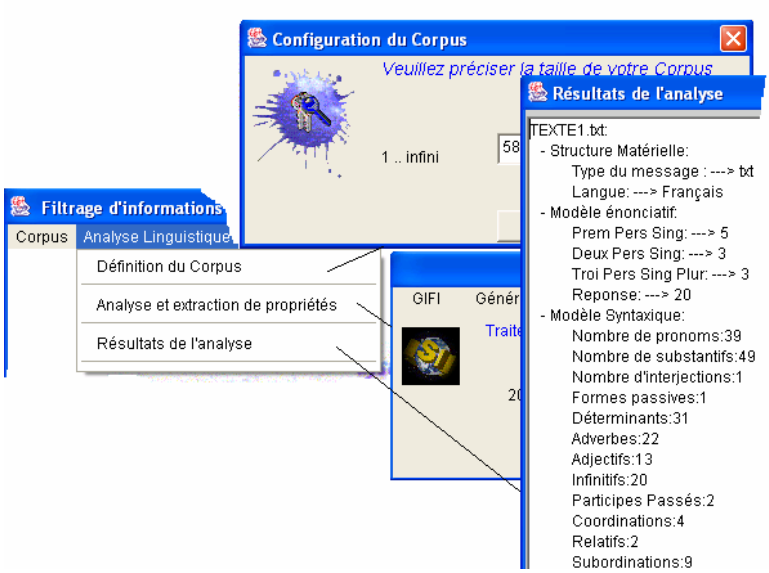


Figure IV.20 : Interface d'analyse linguistique

Cette interface permet de faire passer chaque texte du corpus par les différents modèles linguistiques réduits et d'extraire pour chaque niveau d'analyse un ensemble de caractéristiques. En sortie, les textes sont représentés vectoriellement par un ensemble d'entités lexicales et d'une suite de propriétés linguistiques (architecturales, énonciatives, structurelles et syntaxiques).
L'utilisateur a la possibilité de visualiser les résultats de l'analyse pour chaque texte traité. La Figure IV.21 présente un exemple de résultats de cette analyse.

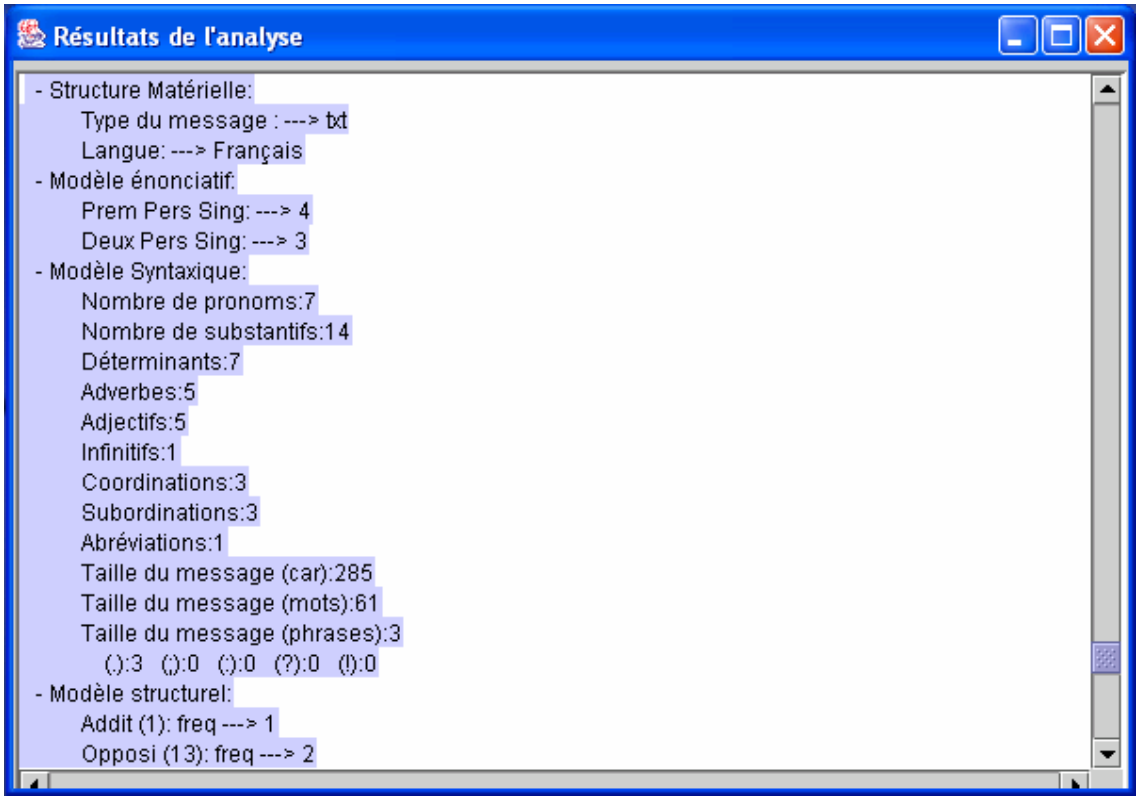


Figure IV.21 : Résultat de l'analyse

Le menu *représentation des données* permet la réduction, le codage, la normalisation et l'affichage des données (figure IV.22).

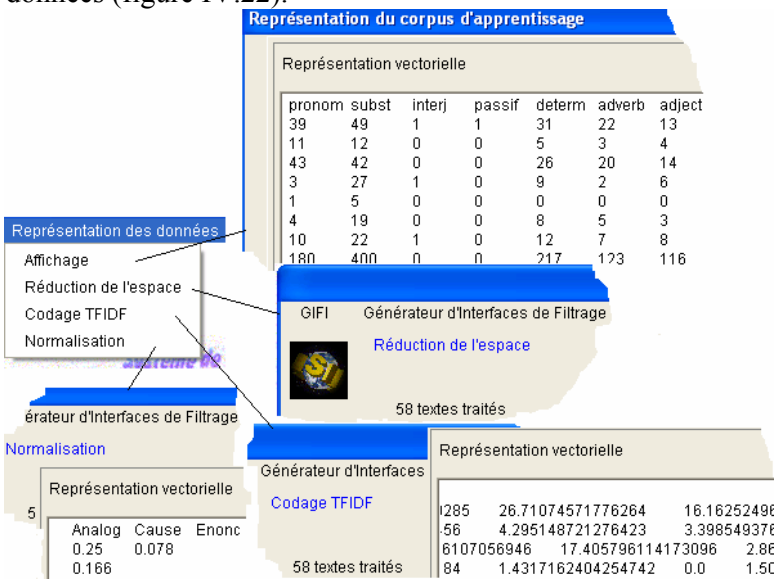


Figure IV.22 : Interface pour la représentation des données

La *réduction* permet d'éliminer toutes les propriétés dont la valeur est nulle, c'est-à-dire les propriétés qui n'apparaissent dans aucun texte du corpus. Les figures IV.23 et IV.24 présentent respectivement un aperçu du corpus avant et après la réduction.

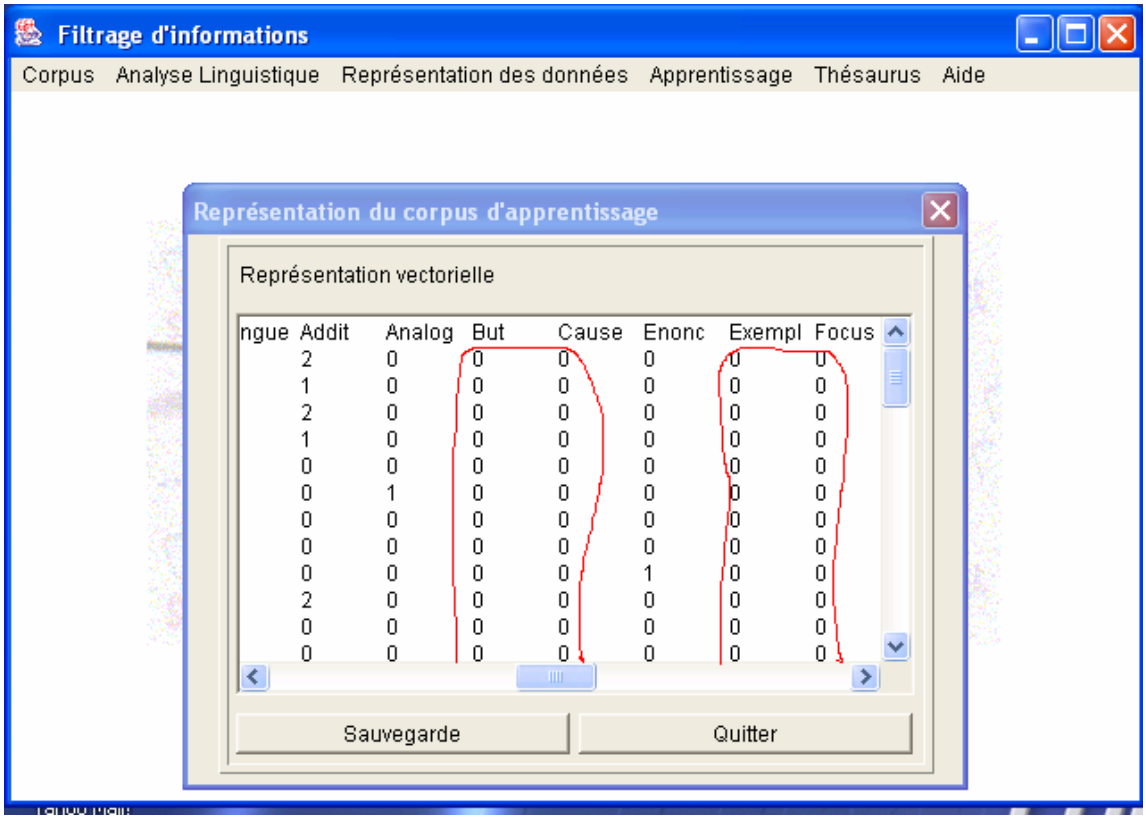


Figure IV.23 : Aperçu avant réduction

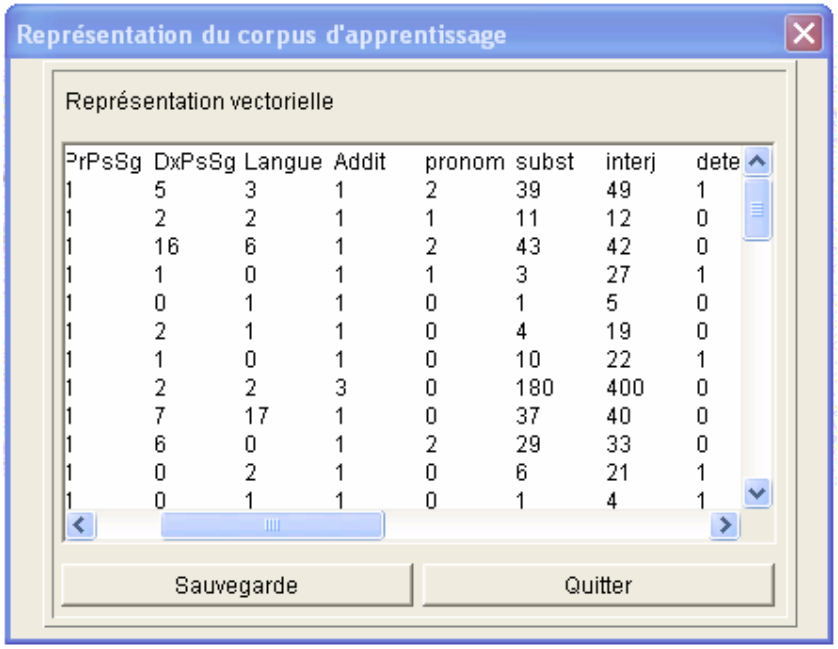


Figure IV.24 : Aperçu après réduction (~34%)

Le *codage* permet d'attribuer un poids à chaque propriété de chaque texte du corpus. La figure IV.25 présente un aperçu du codage TFIDF.

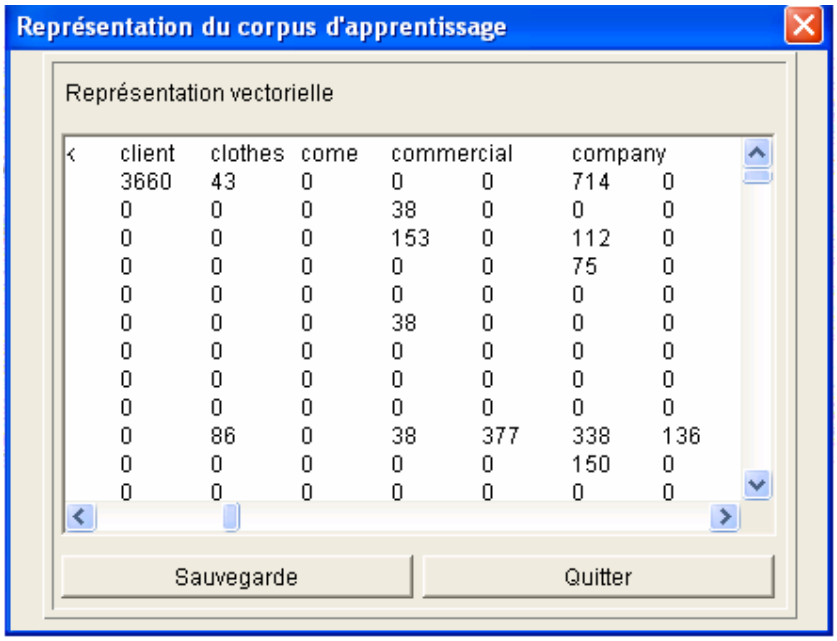


Figure IV.25 : Aperçu après codage

La *normalisation* des valeurs de la pondération permet de donner une chance identique aux textes quelle que soit leur taille. Elle consiste à diviser chaque poids par la valeur maximale de la même caractéristique dans les textes du corpus. La figure IV.26 présente un aperçu du corpus après normalisation.

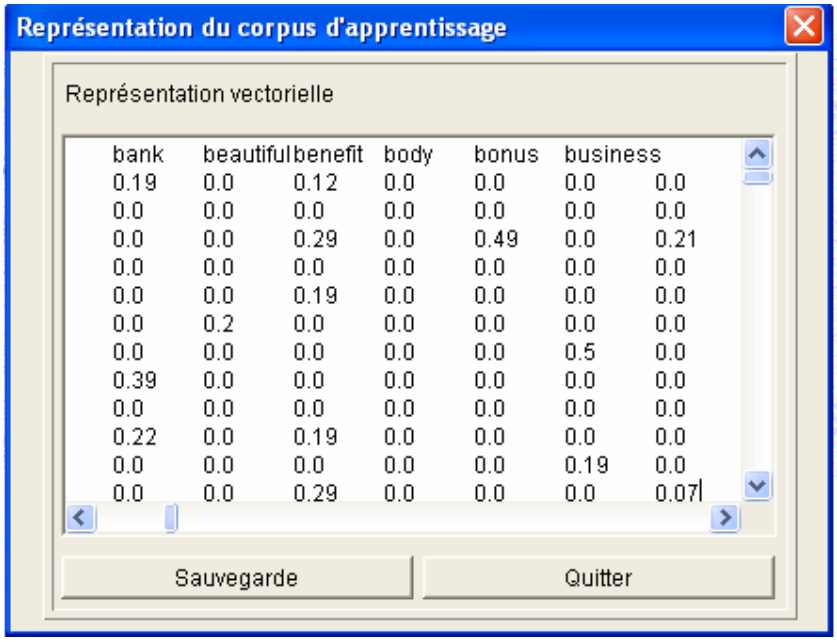


Figure IV.26 : Aperçu après normalisation

Le menu *apprentissage* (figure IV.27) permet l'estimation des paramètres de la fonction d'apprentissage (réseaux de neurones et Rocchio).

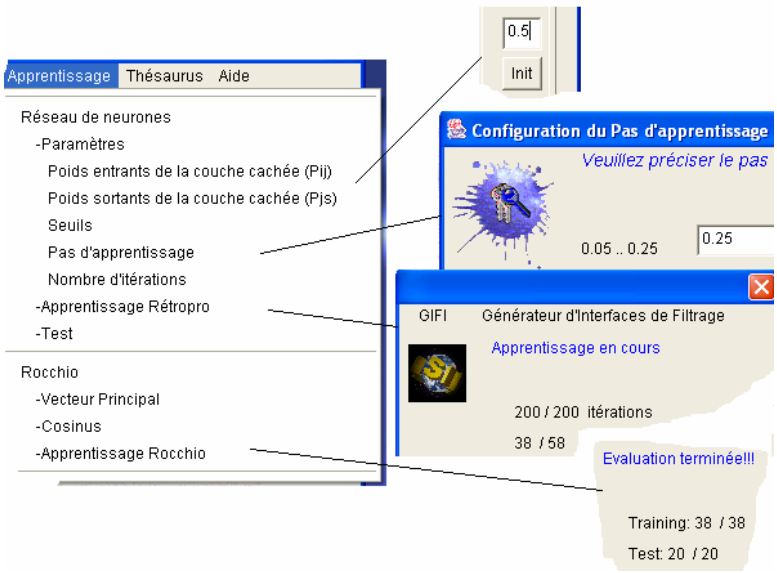


Figure IV.27 : Interface d'apprentissage

Le menu *thésaurus* (figure IV.28) permet un apprentissage pseudo sémantique (construction du thésaurus).

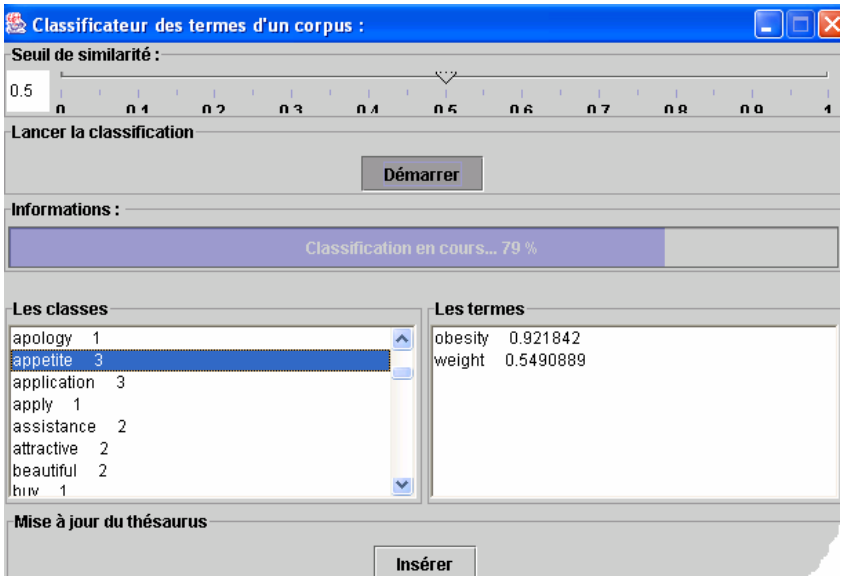


Figure IV.28 : Interface Sémantique

Enfin, le menu *aide* permet d'afficher l'aide de GIFI ainsi que les informations sur l'application et ses auteurs.

9 Conclusion

Nous avons présenté dans cette partie, l'architecture globale de notre système de filtrage. Nous avons décris les principaux modules ainsi que les différentes connaissances et outils utilisés par le système. Il est basé sur une approche qui exploite un maximum d'informations pour améliorer l'efficacité du processus de filtrage. Une approche qui permet, d'une part, de représenter un texte en ajoutant, à la représentation lexicale, d'autres propriétés sous forme d'indices. L'extraction de ces indices est basée sur des modèles linguistiques réduits. Cela nous permet de définir un indice de confiance sur le fait que le texte soit d'un type donné ou non. D'autre part, l'introduction de l'aspect sémantique au processus de filtrage. Il permet de relier des termes même s'il n'existe pas de lien visible entre ces termes. En effet, deux termes sont utilisés dans des contextes similaires, vont avoir des représentations similaires ou proches. L'approche exploite deux connaissances pour aborder l'aspect sémantique : un réseau lexical et la cooccurrence des termes. Le réseau lexical permet d'améliorer la représentation d'un texte. Il est implicite, adaptatif, et propre à chaque utilisateur. En effet, il repose beaucoup sur l'utilisateur qui doit le mettre à jour régulièrement et aider le module d'apprentissage feed-back pour l'affiner. La cooccurrence des termes permet de bien filtrer des textes, même si ils ne partagent pas beaucoup de termes avec le profil de l'utilisateur.
Enfin, nous avons présenté GIFI, un assistant à la génération d'interfaces de filtrage.

Chapitre V

Filtrage, typologie et caractéristiques des messages

Dans cette partie, nous décrivons une application pratique du domaine de filtrage de l'information : le courrier électronique. En effet, les moyens de communication électronique sont en rapide expansion, ils permettent facilement de créer et de diffuser de l'information auprès des utilisateurs. Aujourd'hui, le courrier électronique est le mode de communication le plus populaire. Il est devenu un moyen rapide et économique pour échanger des informations. Cependant, les utilisateurs d'Internet se retrouvent assez vite submergés de quantités astronomiques de messages dont le traitement nécessite un temps considérable. Devant l'importance de ce phénomène, il est donc nécessaire aujourd'hui d'élaborer des outils efficaces capables de traiter et de filtrer l'email.
Nous présentons quelques systèmes commerciaux et prototypes de filtrage de messages. Nous décrivons quelques stratégies et approches de traitement et de filtrage de messages. Ensuite, nous décrivons le corpus de messages émails utilisé, ainsi que la typologie et les caractéristiques associées. Nous terminons par une présentation de notre système de courriers électroniques paramétrable avec plusieurs niveaux de filtrage, en donnant quelques mesures chiffrées de performance de notre approche de filtrage.

1 Messagerie électronique

1.1 Anatomie ou format d'un message électronique

Les messages électroniques, contrairement aux autres types de documents circulant sur Internet sont des entités dites « semi-structurées ». Elles comportent une partie structurée ou *header* et une partie non structurée (texte quasi-brut) ou *body* (figure 1). Les messages sont rédigés dans un style assez simple. Ils suivent tous le même format :

1.1.1 Partie Structurée (Header)

Elle est destinée à l'identification et à l'aiguillage des messages, elle est constituée des champs suivants :

- To : L'adresse électronique du destinataire du message qui se compose de deux parties : le nom de la boite qui est un nom personnel que choisit l'utilisateur et le nom du domaine qui

caractérise l'organisme ou la société qui héberge cette boite aux lettres (onouali@cerist.dz, onouali@yahoo.fr, etc.).

- From : Le ou les rédacteurs du message. Ce champ Indique l'adresse électronique de l'expéditeur du message, qui peut être un news group, un organisme ou tout simplement une personne.

- Subject : La ligne qui décrit le sujet du message.

- Date : La date d'envoie du message.

- Cc (Carbon Copy) : La liste des destinataires en copie.

- Bcc (Blind Carbon Copy) : La liste des destinataires en copie, mais chaque destinataire ne dispose pas des adresses des autres destinataires en copie (adresses masquées).

- Attachements : La liste des pièces jointes au message. Elles peuvent être de n'importe quel format (Txt, Pdf, Jpg, Doc, etc.).

1.1.2 Partie non structurée (texte libre)

Elle ne respecte aucun formalisme, elle est formée de deux zones de texte qui sont :

- Corps du message : C'est le message proprement dit, il contient les informations que veut transmettre l'expéditeur au destinataire sous forme textuelle, en fait les nouvelles normes régissant la messagerie électronique autorisent aujourd'hui un corps du message sous forme *html* en raison de l'étendue de ce format qui est le langage d'échange d'informations sur Internet.

- Signature : C'est un texte libre qui sert à contenir la signature de l'expéditeur ou tout simplement quelques mots pour terminer son courrier.

Exemple : *Nouali Omar,*
Chargé de recherche,
Laboratoire des logiciels de base, CERIST,
Rue des 3 frères Aissiou, Ben Aknoun, Alger, Algerie,
Tél : (00 213) (0) 21 91 62 11 Fax : (00 213) (0) 21 91 21 26
E-mail : onouali@cerist.dz

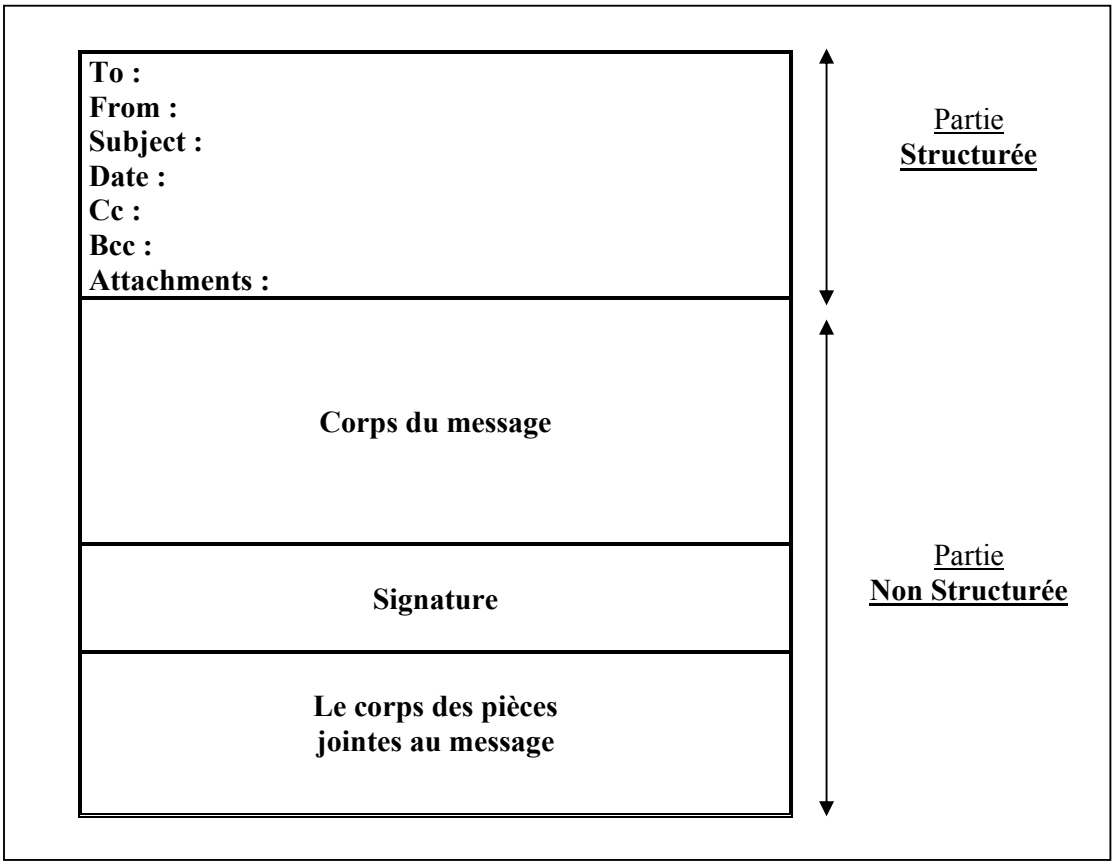


Figure V.1 : Anatomie d'un message électronique

2 Filtrage d'emails

2.1 Définition de filtrage de messages

Actuellement, la notion de filtrage de messages est exprimée par un sens qui découle directement de la sémantique de deux verbes qui sont *sélectionner* et *trier*. Filtrer des messages consiste donc à sélectionner tous les messages intéressants non lus, triés par ordre d'importance [KIL 97b]. Il s'agit donc de bien définir le mot *intéressant* et le mot *importance,* à la machine. Aujourd'hui, la technique, la plus utilisée, pour définir *intéressant* est sous forme d'une liste de mots clés. L'*importance* par contre est définie par le nombre d'occurrences de ces mots clés dans un message.

Nous distinguons deux opérations principales dans le processus de filtrage de messages [TUR 00]:

- **Classement** : modifie et affecte les propriétés intrinsèques des messages (propriétés de stockage) : déplacer les messages vers un répertoire, supprimer un message, générer automatiquement un "faire suivre" ou "répondre", etc.

Exemples de filtres de classement:
si *From* **contient** "@mail.cerist.dz", **déplacer** dans le répertoire "travail".

si *Received* **contient** "205.199.212.", **déplacer** dans le répertoire "spammers".

- **Affichage** : modifie ou génère les propriétés extrinsèques, en analysant certaines propriétés dynamiques d'un message : le score, le poids ou la propriété d'un message déterminé sur la base d'un tri, la couleur à utiliser pour afficher les lignes d'informations d'un message.

Exemples de filtres d'affichage:
si *Subject* **contient** "filtrage d'information", **rapprocher** son score de la tête de liste d'affichage.
si *taille* > 100 Ko, **diminuer** fortement son score.
si *priorité* = 3, **l'afficher** en rouge.

2.2 Quelques systèmes de filtrage du courrier électronique

La plupart des outils commerciaux existants sur le marché, qui proposent des fonctions de filtrage automatique de contenu, présupposent une énonciation de règles robustes de la part de l'utilisateur. C'est rarement le cas puisque l'utilisateur se contente de déclarer quelques mots clés recouvrant faiblement la thématique correspondante [TUR 00].
Nous présentons brièvement les fonctions de filtrage des trois gestionnaires de mails les plus utilisés (Outlook Express de Microsoft, Netscape Messenger et Eudora Pro).

a)- Outlook Express

Le filtrage consiste en un ensemble de règles qui permet d'éliminer purement et simplement les messages de certains expéditeurs, de faire un filtrage plus fin, en combinant plusieurs conditions complémentaires ou hétéroclites : reconnaissance d'un texte spécifié dans une partie du mail (expéditeur, objet, corps du message, copie vers, etc.), taille du message, pièces jointes, etc. Les actions de filtrage sont (plusieurs actions possibles sur le message ainsi filtré) : le copier ou le déplacer vers un dossier spécifié (classement automatique ou effacement s'il s'agit de la poubelle), le mettre en sur brillance et en couleur (très utile pour repérer d'un seul coup d'œil les messages importants, marquer comme lu (pour les messages secondaires à consulter à l'occasion), le supprimer du serveur, etc.

Figure V.2: Outlook Express

b)-Netscape Messenger

Le filtrage consiste à définir un ensemble de filtres : un filtre est construit comme suit : sélectionner un champ d'une liste déroulante contenant les différents champs sur quoi va s'appliquer le filtre (objet du message, expéditeur, date, corps, etc.), puis le type de critère (contient, ne contient pas, commence par, etc.) avant de fournir le texte à rechercher. Un clic sur *Davantage* permet d'introduire un nouveau critère simultané, la recherche multicritères pouvant ainsi accepter jusqu'à cinq critères, avec opérateurs logiques OU et ET, qui s'appliquent toutefois sur l'ensemble des conditions (pas de panachage possible et/ou). La ligne "alors" permet de définir le devenir du message filtré : déplacement dans un dossier spécifié, suppression, mention lu, changement de priorité, etc. Là encore il est possible de combiner plusieurs filtres, qui seront appliqués successivement.

Figure V.3: Netscape Messenger

c)- Eudora

Le filtrage est le même que celui dans Netscape messenger, par l'option filtres du menu Outils. Cliquez sur Nouveau et laissez cochée la case "entrant" (Eudora peut en effet effectuer un filtrage dans les deux sens, réception et émission de mails). Dans la liste déroulante En-tête, nous devons choisir le segment du mail à analyser : expéditeur, sujet, etc. La liste déroulante suivante détermine le mode de recherche du texte (contient, ne contient pas, n'est pas, etc.), le texte lui-même devant être tapé dans la ligne en regard. Eudora autorise la recherche bicritère, avec opérateurs et/ou/sauf si, les opérations précédentes devant alors être doublées. Les actions à effectuer (jusqu'à 5 par filtre) doivent être sélectionnées dans la liste : copie ou transfert vers un boîte, étiquette, impression, signal sonore, etc.

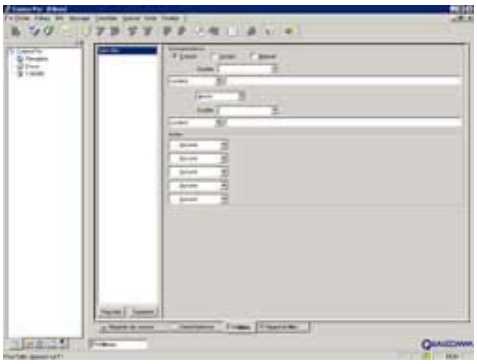


Figure V.4: Eudora

En parallèle, divers prototypes ont été proposé, tels que *IFILE* [REN 98], *RE:AGENT* [BOO 98], etc. Par exemple, l'outil *IFILE* développé par Rennie [REN 98] utilise une méthode Bayésienne naïve pour créer des règles de filtrage. Il utilise la notion d'age. Par exemple, un mot est éliminé du profil quand l'age est supérieur à une fréquence donnée (nombre de messages filtrés depuis que le mot a été rencontré). Les champs utilisés sont *sujet, expéditeur, destinataire* et *corps* du message. Les mots vides sont éliminés et une troncature est réalisée. L'outil filtre le message en lui attribuant une catégorie (score maximal) en calculant la probabilité P (catégorie i / message).

2.3 Stratégies de filtrage

Actuellement, il existe deux principales stratégies pour filtrer les messages [KIL 97b] : sélectionner les messages rejetés (Killfiles) ou sélectionner les messages intéressants. En effet, dans certains systèmes simples, le filtrage consiste à identifier les messages à effacer et de considérer le reste des messages comme étant intéressants ou acceptables. L'utilisateur doit spécifier au système comment reconnaître les messages à rejeter. L'inconvénient majeur de cette méthode est que le système peut accepter des messages indésirables non prévus par l'utilisateur. Inversement, la deuxième méthode consiste à identifier les messages intéressants et d'effacer automatiquement les autres non sélectionnés. Même inconvénients que la méthode précédente, le système peut effacer un message intéressant pour l'utilisateur.

2.4 Approches pour traiter le courrier électronique

Cette partie présente les approches possibles pour traiter le courrier électronique [KOS 01].

2.4.1 La classification de textes

Cette approche est appropriée lorsque les messages sont assez longs et stéréotypés [KOS 01]. L'hypothèse de cette approche est que le nombre de types de messages est petit. Les messages sont catégorisés selon leur type.
Le système de filtrage affecte automatiquement une catégorie à chaque message reçu, ce qui permet à l'utilisateur d'ordonner la lecture de son courrier, par exemple en fonction de la priorité qu'il attribue à chaque catégorie. On peut imaginer au moins deux principes de classification de messages :

- Une classification par *thème*, où l'utilisateur définit une nomenclature de messages qui permet au système de classer chaque nouveau message dans une catégorie représentant son thème. Cette classification par thème aidera l'utilisateur à organiser ses messages en groupes cohérents et faciles à archiver par exemple.
Un exemple de telle nomenclature pourrait être : Personnel, projet A, projet B, appel à communication, invitation, réunion de travail, spam, etc.

- Une classification par *importance*, où le système affecte au nouveau message reçu une priorité qui permet à l'utilisateur de déterminer dans quel ordre consulter ses messages.
De nombreux travaux ont porté sur la classification de textes, s'appuyant sur des techniques d'apprentissage automatique [COH 96c] [YAN 99] ou sur des approches linguistiques [CHA03].

2.4.2 L'extraction d'information

Consiste à identifier de l'information bien précise à partir d'un document et à la représenter sous forme structurée [MUC 7]. L'extraction d'information s'avère très pratique dans l'industrie où des opérations d'extraction y sont quotidiennement effectuées à la main. Par exemple, le traitement de rapports de filature d'une agence de surveillance, la gestion de dépêches d'une agence de presse, la manipulation de rapports d'incidents d'une compagnie d'assurances, etc. L'un des grands succès de l'extraction d'information est le domaine de l'extraction d'entités nommées, car il est indépendant du domaine du discours. Le système permet d'extraire des noms (de personnes, départements, etc.), des lieux et des dates en utilisant des dictionnaires de noms connus et d'indicateurs lexicaux (Mr., Mme, inc, etc.) et des expressions régulières. L'apprentissage à partir de corpus est le moyen utilisé pour l'automatisation de la tâche d'extraction automatique d'informations [ARC 02]. En effet, le corpus de base, une fois normalisé (étiqueté et/ou lemmatisé), doit contenir des amorces permettant d'initier un processus d'acquisition dynamique et itératif. Ces amorces sont des mots ou groupes de mots, représentatifs de l'information à extraire : patrons syntaxiques. Par un processus récursif et d'extraction, le système cherche dans le corpus et par des calculs statistiques augmente sa liste d'amorces.
Les méthodes actuelles ont démontré leur efficacité dans l'analyse de textes formels (ex : textes journalistiques) où les critères rédactionnels sont stricts. Par exemple, à partir d'un rapport sur un accident automobile, un système d'extraction d'information sera capable d'identifier la date et le lieu de l'accident ainsi que les noms des victimes. Par contre, les messages électroniques ne suivent pas des critères rédactionnels stricts et sont généralement informels et contiennent du bruit. Le bruit textuel peut venir de la typographie (ex. des tables et des dessins faits avec des caractères), de la terminologie (ex. des abréviations informelles), l'orthographe et des irrégularités grammaticales et aussi des particularités rhétoriques (ironie, humour, etc.) dont le sens est difficile à cerner. Mais lorsque le domaine de discours est restreint, l'extraction de scénarios peut être utilisée pour représenter le contenu du courrier de façon structurée. Par exemple, le contenu des messages d'appels à communication peut être représenté de façon structurée sous forme de patrons (lieu de conférence, date, deadline, etc.). L'utilisateur peut même utiliser certains patrons pour décrire ses propres règles de filtrage (ne retenir que les appels dont la date est inférieure à la date courante, etc.).

2.4.3 Le raisonnement par cas

C'est de calculer la similarité entre un nouveau message et les messages déjà reçus et lui appliquer les mêmes actions de filtrage que le message le plus proche.

2.4.4 La recherche d'information

Considère le message comme une requête à faire correspondre à une base documentaire (base des profils). Filtrer revient alors à identifier le profil qui ressemble le plus au message (requête).

2.4.5 Question-Réponse

Le but de la question-réponse est d'identifier dans une base documentaire une réponse factuelle à une question courte et explicite (ex. *Quel est le nom de jeune fille de Hillary Clinton*?). Les systèmes de question-réponse effectuent généralement une recherche d'information (à partir des termes de la question et souvent d'un thésaurus) pour identifier les passages de la base documentaire les plus susceptibles de contenir la réponse. Ces passages sont ensuite analysés pour extraire des entités sémantiques particulières (noms de personnes, dates, etc.) et en fonction du type de réponse recherchée (date, lieu, etc.) l'entité la plus probable est extraite. Ce domaine, a l'avantage d'être indépendant du domaine de discours. Cette approche est utilisée dans des applications de suivi d'emails.

3 Typologie et choix du corpus

Pour le choix du corpus, l'existence d'une typologie de messages et d'un corpus de référence faisant défaut, nous nous sommes limités à trois types génériques de messages bien particuliers (figure V.5) pour construire le modèle de connaissances ou utilisateur (profil de base) : les messages *personnels*, *professionnels* et les messages indésirables (appelés *Spam*). Les messages personnels regroupent tous les messages familiaux, ceux provenant d'amis, ainsi que les messages personnels-professionnels (collègue-collègue, étudiant-professeur, etc.). Les messages *professionnels* regroupent les appels à communication, les annonces de livres, les articles, les messages de direction, institution, etc. Enfin, les messages non sollicités et indésirables appelés *Spam*, qui continuent à polluer nos boîtes emails de façon croissante. Ils constituent le centre de contre-intérêt de l'utilisateur. Il s'agit de messages publicitaires proposant des services, des produits miraculeux (maigrir en un temps record, etc.), offres de voyages à prix attractif, opportunités d'investissement pour devenir riche en peu de temps, propositions de cartes de crédit à taux d'intérêt réduit, messages pornographiques, etc. Le *spam* est un phénomène mondial et massif. Il cause de multiples désagréments tels que l'engorgement des boites d'emails et des serveurs d'emails, dilution des messages utiles, perte de temps et d'espace, etc. Devant l'importance de ce phénomène, il est donc nécessaire aujourd'hui, à fin d'aider l'utilisateur submergé de mails, d'élaborer des outils efficaces capables de traiter et de filtrer le courrier électronique, et plus particulièrement le courrier *spam*.

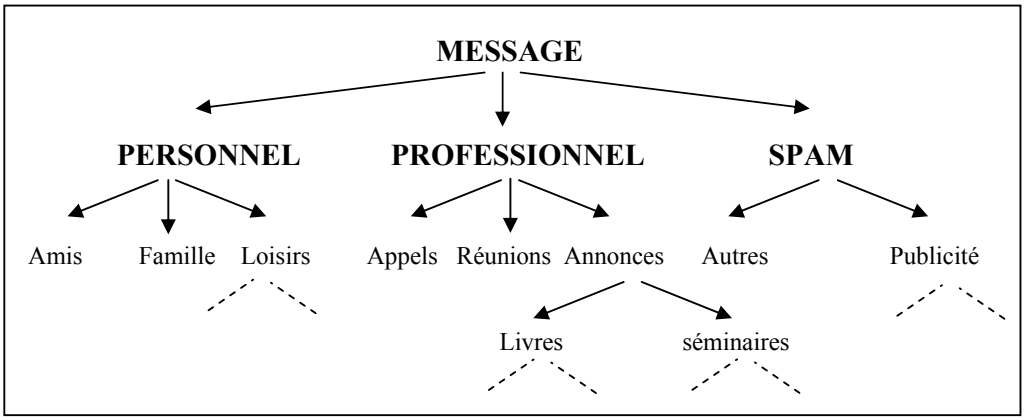


Figure V.5: Typologie de messages

4 Caractéristiques des mails

Les caractéristiques des mails (ou critères de filtrage) sont sélectionnées automatiquement par le système à partir du corpus d'apprentissage. Elles peuvent être aussi décrites et introduites dans le système sous forme manuelle : l'utilisateur introduit pour chaque type de profils une liste de critères lexicaux.
L'étude statistique menée sur notre corpus nous a donné les résultats suivants: un vocabulaire lexical et un ensemble de caractéristiques linguistiques supplémentaires.

4.1 Caractéristiques lexicales

Nous avons défini et identifié d'une façon automatique un vocabulaire lexical (annexe C), constitué de mots simples (vocabulaire de base) et de mots composés.
La construction du vocabulaire de base suit les étapes suivantes :

(1) : Elimination des mots communs.

(2) : Normalisation des termes.

(3) : Réduction du vocabulaire.

Le critère utilisé pour réduire le vocabulaire est la mesure de l'information mutuelle (chapitre3).
Le vocabulaire construit lors du traitement de notre corpus de mails comprend initialement 54760 mots. Ensuite, il est réduit à 600 termes dont les informations mutuelles sont les plus élevées. Voici un extrait du vocabulaire sélectionné pour les domaines considérés (table V.1):

Spam
business, time, money, free, price, product, credit, order, opportunity, guarantee, click, marketing, investment, risk, advertisement, sex, travel, miracle, etc.
Personnel
a+ , absence, actuellement, besoin, beaucoup, bises, bisous, bonheur, bonjour, bonne, boulot, contacter, courage, courrier, dérangement, désolé, dieu, dommage, embrasser, espérer, essayer, excuser, famille, galère, heureuse, job, joie, maman, médecin, merci, nouvelles, ok, papa, plaisir, salut, samedi, soin, super, vacances, vie, visa, visite, vœux, voiture, voix, voyage, etc.
Professionnel
actes, appel, calendrier, cher, collègue, comite, communication, conférence, contribution, cordial, critères, date, format, langage, langue, madame, monsieur, plaisir, salutation, soumission, veuillez, etc.

Table V.1 : Le vocabulaire de base

Les mots composés sont définis et identifiés d'une façon automatique à partir des listes *bigrammes* et *trigrammes* apprises par le système. Voici un extrait (table V.2):

Spam
bulk email, business opportunity, credit card, credit repair, financial news, free erotic, free investment, half price, home business, home worker, immediate release, investment report, limited time, live sex, low price, major credit, money order, offer valid, order by phone, order report, phone number, return address, sex stories, special bonus, take action, time offer, xxx video, etc.
Personnel
à bientôt, à plus, à toute, après-midi, as-tu, aurais-je, deviens-tu, dis-moi, es-tu, fais-tu, grosso-modo, mi-temps, parce que, peux-tu, puis-je, rendez-vous, sais-tu, week-end, etc.
Professionnel
appel a communication, cher collègue, comite de lecture, comite de programme, comite d'organisation, critères de sélection, date limite de soumission, journées d'étude, salutations distinguées,..., final camera ready copy, method of submission, notification of acceptance, notification of workshops, notification to authors, organized by, organizing committee, paper submission, paper submission form, selection criteria, submitted papers, etc.

Table V.2 : Le vocabulaire composé

4.2 Caractéristiques supplémentaires

Lorsque nous analysons un message donné, nous déterminons également, en plus des caractéristiques lexicales, d'autres caractéristiques supplémentaires qui constituent un certain ensemble d'indices que nous ajoutons au vocabulaire lexical. Les résultats de l'étude sont :

- L'auteur du texte : L'auteur du texte est intuitivement un critère important. L'auteur peut être une organisation, une personne, etc. Dans notre cas, l'auteur du message ou expéditeur est généralement inconnu, il est donc intéressant d'analyser l'adresse émettrice mail qui est précisée dans le champ *From*, ou ce que nous appelons le domaine (.com, .gov, .edu, etc.). Ce domaine peut symboliser une société, une université ou simplement un pays, ce qui peut être exploiter pour arriver à une classification approximative du message. Nous observons par exemple, 52% de *spam* contre 6% de non *spam* pour le domaine *com,* 13% contre aucun pour le domaine *net,* etc.

Exemple :

rita78@msn.com
henry106@augsburg.net

Par exemple, si le domaine (dans l'adresse émettrice) est *« edu », «dz»,* le message a une probabilité faible pour être du type *spam*. Par contre, si le domaine est *« com », « net», « undisclosed », « recipients »* la probabilité d'être un *spam* augmente.

- La longueur du texte: L'exploitation de ce critère diverge. Certains pensent que la longueur d'un texte est vraiment discriminante et d'autre non. En effet, certains pensent qu'un message volumineux consomme beaucoup de temps pour être lu ou analysé et présente un risque de contamination (virus, cheval de Troie, etc.). D'autres pensent carrément le contraire, en affirmant que les messages courts sont souvent d'importance minime et doivent donc être ignorés [KIL 97b]. La taille est évaluée par le nombre de mots. 85% de *spam* sont de taille

relativement courte contre 95% pour *non spam*. Dans notre corpus, la longueur du message n'est pas vraiment discriminante.

- Le type du contenu (html/txt) : Nous ajoutons également comme caractéristique le type du contenu. 45% de *spam* sont de type *html*, contre aucun pour *non spam*. Ce résultat en réalité n'est pas toujours vrai car des *non spams* peuvent également contenir de l'*html.* 35% des mails *spams* sont des images, contre aucun pour *non spam.*

- La langue du message : 100% de *spam* sont en anglais contre 80% de *non spam* sont en français. Il faut nuancer ce résultat, car des *spams* peuvent également être en français.

- Le nombre de destinataires: La liste des destinataires du message joue un rôle important. Si vous êtes le seul destinataire du message, ce dernier a de grandes chances d'être d'ordre personnel, tandis que si vous faite partie d'un ensemble de destinataires (des abonnés à une liste de distribution par exemple), cette possibilité est à écarter. En sachant que d'autres gens figurent dans la liste des destinataires du message et en connaissant les intérêts de ses gens, nous pouvons arriver à déterminer le domaine auquel appartient ce message.

- Mots en majuscule: Nous constatons que les pourcentages de mails *spam* et *non spam* contenant des mots qui commencent par une majuscule dans leur champ *subject* sont équivalents. Par contre, nous observons que les mails qui contiennent plusieurs majuscules dans leur champ *body* ont une probabilité plus forte d'être un *spam :* 86% de *spam* contre 5% pour *non spam*.

- **Abréviations** : 35% de *spam* contre 10% pour *non spam.*

- Les caractères non alphanumérique ($, !, #, %, *, &, etc.): 65% de *spam* contiennent des caractères non alphanumériques contre 2% pour non *spam*. 76 % de *spam* contiennent le point d'exclamation (ex : *get rich quick*!) contre 90% de non spam ne le contiennent pas. 43 % de *spam* contiennent le caractère $ (90% dans le champ subject) contre aucun pour *non spam*. La probabilité pour qu'un mail soit un *spam* est donc plus grande si le champ *subject* du mail contient au moins un caractère non alphanumérique.

- Les caractères numériques : Nous constatons que la présence de chiffres dans le corps des messages n'est pas très discriminante. Nous pouvons tout de même considérer que si le champ *subject* contient au moins un chiffre, la probabilité d'être un message *spam* est plus forte : 80% de *spam* contre 10% pour *non spam*.

- **La taille des phrases** : 76% de *spam* contiennent des phrases courtes (<10 mots) contre 92 % pour *non spam*.

- La longueur de l'entête des messages: Nous constatons que l'entête du message constitue un critère qui semble particulièrement intéressant pour caractériser un message *spam*. En effet, les messages *spam* subissent avant d'être reçus par le destinataire, un certain nombre de relais par des serveurs de mails de façon à atteindre un maximum d'utilisateurs. 96 % de *spam* de la base subissent des relais contre aucun pour non *spam*[1].

[1] Un relais est identifié dans l'entête du message par le mot clé : « Received : from… »

Voici pour exemple l'entête d'un message caractérisé comme étant un *spam*. Nous observons que ce *spam* a une entête contenant 15 lignes. Les lignes 3 à 6 sont des relais. Ce message a subi 04 relais.

Exemple:

+ok 1463 octets
return-path: <onouali@hotmail.com>
received: from hotmail.com (f70.law8.hotmail.com [216.33.241.70])
by mail01.wissal.dz (8.9.3+sun/8.9.3) with esmtp id oaa29554
for <onouali@wissal.dz>; wed, 13 feb 2002 14:29:44 -0100 (gmt)
received: from mail pickup service by hotmail.com with microsoft smtpsvc;
wed, 13 feb 2002 05:25:57 -0800
received: from 194.57.187.15 by lw8fd.law8.hotmail.msn.com with http;
wed, 13 feb 2002 13:25:56 gmt
received: from pop3 (f70.law8.hotmail.com [216.33.241.70])
by hanimail.com with esmtp id oaa29554
for moneynow22@hanimail.com; wed, 13 feb 2002 12:20:15 -0100 (gmt)
from: "Bessemer" < bessemer@zworg.com >
to: onouali@wissal.dz
subject: Advertise Via The Net, Earn Six figures This Year!
date: wed, 13 feb 2002 13:25:56
mime-version: 1.0
content-type: text/html; charset=iso-8859-1
message-id: <f70kh2vms1medynr7aj00018033@hotmail.com>
x-originalarrivaltime: 13 feb 2002 13:25:57.0284 (utc) filetime=[f857b640:01c1b491]
status: ro

La probabilité d'avoir un mail de type *spam* est donc plus forte quand la taille de l'entête est grande.

- Les fichiers attachés : 98% de *spam* n'ont pas de fichiers attachés contre 92% pour *non spam*.

- **Horaire d'envoi (nuit/jour)** : L'importance de ce critère réside, en plus de la capacité d'ordonnancement chronologique des messages. 65% de *spam* sont envoyés la nuit contre 88% de *non spam,* sont envoyés le jour.

- etc.

5 Filtrage automatique d'emails, une approche adaptative et multi niveaux

La plupart des systèmes de filtrage du courrier électronique existants enregistrent des lacunes ou faiblesses du point de vue efficacité de filtrage. Certains systèmes sont basés seulement sur le traitement de la partie structurée (par exemple, dans le cas de messages non sollicités appelés *spam*, le filtrage opère généralement sur les adresses émettrices en se basant sur une liste noire des *spammeurs*), et d'autres sont basés sur un balayage superficiel de la partie texte du message en permettant aux utilisateurs d'écrire manuellement des règles logiques de filtrage à base de mots clés.
Le problème, avec ces systèmes, est qu'ils ne sont pas précis car l'aspect sémantique est négligé et le processus de filtrage est une classification basée simplement sur une propriété lexicale, la présence ou l'absence de mots-clés que l'utilisateur doit indiquer au logiciel. Nous proposons une amélioration de ces systèmes : un système paramétrable avec plusieurs niveaux de filtrage :

(1) un filtrage simple basé sur l'information structurée, qui offre à l'utilisateur la possibilité de définir un ensemble de règles de filtrage sur l'information contenu dans l'ensemble de l'entête du message.

(2) Un filtrage booléen appelé superficiel basé sur l'existence ou non de mots clés dans le corps du message.

(3) Un filtrage vectoriel appelé intermédiaire, basé sur le poids de contribution des mots clés du message.

(4) Un filtrage approfondi basé sur les propriétés linguistiques caractérisant le contenu ainsi que l'utilisation de connaissances lexicales spécifiques permettant d'améliorer la représentation du message en prenant en considération l'aspect sémantique (expansion).

5.1 Architecture générale

L'interface de filtrage d'emails a été développée en utilisant l'assistant GIFI (chapitre 4) : certaines connaissances sont construites automatiquement. Le système est constitué principalement des modules suivants :

5.1.1 Pré-traitement

Ce module est lancé pour préparer les messages récupérés de la boite d'emails, aux différentes étapes ultérieures de l'analyse. Il consiste à isoler les différents champs et à identifier la langue (figure V.6).

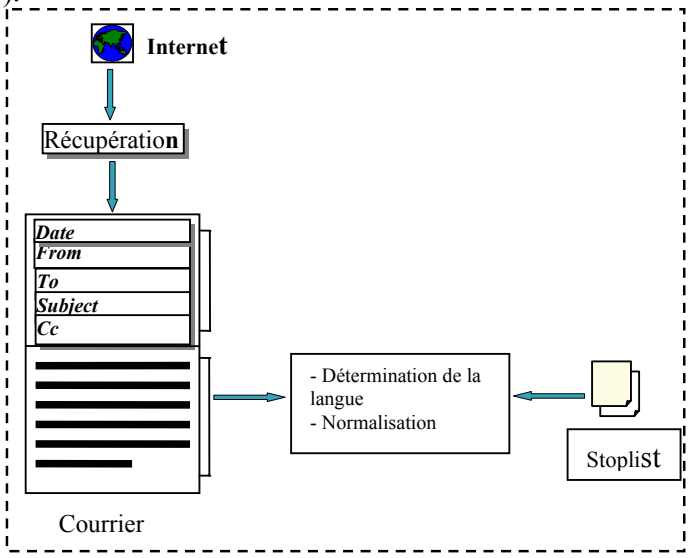


Figure V.6: Prétraitement du message

Puis, ayant connaissance de la langue, un ensemble de règles de lemmatisation est appliqué sur les différents mots du message (normalisation) pour réduire les variantes morphologiques à une forme commune (rendre les verbes à l'infinitif, supprimer les formes plurielles, etc.).

5.1.2 Analyseur automatique

A l'issue de l'étape de pré-traitement, le message est donné à un analyseur automatique, qui a pour but d'identifier et d'extraire les informations pertinentes à représenter, permettant de caractériser le contenu du message. Il délivre en sortie une représentation vectorielle associée, selon le niveau de filtrage choisi : par exemple, dans le cas du filtrage approfondi, le message passe par plusieurs niveaux d'analyse (figure V.7). L'objectif de chaque niveau est d'analyser le message et d'extraire un ensemble de propriétés sous forme d'indices. La représentation interne du message est construite au fur et à mesure de l'analyse.

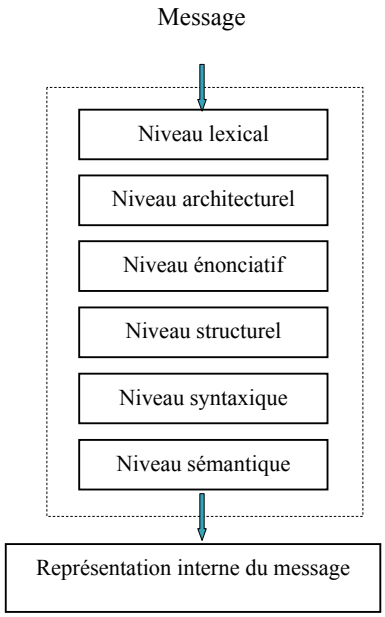


Figure V.7: Analyse linguistique du message

Ensuite, la représentation obtenue est complétée éventuellement par le processus d'expansion (chapitre 4). Cette représentation finale constitue l'entrée du processus de filtrage. Elle est donc crée dynamiquement à chaque récupération d'un nouveau message.

5.1.3 Processus de filtrage

Il est composé de deux principaux modules qui coopèrent entre eux pour effectuer le filtrage :

a)- Réseau de neurones

Il modélise les différents types de messages considérés. Il constitue la connaissance du système de classification. Il permet de classer un nouveau message analysé par le module précédent. Le modèle utilisé est un réseau de neurones non récurrents (absence de boucles) à trois couches (chapitre 4). Le réseau est entraîné sur un corpus de messages que nous avons collecté pendant une certaine période, dans le but de catégoriser correctement un nouveau message.
La représentation interne obtenue par l'analyseur précédent sera propagée à travers les différentes couches du réseau pour donner en sortie le type du message (figure V.8).

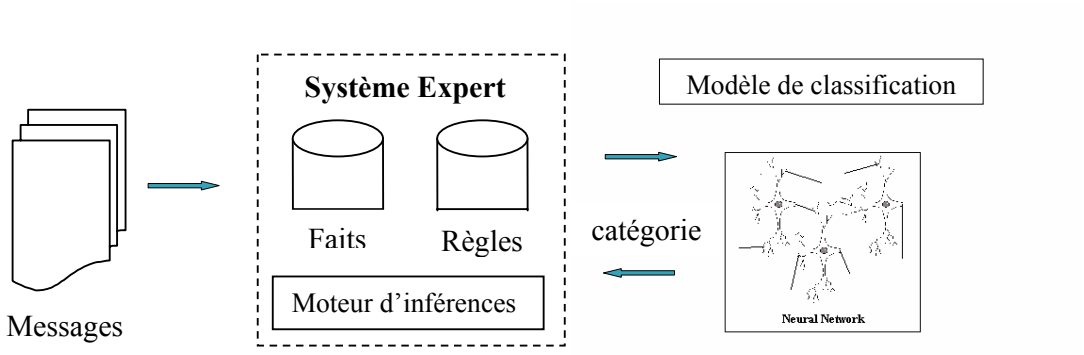


Figure V.8: Le processus de filtrage

b)- Système Expert

Le système expert pilote le processus de filtrage : il coopère, selon le niveau de filtrage choisi par l'utilisateur, avec les autres modules du système. Il permet de prendre en conséquence, des actions de filtrage tel que supprimer, sauvegarder, signaler, etc., définies par l'utilisateur.

Les connaissances du SE sont organisées sous forme de règles de production de la forme:
Si *(suite de conditions)* ***Alors*** *(suite d'actions)*

Les conditions portent sur les différents champs du courrier (*From, Subject, to, date*, etc.) et le critère concernant la catégorie du message. Les actions à entreprendre sont: sauvegarder ou classer, déliter, générer une réponse automatique, etc. Voici quelques exemples de règles :
Règle 1: *if FROM= 'omar@tassili.cerist.dz' THEN delete (message);*
Si le message provient de la personne Omar, alors effacer le message.

Règle 2: *IF SUBJECT Contains 'sport' THEN save (message) in sport folder;*
Si le mot 'sport' apparaît dans le champ subject du message alors sauvegarder le message dans le répertoire 'sport'.

Les différents types de conditions et actions, ainsi que le mode de fonctionnement du moteur d'inférence sont présentés en annexe D.

L'avantage d'utiliser un système expert est qu'il peut justifier les décisions concernant l'opération de filtrage auprès de l'utilisateur en lui présentant l'ensemble des règles ayant contribué à la prise de chaque décision. Ce qui permet à l'utilisateur, à tout moment, de mettre à jour la base de règles.

5.2 Niveaux de filtrage

Le système est destiné à traiter des informations textuelles semi-structurées, il s'agit de :

- une entête bien définie (Header) : From, To, Subject, etc.
- un corps libre non structuré.

Notre système est paramétrable avec plusieurs niveaux de filtrage: un filtrage basé header et trois types de filtrage basé contenu (superficiel, intermédiaire et approfondi).

-Filtrage basé sur l'information structurée (header): c'est un filtrage simple, qui offre à l'utilisateur la possibilité de définir un ensemble de règles de filtrage sur l'information contenu dans l'ensemble de l'entête du message.

- Filtrage superficiel : c'est un filtrage booléen, qui traite le contenu du message, mais d'une façon très superficielle. Il est basé sur la comparaison exacte entre le profil et les messages. Il est basé sur l'existence ou non de mots clés dans le corps du message. L'utilisateur exprime ses profils par des mots qui doivent exister ou ne doivent pas exister dans le message à recevoir.
Le système sélectionne les messages qui satisfont une expression logique sur les termes du profile. Les opérations de base pour ce modèle sont les connecteurs logiques : ET (AND), OU (OR) et SAUF (NOT). Par exemple, le profil exprimé par l'expression logique P= (« intelligence » OU « raisonnement ») ET « artificiel », permet de sélectionner tous les messages contenant le terme « artificiel » et un des termes « intelligence » ou « raisonnement ».

- Filtrage intermédiaire : C'est un filtrage vectoriel, basé sur le poids de contribution des différents termes du message. Il utilise comme support de base un modèle utilisateur représentant les différents profils. Chaque profil est représenté par un ensemble d'entités lexicales les plus pertinentes. Le profil est soit généré automatiquement à partir d'un ensemble de messages, soit introduit par l'utilisateur sous forme de mots clés.

- Filtrage approfondi : Le filtrage est basé sur des indices qui portent généralement sur la structure et le contenu des messages. L'objectif est donc d'élargir l'éventail des propriétés du modèle précédent. Ce type de filtrage nécessite une analyse linguistique du message qui délivre en sortie un ensemble de propriétés le caractérisant. Pour cela, le message passe par plusieurs niveaux d'analyse.

5.3 Connaissances utilisées

La mise en pratique d'un modèle utilisateur est difficile car, l'utilisateur lui même a des difficultés à décrire ses attentes de manière formelle et explicite. Nous offrons à l'utilisateur deux possibilités pour décrire ses propres intérêts :

- Modélisation manuelle : l'utilisateur introduit pour chaque type de profils une liste de critères sous forme de mots clés. Il associe, à chaque critère, un poids qui représente son degré d'importance. Les profils sont améliorés et augmentés de propriétés linguistiques par un apprentissage.

- Modélisation automatique : le système se charge de la modélisation en se basant sur un apprentissage à partir de corpus. En effet, le système se charge d'analyser et d'extraire des mots clés et des propriétés linguistiques et de leur attribuer un poids.

Le système de filtrage utilise un ensemble de connaissances linguistiques de base (chapitre 4) pour analyser, extraire les différentes propriétés et construire la représentation interne de chaque message. Ces connaissances sont indépendantes du domaine d'application. De plus,

un modèle utilisateur de base, sous forme d'une typologie générale, permet d'aider l'utilisateur à décrire ses propres profils. Il est généré à l'aide de l'outil GIFI : en effet, pour déterminer donc les propriétés linguistiques utiles pour notre modèle de base, nous faisons passer chaque message du corpus par les différents modèles linguistiques réduits de base. En sortie de cet apprentissage, nous obtenons une typologie constituée seulement de propriétés linguistiques utiles (poids supérieur à un certain seuil) pour chaque type considéré. Elle est réalisée (modélisée) par un réseau de neurones (figure V.9) dont la connaissance est construite automatiquement à partir d'un ensemble de corpus et de ces modèles linguistiques de base. Cette typologie constitue donc une sorte de profil de base, qui permettra d'aider l'utilisateur à décrire et élargir ses propres profils.

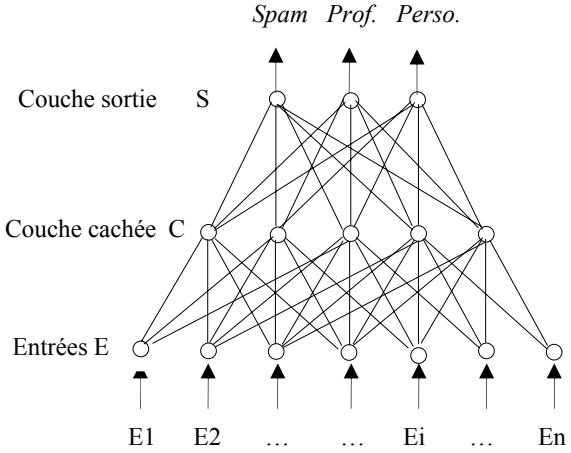


Figure V.9 : Architecture d'un réseau à trois couches

Une couche en entrée qui reçoit les entrées du réseau. Une couche cachée représentant l'ensemble des connaissances. Une couche de sortie qui représente les types de textes (*Spam, professionnel* et *Personnel*).

5.4 La correction

Une fois un message reclassé, le système doit réorganiser les messages en fonction des courriels lus par l'utilisateur. Le système dispose d'un **apprentissage assisté** appelé ***feed-back*** où l'utilisateur peut soit donner un avis direct sur le message lu, soit déplacer un message d'une classe à une autre (cas de mauvaise classification par le système), ce qui lui permet d'approcher la pertinence de l'utilisateur et de s'adapter ainsi à ses besoins. L'apprentissage agit sur le modèle, qui consiste à modifier les poids dans le but d'améliorer la réponse du système. L'utilisateur peut aussi ajouter et supprimer des mots et des profils à sa demande.

6 Evaluation

Cette partie est consacrée à l'évaluation des performances de notre approche de filtrage, sur un corpus de messages électroniques. Nous présentons, dans un premier temps, le corpus utilisé, un corpus de messages électroniques issu d'une collecte effective pendant une certaine

période. Dans un deuxième temps, nous donnons quelques mesures chiffrées de performance de notre approche du FI. En effet, nous avons mené des tests pour :

i) mesurer l'importance et le rôle de l'information linguistique dans la représentation des messages,

ii) mesurer les performances du système de classification du point de vue précision et rappel,

iii) et montrer comment l'opération d'apprentissage agit sur l'efficacité du filtrage.

6.1 Le corpus

Pour effectuer nos tests nous avons travaillé avec un corpus de 1200 messages construit à partir d'un ensemble de messages que nous avons collectés pendant quatre mois (10 messages en moyenne par jour), ce qui représente 5,6 Mégaoctets de texte. Il regroupe 700 mails de classe *spam* et 500 non *spam* (35% de messages sont de type *personnel* et 65% de type *professionnel*).

Nous avons divisé le corpus en une base d'apprentissage (ou de paramétrage) et une base de tests selon le découpage suivant (table V.3):

Catégorie	% de messages par catégorie	Base d'apprentissage (2/3)	Base de test (1/3)
Spam	58,33	470	230
Personnel	16,67	135	65
Professionnel	25	200	100

Table V.3. Découpage du corpus de travail

Le principe d'une telle répartition est de fournir au système évalué un sous-ensemble des données de référence, qui servira au paramétrage, sans limite de temps ou d'itérations (l'apprentissage automatique peut subir plusieurs présentations du même corpus d'apprentissage), ainsi qu'un sous-ensemble de test, constitué de données inconnues du système. Le corpus de test sert à vérifier l'adéquation du paramétrage, il est donc nécessaire, afin d'obtenir des résultats interprétables, que les deux ensembles de données soient comparables (i.e. même domaine).

6.2 Les critères d'évaluation

Pour l'évaluation de notre système, nous avons suivi un protocole de type « boîte noire », où seule la différence entre le nombre de réponses attendues et celles observées est prise en compte.
Les conférences d'évaluation TREC, ont montré que le domaine du FI se caractérise par un flottement terminologique et conceptuel, qui se traduit par une absence regrettable de cadre méthodologique stable pour l'évaluation des systèmes automatiques de filtrage. C'est-à-dire, aucune métrique d'évaluation TREC ne semble faire l'unanimité, essentiellement, en raison de l'absence d'un ensemble de données de référence issu d'une pratique effective du filtrage d'information. Dans notre cas, les corpus d'apprentissage et de test constituent des ensembles

bornés, pour lesquels nous connaissons exactement la répartition en classes de chaque message (personnel, professionnel, etc.). Par ailleurs, le volume de données traité, de l'ordre du Mégaoctet, reste manipulable, contrairement aux volumes titanesques de TREC, qui justifient les méthodes d'échantillonnage (sampling, pooling, etc.).
Pour mesurer les performances du système, nous choisissons d'utiliser les deux métriques de performance standard en recherche d'information : les mesures de *précision* et de *rappel*. Les scores de silence et de bruit, sur lesquels reposent la précision et le rappel, sont donc calculés simplement en faisant la différence entre les réponses observées et les réponses attendues. L'idéal théorique étant de minimiser les deux taux conjointement (taux de silence et de bruit tendant vers 0%). Nous déterminons également la performance globale du système en calculant le pourcentage d'erreurs et de succès.

Jugement du Système ⇓	Jugement de l'utilisateur	
	C	$\neg C$
$\boldsymbol{C}$	α	β
$\neg C$	γ	δ

Table V.4 : Critères d'évaluation

Avec :
α : messages de classe *C*, correctement filtrés (classés) par le système.
β : messages n'appartenant pas à la classe *C*, incorrectement filtrés par le système.
γ : messages de classe *C*, incorrectement non filtrés (rejetés) par le système.
δ : messages n'appartenant pas à la classe *C*, correctement non filtrés par le système.

Les mesures *rappel* et *précision* pour le filtrage de messages de classe *C* sont :

$$Rappel = \frac{\alpha}{\alpha+\gamma} \qquad Précision = \frac{\alpha}{\alpha+\beta}$$

Les mesures globales *erreur* et *précision* du système sont *:*

$Erreur\_globale = \frac{\beta+\gamma}{\alpha+\beta+\gamma+\delta}$, est le rapport entre le nombre total de messages incorrectement filtrés et incorrectement non filtrés et le nombre total de messages de la base de test.

$Précision\_globale = \frac{\alpha+\delta}{\alpha+\beta+\gamma+\delta}$, est le rapport entre le nombre total de messages correctement filtrés et correctement non filtrés et le nombre total de messages de la base de test.

6.3 Expériences

Nous présentons et nous discutons, dans ce qui suit, les résultats des performances, au cours d'une évaluation quantitative de notre système de filtrage, dans plusieurs cas de configuration :

6.3.1 Performances en fonction des caractéristiques lexicales seulement

Nous mesurons les performances du système en considérant tout d'abord un modèle de base (MB) constitué uniquement de mots simples. Les connaissances du système sont décrites par trois profils et introduites dans le système en deux cas différents : modélisation manuelle et modélisation automatique.
Les résultats de filtrage dans les deux cas de figure sont donnés dans la table V.5 :

Catégories	**Modélisation manuelle**	**Modélisation automatique**	
		Avant modification	**Après modification**
Personnel	45%	83%	85%
Professionnel	72%	88%	91%
Spam	67%	87,7%	90%

Table V.5 : Performances en fonction des caractéristiques lexicales

Les résultats obtenus avec une modélisation automatique du profil sont nettement meilleurs qu'avec une modélisation manuelle (ce qui montre la difficulté de l'utilisateur à décrire ses propres profils). De plus, nous constatons que certains mots corrèlent avec certains types de messages considérés, mais statistiquement sont insignifiants (valeur faible). Ce qui nous a poussé à modifier l'importance des différents mots tout en gardant un traitement générique sans l'intervention de l'utilisateur. Le système attribue une forte valeur du poids aux mots qui sont uniques dans chaque catégorie, par rapport à ceux qui se trouvent dans plusieurs catégories. Les résultats des tests étaient meilleurs.

6.3.2 Performances en fonction des mots composés

Au début de l'expérience, nous ajoutons un ensemble de mots composés (MC) au modèle de base constitué initialement de mots simples (MB).

Caractéristiques	**Performance globale**					
	Personnel		*Professionnel*		*Spam*	
	Erreur Globale	Précision Globale	Erreur Globale	Précision Globale	Erreur Globale	Précision Globale
MB	17%	83%	12%	88%	12,3%	87,7%
MB + MC	17%	83%	11%	89%	13%	87%
MB + MC + Pondération	17%	83%	9%	91%	8,6%	91,4%

Table V.6 : Performances en fonction des mots composés

Nous ne constatons pas une amélioration des performances. En effet, les mots composés corrèlent avec les types de messages considérés, mais statistiquement sont insignifiants (valeur faible). Ensuite, nous avons donc modifié l'importance de ces différents mots composés, en leur attribuant une forte valeur du poids. Les résultats des tests étaient nettement meilleurs (ex : 91% pour le profil *spam*).

6.3.3 Performances en fonction du nombre de caractéristiques linguistiques

Dans cette partie, nous étudions les caractéristiques supplémentaires que nous ajoutons au modèle de base. Nous mesurons les performances du système en montrant l'importance et le rôle de ces caractéristiques dans la représentation globale des textes.
Nous avons défini et identifié un ensemble de propriétés automatisables, qui servent à caractériser les textes. Il s'agit d'un ensemble d'indicateurs sur le texte qui permettent de situer un texte par rapport aux autres, de rapprocher les textes qui appartiennent à la même classe ou éloigner ceux qui appartiennent à des classes différentes. Ces connaissances sont indépendantes du domaine d'application. Nous les avons classées en plusieurs niveaux : matériel, énonciatif, structurel et syntaxique (chapitre 4).
Dans un premier temps, nous avons considéré toutes les propriétés linguistiques sans restriction pour la construction du modèle. Les résultats sont résumés dans la table V.7:

Catégories	**Performances**
Personnel	74%
Professionnel	79%
Spam	82%

Table V.7 : Les performances du modèle sans restriction

Nous constatons que les performances du système sont moyens. En effet, certaines caractéristiques ne corrèlent pas avec certains types de messages (valeur=0). Nous avons donc testé les performances lorsque nous réduisons le nombre de propriétés du vocabulaire en imposant un seuil sur les occurrences des propriétés à considérer pour les différentes représentations des mails.
En effet, pour déterminer donc les propriétés linguistiques utiles pour notre modèle de base, nous faisons passer chaque message du corpus par les différents modèles linguistiques réduits. En sortie de cet apprentissage, nous obtenons une liste constituée seulement de propriétés linguistiques utiles (poids supérieur à un certain seuil).

Les résultats de filtrage sont donnés dans la table V.8.

Catégories	**Modèle de base**	**Modèle de base + PL**
Personnel	85%	92%
Professionnel	91%	93%
Spam	90%	95%

Table V.8 : Les performances avec restriction

Nous constatons que les performances globales du système sont améliorées. Ceci s'explique par le fait que les messages rejetés par le système (1ère expérience) par absence de mots clés ou valeur très faible, sont acceptés cette fois ci, et ceci à cause de la présence de certaines propriétés linguistiques (PL). Par exemple, les messages personnels sont caractérisés par l'utilisation de pronoms personnels (1ère et 2ème personne).

6.3.4 Mesurer l'importance et le rôle de l'apprentissage assisté

Dans cette expérience, l'utilisateur a la possibilité de créer ses propres profils :

- soit utiliser ou combiner des types prédéfinis,

- soit proposer de nouveaux types.

Pour des raisons ergonomiques, une solution simple est de présenter à l'utilisateur des types ou classes de messages clairement identifiables pour lui plutôt qu'un ensemble de propriétés linguistiques, complexes et inexploitables pour lui.

Nous considérons un utilisateur avec trois profils différents :

- un profil *P1* choisi parmi les 3 proposés par le système (*Spam, Personnel* et *Professionnel*),

- un nouveau *P2* crée en proposant une liste de mots clés,

- et enfin un autre profil *P3* choisi et modifié par l'utilisateur.

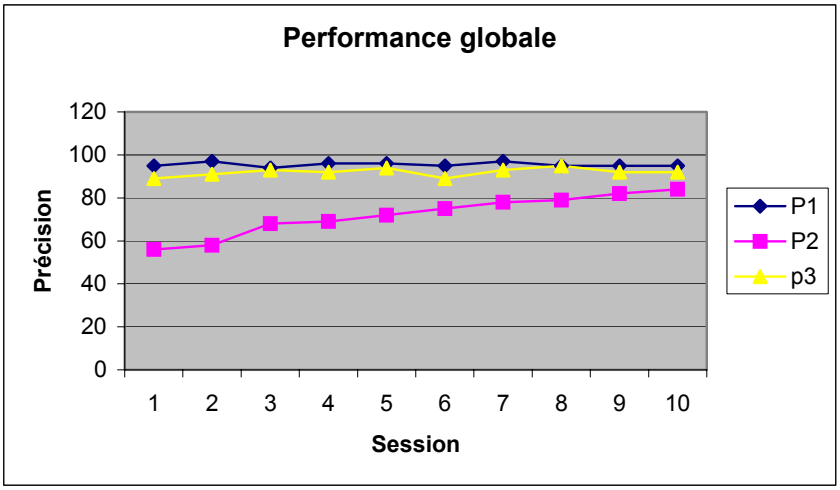


Figure V.10 : Apprentissage assisté

Nous constatons que les résultats varient d'un profil à l'autre. Après un certain temps d'apprentissage, les résultats des profils *P1* et *P3* restent presque stables, l'apprentissage n'améliore pas vraiment les résultats (profils satisfaisants). Par contre, dans le cas du profil *P2*, le modèle converge vers un modèle de filtrage satisfaisant, mais lentement. En effet, le modèle nécessite plusieurs sessions d'apprentissages assistés pour améliorer la qualité de ses résultats. Il est donc nécessaire de lancer l'apprentissage *feedback* régulièrement, par exemple après chaque session de filtrage.

6.3.5 Mesurer l'importance et le rôle du filtrage avec propagation (horizontal)

L'expérience consiste à présenter au système en deux cas différents, un ensemble de courriers à filtrer en plusieurs sessions. Puis mesurer à chaque fois la précision et le rappel et effectuer un apprentissage assisté pour mesurer son efficacité et son influence sur les deux facteurs.

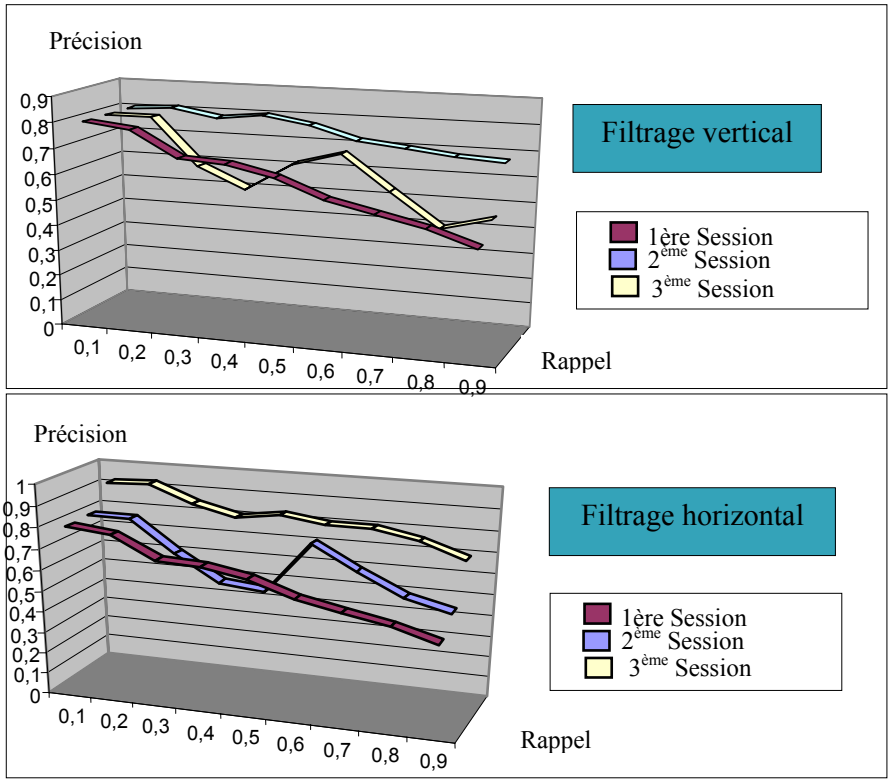


Figure V.11 : Filtrage horizontal vs filtrage vertical

Après plusieurs sessions d'apprentissages assistés, nous constatons que la convergence du *filtrage horizontal* vers un modèle satisfaisant est plus rapide que celle du *filtrage vertical*. En effet, dans le filtrage avec propagation, la cooccurrence des critères est prise en considération, ce qui permet d'augmenter le taux de *rappel* tout en gardant une bonne précision. Par exemple, en ce qui concerne les messages de type *personnel*, on obtient un taux de filtrage avec 92% de précision pour un rappel de 70%.

6.4 Discussion

Les principales techniques actuelles employées dans le domaine du filtrage sont basées d'une façon directe ou indirecte sur les techniques des méthodes traditionnelles de recherche d'information. Elles se basent sur l'occurrence d'un ensemble de mots clés pour identifier ou reconnaître l'information pertinente (modèle booléen, modèle vectoriel, etc.).
Pour notre part, nous insistons sur la nécessité d'inclure des traitements linguistiques ou offrir des fonctionnalités de traitement automatique des langues, dans le cadre d'application de filtrage d'information. En effet, l'avantage de l'approche statistique repose principalement sur sa simplicité, mais elle n'est pas très précise car l'aspect sémantique est négligé.

A travers ces différentes expériences, nous avons montré l'applicabilité et l'adaptabilité d'une approche linguistique au processus de filtrage. En effet, après les tout premiers tests, nous avons remarqué que les résultats semblaient plutôt satisfaisants. Mais nous ne pouvons pas affirmer que certaines propriétés telles que les propriétés concernant la structure matérielle constituent une connaissance suffisante de discrimination de messages. Néanmoins, la probabilité d'avoir un mail d'un certain type est plus forte quand ces caractéristiques sont vérifiées. Les résultats obtenus sur notre corpus semblent intéressants. Néanmoins, il serait intéressant d'étendre l'étude sur d'autres types de textes pour étendre la liste des critères et tester l'adaptabilité de l'approche.
Ainsi, à travers notre modeste expérience, la conclusion générale que l'on peut évoquer est que les méthodes linguistiques combinées aux méthodes statistiques semblent prometteuses pour avoir un filtrage efficace de l'information sur les réseaux de communication.

7 Conclusion

Ce chapitre propose un système de courrier électronique qui exploite le maximum d'informations et paramétrable avec plusieurs niveaux de filtrage:
(1) un filtrage simple basé sur l'information structurée, qui offre à l'utilisateur la possibilité de définir un ensemble de règles de filtrage sur l'information contenu dans l'ensemble de l'entête du message.
(2) Un filtrage booléen appelé superficiel basé sur l'existence ou non de mots clés dans le corps du message.
(3) Un filtrage vectoriel appelé intermédiaire, basé sur le poids de contribution des mots clés du message.
(4) Un filtrage approfondi basé sur les propriétés linguistiques caractérisant le contenu ainsi que l'utilisation d'un réseau lexical permettant d'améliorer la représentation du message en prenant en considération l'aspect sémantique. Ces propriétés constituent un ensemble d'indices qui portent généralement sur la structure et le contenu des messages.
Pour la mise en pratique du modèle de connaissances, une typologie de messages est présentée à l'utilisateur lui permettant de l'aider dans la tâche de création de ses propres profils.
Nous proposons une solution adaptative et évolutive qui permet un filtrage nettement meilleur en qualité : nous utilisons une méthode d'apprentissage automatique permettant au système d'apprendre à partir de données, de modifier ses connaissances et de s'adapter à l'évolution des intérêts de l'utilisateur (profils) et à la variation de la nature des mails dans le temps. En effet, un apprentissage du modèle utilisateur est entretenu au fur et à mesure des nouveaux courriels entrants mais aussi au moyen de corrections apportées par l'utilisateur.

Dans la partie, consacrée à l'évaluation des performances de notre système de filtrage, nous avons abordé les aspects techniques et opérationnels de l'implémentation du système réalisé au sein du laboratoire des logiciels de base (CERIST), ainsi qu'au laboratoire LPL (Université de Provence).
Nous avons évalué une approche évolutive, qui s'adapte bien à la nature des mails au cours du temps et exploite le maximum d'informations pour filtrer l'email. Elle fait essentiellement appel à des propriétés linguistiques qui permettent d'aider à améliorer les résultats de filtrage. Ces propriétés sont basées sur des modèles linguistiques réduits. Pour la mise en pratique du modèle de connaissances, dans le cas d'utilisateurs néophytes par exemple, une typologie de messages peut être présentée leur permettant de les aider dans la tâche de création de leurs propres profils. Un apprentissage des caractéristiques linguistiques est entretenu au fur et à

mesure des nouveaux courriels entrants mais aussi au moyen de corrections apportées par l'utilisateur.

Les expériences menées sur notre corpus de messages, très modeste, nous ont permis de valider :

- le recours aux connaissances linguistiques, sous forme de modèles linguistiques réduits, pour améliorer les performances d'un système de filtrage d'information. Ces connaissances, portant sur la structure et le contenu, sont classées en plusieurs niveaux linguistiques ;

- l'approche évolutive par apprentissage automatique, passage obligé dans la conception et l'amélioration des performances d'un système de filtrage d'information dynamique ;

- la prise en compte de l'aspect sémantique dans le processus de filtrage, en utilisant deux connaissances : un réseau lexical et la cooccurrence des critères de filtrage ;

- la portabilité du système : Les connaissances de base sont indépendantes du domaine d'application (les modèles linguistiques réduits). Les connaissances spécifiques à l'application (email) sont générées automatiquement. En effet, le profil de l'utilisateur est calculé par analyse automatique du contenu qui permet de produire un ensemble de termes et de propriétés linguistiques le caractérisant. De plus, notre système est complètement indépendant du domaine de connaissances. Il a une structure modulaire, lui permettant éventuellement de s'adapter à toute extension et modification.

Ces expériences ont également montré la nécessité de mettre en œuvre des interfaces intelligentes, adaptables en fonction de l'utilisateur. C'est-à-dire développer des systèmes ″boite noire dans une boite de verre″ (a black box in a glass box) où seuls les niveaux conceptuels les plus élevés sont accessibles à l'utilisateur, la complexité linguistique reste cachée. En effet, pour des raisons ergonomiques, le système présente à l'utilisateur des types ou classes de messages clairement identifiables pour lui plutôt qu'un ensemble de propriétés linguistiques, complexes et inexploitables pour lui (expérience 4).

Conclusion

1. Pour une approche guidée par les observables

Les études sur corpus nous semblent un passage obligé dans la conception d'un système d'analyse automatique. Ceci vient du constat de la prépondérance des approches guidées par les observables dans les domaines applicatifs.
Dans notre cas, nous avons examiné une application d'un principe d'analyse automatisée, reposant sur des traitements linguistiques faibles, au problème du filtrage d'information. Nous avons tenté de déterminer la relation et l'adéquation entre discrimination textuelle et occurrence de propriétés linguistiques. Nous avons donc montré quel pouvait être l'apport d'une étude linguistique des corpus dans un domaine applicatif. Toutefois, les résultats acceptables enregistrés dans notre expérience ne doivent pas occulter le fait que, l'adéquation entre propriétés linguistiques et types de textes n'est pas parfaite.

2. Intégration de techniques de TAL au service du filtrage d'information

Les ressources linguistiques et les différents outils et méthodes issus du TAL constituent une source importante d'amélioration de la qualité des systèmes d'analyse automatique et plus particulièrement de filtrage d'information, permettant en particulier d'étendre la représentation de textes et d'offrir ainsi des performances supérieures. En effet, des outils tels que les traitements linguistiques et les ressources lexicales existantes ou acquises sur corpus (la lemmatisation, le repérage des termes simples et complexes, etc.) autorisent la prise en compte de variantes et peuvent donc aider à la désambiguisation des mots, améliorer la représentation textuelle et par conséquent les performances.
Pour conclure, nous pouvons donc noter que la façon d'exploiter pleinement les informations de TAL dans plusieurs domaines et plus particulièrement en filtrage d'information est encore à trouver.

3. Approche proposée

Dans ce travail, nous avons proposé une approche évolutive qui s'adapte à la nature des informations au cours du temps et qui exploite le maximum d'informations pour filtrer l'information :

- Elle fait essentiellement appel à des propriétés linguistiques qui permettent d'aider à améliorer les résultats de filtrage. Nous avons défini et identifié un ensemble de

connaissances linguistiques (sous forme de modèles réduits), c'est-à-dire un ensemble de propriétés (indicateurs ou indices) portant sur la structure et le contenu des textes, qui servent à caractériser les textes et les situer les uns par rapport aux autres. Ces connaissances sont automatisables et indépendantes du domaine d'application. Nous les avons classées en plusieurs niveaux (matériel, énonciatif, structurel et syntaxique). Dans le cadre de ce travail, nous ne cherchons pas à faire une analyse complète et profonde du contenu des textes, mais plutôt, une analyse partielle utilisant plusieurs niveaux d'analyse qui permettent de dégager des propriétés linguistiques qui devraient permettre de distinguer les différents types de textes et de classer ensuite les nouveaux textes.

- Elle fait recours à un réseau lexical, pour l'aspect sémantique, regroupant les mots sémantiquement proches, qui permet d'améliorer la représentation de textes à filtrer et d'augmenter donc les chances d'apparier un texte et un profil. En effet, les termes du texte sont automatiquement propagés en suivant les liens exprimés dans le réseau, de manière à disposer d'une description plus étendue de ce texte. Notre modeste expérience (chapitre 5) nous amène à plaider pour l'utilisation de ressources lexicales dans un système de filtrage d'information. En effet, le traitement d'équivalence sémantique permet d'augmenter les performances du système.

- Pour la mise en pratique du modèle de connaissances, nous proposons, à l'utilisateur néophyte (ou concepteur), une boite à outils (ou générateur automatique d'interfaces), lui permettant de l'aider dans la tâche de création de ses propres interfaces de filtrage. Dans ce contexte, il est devenu important d'être capable de traîter de grandes quantités de données textuelles, d'apporter des solutions diversifiées aux nouvelles demandes des utilisateurs, et d'automatiser les outils qui permettent d'exploiter l'information textuelle. Notre boite à outils permet d'assister l'utilisateur dans le processus d'acquisition et de génération. Parmi les services offerts, l'analyseur automatique du contenu des textes : il consiste à associer aux textes un ensemble de termes et de propriétés linguistiques servant à les caractériser. Nous partons de propriétés issues de modèles linguistiques de textes (sous forme de modèles réduits), la fiabilité repose sur l'opération d'apprentissage.

Le domaine de filtrage de l'information reste un domaine très ouvert. Certains chercheurs utilisent les techniques traditionnelles en essayant toujours de les améliorer, d'autres proposent de nouvelles approches. Il n'y a pas encore de conclusion concrète. A travers notre modeste expérience, nous pensons que les méthodes linguistiques combinées aux méthodes statistiques semblent prometteuses pour avoir un filtrage efficace de l'information sur les réseaux de communication.

4. Perspectives

a)- Définir, pour la typologie du domaine d'application, une architecture générale et évolutive : l'architecture la plus adéquate est de structurer les différents types du domaine sous forme d'arborescence (hiérarchie) ouverte, appelée arbre de classification. Cette classification doit être la plus complète possible présentant les différents types, du type le plus général au plus spécifique. Cet arbre doit avoir les caractéristiques suivantes :

- Permettre l'héritage entre les types possédant des attributs communs.

- Permettre une identification du type par navigation dans l'arborescence jusqu'à atteindre les feuilles, représentant les types atomiques.

- Permettre d'ajouter (ou supprimer) des types jugés importants (ou non importants) par l'utilisateur.

b)- Nécessité de traiter les anaphores pour diminuer les biais de calcul des cooccurences.

c)- Il serait intéressant d'intégrer, dans le processus de modélisation, les méthodes d'apprentissage ou de *classification non supervisée.* Ces méthodes sont dites méthodes de structuration. Elles sont caractérisées par la non disponibilité d'aucune autre information préalable que la description des exemples. Elles sont destinées à produire des groupements d'objets, selon un critère de similarité ou dissimilarité à partir d'une description sur ces objets (traits, caractéristiques, propriétés, etc.). Dans ce cadre, il y a un travail en cours : il s'agit de la réalisation d'un outil qui permet de construire automatiquement un profil utilisateur à partir d'un ensemble de documents jugés pertinents par ce dernier.

d)- Etendre ou enrichir les minis modèles

e)- Expérimenter l'approche sur d'autres types de textes.

Bibliographie personnelle

Communications Internationales avec comité de lecture

[2003] Nouali O.,
Sélection de critères pour le filtrage automatique de messages, TALN/RECITAL 2003, 11-14 Juin 2003, Batz sur mer, France.

[2003] Nouali O.,
Interface en langage naturel pour une base de données, Colloque l'écriture dans tous ses états, approches en sciences cognitives, 20-21 Mai 2003, université de Provence, France.

[2003] Nouali O., Blache P.,
Filtrage de messages SPAM, Sixth International Symposium on Programming and System, ISPS'2003, 5-7 Mai, 2003, Alger, Algérie.

[2002] Nouali O., Blache P.,
Linguistic-Based Automatic E-mail Classification, International Conference on Artificial Intelligence and Soft Computing, ASC 2002, July 17 to July 19, 2002, in Banff, Canada.

[2002] Nouali O.,
Classification Automatique de Messages: une approche hybride, TALN/RECITAL 2002, 24-27 Juin 2002, Nancy, France.

[2000] Nouali O., Blache P.,
Automatic Classification and Filtering of Electronic Mail, International Conference on Artificial Intelligence and Soft Computing, ASC 2000, July 24-26 July 2000, Banff, Canada.

[1999] Nouali O., Blache P.,
E-mail Intelligent Filtering Tool, Forth International Conference on recent Trends in computer science applications & information systems, 13-14 July 1999, Philadelphia University, Faculty of Science, Amman, Hashemite Kingdom of Jordan.

[1998] Nouali O.,
Automatic Filtering of electronic information, à la 5ème Conference On Computer Communications, AFRICOM-CCDC'98, INTERNET AND GLOBAL NETWORKING, Tunis, Tunisie, 20-22 Octobre 1998.

Communications Nationales

[2001] Nouali O.,
Filtrage et Sécurité des messages sur Internet, Cinquièmes Journées Informatiques de l'Entreprise portant sur les Technologies de l'Information Intranet-Internet au service de l'entreprise, Sonelgaz, 1-2 Avril 2001, Hôtel SAFIR-MAZAFRAN, ZERALDA, Algérie.

Publications Internationales avec comité de lecture

[2004] Nouali O., Blache P.,
"Automatic Classification and Filtering of Electronic Information: Knowledge-Based Filtering Approach", at the International Arab Journal of Information Technology, IAJIT, Vol. 1, n° 1, January, 2004, Pages 86-94, ISSN: 1683-3198.

[2004] Nouali O., Blache P.,
"A Semantic Vector Space and Features-Based Approach for Automatic Information Filtering", Expert Systems with Applications, ESWA, An International Journal, Volume 26, Issue 2, Elsevier Ltd., February 2004, Pages 171-179.
http://authors.elsevier.com/sd/article/S0957417403001180

Nouali O., Regnier A., Blache P., à paraître,
Classification de courriers électroniques : une approche par apprentissage basée sur des modèles linguistiques, Revue d'Intelligence Artificielle, Hermes Sciences Publications.

Nouali O., Blache P., à paraître,
Filtrage Automatique d'EMAILS : une approche adaptative et multi niveaux, Revue des Annales des télécommunications, Hermes Sciences Publications.

Publications Nationales avec comité de lecture

[1999] Nouali O., Azzouz R., Benyahia C.,
Un système multi agents pour le filtrage automatique du courrier électronique, publié dans la revue RIST, vol.9 N°01, ISSN 1111-0015, CERIST 1999.

[1999] Nouali O.,
Filtrage d'information,
publié dans la revue RIST, vol.7 N°02, ISSN 1111-0015, CERIST 1997.

Bibliographie

[ABN 96a] Abney S.,
Partial parsing via finite state cascades, *Actes de ESSLLI'96*, Robust parsing workshop, 1996.

[ABN 96b] Abney S.,
Statistical Methods and Linguistics, In J. Klavans, P. Resnik (eds.), *The Balancing Act: Combining Symbolic and Statistical Approaches to Language*, The MIT Press, Cambridge, MA, 1996.

[ABN 91] Abney S.,
Parsing by chunks, Principle-Based Parsing, Berwick R., Abney S., and Tenny C., eds., Dordrecht: Kluwer Academic Publishers, 1991.

[ABN 90] Abney S.,
Rapid Incremental Parsing with Repair, In Proceedings of the *6th New OED Conference*: Electronic Text Research, pp. 1-9, University of Waterloo, Waterloo, Ontario, 1990.

[ADA 85] Adam J. M.,
Quels types de textes ? *Le français dans le monde*, (192), 1985.

[AIT 97] Aït-Mokhtar S., Chanod J.-P.,
Xerox Incremental Parser (XIP), Proceedings of the *Fifth Conference on Applied Natural Language Processing*, 72-79, 1997.

[AMI 01] Amini M. R.,
Apprentissage automatique et recherche de l'information : application à l'extraction d'information de surface et au résumé de texte, *Thèse de doctorat,* Université de Paris 6, 2001.

[ANT 1999] Antoine J. Y., Genthial D.,
Méthodes hybrides issues du TALN et du TAL Parlé : état des lieux et perspectives, *Atelier Thématique TALN 1999*, Cargèse, 12-17 juillet 1999.

[APT 94] Apte C., Damerau F., Weiss S. M.,
Automated Learning of Decision Rules for Text Categorization, *ACM Transactions on Information Systems*, vol. 12, n° 3, pp. 233-251, 1994.

[ARC 02] Arcouteil A.,
Complémentarité de deux technologies pour l'extraction d'informations : apprentissage automatique et RDF, *CRIM Langues'O* – DESS Ingénierie Multilingue, 31 Janvier 2002.

[ASS 98] Assadi H.,
Construction d'onthologie à partir de textes techniques, Application aux systèmes documentaires, *Thèse de doctorat*, Université Pierre et Marie Curie (Paris IV), octobre 1998.

[BAC 92] Baclace, Paul E.,
Competitive Agents for Information Filtering, *Commun. ACM* 35, 12 December 1992.

[BAL 02] Balvet A.,
Approches Catégoriques et Non Catégoriques en Linguistique des corpus spécialisés : Application à un Système de Filtrage d'Information, *Thèse de doctorat*, Université Paris X-Nanterre, Décembre 2002.

[BAL 01a] Balvet A.,
Filtrage d'information par analyse partielle Grammaires locales, dictionnaires électronique et lexique grammaire pour la recherche d'information, *TALN 2001 Récital*, pp. 421-430, Tours, 2-5 Juillet 2001.

[BAL 01b] Balvet A., Bizouard S.,
Ressources linguistiques électroniques pour le filtrage d'information, *4èmes Journées INTEX*, Bordeaux, 2001.

[BAL 95] Balabanovic M., Shoham Y.,
Learnin Information retrieval agents: experiments with automated web browsing, Standford University, Department of Computer Science, 1995.

[BAU 70] Baum L. E., Petrie T., Soules G. and Weiss N.,
A maximization technique occuring in statistical analysis of probabilistic functions in Markov chains, *The Annal of Mathematical Statistics*, 41(1):164-171, 1970.

[BEL 92] Belkin N. J., Croft W. B.,
information filtering and information retrieval: two sides of the same coin? In *communication of the ACM*, vol. 35, N° 12, pp. 29-38, December 1992.

[BEL 01] Bellot P., El-Bèze M.,
Classification locale non supervisée pour la recherche documentaire, *Traitement Automatique des Langues, T.A.L.*, 2001, vol. 42, n° 2, Hermès, pp. 335-366, janvier 2001.

[BEL 00] Bellot P.,
Méthodes de classification et de segmentation locales non supervisées pour la recherche documentaire, *thèse de doctorat*, université d'avignon, 2000.

[BEN 02] BEN HAZEZ S.,
Modèles et langages pour la construction dynamique de systèmes de filtrage et de résumé de textes, *Le résumé automatique de texte, ATALA, 2002.*

[BER 92] Berry M. W.,
Large Scale Singular Value Computations, *International Journal of Supercomputers*, pp. 13-49, 1992.

[BER 02] Bernard P., Lecomte J., Dendien J., Pierrel J.M.,
Computerized linguistic resources of the research laboratory ATILF for lexical and textual analysis: Frantext, TLFi, and the software Stella, Proceedings of *the 3rd conference Language Resources and Evaluation Conference*, Las Palmas, 2002.

[BER 97] Cathy Berthouzoz C., Merlo P.,
Filtrage structurel versus Filtrage statistique : Expériences sur la résolution de l'ambiguïté syntaxique, *Actes des JST'97*, Avignon, France, 15-16 avril 1997.

[[BES 02] Besançon R., Rajmin M.,
Filtrage syntaxique de co-occurrences pour la représentation vectorielle de documents, *9ème Conférence Annuelle sur le Traitement Automatique des Langues Naturelles, TALN'2002*, tome 1, Nancy, France, 24-27 Juin 2002.

[BIB 95] Biber D.,
Dimensions of register variation : a cross-linguistic comparison, Cambridge University Press, Cambridge, 1995.

[BIB 93] Biber D.,
Using register-diversified corpora for general language studies, *Computational Linguistics*, vol. 19, N° 2, pp. 243-258, 1993.

[BIB 88] Biber D.,
Variation Across Speech and Writing, University Press, Cambridge, 1988.

[BLA 00a] Blache P.,
Le rôle des contraintes dans les théories linguistiques et leur intérêt pour l'analyse automatique : les Grammaires de Propriétés, in actes de *TALN-2000.*

[BLA 00b] Blache P.,
Contraintes et théories linguistiques : des Grammaires d'Unification aux Grammaires de Propriétés, *Habilitation à diriger des recherches*, 2000.

[BLA 95] Blache P.,
Une introduction à HPSG., 1995.
http://www.lpl.univ-aix.fr/~blache/publis.html

[BON 98] Boone G.,
Concept Features in RE: AGENT, an intelligent email agent, 1998.

[BON 84] BONNET A.,
l'intelligence artificielle promesses et realités, *inter-édition*, Paris, 1984.

[BON 80] BONNET A.,
Les grammaires sémantiques, outil puissant pour interroger les bases de données en langage naturel, *RAIRO informatique/computer science*, vol. 14, N° 2, pp. 137-148, 1980.

[BOO 73] Booth T.L., Thompson R.A.,
Applying probability measures to abstract languages, *IEEE Transactions on Computers*, C-22(5), pp.: 442-450, 1973.

[BOU 98] Bouillon P.,
Traitement automatique des langues naturelles, *Champs linguistiques Recueils, Editions Duculot,* 1998.

[BOU 97] Bouaud J., Habert B., Nazarenko A., Zweigenbaum P.,
Regroupements issus de dépendances syntaxiques en corpus : catégorisation et confrontation à deux modélisations conceptuelles, In *Actes des 1 Journées Ingénierie des Connaissances*, pp. 207-223, Roscoff, France, 20-22 mai, 1997.

[BOY 02] Boyé A., Comairas M.C.,
Moyenne, médiane, écart-type : Quelques regards sur l'histoire pour éclairer l'enseignement des statistiques. *Reperes-IREM*, 48, pp. 27-39, 2002.

[BRE 84] Breiman L., Friedman J. H., Olshen R. A., Stone C. J.,
Classification and Regression Trees, Belmont, CA, Wadsworth, 1984.

[BRE 82] Bresnan J., Kaplan R. M.,
Lexical-Functional Grammar: A formal system for grammatical representation, In *J. Bresnan, Ed.*, The Mental Representation of Grammatical Relations, chapter 4, pp. 173-281, Cambridge, Mass, MIT Press, 1982.

[BRI 95] Brill E.,
Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging, *Computational Linguistics*, 21(4):543-565, 1995.

[BRI 92a] Brill E.,
A Simple Rule-based Part of Speech Tagger, Proceedings of *the Third Conference on Applied Natural Language Processing*, ACL, pp.152-155, 1992.

[BRI 92b] Brill E., Marcus M.,
Automatically Acquiring Phrase Structure Using Distributional Analysis, *Proceedings of the DARPA Speech and Natural Language Workshop*, 1992.

[BRO 96a] Bronckart J.-P.,
Activité langagière, textes et discours: pour un interactionisme socio-discursif, *Sciences des discours, Delachaux et Niestlé*, 1996.

[BRO 96b] Bronckart J.-P.,
Genres de textes, types de discours et opérations discursives. *Enjeux*, (37-38), pp. 31-47, Namur, 1996.

[BRO 85] Bronckart J.-P., Bain D., Schneuwly B., Davaud C., Pasquier A.,
Le fonctionnement des discours : un modèle psychologique et une méthode d'analyse, Lausanne: Delachaux & Niestlé, 1985.

[BRU 02] Brun A., Smaili K., Haton J. P.,
WSIM: une méthode de detection de theme fondée sur la similarité entre mots, 9ème *Conférence Annuelle sur le Traitement Automatique des Langues Naturelles, TALN'2002*, tome 1, Nancy, France, 24-27 Juin 2002.

[BUC 92] Buckley C., Salton G., Allan J.,
Automatic Retrieval with locality information using SMART, In *TREC-1* Proceedings, pp. 69-72, 1992.

[CAR 01] Caropreso M., Matwin S., Sebastiani F.,
A learner-independant evaluation of the usefulnessof statistical phrases for automatic text categorization, pp. 78–102, Hershey, US. 2001.

[CHA 03] Chauché J., Prince V., Jaillet S., Teisseire M.,
Classification automatique de textes à partir de leur analyse syntaxico-sémantique, *TALN 2003*, Batz-sur-Mer, 11–14 juin 2003.

[CHA 98] Chandrasekar R., Srinivas B.,
GLEAN : using syntactic information in document filtering, In *Information Processing&Management* vol. 34, n° 5, pp. 623-640, 1998.

[CHA 94] Chanod J.P.,
Développements en analyse syntaxique automatique, actes *TALN'94*, Marseille, France, pp. : 87-92, 1994.

[CHA 97a] Charniak E.,
Statistical Parsing with a Context-free Grammar and Word Statistics, In Proceedings of the 14th National Conference on Artificial Intelligence AAAI97, Menlo Park, CA, 1997, AAAI Press.

[CHA 97b] Charniak E.,
Statistical techniques for natural language parsing, *AI Magazin,* 1997.

[CHA 93] Charniak E., Hendrickson C., Jacobson N., Perkowitz M.,
Equations for part-of-speech tagging, In Proceedings of *the eleventh national conference on artificial intelligence, the American Association for Artificial Intelligence, AAAI93,* pp. 84-89, Washington, D.C., July, 1993.

[CHO 57] CHOMSKY N.,
Syntactic Structures, *Mouton*, La Haye, 1957.

[CHO 89] Chow Y., Schwartz R.,
The N-best algorithm: an efficient procedure for finding top N-séquences hypotheses", proc. *ARPA Workshop on Speech and Natural Language*, pp. 199-202, 1989.

[CHO 57] CHOMSKY N.,
Syntactic structures, Mouton La haye, 1957.

[CHU 88] Church K.,
A stochastic parts program and noun phrase parser for unrestricted text, proc. 2nd Conference on Applied NLP, *ANLP'88*, Austin, Texas, pp.: 136-143, 1988.

[COH 96a] Cohen W.,
Learning trees and rules with set-valued features, In Proceedings of *the Thirteenth National Conference on Artificial Intelligence, AAAI96*, 1996.

[COH 96b] Cohen W.,
Learn rules that classify e-mail, In *AAAI Spring Symposium on Machine Learning in Information Access*, Stanford, March 25-27 1996.

[COH 96c] Cohen W., Singer Y.,
Context-sensitive learning methods for text categorization, In Proc. of *the 19th Annual International ACM/SIGIR Conference,* pp. 307-315, 1996.

[COL 96] Collins M. J.,
A New Statistical Parser Based on Bigram Lexical Dependencies, In Proceedings of *the 34th Annual Meeting of the ACL*, Santa Cruz, CA, June 1996.

[CON 03] Constant M.,
Grammaires locales pour l'analyse automatique de textes : méthodes de construction et outils de gestion, *thèse de doctorat*, Université de Marne-La-Vallée, Septembre 2003.

[COP 00] Copeck T., Barker K., Delisle S., Szpakowicz S.,
Automating the Measurement of Linguistic Features to Help Classify Texts as Technical, Conference *TALN2000*, Lausanne, 2000.

[COR 97] Coriat S., Pichon C., Mazeran V., Lequang P.,
Recherche Linguistique d'Information, Synthèse bibliographique, Février 1997.

[COU 86] Coulon A.,
Informatique et langage naturel, *téchnique et série en informatique*, vol. 5, N° 2 1986.

[CRI 99] Crispino G., Ben Hazez S., Minel J. L.,
ConTextO, un outil du projet FilText orienté vers le filtrage sémantique de textes, *VEXTAL'99*, Venezia, San Servolo, V.I.U, 1999.

[CRO 79] Croft W. B., Harper D. J.,
Using Probabilistic Models of Document Retrieval without relevance information, *Journal of documentation*, pp. 285-295, vol. 35, 1979.

[CYR 02] Cyr S., DeBlois L.,
Donner du sens à la notion de corrélation à partir des connaissances antérieures, *Actes du 45e congrès du GRMS*, 38-42, 2002.

[DAG 97] Dagan I., Karov Y., Roth D.,
Mistake-driven learning in text categorization. In Proceedings of *the EMNLP-97, 2nd Conference on Empirical Methods in Natural Language Processing*, pp. 55–63. 1997.

[DAI 01] Daille B., Romary L.,
Linguistique de corpus, *Traitement automatique des langues*, vol. 42, N° 2, 2001.

[DAV 93] Davalo E., Naim P.,
Des réseaux de neurones, *Edition Eyrolles*, 1993.

[DEE 90] Deerwester S., Dumais S.T., Furnas G.W., Landauer T. K., Harshman R.,
Indexing by latent semantic indexing, *Journal of the American Society for information Science* 41, 6, pp. 391-407, 1990.

[DEL 02] Delichère M., Memmi D.,
Analyse Factorielle Neuronale pour documents Textuels, *9ème Conférence Annuelle sur le Traitement Automatique des Langues Naturelles, TALN'2002*, tome 1, Nancy, France, 24-27 Juin 2002.

[DEN 00] Denis F., Gilleron R.,
Apprentissage à partir d'exemples, *Notes de cours*, Université Charles de Gaulle, Lille3, 14 Avril 2000.

[DEN 92] Denning P. J.,
Electronic Junk, *Communication Of The ACM*, March 1992.

[DES 93] Desclés J. P., Jouis C.,
L'exploration Contextuelle: une méthode linguistique et informatique pour l'analyse automatique de textes, *Actes ILN'93*, 1993.

[DRE 02] Dreyfus G., Martinez J.M., Samuelides M., Gordon M.B., Badran F.,
Thiria S., Hérault L.,
Réseaux de neurones : méthodologie et applications, *Eyrolles*, 2002.

[DRI 01] Drias H.,
La technologie des agents intelligents et la recherche d'information sur internet, *Séminaire international sur l'automatisation du trésor de la langue arabe*, Alger, Algérie, 03-05 Novembre 2001.

[DRU 99] Drucker H., Wu D., Vapnik V.,
Support vector machines for spam categorization, *IEEE Transactions on Neural Networks* 10 (5), pp.1048-1054, 1999.

[DUD 83] Duda R. O.,
Expert systems research, in *Science*, 1983, pp.: 220-261, 1983.

[DUM 98] Dumais S., Platt, Heckerman D., Sahami M.,
Inductive Learning Algorithms and Representations for Text Categorization, In proceedings of *the 7th International Conference on Information and Knowledge Management (CIKM98)*, pp. 148-155, 1998.

[FAI 98] Faiz R.,
Filtrage automatique des phrases temporelles d'un texte, In *RIFRA 98*, Rencontre Internationale sur l'Extraction, le Filtrage et le Résumé Automatique, pp 55-63, 1998.

[FAR 89] Fargues J.,
Des graphes pour coder le sens des phrases, pour la science, N° 137, Mars, 1989.

[FIL 68] Filmore C.,
The case for case, *Universals Linguistic Theory*, 1968.

[FER 84] Ferber J.,
La compréhension du langage naturel, Une affaire de syntaxe, des phrases pleines de sens, la structure du recit, *Micro-système*, sept, oct, nov. 1984.

[FOL 90] Foltz P. W.,
Using Latent Semantic Indexing for information filtering, In Proceedings of *the ACM Conference on Office Information Systems, ACM/SIGOIS*, New York, pp. 40-47, Avril 1990.

[FOL 92] Foltz P. W., Dumais S. T.,
Personalized information delivery: an analysis in information filtering methods, In *communication of the ACM* vol. 35, N° 12, pp. 51-60, December 1992.

[FUH 92] Fuhr N.,
Probabilistic models in information retrieval, *The computer journal,*vol. 35, N° 3, pp. 243-25, 1992.

[FUR 87] Furnas G. W., Landauer T. K., Gomez L. M., Dumais S. T.,
The vocabulary problem in human-system, In *communication of the ACM*, vol. 30, N° 11, pp. 964-971, 1987.

[GAR 98] Garcia D.,
Exploitation pour l'élaboration de requêtes de filtrage de texte, des connaissances causales détecté par COATIS, In *RIFRA'98 Rencontre internationale sur l'extraction, le filtrage et le résumé automatique*, pp 44-54, 1998.

[GAR 98] Garcia D., Aussenac-Gilles N., Courcelle A.,
Exploitation pour la modélisation, des connaissances causales détectées par COATIS dans des textes, In *Ingénierie des connaissances. Eds Kassel G., Charlet J., M. Zacklad M., Edition Eyrolles*, 1999.

[GOM 02] Gomaz Hidalgo J. M., Puertas Sanz E., Mana Lopez M. L.,
Evaluating Cost-Sensitive Unsolicited Bulk Email Categorization, *JADT2002, 6eme Journées internationales d'analyse statistique des données textuelles*, 2002.

[GRE 96] Grefenstette G.,
Evaluation techniques for automatic semantic extraction: Comparing syntactic and window based approaches, In Boguraev, B. and Pustejovsky, J., editors, *Corpus Processing for Lexical Acquisition*, Language, Speech and Communication, chapter 11, pp. 205–216, The MIT Press, Cambridge, Massachusetts, 1996.

[GUT 94] GUTHRIE L., WALKER E.,
Document classification by machine: theory and practice, in proccedings of *COLING 94, The 15th International Conference on Computational Linguistics*, Kyoto Japan, 1994.

[HAB 02] Habert B., Zweigenbaum P.,
Régler les règles, *TAL.,* Vol. 43, N° 3, pp. 83-105, 2002.

[HAB 01] Habert B.,
"Typologies de textes",
http://www.limsi.fr/Individu/habert/Publications/Fichiers/Fichiers/Perpignan00/index.html, 15/05/2001.

[HAB 00a] Habert B.,
Des corpus représentatifs : de quoi, pour quoi, comment ? *In: Linguistique sur corpus. Études et réflexions*, éd. par Bilger (Mireille), pp. 11-58, Perpignan, Presses Universitaires de Perpignan, 2000.

[HAB 00b] Habert B., Illouz G., Lafon P., Fleury S., Folch H., Heiden S., Prévost S.,
Profilage de textes : cadre de travail et expérience, *JADT 2000 : 5eme Journées Internationales d'Analyse Statistique des Données Textuelles*, 2000.

[HAB 97] Habert B., Nazarenko A., Salem A.,
Les linguistiques de corpus, *Armand Colin & Masson*, ISBN : 2-200-01775-8, Paris 1997.

[HAB 96] Habert B., Nazarenko A.,
La syntaxe comme marche-pied de l'acquisition des connaissances: Bilan critiques d'une expérience. Actes des *septièmes journées Aquisition des connaissances JAC'96*, pp. 137-148, 1996.

[HAR 91] Harris Z. S.,
A theory of language and information, *A mathematical approach*, Oxford University Press, Oxford, 1991.

[HAR 85] Harris M. D.,
Introduction to naturel language processing, Loyala university New Orleans, édition 1985.

[HAR 68] Harris Z.S.,
Mathematical Structures of Language, John Wiley and sons, New York, 1968.

[HAR 51] Harris Z.S.,
Methods in Structural Linguistics, The University of Chicago Press, Chicago, 1951.

[HAY 94] Haykin S.,
Neural Networks: A Comprehensive Foundation, *Mc Millan College Publishing Co.*, 1994.

[HEB 49] Hebb D. O.,
The Organization of Behavior, *John Wiley & Sons*, New York, 1949.

[HIN 83] Hindle D.,
User manual for Fidditch, a deterministic parser, Technical Report Memorandum 7590-142, Naval Research Laboratory, Washington, D.C., 1983.

[HOA 00] Hoashi K., Matsumoto K., Inoue N., Hashimoto K.,
Document Filtering Method Using Non-Relevant Information Profile, Proceedings of *ACM-SIGIR 2000*, pp. 176-183, 2000.

[HUL 94] Hull D.,
Information Retrieval using statistical classification, *Thèse de doctorat*, Université de Stanford, 1994.

[HUL 98] Hull D.,
The TREC-7 Filtering Track: Description and Analysis, dans les actes de *Text Retrieval Conference (TREC)*, University of Maryland, 1998.

[ILL 00] Illouz G., Habert B., Fleury S., Folch H., Heiden S., Lafon P., Prévost S.,
TyPTex: Generic Features for text profiler, Proceedings of the *RIAO-2000 Conference*, pp. 1526-1540, 2000.

[ILL 99] Illouz G., Habert B., Fleury S., Folch H., Heiden S., Lafon P.,
Maîtriser les déluges de données hétérogènes, *Atelier Thématique TALN 1999*, Cargèse, 12-17 Juillet 1999.

[IWA 95] Iwayama M., Tokunaga T.,
Cluster-based Text Categorization : a comparison of Category Search Strategies , Proceedings of the annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'95), pp. 273-280, 1995.

[JAC 98] Jackiewics A.,
L'expression de la causalité dans les textes, Contribution au filtrage sémantique par une méthode informatique d'exploration contextuelle, *thèse de doctorat*, Paris-Sorbonne , 1998.

[JAL 02] Jalam R., Chauchat J. H.,
Pourquoi les n-grammes permettent de classer des textes? Recherche de mots-clefs pertinents à l'aide des n-grammes caractéristiques, *JADT 2002, 6eme Journées internationales d'analyse statistique des données textuelles*, 2002.

[JIA 93] Jiang Z.,
Understanding information filtering and providing an information filtering model, *Master's thesis*, University of Missouri 1993

[JOA 97] Joachims T.,
A probabilistic analysis of the rocchio algorithm with tfidf for text categorization, In *Proceedings of ICML-97, 14th International Conference on Machine Learning*, pp. 143–151, 1997.

[JOA 98] Joachims T.,
Text categorization with support vector machines: learning with many relevant Features, In *Proceeding of ECML-99, 16th European Conference on Machine Learning*, pp. 137–142. 1998.

[JOH 94] John G., Kohavir, Pfleger K.,
Irrelevant features and the subset selection problem in machine learning, Proceedings of *the Eleventh international conference*, pp. 121-129, Morgan Kaufman Publishers, San Francisco, CA, 1994.

[JUN 97] Junker M., Abecker A.,
Exploiting Thesaurus Knowledge in Rule Induction for Text Classification, Proceedings of *the RANLP-97 Conference* , pp.202-207, 1997.

[KAY 01] Kayser D., Levrat B.,
Traitement automatique du langage naturel, *Technique et Science Informatiques, TSI, Hermes,* vol. 20, N° 3, 2001.

[KAY 86] Kayser D.,
Présentation générale des méthodes d'interprétation des textes écrits, Téchniques et sciences informatiques, vol. 5, N° 2, 1986.

[KES 96] Kessler B., Nunberg G., Schutze H.,
Automatic Detection of Text Genre,
ftp://parcftp.xerox.com/pub/qca/papers/genre

[KIL 97a] Kilander F., Fahraeus E., Palme J.,
Intelligent Information Filtering, *Technical report 97-002*, Dpt of Computer and Systems Sciences, Stockholm University, February 17, 1997.

[KIL 97b] Kilander F., Palme J.,
The private filtering news agent, Department of computer and system science, *Technical report 97-004*, Stockholm university,1997.

[KOH 84] Kohonen T.,
Self-Organization and Associative Memory, Springer-Verlag, New York, 1984.

[KOH 99] Kohrs A., Merialdo B.,
Clustering for collaborative filtering application, *actes de CIMCA*, 1999.

[KOS 01] Kosseim L., Lapalme G.,
Critères de selection d'une approche pour le suivi automatique du courriel, *TALN 2001Récital*, Tome 1, pp. 357-363Tours, 2-5 Juillet 2001, RALI, DIRO, Université de Montréal, 2001.

[KOS 98] Kosseim L., Lapalme G.,
EXIBUM: Un système experimental d'extraction bilingue, *Actes des rencontres internationales sur l'extraction, le filtrage et le résumé automatique (RIFRA'98)*, Sfax, Tunisie, pp. 129-140, 1998.

[KOS 91] Koskiennnemi K.,
Finite state parsing and disambiguation, proc. 13th Conference on Computational Linguistics, *COLING'90*, Helsinki, Finlande, vol. 2, pp.: 229-232, 1991.

[LAF 92] Lafferty J., Sleator D., Temperley D.,
Grammatical trigrams: a probabilistic model of link grammar, in *Goldman R. (Ed.) AAAI Fall symposium on probabilistic approaches to Natural Language Processing*, Cambride, Mass., AAAI Press, 1992.

[LAR 98] Larkey L. S.,
Automatic essay grading using text categorization techniques. In prpceedings of *SIGIR'98, 21st ACM international conference on research and development in information retrieval*, Melbourne, AU, pp. 90-95, 1998.

[LAR 98] Larson R., Hearst M.,
SIMS 202: Information Organization and Retreival (Lecture 17), University of California, Berkley School of Information Management and Systems, 1998.

[LAS 02] Laskri M. T., Meftouh K.,
Extraction automatique du sens d'une phrase en langue française par une approche neuronale, *JADT 2002, 6eme Journées internationales d'analyse statistique des données textuelles*, 2002.

[LAS 86] Lasguignes J.,
Approche des méthodes et problèmes posés par la compréhension automatique du langage naturel, CERFIA, *D.E.A informatique*, 1986.

[LEC 98] Leconte J.,
Le Catégoriseur Brill14-JL5/Winbrill-0.3, INALF/CNRS, Université de Pennsylvanie, Décembre 1998.

[LEW 92a] Lewis D.D., Tong R. M.,
Text Filtering in MUC-3 and MUC-4, Proceedings of *the* Fourth Message Understanding Conference (MUC-4), pp.51-66, 1992.

[LEW 92b] Lewis D.D.,
An evaluation of phrasal and clustered representations on a text categorization task, In proceedings of *SIGIR-92, 15th ACM International Conference on Research and Development in Information Retrieval?* Copenhagen, Denmark, pp: 35-50, 1992.

[LEW 94] Lewis, D.D., Ringuette M.,
Comparaison of two learning algorithms for text categorization, In proceedings of *the Third Annual Symposium on Document Analysis and Information Retrieval SDAIR'94*, 1994.

[LOE 92] Loeb S.,
Architecting personalized delivery of multimedia information, In *communication of the ACM*, vol. 35, N°12, pp. 39-48, December 1992.

[LUH 58] Luhn H. P.,
A business intelligent system, IBM Journal of Research and Development, October 1958.

[MAC 89] Mackay W.E., Malone T. W., Crowston K., RAO R., Rosenblitt D., Card S. K.,
How be experienced information lens user use rules? In proceeding of *the ACM CHI'91 Conference on Human Factors in Computing Systems, ACM/SIGCHI*, New York, pp. 211-216, 1989.

[MAE 94] Maes Pattie,
Agents that reduce Work and Information Overload, *Commun. ACM* 37, pp.31-40, 7 July 1994.

[MAN 99] Manning C. D., Schütze, H.,
Foundations of Statistical Natural Language Processing, Cambridge, Massachusetts: MIT Press, 1999.

[MAR 97] Marcu D.,
From discourse structures to text summaries, In *Workshop Intelligent Scalable Text Summarization, EACL 97*, Madrid, pp. 82-88, 1997.

[MAR 60] Maron M. E., Kuhns K. L.,
On Relevance probabilistic indexing and information retrieval, *Journal of the Association of Computing Machinery*, N°7, pp. 216-244, 1960.

[MAS 98] Masson N.,
Méthodes pour une génération variable de résumé automatique : Vers un système de réduction de textes, *Thèse de Doctorat*, Université Paris-11, 1998.

[MCC 98] Mc Callum A., Nigam K.,
A comparison of event models for naïve Bayes Text classification, In *learning for text categorization*, 1998.

[MCD 94] Mc Donough J., Ng K.,
Approaches to topic identification on the switchboard corpus, In *International Conference on Acoustics, Speech and Signal Processing*, pp. 385–388, Yokohama, Japan, 1994.

[MER 95] Merialdo B.,
Modèles Probabilistes et étiquetage automatique, *TAL*, vol.36, n° 1-2, pp. 7-22, 1995.

[MIC 99] Michel C.,
Evaluation de systèmes de recherche d'information, comportant une fonctionnalité de filtrage, par des mesures endogenes, *thèse de doctorat en sciences de l'information et de la communication,* Université Lumière LyonII, Janvier 1999.

[MIG 02] Miguel E. Ruiz, Padmini Srinivasan,
Hierarchical Text Categorization Using Neural Networks, *Kluwer Academic Publishers*, Netherlands, 2002.

[MIL 99] Miller D. T. H., Leek T., Schwartz R.M.,
BBN at TREC-7: using hidden Markov models for information retrieval, In proceedings of *the seventh Text Retrieval Conference, TREC7*, NIST special publications, 1999.

[MIL 90] Miller G,
WordNet: An On-line Lexical Database, *International Journal of Lexicography,* 1990.

[MIN 01] Minel J.L., Desclés J.P., Cartier E., Ben Hazez S., Crispino G., Jackiewicz A.,
Résumé automatique par filtrage sémantique d'informations dans des textes, *Technique et Science Informatiques,* n° 3, Paris, 2001.

[MIT 96] Mitchell T.,
Machine Learning, chapter 3, Mc Graw Hill, 1996.

[LEM 02] Le Moal J. C., Hidoine B., Calderan L.,
La recherche d'information sur les réseaux, cours INRIA, Collection Sciences de l'information, *ADBS Editions,* 2002.

[MUC-7] Proceeding *Seventh Message Understanding Conference*, 1998, http://www.muc.saic.com.

[MUC-6] Proceeding *Sixth Message Understanding Conference* (DARPA), Morgan Kaufmann Publishers, San Francisco, 1995.

[MUK 97] Mukhopadhyay S., Rajeev R. Raje, Boyles M., Patel N., D-SIFTER: A Collaborative Information Classifier, *International Conference on Information, Communications and Signal Processing ICICS'97*, Singapore, Septembre 1997.

[NAM 00] Namer F., FLEMM : Un analyseur flexionnel du français à base de règles, *TAL*, Vol. 41, n°2, décembre 2000.

[NG 97] Ng H.T., Goh W. B., Low K. L., Feature selection, perceptron learning and a usability case study for text categorization, In Nicholas Belkin, A. Desai Narasimhalu, andpetter Willett, editors, Proceedings of *the 20 th Annual International ACM SIGIR Conference on Research and Developpment in Information Retrieval*, Philadelphia, PA, pp 67-73, July 1997.

[OAR 99] Oard D. W., Adaptive Vector Space Text Filtering for Monolingual and Cross-language Applications, *PhD thesis*, University of Maryland, College Park 1999.

[PAS 96] Pascual E., Virbel J., Semantic and Layout Properties of Text Punctuation, *SIGPARSE 96 : ACL-96 Workshop on Punctuation in Computational Linguistics*, University of California, Santa Cruz, USA, 1996.

[PAU 95] Marie Paule, Péry Woodley, Quels Corpus pour quels traitements automatiques ? *TAL*, vol. 36, n° 1-2, pp. 213-232, 1995.

[PIL 00] Pillet V., Méthodologie d'extraction automatique d'information à partir de la littérature scientifique en vue d'alimenter un nouveau système d'information, *thèse de doctorat en sciences*, Aix-Marseille III, Janvier 2000.

[PIT 85] Pitrat J., textes, ordinateurs et compréhension, *EYROLLES,* 1985.

[POI 99a] Poibeau T., Nazarenko A., L'extraction d'information, une nouvelle conception de la compréhension de textes? *Traitement Automatique des Langues (TAL)*, 40(2), novembre 1999.

[POI 99b] Poibeau T., Evaluation des systèmes d'extraction d'information: une expérience sur le français, *Langues*, vol. 2, n°2, pp. 110—118, 1999.

[POL 88] Pollock S.,
A rule based filtering system, *ACM transactions on office information systems*, 1988.

[PON 98] Ponte J. M., Croft W. B.,
A language Modeling approach to information retrieval, In proceedings of *the 21 st ACM SIGIR conference on research and development in information retrieval*, pp. 275-281, 1998.

[POR 80] Porter M. F.,
An algorithm for suffix stripping, *Program*, vol. 14, pp. 130–137, 1980.

[QUI 93] Quinlan J. R.,
C4.5 : Programs for Machine Learning, Morgan Kaufman, 1993.

[QUI 89] Quinlan J. R., Rivest R. L.,
Inferring Decision Trees using the Minimum Description Length Principle, *Information and Computation* **80(3)**, pp. 227-248, 1989.

[RAB 89] Rabiner L. R.,
A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, Proceedings of the *IEEE*, 77(2):257-285, 1989.

[RAJ 97] Rajman M., Faltings B.,
A la poursuite de l'information : techniques de recherche et d'analyse pour données textuelles, EPFL-DI, Laboratoire d'intelligence artificielle, 1997,
http://sawww.epfl.ch/SIC/SA/publications/FI97/fi-sp-97/sp-97-page34.html

[REN 98] Rennie J.,
IFILE : An application of Machine Learning to Email,
http://www.cs.cmu.edu/~jr6b/papers/ifile98.html, 1998.

[RIL 99] Riloff E., Jones R.,
"Learning Dictionaries for Information Extraction by Multi-Level Bootstrapping", *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI-99)*, 1999, pp. 474-479.

[ROB 94] Robertson S. E., Walker S.,
Some simple effective approximation of the 2-Poisson model for Probabilistic weighted retrieval, In *ACM SIGIR*, pp. 232-241, 1994.

[ROB 76] Robertson S. E., Sparck-Jones K.,
Relevance Weighting of search terms, *Journal of American Society for information Science*, N°27, pp. 129-146, 1976.

[ROC 71] Rocchio J. J.,
The SMART Retrieval System: Experiments in Automatic Document Processing, chapter 14, *Relevance Feedback in Information Retrieval*, pp. 313–323. Gerard Salton (editor), Prentice-Hall Inc., New Jersey, 1971.

[ROC 97] Roche E., Schabes Y.,
Finite-state language processing, Cambridge, Massachusetts: The MIT Press, 1997.

[RUI 99] Ruiz M. E., Srinivasan P.,
Hierarchical neural networks for text categorization, In proceedings of *SIGIR-99, 22nd ACM international conference research and development in information retrieval*, Berkeley, US, pp. 281-282, 1999.

[SAL 88] Salton G., Buckley C.,
Term Weighting Approaches in Automatic Text Retrieval, *Information Precessing and Management*, vol. 24, N° 5, pp. 513-523, 1988.

[SAL 83] Salton G., Wong A., Yang C. S.,
A vector space model for automatic indexing, *Communications of the ACM*, 18 (11), 613-620, 1975.

[SAL 75] Salton G., Wong A., Yang C.,
A vector space model for information retrieval, *Communications of the ACM*, 18(11), 613–620, 1975.

[SCH 92] Schabes Y.,
Stochastic lexicalized tree-adjoining grammars, proc. of *the 14th International Conference on Computational Linguistics, COLING'92*, Nantes, France, 1992.

[SCH 70] Schank R.,
Identification of conceptualisation understanding natural language, *computer models of thought and language*, 1970.

[SCH 98] Schapire R. E., Singer Y., Singhal A.,
Boosting and Rocchio applied to text filtering, In proceedings of *SIGIR-98, 21st ACM International Conference on Researche and Development in Information Retrieval*, pp.215-22, Melbourne, Australia, 1998.

[SCH 94] Schmid H.,
Probablistic Part-of-Speech Tagging Using Decision Tree*s*, actes du *First International Conference on New Methods in Natural Language Processing (NemLap-94)*, Manchester, U.K., pp. 44-49, 1994.

[SCH 95] Schutz H., Hull D. A., Pedersen J. O.,
A comparison of classifiers and document representations fort he routing problem, In proceedings of *SIGIR'95, 18th ACM international conference on research and development in information retrieval*, Seattle, US, pp. 229-237, 1995.

[SEB 99a] Sebastiani F.,
A Tutorial on Automated Text Categorisation, Proceedings of *ASAI-99, 1st Argentinean Symposium on Artificial Intelligence,* 1999.

[SEB 99b] Sébillot P., Bouillon P., Fabre C., Jacqmin L.,
Apprentissage de ressources lexicales pour l'extension de requêtes, *TAL*, vol 0, n°0, pp.1-25, 1999.

[SEH 93] Sheth, Beerud, Maes, Pattie,
Evolving Agents for Personalized Information Filtering, In Procedings of *the Ninth IEEE Conference on Artificial Intelligence for applications*, pp. 1-7, 1993.

[SCH 94] Schmid H.,
Probabilistic Part of Speech Tagging using Decision Trees, *International Conference on New Methods in Language Processing, Manchester*, UK, 1994.

[SIN 96a] Singhal A., Buckley C., Mitra M.,
Pivoted document length normalization, In *ACM SIGIR*, pp. 21-29, 1996.

[SIN 96b] Sinclair J. McH.,
EAGLES: Preliminary Recommendations on Text Typology, Eagles Document EAG-TCWG-TTYP/P, Jun 1996.

[SMA 93] Smadja F.,
Retrieving collocations from text: Xtract, *Computational Linguistics*, 19(1):143-178., 1993.

[SPR 98] Spriet T., El-Beze M.,
Introduction of Rules into a Stochastic Approach for Language Modelling, *Computational Models of Speech Pattern Processing*, NATO ASI Series F, vol. 169, ed. Keith Ponting, pp. 350-355,1998.

[STA 86] Stanfill C., Waltz D.L.
Toward memory-based reasoning, in Journal of the Association for Computing Machinery, Vol. 29, N°12, pp. 1213-1228, 1986.

[TAK 97] Takkinen J.,
Towards a conceptual model for information management in electronic mail, *Thesis N°640, Linköping Studies in Science and Technology Sweden*, ISBN 91-7219-015-9, 1997.

[TAP 97] Tapanainen P., Järvinen T.,
A non-projective dependency parser, In Proceedings of *the 5th Conference on Applied Natural Language Processing*, Washington, DC, USA, pp. 64-71, April, 1997.

[TOD 01] Todirascu A.,
Semantic Indexing for Information Retrieval Systems, *Thèse*, Université Louis Pasteur, Strasbourg, Mars 2001.

[TREC-9] TREC-9, This report constitutes the proceedings of the *Ninth Text REtrieval Conference (TREC-9)* held in Gaithersburg, Maryland, November 13-16, 2000.

[TUR 91] Turtle H., Croft W. B.,
Efficient probabilistic inference for text retrieval, Proceedings of *RIAO* 3, 1991.

[TUR 00] Turenne N.,
Apprentissage statistique pour l'extraction de concepts à partir de textes. Applications au filtrage d'informations textuelles, *Thèse de doctorat*, Université de Louis-Pasteur, Strasbourg (ENSAIS), Novembre 2000.

[VAP 82] Vapnik V.,
Estimation of Dependences Based on Empirical Data, Springer Verlag, New York, 1982.

[VAP 95] Vapnik V.,
The Nature of Statistical learning theory, Springer-Verlag, New York, 1995.

[VIR 97] Virbel J.,
Aspects du contrôle des structures textuelles, In J. Lambert and J.-L. Nespoulous, éditeurs, *Perception auditive et compréhension du langage*, pp. 251-272, Solal, 1997.

[VIR 98] Virbel J.,
Symbolic Aspects of Layout Properties of Text Inscription, In M. Borillo, R. Pouivet, and J. Virbel, editors, *Esthétique Cognitive*, Hermès, 1998.

[VOO 98] VOORHEES E. M.,
Using WordNet for text retrieval, In C. FELLBAUM, Ed., *WordNet: an electronic lexical data*base*, Language, Speech and Communication,* chapter 12, pp. 285-303, Cambridge, Massachusetts: The MIT Press., 1998.

[VOO 94] Voorhees E. M.,
Query expansion using lexical-semantic relations, Research and Development on Information Retrieval, *ACM-SIGIR*, Dublin, 61-70., 1994.

[VOS 98] Vossen P.,
EuroWordNet: Construction d'une base de données multilingue organisée autour de réseaux de mots pour les langues européennes, *ELRA, la lettre d'information*, vol 3. n°1, pp. 7-10, Février 1998.

[WEI 90] Weiss S. M., Kulikowski C. A.,
Computer Systems That Learn, Morgan Kaufman, 1990.

[WID 60] Widrow G., Hoff M. E.,
Adaptive switching circuits, *1960 IRE WESCON Convention Record*, New York: IRE, 1960.

[WOO 75] Wood W.,
What's in a link: foundations for semantic networks, in *Brobrow & Collins*, pp. 35-82, 1975.

[YAN 00] Yan T.W., Garcia-Molina H.,
The SIFT Information Dissemination System, *ACM TODS*, 2000.

[YAN 93] Yan T. W., Garcia-Molina H.,
Index structures for selective dissemination of information under the Boolean mode, Department of Computer Science, Stanford University, Stanford, *CA 94305*, December 15, 1993.

[YAN 94] Yang Y., Chute C. G.,
An example-based mapping method for text categorization and retrieval, *ACM Transaction on Information Systems (TOIS)*, pp. 253-277, 1994.

[YAN 97] Yang Y., Pedersen J.O.,
A comparative Study on Feature Selection in Text Categorization, *International Conference on Machine Learning ICML 1997*, Nashville, TN, USA, pp. 412-420, 1997.

[YAN 99] Yang Y. Liu X.,
A re-examination of text categorization methods, In proceedings of *SIGIR'99, 22 nd ACM International Conference on Research and development in information retrieval*, Berkeley, US, pp. 42-49, 1999.

[ZAR 98] Zaragoza H., Gallinari P.,
Modèle Hiérarchique de Recherche et d'Extraction de l'Information Textuelle de Surface, Journées Francophones d'Apprentissage (JFA'98), Arras 1998.

[ZWE 03] Zweigenbaum P., Hadouche F., Grabar N.,
Apprentissage de relations morphologiques en corpus, TALN 2003, Batz-sur-Mer, 11–14 juin 2003.

Le jeu d'étiquettes de la version BRILL1.4-JL5 / WINBRILL-0.3

Etiquettes	Signification	Exemples
ABR	Abréviation	*ex. pp. chap.*
ADJ :sg	Adjectif (sauf Participe passé) au singulier	le *4ème*/ADJ:sg chapitre
ADJ :pl	Adjectif (sauf Participe Passé) au pluriel	des besoins *immédiats*/ADJ:pl
ADV	Adverbe	*ne*/ADV *jamais*/ADV
CAR	Cardinal (en chiffres ou en lettres)	*89*/CAR *cent*/CAR *mille*/CAR
COO	Coordonnant	*et, ou, ni, mais, or, car*
DTN :sg	Déterminant de groupe nominal, au singulier, non contracté	La/DTN:sg pédagogie/SBC:sg
DTN :pl	Déterminant de groupe nominal, au pluriel, non contracté	*les*/DTN:pl mineurs avec *leurs*/DTN:pl lampes
DTC :sg	Déterminant de groupe nominal, au singulier, contracté	*jusqu' au*/DTC:sg pavé
DTC :pl	Déterminant de groupe nominal, au pluriel, contracté	je t'apporte des/DTC:pl tomates du/DTC:sg jardin.
FGW	Mot étranger	*book*
INJ	Interjection, Onomatopée, etc.	*hélas (!), chut, oui, non, ben*
PFX	Préfixe détaché	mon ex/PFX - fiancé/SBC:sg
PREP	Préposition	*à*/PREP travers/SBC:sg
PRV :sg	Pronom « supporté » par le verbe (conjoint, clitique) au singulier	*je*/PRV:sg *vous*/PRV:pl *le*/PRV:sg conseille vivement
PRV :pl	Pronom « supporté » par le verbe (conjoint, clitique) au pluriel	*je*/PRV:sg *vous*/PRV:pl *le*/PRV:sg conseille vivement
PRV :++	Pronom « supporté » par le verbe (clitique, réfléchi) genre indéterminé	il faudra *s'*/PRV:++ habituer
PRO :sg	Autre Pronom, singulier	*toi*/PRO:sg qui parles si bien, qui/PRO:sg es - *tu*/PRV:sg ?
PRO :pl	Autre Pronom, pluriel	*les*/PRO:pl voici/PREP qui arrivent !
PRO :++	Autre Pronom, genre indéterminé	*j'*/PRV:sg en/PRO:++ vois *plusieurs*/PRO:pl, mais ne sais pas *qui*/PRO:sg viendra
PUL	Particule non indépendante	*quant*/PUL
REL	Relatif (Pronom, Adjectif ou Adverbe)	je le vois *qui*/REL vient.
SUB	Subordonnant	*parce_que, afin_que*
SUB$	Subordonnant possible. = Code par défaut de « que »	je tiens à ce *que*/SUB$ tu viennes
SBC :sg	Substantif, nom commun, singulier	il roule en *voiture*/SBC:sg
SBC :pl	Substantif, nom commun pluriel	dans les *bois*/SBC:pl
SBP :sg	Substantif, nom propre ou à majuscule, singulier	Amérique/SBP:sg
SBP :pl	Substantif, nom propre ou à majuscule, pluriel	
SYM	Symbole ou Signe mathématique	*° $ % + x*
ACJ :sg	Verbe « avoir », conjugué, singulier	elle l'*aura*/ACJ:sg *voulu*/VPAR:sg

ACJ :pl	Verbe « avoir », conjugué, pluriel	
ANCFF	Verbe « avoir », non conjugué, infinitif	sans *avoir*/ANCFF *pu*/VPAR:sg y *aller*/VNCFF
ANCNT	Verbe « avoir », non conjugué, gérondif ou participe présent	n' *ayant*/ANCNT pas voulu cela
APAR :sg	Verbe « avoir »,non conjugué, participe passé, singulier	elle a *eu*/APAR:sg faim
APAR :pl	Verbe « avoir »,non conjugué, participe passé, pluriel	
ECJ :sg	Verbe « être », conjugué, singulier	il es*t*/ECJ:sg *parti*
ECJ :pl	Verbe « être », conjugué, pluriel	ils *sont*/ECJ:pl *partis*/ADJ1PAR:pl
ENCFF	Verbe « être », non conjugué, infinitif	ne doit-elle pas être/ENCFF considérée
ENCNT	Verbe « être », non conjugué, gérondif ou participe présent	*étant*/ENCNT concerné par ce problème
EPAR :sg	Verbe « être », non conjugué, participe passé, singulier (pas de pluriel)	elle a *été*/EPAR:sg mangée
VCJ :sg	Autre Verbe, conjugué, singulier	Il indique /VCJ:sg
VCJ :pl	Autre Verbe, conjugué, pluriel	Ils indiquent /VCJ:pl
VNCFF	Autre Verbe, non conjugué, infinitif	sans *vouloir*/VNCFF *aller*/VNCFF le *dénoncer*/VNCFF
VNCNT	Autre Verbe, non conjugué, gérondif ou participe présent	ils restèrent muets , n' *osant*/VNCNT plus remuer
VPAR :sg	Autre Verbe, non conjugué, participe passé après « avoir », singulier	elle a *mangé*/VPAR:sg
VPAR :pl	Autre Verbe, non conjugué, participe passé après « avoir », pluriel	les gens que j' ai *vus*/VPAR:pl
ADJ1PAR :sg	Participe passé après « être », adjectival ou verbal, au singulier	elle était *fatiguée*/ ADJ1PAR:sg
ADJ1PAR :pl	Participe passé après « être », adjectival ou verbal, au pluriel	elles étaient *fatiguées*/ADJ1PAR:pl
ADJ2PAR :sg	Participe passé adjectival, singulier (non après auxiliaire)	il dormait *assis*/ADJ2PAR:sg
ADJ2PAR :pl	Participe passé adjectival, pluriel (non après auxiliaire)	il dormait *assis*/ADJ2PAR:sg

Liste de mots vides en langue Française

afin	certains	excepté	lors	ont	quels
ailleurs	chacun	flac	lorsque	être	quelqu'un
ah	chacune	floc	lui	ou	quelqu'une
ai	chaque	fors	là	ouais	quelque
aie	chez	foutre	lès	ouf	quelques
ainsi	combien	grâce	ma	ouille	qui
allô	comme	gué	mais	oust	quiconque
alors	concernant	ha	malgré	ouste	quoi
après	ces	hein	me	outre	quoique
attendant	cet	hem	même	ouïe	revoici
au	cette	hep	mêmes	où	revoilà
aucun	ceux	heu	merci	paf	sa
aucune	chez	hi	mes	par	sans
au-dessous	chiche	ho	mien	pardieu	sauf
au-dessus	chut	holà	mienne	parmi	se
Auprès	ci	hop	miennes	partant	selon
aussitôt	dans	hormis	miens	pas	seront
aujourd	de	hors	mince	passé	ses
auquel	dedans	hou	moi	patapouf	si
aussi	dehors	houp	moins	patatras	sien
autant	déjà	hue	moment	pendant	sienne
autour	delà	hum	mon	peuh	siennes
autre	depuis	hé	motus	peut	siens
autres	des	hélas	moyennant	pif	sinon
aux	desquelles	ici	même	pin	soi
auxquelles	desquels	il	mêmes	plein	soit
auxquels	dessus	ils	n	plouf	son
avant	dès	jadis	ne	plus	sont
avec	donc	je	ni	plusieurs	sous
avoir	dont	jusqu	non	pouf	stop
beaucoup	du	jusque	nos	pour	suivant
boum	duquel	l	notamment	pourquoi	suis
bravo	durant	la	notre	proche	sur
car	et	là	nous	près	ta
ce	elle	laquelle	néanmoins	puis	Tandis
ceci	elles	las	nôtre	puisque	tant
cela	en	le	nôtres	ô	te
celle	encore	lequel	nulle	qu	tes
celles	entre	les	nulles	quand	telle
celui	étant	lesquelles	oh	quant	telles
cependant	etc	lesquels	ohé	que	tien
certain	être	leur	ollé	quel	tienne
certaine	euh	leurs	olé	quelle	tiennes
certaines	eux	lez	on	quelles	tiens

toi
ton
tope
touchant
toujours
tous
tout
toute
toutes
très
trop
soin
tu
un
une
va
vers
via
vivat
vive
vivent
voici
voilà
vos
votre
vous
vu
vôtre
vôtres
y
zest
zeste
zou
zut
à
ça
échange
été

Liste de mots vides en langue Anglaise

as	associated	contains	five	hi
able	at	corresponding	followed	him
about	available	could	following	himself
above	away	couldn't	follows	his
according	awfully	course	for	hither
accordingly	be	currently	former	hopefully
across	became	definitely	formerly	how
actually	because	described	forth	howbeit
after	become	despite	four	however
afterwards	becomes	did	from	i'd
again	becoming	didn't	further	i'll
against	been	different	furthermore	i'm
all	before	do	get	i've
allow	beforehand	does	gets	ie
allows	behind	doesn't	getting	if
almost	being	doing	given	ignored
alone	believe	don't	gives	immediate
along	below	done	go	in
already	beside	down	goes	inasmuch
also	besides	downwards	going	inc
although	best	during	gone	indeed
always	better	each	got	indicate
am	between	edu	gotten	indicated
among	beyond	eight	greetings	indicates
amongst	both	either	had	inner
an	brief	else	hadn't	insofar
and	but	elsewhere	happens	instead
another	by	enough	hardly	into
any	came	entirely	has	inward
anybody	can	especially	hasn't	is
anyhow	can't	et	have	isn't
anyone	cannot	etc	haven't	it
anything	cant	even	having	it'd
anyway	cause	ever	he	it'll
anyways	causes	every	he's	it's
anywhere	certain	everybody	hello	its
apart	certainly	everyone	help	itself
appear	changes	everything	hence	just
appreciate	clearly	everywhere	her	keep
appropriate	come	ex	here	keeps
are	comes	exactly	here's	kept
aren't	concerning	example	hereafter	know
around	consequently	except	hereby	knows
as	consider	far	herein	known
aside	considering	few	hereupon	last
ask	contain	fifth	hers	lately
asking	containing	first	Herself	later

latter
latterly
least
less
lest
let
let's
like
liked
likely
little
look
looking
looks
ltd
mainly
many
may
maybe
me
mean
meanwhile
merely
might
more
moreover
most
mostly
much
must
my
myself
name
namely
nd
near
nearly
necessary
need
needs
neither
never
nevertheless
new
next
nine
no
nobody
non
none
noone
nor
normally
not
nothing
novel
now
nowhere
obviously
of
off
often
oh
ok
okay
old
on
once
one
ones
only
onto
or
other
others
otherwise
ought
our
ours
ourselves
out
outside
over
overall
own
particular
particularly
per
perhaps
placed
please
plus
possible
presumably
probably
provides
que
quite
rather
rd
re
really
reasonably
regarding
regardless
regards
relatively
respectively
right
said
same
saw
say
saying
says
second
secondly
see
seeing
seem
seemed
seeming
seems
seen
self
selves
sensible
sent
serious
seriously
seven
several
shall
she
should
shouldn't
since
six
so
some
somebody
somehow
someone
something
sometime
sometimes
somewhat
somewhere
soon
sorry
thats
the
their
theirs
them
themselves
then
thence
there
there'
specified
specify
specifying
still
sub
such
sup
sure
take
taken
tell
tends
th
than
thank
thanks
thanx
that
that's
thereafter
thereby
therefore
therein
thereupon
these
they
they'd
they'll
they're
they've
think
third
this
thorough
thoroughly
those
though
three
through
throughout
thru
thus
to
together
too
took
toward
towards
tried
tries
truly
try
trying
twice
two
un
under
unfortunately
unless
unlikely
until
unto
up
upon
us
use
used
useful
uses
using
usually
value
various
very
via
vs
want
wants
was
wasn't
way
we
we'd
we'll
we're
we've
welcome
well
went
were

weren't
what
what's
whatever
when
whence
whenever
where
where's
whereafter
whereas
whereby
wherein
whereupon
wherever
whether
which
while
whither
who
who's
whoever
whole
whom
whose
why
will
willing
wish
with
within
without
won't
wonder
would
would
wouldn't
yes
yet
you
you'd
you'll
you're
you've
your
yours
yourself
yourselves
zero

Structure Matérielle

Code	La propriété linguistique	Observations
PSM1	Titres, section, etc.	
PSM2	Introduction, conclusion	
PSM3	Images, dessins, etc.	
PSM4	Commentaires, legendes	
PSM5	Tirets au début de phrases (puces word), etc.	
PSM6	Indentation (tabulation)	
PSM7	Ligne séparant deux paragraphes	
PSM8	Souligné, italique, gras	
PSM9	Ponctuation « : »	
PSM10	P.S. à la fin	
PSM11	Tableaux	
PSM12	Police, taille, type	
PSM13	Couleur des caractères	
PSM14	Bordures	
PSM15	Colonnes	
PSM16	Cadres	
PSM17	Paragraphes	
PSM18	Type: text/html	
PSM19	Fichier attaché	
PSM20	La langue	FR/ANG
PSM21	Adresse URL	
PSM22	Nombre de destinataires	
PSM23	Auteur du texte	
PSM24	Longueur du texte	
PSM25	Longueur moyenne des phrases	
PSM26	Mots en majuscule	
PSM27	Caractères non alphanumériques	&, $, %, *, #, etc.
PSM28	Caractères numériques	
PSM29	Horaire d'envoi	Nuit / jour

Modèle énonciatif

Code	La propriété linguistique	Observations
PEN1	1ere pers du singulier : je, j', me, m', moi, mon, ma, mes, mien, miens, mienne, miennes, etc.	
PEN2	2eme pers du singulier : tu, t', te, toi, ton, ta, tes, tien, tiens, tienne, tiennes, etc.	
PEN3	1ere pers du pluriel : nous, notre nos, nôtre, nôtres, etc.	
PEN4	2eme pers du pluriel : vous, votre, votre, vôtre, vôtres, etc.	
PEN5	3 pers (singulier, pluriel) : il, elle, ils, elles, lui, son, sa, ses, leur, leurs, le, la, l', etc.	le, la, l' : ambigue
PEN6	Discours rapporté direct : ¨ ¨	Ne pas prendre en compte (supprimer du texte)
PEN7	Retours de courriers (réponses) : Présence de (Re:) dans subject et la présence du symbole > au début de la ligne, etc.	Ne pas prendre en compte (supprimer du texte)
PEN8	On	
PEN9	Enoncer : VCJ VNCFF ADJ2PAR, Admet*, admi*, dire, disons, dit, disent, déclar*, remarqu, protest*	

Modèle Structurel[1]

Pattern	Code	Catégorie	après virgule	autres	début de phrase
à cela s'ajoute qu	1	Addition	50	25	100
ainsi qu	1	addition	50	0	100
ajoutons qu	1	addition	50	0	100
aussi	1	addition	50	25	100
d'autre part	1	addition	50	25	100
de plus	1	addition	50	0	100
de surcroît	1	addition	50	25	100
et qui plus est	1	addition	50	0	100
plus	1	addition	50	0	100
quant à	1	addition	50	25	100
voire	1	addition	50	25	0
à l'instar	2	analogie	50	25	100
autrement dit	2	analogie	50	25	100
c'est ainsi qu	2	analogie	50	25	100
c'est-à-dire	2	analogie	50	25	100
comme	2	analogie	50	25	100
d'ailleurs	2	analogie	50	25	100
de la même façon	2	analogie	50	25	100
de manière (plus) générale	2	analogie	50	25	100
de même	2	analogie	50	0	100
de même qu	2	analogie	50	25	100
d'une manière approchante	2	analogie	50	25	100
d'une manière semblable	2	analogie	50	25	100
également	2	analogie	50	0	100
en comparaison de	2	analogie	50	25	100
similairement	2	analogie	50	25	100
à cette fin	3	but	50	25	100
dans cette optiqu	3	but	50	25	100
dans cette perspective	3	but	50	25	100
dans l'intention de	3	but	50	25	100
de sorte qu	3	but	50	25	100
en sorte qu	3	but	50	25	100
pour qu	3	but	50	25	100
afin de	4	cause	50	25	100
afin qu	4	cause	50	25	100
ainsi	4	cause	50	0	100
ceci fait qu	4	cause	50	25	100
c'est pourquoi	4	cause	50	25	100
de ce fait	4	cause	50	25	100
de crainte qu	4	cause	50	25	100

[1] La pondération des patterns est proposée par Alain Regnier, linguiste, laboratoire Parôle et Langage, Université de Provence, France.

de peur qu	4	cause	50	25	100
d'où	4	cause	50	0	100
en effet	4	cause	50	25	100
étant donné qu	4	cause	50	25	100
faute de	4	cause	50	0	100
finalement	4	cause	50	25	100
grâce à	4	cause	50	25	100
par conséquent	4	cause	50	25	100
par voie de conséquence	4	cause	50	25	100
parce qu	4	cause	50	25	100
pour cette raison	4	cause	50	25	100
pour toutes ces raisons	4	cause	50	25	100
puisqu	4	cause	50	25	100
sous l'effet de	4	cause	50	25	100
sous prétexte qu	4	cause	50	25	100
voilà pourquoi	4	cause	50	25	100
d'après	5	énonciatif	50	25	100
selon	5	énonciation	50	0	100
à savoir	6	exemple	50	0	100
par exemple	6	exemple	50	25	100
prenons le cas de	6	exemple	50	25	100
(plus) particulièrement	7	focus	50	25	100
(plus) précisément	7	focus	50	25	100
en particulier	7	focus	50	25	100
de deux choses l'une	8	hypothèse	50	25	100
partant	8	hypothèse	50	0	100
quel qu	8	hypothèse	50	25	100
quoi qu	8	hypothèse	50	25	100
quoi qu'il en soit	8	hypothèse	50	25	100
quoiqu	8	hypothèse	50	25	100
soit	8	hypothèse	50	0	100
à plus forte raison	9	hypothèse intensité	50	25	100
à ce point qu	9	intensité	50	25	100
assez	9	intensité	0	25	0
au même degré qu	9	intensité	50	25	100
au point qu	9	intensité	50	25	100
au total	9	intensité	50	0	100
autant	9	intensité	50	0	100
autant qu	9	intensité	50	0	100
d'autant plus qu	9	intensité	50	25	100
non moins qu	9	intensité	50	0	100
si bien qu	9	intensité	50	25	100
tellement qu	9	intensité	50	25	100
ce qui précède indique qu	10	méta	50	25	100
ce qui revient à dire qu	10	méta	50	25	100
ceci dit	10	méta	50	25	100
cela dit	10	méta	50	25	100
en d'autres termes	10	méta	50	25	100
en résumé	10	méta	50	25	100
en un mot	10	méta	50	25	100

pour conclure	10	méta	50	25	100
à condition de	11	modal	50	25	100
à la condition de	11	modal	50	25	100
à supposer qu	11	modal	50	25	100
dans la mesure où	11	modal	50	25	100
de toute manière	11	modal	50	25	100
en fait	11	modal	50	0	100
en réalité	11	modal	50	25	100
en tout état de cause	11	modal	50	25	100
il est vrai qu	11	modal	50	25	100
pourvu qu	11	modal	50	25	100
sans doute	11	modal	50	25	100
si	11	modal	50	0	100
mais parce qu	12	opp/cause	50	25	100
à l'inverse	13	opposition	50	25	100
à l'opposé	13	opposition	50	25	100
au contraire	13	opposition	50	25	100
bien qu	13	opposition	50	25	100
cependant	13	opposition	50	25	100
certes	13	opposition	50	25	100
contrairement à	13	opposition	50	25	100
contre	13	opposition	50	25	100
différemment de	13	opposition	50	0	100
en contrepoint	13	opposition	50	0	100
en dépit du fait qu	13	opposition	50	25	100
en revanche	13	opposition	50	25	100
inversement	13	opposition	50	25	100
mais	13	opposition	50	25	100
malgré	13	opposition	50	25	100
par contre	13	oppostion	50	25	100
par opposition	13	oppostion	50	25	100
dès lors	14	temps	50	25	100
du moment qu	14	temps	50	25	100
par suite	14	temps	50	0	100
tandis qu	14	temps	50	25	100
a fortiori	15		50	25	100
à l'exception de	15		50	25	100
à tout le moins	15		50	25	100
attendu qu	15		50	0	100
au demeurant	15		50	25	100
autrement qu	15		50	0	100
c'est qu	15		50	25	100
du moins	15		50	0	100
du reste	15		50	0	100
en ceci qu	15		50	25	100
en contrepartie	15		50	0	100
en définitive	15		50	25	100
en somme	15		50	25	100
encore qu	15		50	25	100
et même	15		50	0	100
étant entendu qu	15		50	25	100

excepté	15	50	25	100
hormis	15	50	25	100
même si	15	50	25	100
mieux encore	15	50	0	100
moins	15	50	0	100
moyennant quoi	15	50	25	100
néanmoins	15	50	25	100
non qu	15	50	0	100
nonobstant	15	50	25	100
notamment	15	50	25	100
on doit bien admettre qu	15	50	25	100
on notera qu	15	50	25	100
or	15	50	0	100
par ailleurs	15	50	25	100
par le fait qu	15	50	0	100
pour autant qu	15	50	25	100
pourtant	15	50	25	100
sans	15	50	0	100
sauf	15	50	0	100
somme toute	15	50	0	100
toujours est-il	15	50	25	100
tout au moins	15	50	0	100
tout bien considéré	15	50	25	100
tout compte fait	15	50	25	100
toutefois	15	50	25	100
vu qu	15	50	25	100

Modèle Syntaxique

propriété	**étiquettes de Brill**	**autres conditions**
nominalisation 1	SBC	finissant en tion(s) et age(s)
nominalisation 2	SBC	finissant en isme(s) en ité(s)
nominalisation 3	SBC	finissant en eur(s) et rice(s)
infinitifs	VNCFF ANCFF ENCFF	
Relatif sujet	REL	qui
Relatif objet	Rel	que
Relatives composées	PREP suivie de REL	
passé 1	ACJ	ai as a avons avez ont avais avait avions aviez avaient
passé 2	VCJ	ais ait ions iez aient
passé 3	VCJ	ai is ut umes utes urent irent
futur 1	VCJ	rai ras ra rons rez ront
futur 2	ACJ	rai ras ra rons rez ront
adverbe de temps	ADV	tôt tard après maintenant aujourd'hui demain hier bientôt d'abord ensuite récemment simultanément
adverbes de lieu	ADV	dessus dessous haut bas autour devant derrière dedans dehors localement près auprès
p présent	VNCNT	
P passé	VPAR EPAR APAR	
coordination	COO	et
négation synthétique		aucun(e)(s) nul(le)(s) ni non plus
négation analytique	ADV	pas
existentiel		il y a
syntagme prépositionnel	PREP	suivi de DTN
démonstratifs	DTN	ce cette ces
indéfinis	DTN	Quelque(s) certain(es) aucun(e) nul(les) chaque plusieurs
indéfinis anaphoriques	PRO	quelqu'un chacun quiconque rien personne nul
interrogation		Est ce que ?
subordination	SUB	
interjection	INJ	
abréviation	ABR	
Forme passive	ADJ1PAR	

Vocabulaire Lexical (*personnel*)

à bientot
à plus
à toute
a+
abscence
actuellement
adresse
age
aid
aidkoum
alaik
alaykoum
ami
amicalement
amities
apres-midi
as-tu
aurais-je
besoin
beuacoup
bises
bisous
bonheur
bonjour
bon
bonne
bon courage
bonne chance
boulot
bureau
bye
ca
cad
carte
cas
centre
c'est
chambre
charge
cher
cher collegue
choix
chose
cité
collegue
communication
compte
congé
conseil
contact
cordial
cordialement
cordiaux
côté
courage
courrier
cousine
début
décembre
defaut
demande
départ
dérangement
désolé
deuxiement
deviens-tu
dieu
dimanche
dis-moi
dommage
dossier
email
embrasse
endroit
enervé
enfants
entrée
envie
époque
espère
essaie
es-tu
ete
étudiants
excuse
fac
facon
fais-tu
famille
fax
fille
fin
fois
forme
froid
galère
garçons
genre
gens
grosso-modo
groupe
hamdoulillah
hasard
hauteur
hésite
heure
heureuse
inchallah
ingénieur
invitation
janvier
jeudi
jeunesse
job
joie
jour
journee
jours
kilos
labo
lettre
lieu
lundi
madame
mail
maison
majorité
Maladie
Maman
Manière
marche
mars
matin
medecin
merci
mercredi
message
midi
mi-temps
mois
moment
monde
monnaie
monsieur
moubarek
niveau
nouvelles
numéro
occasion
ok
papa
parceque
paris
part
pays
peine
père
période
personne
peux-tu
photo
place
plage
plaisir
plan
pluie
printemps
problème
promotion
propriétaire
puis-je
quartier
rapport
réalité
reception
recherche
rendez-vous
repas
reponse
retard
retour
réussite
saha
sais-tu
salam
salamat
salle
sallem
salut
salutations
salutations distinguees
samedi
santé
seconde
selam
semaine
septembre
service
situation
société
soin
soir
soleil
sortie
sport
studio
sujet
sup
super
taille
téléphone
temps
tête
tour
train
travail
université
vacances
vendredi
vêtement
veuillez
vie
villa
ville
visa
visite
voeux
voisins
voiture
voix
voyage
weekend
week-end

Vocabulaire Lexical (*professionnel*)

actes
appel
appel a communication
appel a proposition
appel a soumission
article
calendar
calendrier
call
call for papers
call for propositions
call for workshop
call for workshop proposals
camera ready copy
comite
comite de lecture
comite de programme
comite de redaction
comite d'organisation
committee
communication
conference
conference language
conference workshops
contributions
critere
critere de selection
criteria
date
date limite de soumission
date limite
deadline
deadline for paper submission
deadline for workshop proposals
Demonstration
Demo
editorial board
editorial committee
electronic submissions
final camera ready copy
format
forum
important dates
international forum
international program committee
journees d'etude
language
langue
method of submission
notification
notification of acceptance
notification of workshops
notification to authors
organization
organized by
organizing committee
panel
panel proposals
paper submission
paper submission form
paper
post-conference workshops
poster
pre-registration
proceedings
program
program chairs
program committee
proposals
proposals for workshops
referenced demos
registration
reviewers
reviewing
reviewing of papers
selection criteria
submission
submission deadline
submission form
submission of abstracts
submission of papers
submission procedure
submit
submitted papers
sujet
theme
topic
tutorial
workshop
workshop chair
workshop proposals
workshop submission

Vocabulaire (Spam)

ability	credibility	Heart	night	service
accept	credit	Help	note	sex
access	credit card	Helpful	nothing	sex
account	credit repair	Holiday	notice	sex life
accuracy	cum	Home	number	sex stories
accurate	customer	home business	nutrition	sexe
act	dancing	home worker	obesity	sexual
action	database	honey	obligation	sexual potency
address	day	honor	offer	sexuality
adult	deal	hope	offer valid	sexy
advertise	dear	horny	office	shipping
advertisement	decision	horny fulfill all your desires	online	shop
advertising	decline	hot	opportunity	shopping
advice	delay	hot	option	show
age	delete	hot	order	signal
aggressive	delivery	hot horny	order by phone	simple
always	demand	hottest	order form	site
amount	depression	house	order report	size
announce	desire	html	paradise	smoking
announcement	desires	human growth hormone	part	software
answer	disaster	hunt	participation	solicitation
anxiety	discover	idea	party	solution
anything	dollar	immediate	penis	someone
apology	dot	immediate release	people	something
appetite	download	immediately	period	spam
application	dreams	immigration	person	special bonus
apply	drive	improvement	personnel	spend
appreciation	drug	include	phone	sport
area	earn	income	phone number	stock
assistance	earth	inconvenience	photo	store
			photo	
assurance	easy	Increase	pornographique	story
attractive	economy	industry	picture	stress
automatically	effect	instruction	place	study
available	emotional	interest	please	submit
bank	employment	international	popularity	subscribe
beautiful	energy	internet	porn	successful
beginner	enjoy	internet business	porno	supply
benefit	entertainment	invest	post	support
bible	erotic	investigate	postal	surprise
black	everyone	investigation	powerful	system
black ink	everything	investment	price	take
body	excitement	investment report	privacy	take action
bonus	exclusive	investor	private	tax
box	exercise	invitation	prize	teach
brand	experience	invoices	problem	team

bread
building
bulk email
business
business opportunity
business plan
buy
california ibm
call
calorie
cancer
cannabis
capsule
card
carte virtuelle
case
cash
casino
cell
change
check
choice
cholesterol
christmas
city
click
client
clothes
color
come
comment
commercial
company
complete
concern
confidential
confirm
confirmation
congress
consult
consultants
consumer
contact
control
copy
cost
country
couple
court
extra
fact
family
fantasy
fax
fee
feedback
field
film x
finance
financial
financial future
financial news
fitness
form
fortune
free
free application
free erotic
free financial
free hardcore
free hardcore
acceess
free life time
free lotto
free membership
free porn access
free software
free website
freedom
freeinvestment
friend
fulfill
full
future
gain
gambling
game
girl
girl thing
give
god bless
good
government
grant
growth
guarantee
guide
half price
hardcore
issue
job
jody
joy
junk
kind
know
launch
law
legal
letter
life
life style
life time
like
limited
limited time
link
liquidity
live sex
location
look
loss
lot
lottery
love
low
low price
lowest
magic
maintenance
majorcredit
majority
market
marketing
medical
member
membership
memory
men's fitness
message
method
miracle
miss
money
money order
Muscle
Name
Natural
product
productivity
professional
profit
profitable
program
promise
promote
promotion
promotional
protein
provide
purchase
qualification
quality
question
rate
reason
receipt
receive
recommendation
record
refund
registration
remedy
remove
repair
reply
report
request
research
reservation
reserve
respect
responsibility
result
return
return address
review
risk
run
salary
sale
satisfaction
satisfaction
satisfy
search
security
select
tell
test
thank
ticket
time
time offer
title
today
toll
toll free
tombola
tool
touch
trading
training
transaction
transfer
travel
travel club
trouble
trust
unlimited
unsubscribe
urgency
urgent
user
vacation
video
view
virtual card
virus
visit
want
way
wealth
web
website
weight
weightloss
welcome
wife
win
winner
wish
woman
wonderful
work
world
www

crazy
hardcore
hardcore sex
health care
Need
network marketing
sell
sensation
xxx
xxx video

Système Expert

Le système expert est composé d'une base de connaissances, formée de règles et de faits, et d'un moteur d'inférences. Un système expert se caractérise par la souplesse (mise à jour facile de la base : ajout, suppression ou modification de règles) et la justification de ses actions (mécanisme de trace : évolution pas à pas).

A/ Les conditions

La règle dispose de deux types de conditions: conditions directes qui portent sur les différents champs du courrier et conditions indirectes qui portent sur des critères (faits) concernant l'état de l'utilisateur ou sur des faits qui peuvent être générés éventuellement durant l'exécution.

1. Conditions directes

Chaque champ constituant le mail peut subir une condition. La nature de celle-ci dépend de la nature du champ lui-même.

a)- Pour les champs FROM, TO et SUBJECT, la syntaxe de la condition accepte les opérateurs suivants :
- Le 'et' logique entre plusieurs conditions (' '),
- Le 'ou' logique entre plusieurs conditions (;),
- La négation d'une condition (!).
- 'String' : exprime l'équivalence exacte entre le 'String' spécifié par la règle et le champ du mail.
- *String : exige que le champ se termine par la chaîne 'String'.
- 'String*' : exige que le champ commence par la chaîne 'String'.
- *String* : exige que le champ contient la chaîne 'String'.

Exemples :

FROM='nom@hotmail.com' : le champ FROM du courrier doit être exactement identique à cette chaîne.
SUBJECT='* SPAM *' : le champ SUBJECT doit contenir le mot SPAM
FROM='*.dz' : tout ce qui provient de l'Algérie.

b)- Pour le champ DATE, la syntaxe accepte les opérateurs suivants : AVANT, APRES ou PENDANT.

2. Les conditions indirectes

Elles portent sur l'un des deux types de faits suivants :

a)- Faits sur l'état de l'utilisateur : occupé ou absent.

b)- Faits générés durant l'exécution : par exemple, le type du document (Spam, personnel, urgent, etc.).

B/ Les actions

Dans le cas où un courrier vérifie les conditions exigées par l'une des règles, le système expert effectue une action qui peut être l'un des deux types suivants:

1. Action intermédiaire (logique) en positionnant un des critères. Ce type d'action enrichit la base de faits (du courrier).

2. Action finale (physique) qui peut être:

a)- Sauvegarder : cette action caractérise la classification du courrier reçu après le filtrage en précisant l'un des profils déjà définis par l'utilisateur. Les courriers qui subissent l'action 'sauvegarder' sont associés au profil et ordonnés par ordre décroissant de leur degré de similarité. Cette action assure les deux types de classification:

- La classification par profil.

- La classification par importance (à l'intérieur de chaque profil).

b)-Signaler : ce type d'actions consiste à répondre à l'expéditeur (réponse automatique) ou à envoyer le courrier reçu (diffusion) à d'autres personnes (concernées par le message). Par exemple, l'utilisateur peut prévoir une règle qui énonce que durant son absence, chaque courrier reçu mérite une réponse à l'expéditeur en l'informant de son absence pour une certaine durée.

Physiquement le signal est un envoi de courrier électronique vers un ou plusieurs destinataires. Ce qui nécessite une interaction avec le réseau qui consiste à :

(1) Ouvrir une connexion avec le serveur du réseau spécifique à l'envoi de la messagerie appelé SMTP grâce à un socket.

(2) Transmettre au serveur le courrier à envoyer et la liste d'adresses des destinataires en utilisant le protocole SMTP associé au serveur.

(3) Le serveur transmettra un OK exprimant son accord et se chargera du reste de la tâche du signal.

c)- Déliter qui consiste à supprimer les courriers vérifiant une condition donnée.

Les actions contradictoires sont rejetées par le système.

C/ Les faits

A chaque traitement d'un nouveau courrier, la base de faits du système expert est initialisée par l'ensemble des champs extraits de ce courrier et les informations concernant l'état de l'utilisateur (occupé ou absent). Cette base peut être enrichie durant l'exécution par des faits (tels que courrier urgent, courrier personnel, courrier intéressant, etc.) qui seront éventuellement utilisés pour générer d'autres actions (actions physiques tels que la sauvegarde, le signal ou la suppression).

D/ Le moteur d'inférences

Un système expert utilise l'un des deux modes de fonctionnement suivants: le chaînage avant ou le chaînage arrière selon le besoin et le domaine d'application. Dans notre cas, à priori les caractéristiques d'un nouveau courrier et les actions à effectuer sont inconnues. Le moteur procède donc à partir de faits initiaux pour atteindre le but qui consiste à identifier ses caractéristiques et ses actions et agir en conséquence (chaînage avant), plutôt que de les deviner et d'essayer de les prouver (principe du chaînage arrière).
La vérification des règles par le moteur nécessite une à deux passes. Il suit le principe ou raisonnement suivant : tant qu'un nouveau fait est généré, faire une nouvelle passe dans l'espoir de rencontrer une règle qui a était pénalisée par l'absence de ce nouveau fait durant les passes précédentes.

L 'algorithme :

(1) - Chargement de la base de règles.
(2) - Pour chaque nouveau courrier reçu faire :
    Initialiser la base de faits
    Pour toutes les règles faire (1ere passe):
        REGLE. Verif =faux
        tester la validité de la règle
        SI OUI : REGLE.Verif= vrai.
        faire les actions correspondantes.
        si la règle génère un fait alors : enrichir la base des faits.
(3)- Si fait généré durant l'exécution?
    Pour toutes les règles non vérifiées faire (2eme passe):
        tester la validité de la règle
        SI OUI:
            REGLE.Verif= vrai.
            Faire les actions correspondantes.

En plus de la prise de décisions, le moteur d'inférences associe à chaque action une liste de tous les chemins de règles qui ont conduit à l'exécution de celle-ci afin de justifier ses actes auprès de l'utilisateur.
A la fin du traitement du courrier, le moteur l'archive avec les couples associés (actions, liste des chemins de règles vérifiées) pour donner à l'utilisateur la possibilité de le visualiser, connaître les actions qui lui sont soumises et les règles ayant contribué à sa sélection puis lui donner la possibilité d'une mise à jour de certaines règles et le lancement de l'apprentissage dans le but d'approcher au mieux ses besoins.

Liste des Figures

Liste des Tables

Liste des algorithmes

Glossaire

A

Analyse de textes: est définit comme étant le découpage en éléments essentiels, la détermination des rapports entre ces éléments, produire un schéma (une représentation) de l'ensemble et déduire de nouvelles informations des éléments identifiés.

Analyse lexicale : tout traitement de mots dans un texte. Il comprend par exemple la segmentation en mots d'une liste de caractères dans un texte électronique et l'attribution de catégorie lexicale à ces mots.

Analyse syntaxique : analyse de la structure d'une phrase.

Analyse de surface *(shallow parsing*) : analyse syntaxique minimale fondée sur des séquences d'étiquettes morpho-syntaxiques. À ce niveau, le système d'étiquetage n'a généralement pas accès aux informations de sous-catégorisation.

Analyse locale : analyse syntaxique minimale, fondée sur la description de séquences inférieures à la phrase. Ce type d'analyse est souvent réservé aux domaines spécialisés, dans lesquels la phraséologie est plus fixe que dans la langue générale. Ainsi, par exemple, l'expression des dates, ou d'un montant pour une transaction, peuvent être décrits par une grammaire dite locale.

Analyseurs déterministes : ne rendent qu'une seule analyse par énoncé. Ceci est rendu possible par l'utilisation d'heuristiques pour exclure certaines possibilités.

Analyseurs non-déterministes : génèrent plusieurs analyses par énoncé lorsqu'il y a de l'ambiguïté structurale.

Analyseur de dépendances fonctionnelles : extrait des relations fonctionnelles telles que le sujet et l'objet.

Analyseur hybride : utilise plusieurs approches souvent traditionnellement opposantes (comme celles à base de statistiques et celles à base de règles) afin d'obtenir une analyse de qualité.

Analyseur orthographique : segmente en mots une liste de caractères dans un texte.

Analyseur morphologique : associe une catégorie lexicale (et peut être aussi des propriétés) aux mots dans le texte. Plusieurs catégories lexicales peuvent être associées à un seul mot en cas d'ambiguïté. Dans ce cas, il faut utiliser un désambiguïseur afin de déterminer la bonne catégorie lexicale pour le mot dans la phrase en question.

Analyseur syntaxique partiel : outil qui détermine une partie de la structure syntaxique d'une phrase.

Analyseur robuste : analyseur qui est conçu afin de traiter un texte libre plutôt qu'un texte spécifique à un domaine particulier.

Analyseur symbolique : analyseur à base de règles.

Annoter : affecter aux mots des catégories de nature morphologique, syntaxique, sémantique, pragmatique, prosodique, etc.

Apprentissage automatique : paramétrage d'un système automatique par des données à partir desquelles le système induit des règles. Dans le cas d'un apprentissage supervisé, les données à traiter sont accompagnées de la réponse désirée, au cours de la phase de paramétrage. Dans le cas d'un apprentissage non supervisé, les règles induites le sont à partir des seules données fournies au système.

C

Catégorie lexicale : catégorie comme "nom", "verbe" ou "adjectif " etc.

Chomskiens : sont les linguistes qui suivent la théorie linguistique de Noam Chomsky, le linguiste le plus réputé du vingtième siècle. Il postule l'existence d'une faculté de langue innée et cherche à la décrire au biais de règles syntagmatiques et transformationnelles.

Corpus : ensemble de productions linguistiques (ex. : discours transcrit, textes) formant un échantillon d'une langue donnée. Les corpus peuvent être construits de façon à être le plus représentatifs de la langue étudiée, ils peuvent être considérés sous deux points de vue : en tant qu'échantillons, ou bien comme extraits d'une langue. Dans les expérimentations, on distingue généralement entre corpus d'entraînement et corpus d'apprentissage. Le corpus d'entraînement sert au paramétrage des systèmes, le corpus d'apprentissage sert à tester la validité des règles induites au cours de l'apprentissage ; il est constitué de données inconnues du système évalué.

Corpus "texte libre" : est un corpus général qui peut contenir des erreurs orthographiques, grammaticales ou des omissions. De plus, il est composé de textes ayant plusieurs styles et venant de plusieurs domaines.

Couverture linguistique : concerne le nombre et la variété de textes qu'un analyseur est capable d'analyser. Lorsque la couverture est étroite ou spécifique à un domaine, l'analyseur est incapable d'analyser d'autres types de textes.

D

Désambiguïseur de catégories : sélectionne une catégorie à partir de plusieurs possibilités en utilisant de l'information contextuelle : c'est à dire en se basant sur le(s) mot(s) qui précèdent et suivent le mot ambigu en question. Il vise à limiter le nombre d'hypothèses élaborées au cours d'une analyse automatique.

E

Etiqueteur : associe des informations à des mots. Le choix d'étiquettes (morphologiques, syntaxiques, sémantiques, grammaticales, sociologiques, pragmatiques, etc.) varie selon l'objectif recherché. Par exemple, l'étiqueteur peut associer la catégorie lexicale "Nom" à un mot de cette classe comme " Omar ". Il peut également rajouter aux mots des propriétés telles que "nom propre", "troisième personne du singulier", etc.

Etiquetage partiel / intégral : un étiquetage partiel est un étiquetage ou certains mots du texte ne sont pas étiquetés ou étiquetés d'une façon incomplète. Il peut s'agir d'un étiqueteur limité, qui ne peut traiter des mots inconnus (absents des dictionnaires) ou d'un étiqueteur partiel visé en tant que tel, n'étiquetant que les mots du texte jugés pertinents en ignorant le reste des mots.

Explosion combinatoire : concerne le nombre d'analyses (surtout accidentelles) qu'un analyseur peut générer pour une phrase à cause de l'ambiguïté lexicale et structurale. En dehors des mots grammaticaux, comme les articles et les prépositions, la plupart des mots de la langue sont ambigus à plusieurs niveaux (un même mot peut appartenir à plusieurs catégories lexicales telles que les noms ou les verbes, un même mot peut avoir plusieurs sens selon le contexte dans lequel il est exprimé, etc.). L'analyseur peut générer plusieurs résultats d'analyse pour une même phrase en fonction de la catégorie lexicale associée à chacun des mots. Ce problème existe également au niveau des syntagmes. Par exemple, dans la phrase "le garçon voit un homme avec le télescope", le syntagme prépositionnel peut s'attacher au syntagme nominal ou verbal.

Extraction d'information ***(information extractio*****n**) : activité de recherche d'information visant la mise à jour automatique de bases de données (relationnelles ou autres) à partir de textes en langage naturel. Ainsi, un système d'extraction d'information traitant des descriptions d'attentats (MUC-3, MUC-4), viserait à renseigner les champs « nombre de blessés », « localisation géographique », ou encore « type d'arme utilisé », d'un formulaire *(templat*e) fixe.

F

Filtrage d'information : sélection et acheminement de documents extraits d'un flux d'information textuelle (ex. : fil de dépêches journalistiques), sur la base d'une comparaison binaire (correspondance/non correspondance) entre le profil informatif de chaque document et celui du besoin en information exprimé par un ensemble d'utilisateurs. En filtrage d'information, seuls les documents pertinents sont acheminés vers les utilisateurs.

Filtre : dans le cadre d'un système de filtrage d'information, le filtre désigne un (ou des) sous-élément d'un profil d'utilisateur. Un filtre peut être constitué par une séquence d'expressions à rechercher dans les documents, ou une conjonction/disjonction/négation de ces expressions (opérateurs booléens).

G

Grammaire locale *(local grammar)* : grammaire généralement limitée à l'analyse d'éléments dont la productivité syntaxique est limitée. Ainsi, l'expression des dates, en français, peut être analysée par une grammaire locale. Il est possible d'imbriquer ou d'associer des grammaires locales afin d'étendre le degré de localité.

Grammaire de dépendances : est une grammaire syntaxique basée sur les relations (grammaticales, sémantiques, etc.), entre des paires de mots dans la phrase.

Grammaires symboliques : grammaires syntagmatiques et grammaires de dépendances.

H

Heuristique : est une règle basée sur une généralité. Par exemple, l'heuristique de la chaîne la plus longue est souvent utilisée afin de segmenter les syntagmes. Par exemple, il permet d'isoler VP [est bien arrivé] plutôt que SV [est] et SV [bien arrivé] dans la phrase "Il est bien arrivé".

I

Infométrie : analyse quantitative de l'information.

L

Langue configurationnelle : est une langue où le sens d'une phrase dépend de l'ordre des mots (ex : la langue anglaise).

Langue non-configurationnelle : est une langue d'ordre relativement libre qui utilise souvent la déclinaison afin de déterminer le sens (ex : la langue russe).

Linguistes génératifs : sont ceux qui se servent de la grammaire syntagmatique afin de décrire une langue. On "génère" les phrases grammaticales d'une langue à partir des règles de grammaire. Une phrase qui ne peut être générée par la grammaire générative est considérée comme agrammaticale.

M

Moteurs de recherche : sont des programmes informatiques qui permettent aux utilisateurs de faire des recherches sur les documents disponibles dans des sources de données (bases de données, Internet, Intranets particuliers, etc.).

MUC : conférence internationale d'évaluation de systèmes de compréhension automatique de messages en langue naturelle, organisée principalement par le DARPA et le NIST. Cette conférence est essentiellement consacrée aux systèmes d'extraction d'information.

P

Parseur : est un analyseur syntaxique ou structurel.

Précision ***(precisio*****n**) : taux de documents pertinents bien filtrés par un système de filtrage d'information, par rapport à l'effectif des documents filtrés par le système.

Profil utilisateur : modélisation des besoins en information d'un utilisateur donné. Le profil peut être basé sur une explicitation des besoins, ou représenté par l'ensemble des documents consultés et validés.

Profilage de textes : évaluer l'hétérogénéité des données et identifier des parties homogènes (sous corpus). Il s'agit d'un bilan quantitatif fondé sur des indices linguistiques (vocabulaire, catégories et patrons morpho-syntaxiques, syntaxiques, sémantiques, structurels, etc.) pour regrouper des parties d'un grand corpus hétérogène en sous groupes homogènes.

R

Règles de transformation : permettent de dériver la structure de surface d'une phrase à partir de sa structure profonde. Par exemple, la règle de transformation passive sert à expliquer la différence entre une phrase déclarative et la phrase passive correspondante. La forme passive est obtenue à partir de la forme déclarative. Selon la règle, on change le sujet et l'objet de place dans la phrase, on rajoute la préposition "par" devant le sujet et on applique la forme passive au verbe (**" Jean aime Marie ", " Marie est aimée *par* Jean ").**

Relation binaire : est une relation entre une paire de mots dans une phrase.

Rappel ***(recall*****)** : taux de documents pertinents bien filtrés par un système de filtrage d'information par rapport à l'effectif de référence.

Recherche d'information ***(information retrieval*****)** : activité visant à (re)trouver et présenter l'information pertinente à chaque utilisateur des systèmes de recherche d'information. La recherche d'information peut être mise en oeuvre de façon manuelle, semi-automatique (interactive), ou complètement automatique.

Routage d'information *(routing)* : sélection et acheminement de documents extraits d'un flux d'information textuelle (ex. : fil de dépêches journalistiques). L'ensemble des documents sont évalués, en terme de pertinence, par rapport à un besoin en information donné. En routage d'information, l'ensemble des documents traités sont ordonnés en fonction de leur score de pertinence et acheminés vers les utilisateurs.

S

Segment : est un groupe de mots qui sont liés les uns aux autres.

Segmenteur : découpe le texte en constituants : mots, phrases, paragraphes.

Sémantique : est le sens d'un mot, d'un syntagme ou d'une phrase etc.

Syntagme : est une notion linguistique qui désigne l'enchaînement des mots dans la phrase et leur mode d'organisation. Un syntagme consiste en un groupe de mots centré autour d'une tête ou base. Par exemple, "L'homme" constitue un syntagme nominal composé d'un déterminant "Le" et d'une tête "homme".

T

Texte : signifie enchaînement d'idées et suite de mots. Il correspond à la fois à une forme (i.e. un ensemble de *graphies*) et à un contenu (i.e. les *idées* exprimées par le texte).

Texte libre : est un texte quelconque que l'on peut trouver en format électronique, par exemple, sur l'Internet. Il n'est pas limité à un domaine et n'appartient pas à un genre spécifique. Il peut contenir des fautes d'orthographe, de grammaire, des omissions ou de nouveaux mots inventés par l'auteur.

Texte annoté : chaque mot orthographique est étiqueté avec sa catégorie grammaticale (nom, verbe, etc.).

Tokenisation : segmentation en mots d'une liste de caractères dans un texte électronique.

TREC : conférence internationale d'évaluation de systèmes de fouille de textes *(text retrieva*l). Cette conférence reprend le fonctionnement de MUC, elle est consacrée à différentes activités de RI, de l'indexation des documents à l'interrogation vocale de bases de données, en passant par le filtrage d'information. Elle a donné lieu à la diffusion de variantes des moteurs d'indexation et de recherche PRISE et SMART pour l'ensemble des tâches de fouille de textes.

W

WinBrill : version Windows du Catégoriseur de Brill.