Apprentissage statistique pour l'extraction de concepts à partir de textes : application au filtrage d'informations textuelles

Nicolas Turenne

UNIVERSITE LOUIS-PASTEUR STRASBOURG
U.F.R. Mathématiques-Informatique
Ecole Nationale Supérieure des Arts et Industries de Strasbourg (ENSAIS)

N° d'ordre : 3632

THESE

présentée

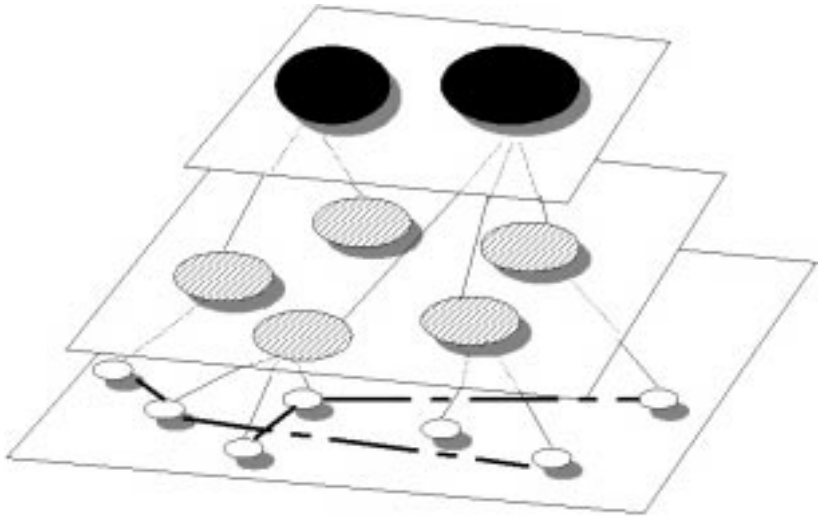
Par **Nicolas TURENNE**

pour l'obtention du grade de
DOCTEUR DE L'UNIVERSITE LOUIS-PASTEUR

Mention *: Sciences*
Spécialité *: Informatique*

Titre :

Apprentissage statistique pour l'extraction de concepts à partir de textes. Application au filtrage d'informations textuelles.



Soutenue publiquement le 24 Novembre 2000,
devant la commission d'examen composée de:

M. Prof.Dr.Fionn	MURTAGH	Rapporteur interne	ULP, Strasbourg / Queens'College, Belfast
M. Prof.Dr.Bernard	KEITH	Directeur de thèse	ENSAIS, Strasbourg
M. Prof.Dr.Yves	KODRATOFF	Rapporteur externe	Univ. Orsay, Paris
M. Dr Martin	RAJMAN	Rapporteur externe	Ecole Polytech. Féd., Lausanne
M. Dr François	ROUSSELOT	Co-Directeur de thèse	Univ. Marc-Bloch, Strasbourg

RESUME

Cette thèse présente un modèle de construction automatique et approximatif de la représentation du sens d'un texte. On adapte des techniques de documentation automatique à des bases documentaires non indexées. Les techniques classiques reposent sur une indexation vectorielle. Chaque document est représenté par un descripteur, on définit une distance entre ces descripteurs. L'accès aux documents pertinents est basé sur des calculs de proximité entre ces descripteurs. Une structuration du domaine, couvert par des documents, est obtenue par une classification (en anglais « clustering ») faisant apparaître des thèmes sémantiques. Il faut améliorer les techniques en leur permettant de traiter les documents non indexés, en améliorant les résultats par une adaptation de connaissances linguistiques et une analyse des relations que marquent les cooccurrences entre termes.

La quantité grandissante d'informations électroniques permet de constituer des échantillons de données variés et significatifs. Les techniques pour décrire les relations entre termes sont issues de méthodes mathématiques usuellement appliquées aux données structurées non textuelles. Le couplage de connaissances propres aux données avec une méthodologie adaptée aux données textuelles devrait apporter une amélioration des résultats. Nous tentons de justifier: d'une part l'utilisation de mécanismes linguistiques réduisant les biais d'une statistique descriptive des occurrences d'un terme, d'autre part l'utilisation d'une méthode basée sur les graphes dont les motifs permettraient de récupérer les relations conceptuelles entre termes. Dans un troisième temps nous facilitons l'interprétation de résultats émanant de traitements automatiques par la qualification consensuelle du thème représenté par une classe. L'interprétation de classes reste difficile, due aux multiples points de vue qu'un lecteur peut se faire des associations entre termes. Des classes de meilleure qualité facilitent l'interprétation, assistée par un thésaurus, que l'on peut attribuer à la structuration conceptuelle des termes d'un domaine.

Le développement d'Internet renforce l'échange de documents électroniques entre les acteurs de différents sites. Le développement de systèmes logiciels d'échanges de documents appelés « workflow » dans les intranets d'entreprise augmente la fluidité des documents entre individus et entre services. Un système qui permet d'apprendre automatiquement des profils d'utilisateur et d'exploiter ces connaissances pour distribuer l'information semble incontournable. Nous essayons de caractériser un centre d'intérêt par des classes de termes.

DISCIPLINE : Informatique, Intelligence Artificielle.

MOTS-CLES : Terminologie, Intelligence Artificielle, Traitement de Corpus, Lexicométrie, Schémas Morphosyntaxiques, Motif de Graphe, Extraction Semi-automatique de Concepts, Classification de Termes, Filtrage de Document, Apprentissage Automatique, Profil Utilisateur, Analyse Statistique des Données, Recherche d'Information.

INTITULE ET ADRESSE DU LABORATOIRE : Laboratoire LIIA, ENSAIS, 24 Bd de la victoire 67000 Strasbourg - France.

TABLE DES MATIERES

CHAPITRE 5
SYSTEME DE FILTRAGE INTELLIGENT DE MESSAGES ELECTRONIQUES. 165

ABREVIATIONS ET SIGLES

IA	Intelligence Artificielle
IC	Ingénierie des Connaissances
BCT	Base de Connaissances Terminologique
TALN	Traitement Automatique des Langues Naturelles
TAL	Traitement Automatique des Langues
GN	Groupe Nominal
SN	Syntagme Nominal
N	Nom
PREP	Préposition
ADJ	Adjectif
Angl.	le terme anglais correspondant

NOTATIONS MATHEMATIQUES

$\cup$	union d'ensembles
$\cap$	intersection d'ensembles
$\subset$	inclusion
$\forall x$	quelque soit x
$\exists x$	il existe au moins un x
$\varnothing$	ensemble vide
$\sum$	somme
$\in$	appartient à
$\notin$	n'appartient pas à
$\wedge$	et logique
$\vee$	ou logique
$\|A\|$	cardinalié d'un ensemble
Π	produit
$p \Rightarrow q$	inférence logique (p implique q)
$p \Leftrightarrow q$	p et q sont équivalents
$f:A \rightarrow B$	fonction f de valeurs de A dans B
n!	factorielle de n
max *f*	valeur maximale de *f*
min *f*	valeur minimale de *f*
P(A|B)	probabilité d'avoir A sachant B
d(x,y)	distance algébrique entre x et y
$\vec{x}.\vec{y}$	produit scalaire
$\vec{x}$, **x**	vecteur à valeurs réelles
x	scalaire
$\|x\|$	valeur absolue de x
I(x;y)	information mutuelle
D(x||y)	distance de Kullback-Leibler
O(n)	complexité d'un algorithme
M^{-1}	matrice inverse de M
M^{T}	matrice M transposée
m_{ij}	élément de la ième ligne et de la jième colonne de la matrice M
var(X)	variance de X
E(X)	espérance de X
cov(x,y)	covariance entre x et y
μ	moyenne
ε	erreur
σ	déviation standard
exp(x),e^{x}	fonction exponentielle
log *a*	logarithme de a
$\int$	intégrale
∂	dérivée partielle
card(x)	nombre d'éléments appartenant à x

LISTE DES FIGURES

INTRODUCTION

Acquisition des connaissances: l'intelligence artificielle au rendez-vous

Dans le cadre de cette thèse, nous abordons un problème difficile et souvent délaissé, celui de la classification automatique. La classification automatique est l'enfant mal aimée des mathématiques modernes et de la statistique parce que peu axiomatisée et peu analytique. Notre bibliographie montre que 40% des sources datent de moins de 5 ans et 60% de moins de 10 ans. Cela signifie 2 choses:

- la classification est une science étudiée depuis longue date et fut la cause d'innombrables travaux théoriques et applicatifs. Elle est encore le foyer de journaux spécialisés et de communautés ardentes à travers le monde.
- les études actuelles démontrent encore l'intérêt actif de la classification automatique, la possibilité de l'appliquer et de l'améliorer.

Difficile de prendre la relève d'un passé si chargé. Des ouvertures sont toutefois possibles en intelligence artificielle et notamment en acquisition des connaissances.

L'intelligence artificielle, science de la modélisation des connaissances, s'est enrichie d'une nouvelle méthodologie avec l'apprentissage automatique, science, entre autre, de l'induction logique. Pour se différencier des méthodes numériques traditionnelles, l'apprentissage, avec une pratique aiguë de la logique, s'est efforcé d'établir un raisonnement en terme symbolique. Les descriptions symboliques, alliées aux opérations de la logique, permettent d'avoir une bonne explicabilité des relations entre les objets. Cette description reste peu efficace lorsqu'elle est appliquée à la réalité pleine de relations et d'objets mal définis. C'est pourquoi un courant est né de l'apprentissage visant à réunir les approches numériques de l'analyse des données classiques et de l'apprentissage symbolique: l'induction symbolique et numérique à partir de données. Le profit mutuel permet de gagner en efficacité grâce aux méthodes numériques et en explicabilité grâce aux méthodes symboliques [Kodratoff, 1991].

C'est dans ce cadre de *méthode numérique* de classification et *d'explicabilité des objets classés* que nous positionnons notre projet de thèse.

Nous ne serions pas les pionniers dans l'art de manipuler la classification avec des attributs symboliques. En quoi pouvons-nous nous différencier des travaux antérieurs, ou comment une étude actuelle sur la classification peut se différencier des travaux antérieurs? Reprenons l'historique de la classification: le XVIIIième siècle a vu renaître la notion de classification des objets naturels avec les biologistes, le XIXième siècle a vu se développer la notion de taxinomie numérique toujours avec les biologistes apportant une rationalisation de la similarité par des coefficients numériques. Avec l'avènement de l'informatique la décennie 50-60 a développé les théories de la classification automatique. La décennie 60-70 a approfondi les théories avec toutes les variantes algorithmiques possibles. La décennies 70-80 a permis de nuancer les méthodes de classification automatique et de les d'appliquer à tous les domaines (finance, archéologie, linguistique, image, biologie, texte…). La décennies 80-90 a apporté la puissance des machines capables d'appliquer et d'expérimenter les algorithmes sur des données massives.
Alors que reste-t-il à innover ? une chose, ce qu'on pourrait appeler la rétro-ingeniérie c'est-à-dire la correction théorique des algorithmes grâce à l'application de ces mêmes algorithmes sur des grosses quantités de données. Les approches standard consistent à poser une méthode

générique et à l'appliquer à des données en l'adaptant au format des données, ou à appliquer ces mêmes méthodes à de petites quantités de données.
La création manuelle d'un thésaurus, entre 1994 et 1996, m'a permis de travailler sur la conception d'un thésaurus de terminologie scientifique (annexe 6). Cela m'a conduit finalement à réfléchir sur la classification automatique de termes et son influence sur l'analyse de contenu thématique.

Nous opérons une approche ascendante consistant à partir des données textuelles pour inférer une méthodologie existante appropriée et l'adapter le plus possible à la structure des données.

Dans ce projet de thèse, nous visons à décrire un domaine à l'aide d'une collection de textes appelée corpus. L'étude des textes est tout aussi ancienne que la classification automatique. 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. On distingue actuellement 2 écoles d'analyse textuelle de contenu: apprentissage statistique et grammaires locales. L'apprentissage statistique vise à utiliser un corpus pour acquérir des connaissances sur les relations entre plusieurs termes en étudiant leurs contextes de cooccurrence. Les grammaires locales essaient d'étudier le rôle syntaxique des constituants de chaque phrase pour étiqueter les corpus qui seront exploités dans l'étude approfondie des contextes d'usage d'un terme en vue, par exemple, de désambiguer son sens.

Partant d'un corpus textuel quelconque, nous allons essayer de tirer partie de la combinaison d'un traitement statistique des associations et de connaissances linguistiques sur les contextes morphosyntaxique et sémantique.

Partitionnement de graphe: une classification adaptée aux textes

Nous trouvons 2 écoles en Analyse des Données: les méthodes hiérarchiques et les méthodes de partitionnement. Les méthodes hiérarchiques visent à élaborer une hiérarchie par un calcul de similarité de chaque paire d'objets et ensuite entre classes pour aboutir à une classe unique.
Les méthodes de partitionnement en général posent le nombre de classes k comme paramètre connu et décident de l'appariement des objets en fonction des k classes.
Les méthodes hiérarchiques sont basées uniquement sur des calculs de similarité avec un seuil fixé arbitrairement. Les méthodes de partitionnement ont le défaut, pour la plupart de fixer un nombre de classes à l'avance.

Nous posons comme hypothèses de travail et de choix de notre méthodologie:

- de ne pas tenir compte de seuil de similarité
  conséquence: exclusion des méthodes hiérarchiques;
- de ne pas fixer le nombre de classes
  conséquence: recherche automatique des classes;
- d'avoir des objets multiclassés
  conséquences: recouvrement des classes et exclusion des méthodes de sériation;
- d'avoir une classification incomplète
  conséquence: autoriser les objets inclassables;
- d'avoir un traitement rétroactif du résultat de la classification

conséquences: exclusion des boîtes noires (réseaux de neurone et analyse factorielle).

Ces hypothèses de méthodologie nous amènent à considérer de plus près les méthodes d'extraction de graphe et des k-moyennes (angl. k-means).

L'analyse de données à partir d'un corpus médical orientera notre méthodologie vers la conception d'un motif de graphe pour l'induction de classes à partir de l'hypergraphe résultant d'une matrice de cooccurrence.

Consensus et thésaurus à 3 niveaux: une sémantique simple et efficace

Les connaissances disponibles dans un corpus (format textuel) sont des connaissances explicites mais impliquent des connaissances implicites. En IA les connaissances doivent être déclarées pour appuyer les traitements inductifs. Une solution possible est une ressource sémantique externe au corpus. Des ressources telles que Wordnet 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 nœuds parcourus entre un terme et un autre. Il faut nécessairement pondérer les relations entre noeuds et fixer un seuil de distance ce qui contredit une de nos hypothèses précédentes. Nous choisissons une structure plus simple qui doit être suffisamment proche de notre objectif de partitionnement: la hiérarchie de classes de termes sur 3 niveaux. La considération des termes à retrouver nous impose un niveau d'instance. Sans généralisation ce niveau ne constituerait qu'une liste linéaire. Un niveau de généralisation partitionne ces termes en catégories. Ce niveau peut être considéré comme ayant une granularité (spécificité) assez importante. Pour tenir compte d'une granularité moins fine on considère également un niveau encore supérieur qui constituera finalement le troisième et dernier niveau. Nos considérations ne demandent pas de structures plus complexes pour les traitements bien que celles-ci soient envisageables.

Ce type de structure sémantique des classes de termes à 3 niveaux de hiérarchie sera utilisé pour caractériser la cohésion lexicale des classes de termes obtenue par notre classifieur.

Filtrage d'information : une affaire d'événement et de contenu

La classification de termes, largement utilisée et testée dans des applications de recherche d'information, a été décriée comme trop grossière et inefficace. Nous pensons qu'il n'y pas d'alternative à *l'analyse locale* induisant une "structure logique de prédicat" et une *analyse globale* induisant une "structure de paquet de mots du 1er ordre". La première forme de structure bien que séduisante s'apparente aux conditions nécessaires et suffisantes d'un concept. Cette forme stricte est difficile à retrouver dans un texte tant l'expressivité de la formulation du langage naturel est variée. La deuxième forme en paquet de mots doit s'avérer puissante si le bruit dans les paquets est réduit à son minimum et le nombre de paquets couvre suffisamment les données sources. Ces paquets de mots, assimilables à des unités de connaissances, signifient en même temps: une sélection des formes significatives (terminologie) et une partition de regroupements significatifs (thématique).
Notre application intègre le processus de classification de termes pour guider un cas spécifique de recherche documentaire: le filtrage d'information. Les systèmes d'information voient de plus en plus l'émergence de données textuelles dynamiques dont le contenu est très rarement analysé. On ne doit pas écarter le fait que l'analyse de contenu ne soit pas la seule forme logique d'analyse d'un flux de document puisque des événements tels que: les métadonnées (date de réception, expéditeur,…), le comportement de l'utilisateur avec le

message (temps de lecture, destruction, routage vers un autre utilisateur, mise en priorité,…) ne sont pas négligeables mais ne concernent pas le projet présenté dans cette thèse.

Notre approche de traitement de l'information dynamique est orientée vers l'apprentissage d'un corpus de données reçues. Ce type d'approche peut s'avérer d'un intérêt majeur pour la mise en oeuvre d'un système de filtrage efficace basé sur le contenu.

Résumé des étapes de notre méthodologie

Le résultat de notre méthodologie est une suite de champs associatifs que l'on qualifie de concepts ou champ sémantiques, explicables par des étiquettes de catégories et des attributs.

On pose les hypothèses suivantes :

hypothèse 1: utilisation d'un corpus dont le contenu est représentatif d'un domaine spécifique et adapté au traitement d'une statistique relationnelle.

hypothèse 2: extraction des termes susceptibles de caractériser le domaine concerné par le corpus.

hypothèse 3: analyse des cooccurrences exprimant significativement la contextualisation du sens.

hypothèse 4: réalisation d'une classification en adoptant une représentation de type tableau de contingence pour aboutir à une classification automatique efficace.

hypothèse 5: possibilité d'attribuer des étiquettes aux classes et au corpus grâce à une ressource sémantique.

hypothèse 6: possibilité de réaliser un filtrage de document performant en exploitant les classes de termes.

L'originalité de notre approche repose sur les 10 points suivants:

- Le couplage de considérations numérique et symbolique pour les données textuelles;
- L'utilisation d'un modèle de classification basé sur l'extraction d'un motif de graphe;
- L'étude de phénomènes linguistiques utiles en classification tels que la réduction des formes linguistiques en formes canoniques;
- L'intégration de type comme attributs symboliques, un cas: le verbe;
- La mise en œuvre d'une méthodologie de classification monothétique[1] basée sur la notion de terme pôle;
- La mise en oeuvre d'une méthodologie de classification qui peut intégrer de nouvelles heuristiques, un cas: l'équivalence distributionnelle;
- La mise en valeur d'une structure sémantique simple et facilement exploitable construite sous forme hiérarchique à 3 niveaux;
- La définition d'une notion de consensus pour caractériser la cohésion lexicale d'une classe de termes;
- L'exploitation d'un thésaurus pour étiqueter des classes de termes;
- L'exploitation de la classification automatique de termes dans un processus dynamique de recherche documentaire, un cas: le filtrage automatique d'information textuelle.

Notre méthodologie d'acquisition est résumée par les étapes suivantes :

Données

a- Un corpus de texte libre.

Pré-traitement

[1] Par classe *polythétique*, il faut entendre une classification où les classes sont formées à partir du comportement "voisin" ou ressemblant des individus sur plusieurs caractéristiques simultanément, par opposition à la notion de classe *monothétique* qui utilise de proche en proche un comportement identique des individus sur une caractéristique à la fois.

b- Extraction des syntagmes nominaux ;
c- Lemmatisation des termes et filtrage des termes fréquents;
d- Transformation du corpus en fichier de position des mots uniques lemmatisés;
e- Extraction d'un fichier d'attributs (verbes).

Traitement

f- Extraction d'un ensemble de termes pôles ;
g- Extraction de la totalité des cliques d'ordre-3 pour chaque terme pôle;
h- Agrégation des cliques-3 en clique-4;
i- Agrégation des cliques-4 en classes grâce à 2 termes pivots communs aux clique-4 agrégées;
j- Intégration de termes dont la distribution des hapax[2] est similaire aux termes pôle de la classe.

Etiquetage

k- Utilisation du thesaurus pour attribuer une étiquette à chaque classe ainsi qu'au corpus.

Plan des chapitres

Chaque chapitre est précédé d'une introduction, qui expose le fil conducteur du chapitre, et se termine par un résumé. Cela permet au lecteur d'avoir une vue globale du contenu de chaque chapitre.

Chapitre 1
Dans ce chapitre, nous allons décrire notre problématique de recherche avec en premier lieu une présentation du contexte.

Chapitre 2
Dans ce chapitre nous présentons les méthodes qui vont nous permettre de dégager des champs sémantiques à partir d'un corpus.

Chapitre 3
Dans ce chapitre nous exposons la méthode de classification en 2 phases: la première phase consiste à extraire les unités qui seront classées par la suite, c'est la phase de collecte des individus et des variables qui doit être satisfaisante pour déboucher sur une classification interprétable. La deuxième phase consiste à classer les unités linguistiques extraites qui sont données en fichier d'entrée au classifieur. L'implémentation algorithmique est détaillée à travers le module GALEX.

Chapitre 4
Dans ce chapitre nous présentons la notion de cohésion lexicale d'une classe de termes et, à travers cette notion, l'évaluation d'une classe par rapport au domaine propre au corpus traité.

Chapitre 5
Dans ce chapitre, nous allons décrire certaines applications qui exploitent des classes dans un processus de traitement de l'information. Nous nous focalisons plus particulièrement sur une application de filtrage d'information ENAÏM avec l'implémentation et l'évaluation de celle-ci.

[2] Un *hapax* est un mot de fréquence 1 dans un texte.

CHAPITRE 1
ANALYSE DE CORPUS, APPRENTISSAGE STATISTIQUE ET FILTRAGE DE DOCUMENTS

Dans ce chapitre, nous allons décrire notre problématique de recherche avec en premier lieu une présentation du contexte.

L'information textuelle prend de plus en plus d'importance dans l'activité quotidienne des chercheurs et des entreprises. Même si cela l'était déjà auparavant la quantité pose des problèmes d'accès et de recherche d'information. La question se pose de structurer une base documentaire et d'exploiter cette structuration avec un maximum de rendement. L'acquisition des connaissances est un moyen privilégié pour tenter de localiser les problèmes d'extraction de connaissances. Mais l'étude des textes requiert une bonne compréhension des mécanismes linguistiques que l'on peut identifier et analyser automatiquement: il s'agit du traitement automatique du langage naturel. Les textes accumulés formeront les données sources à traiter pour tenter de dégager des champs sémantiques par des méthodes statistiques: il s'agit de l'apprentissage statistique. Ces domaines ont souvent évolué de façon autonome, nous présentons leur recouvrement, leur intérêt et leurs limites respectifs.

Dans une premierère partie nous allons exposer les champs d'études dont les méthodologies un intérêt technique pour résoudre un problème de classification à partir de textes à multiple facettes. Nous mettons en jeu des méthodes dites faibles (angl., "knowledge poor") qui demandent des connaissances très limitées voire inexistantes dans certains processus. Ces méthodes ne dépendent que des occurrences des mots rencontrés dans un texte par opposition aux méthodes dites fortes (angl., "knowledge rich") qui font appel à des ressources sémantiques extérieures.

Dans une seconde partie nous discutons l'intérêt d'intégrer une méthodologie de classification automatique dans une application faisant partie d'un système d'information. La quantité d'information ne garantit pas en soi la variété et la qualité. L'intensification des communications ne rend pas non plus nécessairement plus apte à prendre une décision ou à influer sur un processus décisionnel. Un système d'information doit clairement établir la nature du besoin utilisateur en information et une architecture qui vise à produire une recherche d'information appropriée et efficace.

Finalement nous présentons les hypothèses que nous mettons en oeuvre pour développer un système de recherche d'information à partir d'une étude statistique et linguistique du contenu d'une base de documents dans le cadre d'une fonctionnalité particulière d'un système d'information: le filtrage d'information.

1. Contexte de l'acquisition des connaissances à partir de textes

Dans cette partie nous allons présenter les différentes disciplines qui sont nécessaires d'aborder pour atteindre notre objectif. En particulier ce sont l'acquisition des connaissances, l'apprentissage automatique, l'analyse des données et le traitement automatique du langage naturel qui seront décrits.

La démocratisation de l'informatique dans le monde des particuliers, des entreprises et des administrations a permis de créer des volumes conséquents de documents électroniques rédigés en langue naturelle. Il est difficile d'imaginer les quantités de données textuelles crées chaque mois dans les entreprises, ou la quantité de publications scientifiques dans chaque domaine chaque année. Cela a fait naître des besoins d'accès intelligents à cette information textuelle. Le développement parallèle à grande échelle des accès réseaux Internet/Intranet a propulsé ces besoins en nécessité incontournable. D'autant plus que les efforts de normalisation des échanges d'information, de la gestion des fichiers, des formats et du matériel garantit une fluidité de communication. Comment dégager de cette masse d'information: des régularités, des relations sémantiquement contraintes et des ensembles de documents utiles ? C'est à ce niveau que se positionne notre problématique d'extraction de connaissances. La richesse d'information inhérente à de larges collections de textes est virtuellement inexploitée. Les textes encodent cette information dans une forme très difficile à automatiser [Hearst, 1994]. Notre problématique nous oblige à nous placer au confluent de plusieurs disciplines variées et de plus en plus recoupées depuis quelques années (figure 1.1).

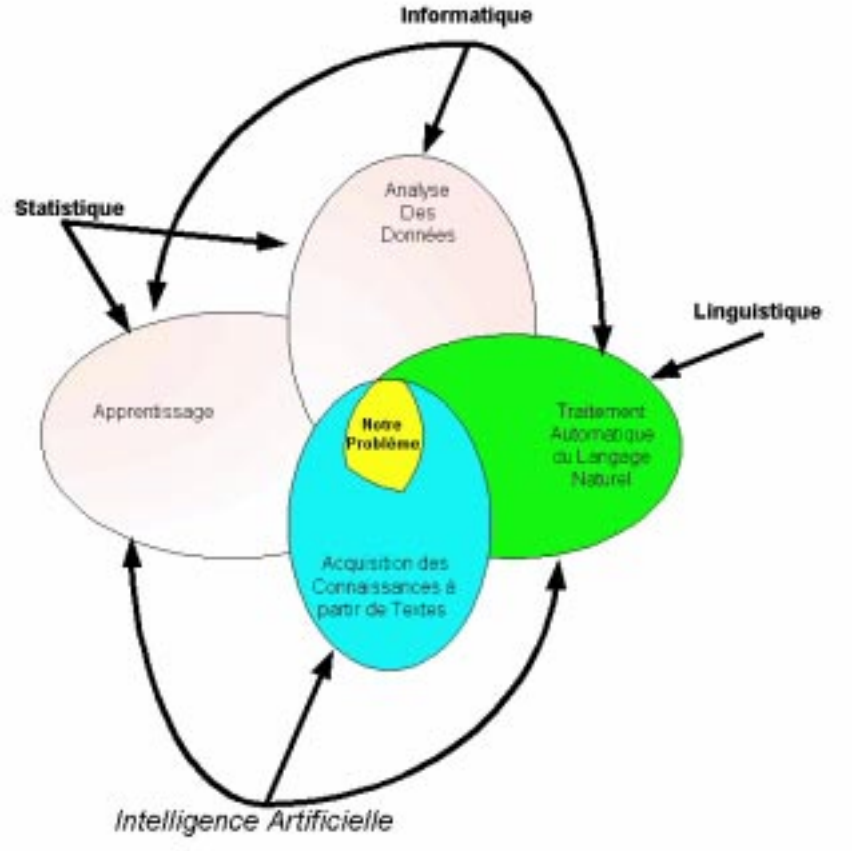


Figure 1.1 Position de notre problème

1.1. Acquisition des connaissances

Le processus d'acquisition des connaissances consiste à traiter des connaissances de bas niveau (données) pour aboutir à des connaissances de haut niveau (structure). (figure 1.2)

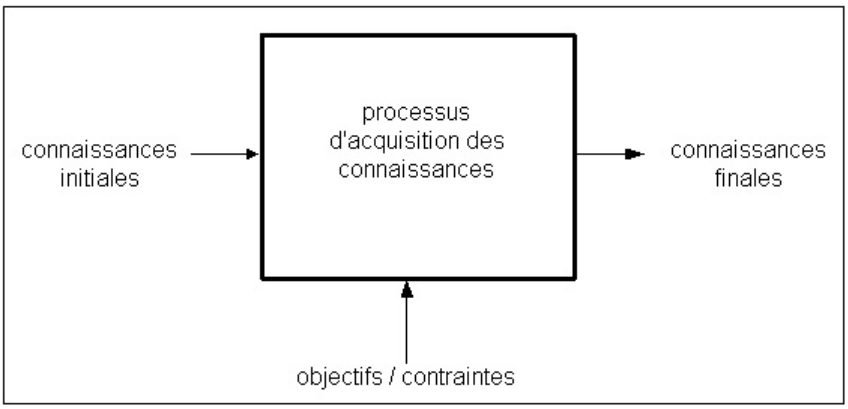


Figure 1.2 Processus d'acquisition des connaissances

Le choix des connaissances ne s'impose pas mais résulte d'une collecte soumise à une série de contraintes ayant une influence fondamentale sur les résultats. La connaissance initiale regroupe l'ensemble des faits et des relations nécessaires à la résolution d'un problème. La connaissance initiale comporte 2 parties :

○ les données, caractérisées par des objets et des variables

● Les variables manipulées peuvent être qualitatives ou quantitatives. Les variables quantitatives sont des variables dont les valeurs appartiennent à un ensemble totalement ordonné, infini et assimilable à R^+.

● Les variables qualitatives sont des variables dont les valeurs appartiennent à un domaine fini ou dénombrable. Ses éléments sont appelés modalités. Leurs valeurs sont purement conventionnelles. Elles peuvent être :

- monovaluées : chaque objet prend une seule valeur par variable (exemple: couleur="bleu")
- multivaluées : chaque objet peut prendre plusieurs valeurs simultanées (exemple: avion: altitude=10 latitude=100 longitude= -15)

○ les connaissances supplémentaires

● informations liées au domaine (expert…)
● informations liées au but (critère à optimiser..)
● informations sur les données :
- relations d'ordre (à côté, au dessus…)
- taxinomies (arborescences…)
- pondérations (rôle modulé de certaines variables…)

1.2. Apprentissage automatique

1.2.1 Définition

L'apprentissage automatique est apparu grâce à la conception du développement de l'enfant, du développement de l'expertise et de l'acquisition des connaissances par instruction ou découverte. Plusieurs tâches sont visées:
- apprendre à reconnaître
- catégoriser
- devenir plus efficace

- acquérir des connaissances

Ces tâches sont réalisées à travers certaines dimensions dans le traitement des connaissances: la compression, la prédiction (on teste l'apprenant dans cette dimension qui valide la performance de l'apprentissage), la généralisation, la compréhension et la résolution de problème inverse (contrainte sur une fonction f pour résoudre x, $y \rightarrow y=f(x)$).

Définition du problème d'apprentissage : acquérir de meilleures ou de nouvelles connaissances et/ou un mécanisme ou une procédure (moteur d'inférence et connaissance).
On caractérise un processus d'apprentissage par une interaction entre l'apprenant et l'environnement (figure 1.3), le critère de succès (ou de performance) pouvant être un écart à un modèle attendu, l'espace d'entrée et l'espace des fonctions cibles (dépendances fonctionnelles).

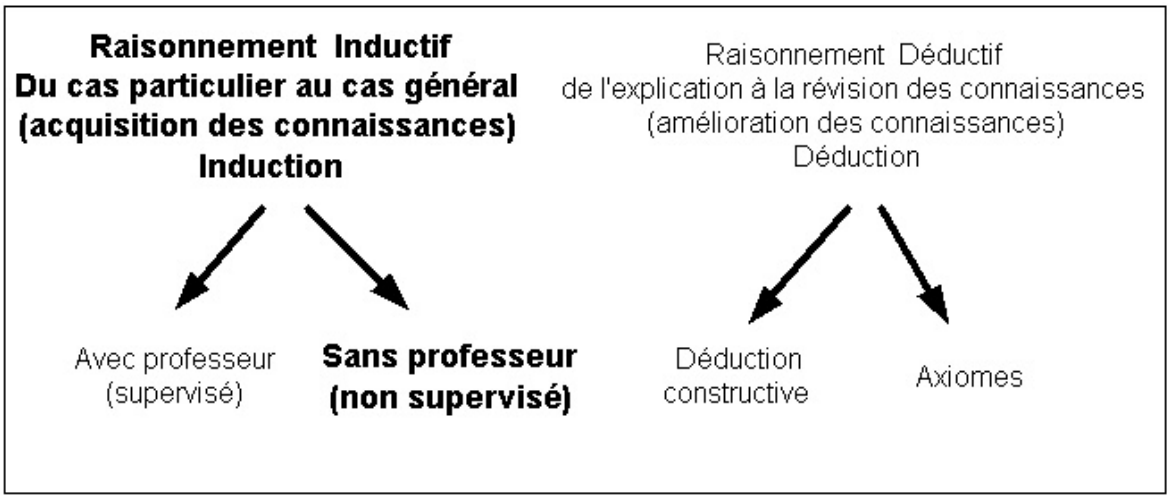


Figure 1.3. Induction/déduction

Le point de vue moderne est davantage tourné vers les approches statistiques. Les méthodes neuronales et les modèles markoviens dominent les techniques utilisées mais d'une manière générale tout apprentissage statistique peut être vu comme apprentissage d'une modèle distributionnel (paramétrique ou non) entre variables. On doit déterminer les conditions générales (sur les données), préciser les principes inductifs et développer un algorithme effectif.

On évalue la performance de l'apprentissage grâce à des critères tels que la probabilité de mauvaise classification, le risque et le nombre d'erreurs.

1.2.2 Types d'apprentissage

Un type d'apprentissage coïncide avec un ou plusieurs objectifs d'apprentissage. Plusieurs objectifs sont possibles:

- acquisition de connaissance
  (apprentissage de concepts)
  - catégorisation (regroupement, loi, …)
  - amélioration/ révision de théorie
    - apprentissage de théorie
    - induction constructive
- amélioration de performance
  - opérationalisation des connaissances

- adaptation
- optimisation

L'objectif ne caractérise pas le type d'apprentissage en soi. En revanche différents protocoles d'apprentissage (modulant les contraintes) permettent de mettre en évidence les types suivants:
- apprentissage supervisé/apprentissage non supervisé (avec/sans professeur);
- apprentissage par renforcement (l'environnement donne un signal, apprentissage difficile);
- apprentissage actif (pratique d'un logiciel, par exemple Oracle);
- apprentissage de type incrémental ou non;
- action de l'environnement (nature indifférente, hostile, professeur plus ou moins coopérant);
- environnement en apprentissage et, environnement et performance.

L'évaluation est conditionnée par la seule grandeur à laquelle on a accès : la taille de l'échantillon (figure 1.4).

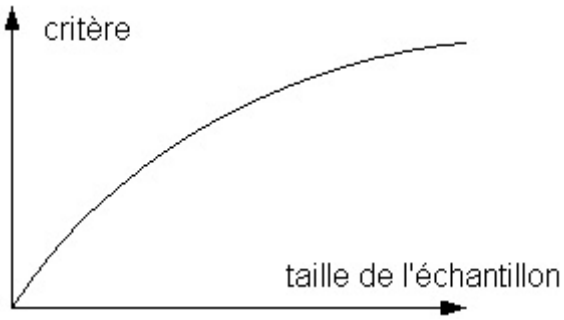


Figure 1.4. Allure possible du critère d'évaluation

L'erreur quand elle est calculable permet de qualifier la performance. L'intelligibilité et la complétude des résultats participent également au critère d'évaluation souvent de façon empirique ou heuristique (interface homme machine, avis d'utilisateurs…).

L'espace des fonctions d'apprentissage cibles est au cœur d'un compromis qui lie la flexibilité pour coller aux exemples et à la pauvreté pour assurer une bonne prédiction:
-si le langage qui modélise l'apprentissage est trop restrictif on est moins prédictif.
-si on n'est pas restrictif, on a plusieurs fonctions de prédiction.
Ce compromis introduit la notion de biais de représentation propre à tout langage de description. Les exemples conduisent aux hypothèses:

Espace des exemples (x) $\rightarrow$ Espace des hypothèses (H)

H peut, par exemple, être un système de règles d'un système expert ou une chaîne de Markov.

Exemples d'apprentissage statistique:

1) Apprentissage bayésien $P(C|x) = \frac{P(x|C).P(C)}{P(x)}$, où x est un élément, C un concept et P(C|x) est la probabilité d'avoir un concept C pour l'élément x;
On apprend des densités de probabilité et on choisit la classe qui maximise cette probabilité.

2) Apprentissage par ensemble de paramètres h(x, a) on apprend une fonction de décision dépendant de paramètres, régression polynomiale h(x)=h(x, w) $w=(w_0,\ldots, w_n)$,

erreur: $\varepsilon = \frac{1}{2}\sum_{i=1}^{p}(h(x_i, w) - t_i)^2$

Plus le nombre de paramètres est grand plus on colle aux données mais moins on est prédictif car on peut approximer du bruit (sur l'apprentissage).

Nous présentons, ci-dessous, un exemple d'apprentissage symbolique permettant de se déplacer dans l'espace des hypothèses pour bien prédire (faire en sorte qu'un exemple positif ne recouvre pas un exemple négatif)
Soient les deux concepts suivants:

$E_1 = \text{triangle} \wedge \text{rayé} \wedge \text{petit}$

$E_2 = \text{carré} \wedge \text{rayé} \wedge \text{grand}$

On définit le plus petit généralisé (au sens faible, c'est-à-dire l'intersection exclusive) $PPG(E_1, E_2)$ = rayé.

Dans le cas d'une implication entre 2 classes non décidable on essaie d'agir par subsomption c'est-à-dire de trouver un algorithme de réduction de clause plus explicative (le risque est de se retrouver avec une complexité de calcul exponentielle).
On définit un exemple positif comme un item représentatif d'un concept donné et un exemple négatif comme un item distinct du concept.
1ère méthode: partir des exemples positifs et calculer leur PPG (plus petit généralisé).
2ième méthode: calcul de l'espace des hypothèses ; on généralise les exemples positifs sans couvrir les exemples négatifs, et ajout d'un nouvel élément positif, on réitère).

1.2.3 Problèmes de l'apprentissage

Plusieurs types de problèmes sont considérés et notamment par rapport aux données et aux hypothèses. Dans un certain nombre de cas réels un mélange de données symboliques et numériques doit s'opérer or les traitements ne sont pas similaires. Une partie des données s'assimile souvent à du bruit, par opposition une partie des attributs est plus pertinente qu'une autre, l'identification de ces parties s'avère d'une très grande importance mais difficile à mettre en œuvre. La taille nécessaire et la surdescription (angl.,"over-fitting") des exemples est souvent empirique et soumise aux biais des expérimentations. Pour éviter ces biais il est possible d'envisager une combinaison de classifieurs et un parcours de l'espace des hypothèses. Cela pose des problèmes d'optimisation algorithmique.

1.3. Traitement automatique du langage naturel

Il existe 3 grands domaines d'application dans le TALN qui sont l'indexation, la traduction automatique et l'interrogation de base de données relationnelles. Nous nous intéressons plus précisément aux problèmes liés à l'indexation de texte intégral.

1.3.1 TALN et syntaxe

Certains outils du TALN permettant de détecter des patrons ou d'attribuer étiquettes syntaxiques fonctionnent à l'aide d'automates à états finis (Finite State Transducer, FST). La technique n'est pas nouvelle mais robuste, rapide, consommant peu d'espace disque et de mémoire. Ces outils peuvent être aussi appliqués au "Shallow Parsing" (analyse grossière) consistant à déterminer les acteurs d'une phrase (sujet, verbe, objet avec leur dépendances dans 60% des cas). Technique semble-t-il largement répandue sans analyse sémantique profonde pour identifier des couples sujet-verbe et verbe-objet. Détection des dépendances avec 80% (précision/rappel) si distance <6 mots du chunk source (morceau de phrase comme un groupe verbal ou nominal). Les moteurs de recherche d'Internet classiques comme Altavista font du TALN, réduit mais efficace (tokénisation de la requête, analyse morphologique, sélection du sens par cooccurrence). Les termes complexes comme « pomme de terre » doivent améliorer la précision mais l'ajout de synonymes (baisse de rappel) ou la

troncature (angl. stemming) (baisse de précision) peuvent être nuisibles. La logique a aussi sa place dans le TALN. La question de l'universalité des hiérarchies de connaissances est posée même pour le langage naturel. Certains, comme J.Sowa, justifient l'usage de la logique comme langage d'expressivité universel et complet (J.Sowa prétend que l'on peut tout décrire avec seulement ∃ et *and*, on peut exprimer l'existence de tous les objets qui nous entourent). J.Sowa prétend que le langage naturel est une structure logique et qu'il est aberrant d'exprimer la logique avec des opérateurs pour prouver les théorèmes :
Sujet + agent + verbe + .. se met sous la forme (∃…,∀…)(…(…))(…)(…).
Cette représentation fait prévaloir 3 règles d'inférences (§1.2) : généralisation, spécialisation et équivalence avec 6 opérations élémentaires. Celles-ci sont applicables au langage naturel avec des expressions logiques.

1.3.2 TALN et sémantique

La désambiguation de sens est relancée depuis 1980 avec l'utilisation de lexiques électroniques (Wordnet) et l'étude de corpus (apprentissage par techniques d'annotation et techniques statistiques). L'évaluation donne 65-90% d'attribution automatique d'étiquettes sémantiques aux mots d'un texte. Les résultats sont variables et très dépendant de l'avis des experts qui évaluent. Tests portés sur quelques mots.
Aucune définition exacte du sens ne fait l'unanimité.
La désambiguation du sens n'est pas une fin en soi mais peut être utilisée en traduction automatique, en recherche documentaire, en analyse thématique et de contenu, analyse grammaticale, traitement de la parole, vérification orthographique. La nécessité de la désambiguation de sens remonte à 1949 avec Weaver en traduction car la TAO (traduction assistée par ordinateur) a besoin d'analyser finement le sens.
Les différentes méthodes de l'intelligence artificielle du TALN sont les suivantes:

- symbolique
- connexionnisme
- à base de connaissances : limité à un domaine
  dictionnaire électronique (Longman, LDOCE, Collins)
  thésaurus (Roget's)
  lexique informatique (Wordnet, Corelex, Acquilex, Cyc…)
- à base de corpus (existait de 1890 à 1940 par des études manuelles, perte de vitesse en 1960 avec les grammaires statiques ou génératives, regain après 1980 avec le renouveau des méthodes statistiques textuelles)

L'affectation automatique de sens se réalise souvent grâce à un corpus étiqueté manuellement. Les modèles sont basés sur des classes lexicales [Brown et al, 1992][Pereira et al 1992][Yarowsky, 1992]. Ces méthodes sont en général s'inspire d'un calcul de similarité. Le projet SENSEVAL se propose d'évaluer l'attribution correcte du sens (seulement quelques mots évalués). Les résultats, variables, donne 65 à 90% d'étiquetage correct. Mais l'évaluation dépend de l'avis des experts.
Certains problèmes restent ouverts comme le rôle du contexte (contexte local, contexte thématique). [Hearst, 1994] segmente les phrases en GN, GV avec contexte de plus ou moins 3 mots en combinant les aspects syntaxique et statistique (capitalisation). (Leacock; Chodorow; Miller) indiquent que pour un classifieur statistique le contexte local est supérieur comme indicateur de sens.
D'autre part le domaine à une incidence sur l'usage d'un sens. C'est la division du sens. Le sens répond à une granularité : 2 modèles, modèle de la banque (consistant à répertorier tous les sens) et modèle de l'usage qui dit que le sens n'apparaît que dans un contexte d'usage [Harris, 1968]. [Manning & Schütze, 1999] adoptent ce point de vue. Par contre le lexique génératif développe une approche dépendant du discours.

1.3.3 Problèmes du TALN

S'il est admis que l'étude de la syntaxe est incontournable pour établir l'interprétation d'un texte, on ne connaît pas encore très bien son impact sur la sémantique. Le traitement de corpus est une voie qui tente de répondre à cette interrogation. L'aspect incrémental syntaxe >> sémantique, sémantique >> syntaxe est à envisager. Il faut aussi considérer le multimedia dans son ensemble comme perspective du TALN pour pouvoir étudier des documents plus complexe faisant intervenir d'autres représentations pour qualifier le sens d'un texte (notamment les graphiques ou images). Il est nécessaire de considérer l'adaptivité d'un système automatique par rapport à l'utilisateur final, au type de tâche, au multilinguisme et au corpus. Deux lacunes majeures sont à prendre en compte: l'absence de méthodologie pour définir un besoin utilisateur et l'absence de perpective d'exploitation des résultats.
La logique n'est peut-être pas le seul moyen incontournable pour raisonner. Les techniques utilisées doivent être proches du but recherché. En résumé il n'y a pas de « méthode » IE (extraction d'information). Il faut peut être construire des bases de connaissances, il faudrait une théorie avant de formaliser et d'axiomatiser. On présente la qualité de détection automatique de différents aspects du TALN (en terme de précision) [Basili et al, 1999]:

Reconnaissance morphologique	97 à 100 %
Etiquetage syntaxique	90 à 97%
Détection de noms propres	85 à 93%
Patron (angl. Template)	50 à 85%
Scénarios	42 à 50%
Coréférence	50 à 60%

Les problèmes les plus ardus à résoudre dans le traitement informatique du sens restent la *polysémie* et la *coréférence*. Nous donnons quelques exemples de polysémie et de coréférence qui posent des problèmes du point de vue informatique [Charniak, 1996].
Exemple d' ambiguïté sémantique de contexte, (entre avocat l'homme de loi et avocat le fruit):
"Si vous tombez sur un bon stage vous allez découvrir tout ce que fait l'avocat en pratique."
Exemple d' ambiguïté sémantique pragmatique:
"les nuages tombent sur la tête"

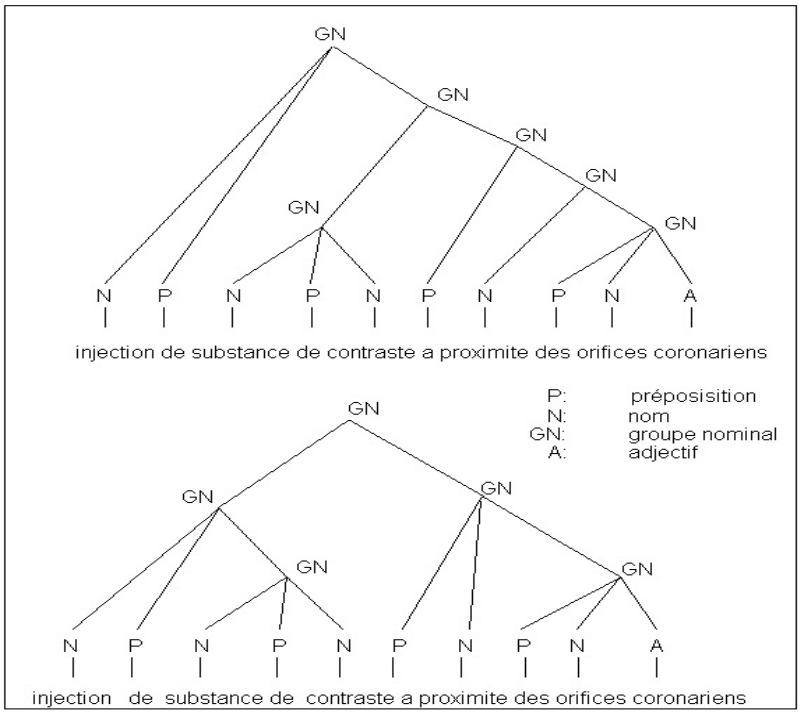


Figure 1.5. Problème du découpage en groupes nominaux

Exemple d' ambiguïté syntaxique de découpage (en groupes syntagmatiques, figure 1.5):
"Injection de substance de contraste à proximité des orifices coronariens."
Exemple d'ambiguïté syntaxique d'étiquetage (figure 1.6):
"Pierre et moi les avions révisés"

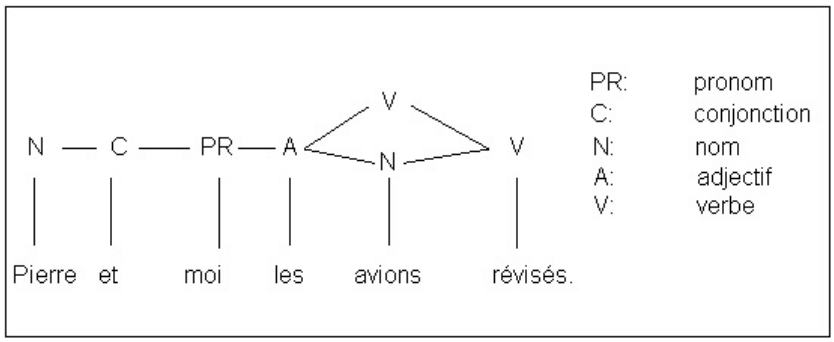


Figure 1.6 Problème d'étiquetage syntaxique

Exemple de coréférence:
"Ces ***renseignements*** ont une valeur pronostique qui ne peut être égalée par aucune autre méthode actuellement connue. De plus, ***ils*** sont essentiels avant toute chirurgie coronarienne ou pontage aortocoronarien."
"Ils" remplace "renseignements", dans la suite de l'énoncé.

La proportion de ces problèmes d'ambiguïté est importante et variable dans les corpus.

1.4. Analyse des données et classification

1.4.1. Aspects statistiques

La classification s'inscrit dans un processus d'apprentissage permettant l'acquisition des connaissances. Depuis Aristote la classification constitue un schéma d'organisation des connaissances. D'abord appliquée en botanique et en zoologie pour classer les espèces, la classification a permis ensuite de structurer les éléments chimiques, l'évolution des étoiles... Elle est maintenant utilisée en informatique (fouille de données) notamment pour obtenir des groupes de termes c'est-à-dire une famille de termes ayant des caractéristiques communes, ou famille sémantique.

La démarche statistique consiste à traiter et interpréter des informations recueillies. Une des grandes branches actives est la statistique exploratoire ou analyse des données [Benzécri, 1973][Benzécri, 1976]. Le but de cette dernière consiste à synthétiser, résumer et structurer l'information contenue dans les données [Murtagh, 1987][Saporta, 1988]. Pour ce faire les données sont réduites sous forme de tableaux, d'indicateurs numériques ou de graphiques. Classiquement on divise les méthodes de l'analyse des données en deux groupes:

- les méthodes de classification, proposant de former des groupes homogènes de l'ensemble des individus;
- les méthodes factorielles, qui réduisent la dimension de l'ensemble des individus en un nombre de composantes synthétiques.

La division en deux groupes de méthodes est plus historique et symbolique dans la mesure où les méthodes factorielles sont plus visuelles que les méthodes de classification. En réalité les méthodes factorielles sont aussi utilisées pour la classification dans la plupart des cas. Ces 2 groupes n'étant pas exclusives, certaines méthodes contiennent les 2 approches.

Ces méthodes sont très consommatrices en temps de calcul mais deviennent plus populaires depuis l'avènement sur le marché de machines puissantes et d'algorithmes optimisés.

Dans la littérature anglo-saxonne le terme de classsification est utilisé dans un double usage par rapport au français, c'est un faux-ami. Si en français la *classification* est la recherche de classes (angl., "clustering"), en anglais elle peut signifier le *classement* c'est-à-dire l'affectation d'un objet à une classe (angl., "categorization" ou "classification").

1.4.2. Démarche de la classification

La figure 1.7 montre le type de tableau exploité pour la recherche de classe d'objets.
Soit $I=\{O_1,\ldots,O_p\}$ l'ensemble des objets ou individus à classer et $J=\{V_1,\ldots V_q\}$ les variables caractérisant les objets. On appellera M la matrice application de $I \times J$ avec m_{ij} l'élément générique de la matrice. Cette représentation est considérée dans toute analyse de classification[3]. A partir de la matrice, différentes stratégies sont menées dépendant des contraintes de l'environnement et de l'application visée pour évaluer la similarité des objets deux à deux. En guise de prétraitement pour aboutir au tableau de données 3 hypothèses essentielles doivent être envisagées:
- la sélection des objets et leur dimensionalité
- la sélection des variables et leur dimensionalité
- la nature de la relation $\Re$ entre un objet et une variable ($o_i \Re v_j$)

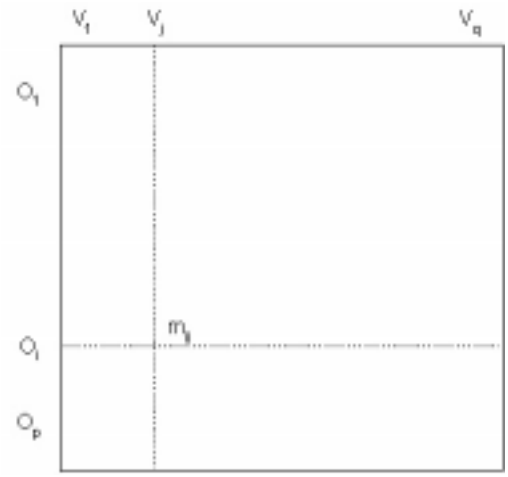


Figure 1.7. Tableau des données

1.4.3. Classification de termes

L'exemple suivant présente un groupe de termes : jardin, guéridon, fontaine, parasol, pelouse. On imagine aisément qu'un martien venu sur terre du jour au lendemain à qui on soumettrait un tel groupe de termes ne pourrait pas interpréter les relations sémantiques de ces termes du simple fait qu'on les a mis ensemble. Notre interprétation humaine est nécessaire pour relier deux termes du groupe à savoir qu'un parasol est utilisé dans un jardin pour se protéger du soleil. Cela vient de notre expérience qu'on appelle aussi connaissance pragmatique.

Les théories actuelles de la sémantique structuraliste ne nous permettent pas d'avoir une qualification précise des liens sémantiques entre deux termes d'un groupe. D'une manière générale la linguistique n'apporte pas de base théorique unifiée sur une formalisation des connaissances pragmatiques. Certains prétendent même que celles-ci sont construites au fil du

[3] Cette structure de données apparaît dans d'autres types d'algorithmes comme l'algorithme d'automates à état finis ou le calcul logique par calcul de tableau.

temps donc furtives. Pour avoir une idée concrète et donc pragmatique des liens entre deux termes on doit disposer de ressources établies par un expert.

On vient de voir deux aspects de la classification : son apport qui est de regrouper des ensembles de termes de même famille et sa lacune pour interpréter ces groupes sans ressource externe. Une classification correctement interprétable agit dans un mode d'acquisition semi-automatique par le biais d'un media de connaissances (collection de texte) et d'une ressource de connaissance (thésaurus).

1.4.4. Applications de la classification

Les applications de la classification en traitement automatique des langues sont multiples :

- obtention d'un profil utilisateur (un corpus utilisateur permet de fixer la terminologie structurée propre à un utilisateur)
- reformulation de requête (des mots-clés de base peuvent amener l'utilisateur à sélectionner des termes de groupes associés aux mot-clés pour affiner sa requête)
- renseignement sur le contenu d'une base (une thématisation hiérarchique peut informer l'utilisateur sur le contenu d'une base de façon à optimiser la formulation de sa requête)
- constitution d'une terminologie structurée d'un domaine spécialisé
- classification de document (création de catégories thématiques pour classer des documents)
- contextualisation conceptuel d'un terme (recherche d'équipe pour un nom propre,…)

1.4.5. Problèmes de la classification

La classification repose sur des représentations des données (objets) très classiques dont les limites de traitement sont connues et souvent très consommateurs de ressources en temps de calcul. La sémantique des objets doit être très clairement définie, non ambiguë. Cette même sémantique doit être typique pour pouvoir trouver les classes d'objet c'est-à-dire proche de relations de type présence/absence. Un biais de traitement se produit dans le cas d'objets ayant un contexte sémantique complexe comme des objets du langage naturel. Les problèmes émanent de la fréquence multicontextuelle des termes et de l'analyse inverse du processus de classification souvent impossible.

2. Définitions

Après s'être intéressé aux disciplines présentant des méthodologies, dans cette partie nous essayons de définir les principales notions qui seront considérées dans notre approche. Cette particularité de rassembler ces définitions tiens compte du fait qu'il n'existe pas de définition unanime pour chaque notion et ce quelque soit le champ de recherche. Nous définissons les notions suivantes: extraction d'information, terme, concept, filtrage d'information, profil d'utilisateur, corpus, recherche documentaire et fouille de textes. Les définitions perdent parfois leur caractère général pour les rapprocher de notre cadre méthodologique.

2.1. Extraction d'information

Processus par lequel un système automatique est capable de traiter des documents par une approche linguistique en dérivant une représentation structurée du contenu. Cette représentation était constituée originellement de cadres d'attributs/arguments mais face à la difficulté de récupérer tous les arguments de façon exacte les efforts se sont portés davantage sur l'extraction de relations plus singulières et souvent guidée par le besoin de l'utilisateur (exemple: la traduction d'un mot dans un paragraphe) [Basili et al, 1999].

2.2. Terme

Pour nous, un terme est une unité syntagmatique. Nous considérons qu'il existe une double typologie des syntagmes : nominale et verbale. Le terme est à la base de la dénomination des entités constituant une ontologie (description d'un domaine). A partir d'un ensemble de termes on peut établir des relations sémantiques [Rastier, 1995].

2.3. Concept

Il existe 3 principales écoles de définition de concept que l'on résume très brièvement de la façon suivante: l'école de la logique qui considère un concept défini par des conditions nécessaires et suffisantes, l'école du stéréotype qui considère qu'un concept est défini par un ensemble d'exemples et l'école du prototype qui définit un concept par rapport à un élément.

On identifie deux types d'extraction de concept C:
- une extraction statique, C est défini par des attributs fixés avec des traits sémantiques
- une extraction dynamique, C possède des termes typiques l'entourant.

Nous sommes plus spécialement focalisés sur le deuxième type d'extraction.

Dans les langages orientés-objet, une classe est un ensemble d'objets possédant une structure, un comportement et des relations similaires. Nous définissons la notion de concept à partir de cette description. Mais a contrario de la plupart des méthodes courantes conduisant à une description polythétique (stéréotypique) nous proposons d'adopter une description monothétique d'un concept.
On définit la description monothétique d'une classe comme étant une description proche de celle d'une d'un concept défini grâce à un prototype. C'est-à-dire qu'un concept sera défini par rapport à une entité et la(les) caractéristique(s) de cette entité.

Voici notre schéma d'extraction de concept:
On donne :
- un ensemble de variables I qui sont les objets à classer (exemple de variable: un terme),
- des règles statistico-linguistiques $R_1,R_2,\ldots R_r \in R$ qui décrivent l'usage des objets à classer (exemple de règle: distribution des termes par rapport aux verbes),
- des heuristiques graphe-statistiques (exemple d'heuristique: agréger 2 graphes qui ont un terme en commun):
  $\{H_1,H_2,\ldots H_h \in R\} \Lambda \{ \exists (a,b,c,\ldots) \in I\} \rightarrow K$ (espace de graphes agrégées).

On trouve :
- une partition C partielle et recouvrante de variables : $C_1,C_2,\ldots C_c \in C$,
- une description monothétique de classes grâce à un terme central ,
- chaque classe est assimilée à un champ sémantique aussi qualifiée de concept et expliquée par des relations internes et un identificateur.

2.4. Filtrage d'information

On appelle filtrage d'information le processus permettant, à partir d'un large volume d'informations dynamiques, d'extraire et de présenter les seuls documents intéressant un utilisateur qui a préalablement décrit ses centres d'intérêt [ADBS, 1998].
Un agent de filtrage est basé sur une correspondance entre un centre d'intérêt et un message. La décision est donc booléenne : le message correspond au profil ou non. Dans le cas d'une correspondance le transfert d'information est autorisé.

2.5. Profil utilisateur

On définit un profil utilisateur par l'ensemble des données identifiant ses centres d'intérêt thématiques. Ce profil est très souvent qualifié par des suites de termes relevant d'un ou plusieurs domaines thématiques.

2.6. Corpus

Pour créer un corpus, deux problèmes sont à considérer : l'homogénéité et la taille. La taille est caractérisée par le nombre de mots. A l'heure actuelle des gros corpus comprennent plusieurs centaines de millions de mots. Un corpus homogène couvre un domaine spécifique dans toute sa diversité. Un corpus est généralement écrit dans une langue bien qu'il puisse être multilingue. Certains, comme [Assadi, 1998], prétendent que les corpus proviennent uniquement de collections de documentation technique parce que la terminologie y est bien établie, stable et nominalisée. En fait comme les documents sont un moyen de communication, s'ils reflètent un échange entre deux membres d'une communauté liés par un thème, nous pouvons considérer que la terminologie reflète aussi la terminologie de cette communauté.

.7. Fouille de textes

Traitement qui permet de passer en revue un ensemble de données textuelles grâce à des techniques statistiques et linguistique pour en déduire de l'information utile par rapport au but fixé par un utilisateur lui évitant une lecture séquentielle de l'information. Cela est représenté par des caractéristiques issues de deux types de traitement dont la recherche documentaire ou indexation (information source ayant même format : texte non structuré) et la Fouille de Données (en anglais Data Mining, information source symbolisée par des ensembles de structure plus précise).

1.8. Recherche documentaire

Extraction de documents textuels, à partir d'une base documentaire, répondant à un besoin d'unutilisateur. Ce besoin se traduit souvent par une requête de mots clés.

3. De l'intelligence économique au filtrage d'information

Après avoir décrit les techniques et après avoir fixé les définitions (souvent variables) de notions utiles et nécessaires à notre approche, nous présentons maintenant un champ applicatif centré vers un utilisateur final. Nous considérons que la classification automatique, cœur de notre méthodologie, possède des qualités lui permettant de s'inscrire dans des rouages d'un système d'information. Ce système d'information s'avère notamment intéressant pour collecter et analyser l'information stratégique au sein d'une entreprise. Finalement nous présentons une fonctionnalité applicative particulière: le filtrage d'informations textuelles.

3.1 Besoin d'information

L'avènement de flux financiers transfrontaliers considérables et les rapides mutations sociologiques et structurelles en cours promettent un XXIième siècle hyper-concurrentiel au travers d'une société de l'information. Le XXIème siècle sera celui de la mondialisation flexible et multipolaire, tirée par une intelligence irriguée d'informations et de renseignements. Le choix de l'Intelligence Economique apparaît forcé, dicté par la marche accélérée et complexe de la mondialisation-globalisation qui favorise les activités de service et l'économie financière. On doit toutefois souligner que l'information transite essentiellement entre les pays technologiquement les plus développés couvrant seulement 25% de la population mondiale.
Autant psychologique que financier, l'engouement pour l'Intelligence Economique et financière, devenue depuis plusieurs années un sujet à la mode et un thème à investissement, a entraîné une nette croissance du nombre d'entreprises françaises et de consultants spécialisés

dans le secteur du renseignement. L'industrie des professionnels de l'information sur mesure pour une intelligence technique, organisationnelle, financière ou autre se signale par sa prospérité et ses perspectives. L'effervescence s'est traduite et consolidée par une série d'articles et d'ouvrages advenus ces dernières années et par des enseignements universitaires (comme aux Universités d'Aix-Marseille, de Poitiers ou de Marne La Vallée) ou par divers organismes de formation. Les termes anglo-saxons de "*Competitive intelligence*", de "*Business intelligence*" ou de "*Corporate intelligence*" en sont devenus plus familiers. Les technologies de l'information, privilégiant des moyens de communication électroniques pour des travaux en réseaux et de la messagerie, sont un passage obligé. La boulimie de miniaturisation des outils favorise *l'informatique communicante*, le télétravail et les externalisations d'activités ainsi que le commerce électronique. L'argent n'est plus la seule loi d'airain puisque l'information est source de pouvoir ("knowledge is power"). Plus l'activité du renseignement s'intensifie et plus les individus cherchent des parades. Plus les acteurs camouflent leurs agissements et plus le besoin d'intelligence au sens de renseignement s'accroît. L'Intelligence Economique s'avère plus proche du monde du renseignement que celui de l'information.

3.2 Veille technologique automatisée

La veille technologique est le bras technique de l'Intelligence Technologique dans son rouage de traitement de l'information. Ses fonctions sont les suivantes : définition des objectifs stratégiques de veille, choix d'une méthodologie logicielle et des sources, qualification de l'information, analyse et synthèse et finalement diffusion de l'information élaborée.
Pour assurer efficacement cette veille face à l'abondance d'informations hétérogènes qui ne sont pas compartimentées selon les domaines auxquels les entreprises s'intéressent, le renfort des outils d'infométrie s'avère décisif. L'infométrie est un terme qui a été utilisé pour la première fois en 1987 par l'International Federation of Documentation (IFD), pour désigner l'analyse de l'évolution des tendances de l'information, couvrant aussi bien la bibliométrie (analyse de notices bibliographiques) que la scientométrie (analyse de l'évolution des sciences par des indicateurs). Différents outils logiciels peuvent être utilisés à chaque étape d'un processus de veille. Le media utilisé dans une veille automatisée consiste en volume de textes libres ou non structurés ou en volume de textes semi-structurés (ayant des champs renvoyant vers certaines valeurs, exemples champ auteur ou date). Les textes semi-structurés sont surtout les notices bibliographiques. A ces deux types de données correspondent deux types d'analyse logicielles:

- l'analyse textuelle de contenu. La génération automatique de représentations simplifiées du contenu des documents offre notamment des aides à la navigation, à la découverte, à la reformulation [Rajman & Besançon, 1997].
- l'analyse de notices bibliographiques. Certains outils d'infométrie permettent de structurer l'information qui n'est pas visible par une simple lecture séquentielle des textes (par exemple, établir une cartographie des équipes de recherche travaillant dans le monde sur un sujet déterminé).

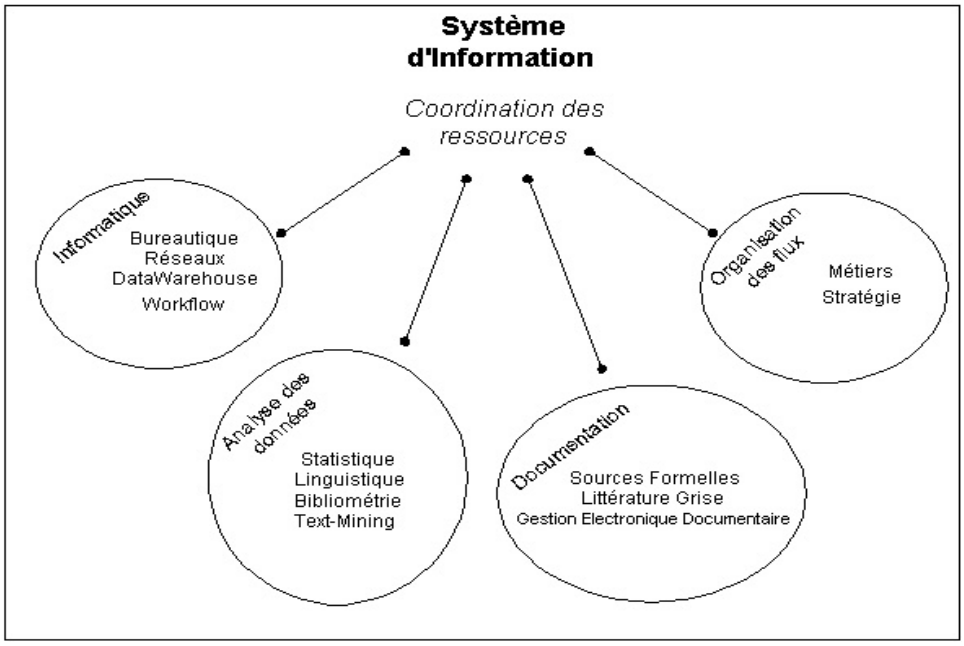


Figure 1.8 Acteurs du filtrage dans un système d'information

Le processus de veille se situe au confluent de la stratégie d'entreprise et des systèmes d'information. L'outil informatique est une brique de ce processus et fait participer les différents services d'une entité économique (figure 1.8). Compte tenu de la complexité croissante des systèmes logiciels, de la quantité d'information périodiquement mise à jour et de la régularité des besoins, la gestion autonome du processus logiciel ira dans le sens du confort de l'utilisateur. C'est l'opposition des *Technologies Push* (l'utilisateur sollicite indirectement) et *Pull* (l'utilisateur sollicite directement). Les technologies *Push* (on parle d'*agent Push*) deviennent de plus en plus convoitées et dans la mesure où elles permettent une adaptation flexible aux besoins de l'utilisateur.

3.3 Agent de filtrage d'information

Le filtrage d'information est un mode de traitement qui assure une sélection d'information suivant un besoin exprimé. Ce besoin n'est pas explicité à chaque fois que l'action de recherche d'information est exécutée.
Un agent automatique qui s'active à un instant donné peut réaliser une certaine tâche (figure 1.9). Un premier type de filtre modifie le message (ou la place à laquelle il est archivé). L'autre type contrôle comment un ensemble de messages doit être affiché ou un message doit être affiché par rapport aux autres. Autrement dit, l'un modifie les propriétés intrinsèques, et l'autre modifie (ou génère) les propriétés extrinsèques ou dérivées. On les appelle filtres de classement et filtres d'affichage.
Les deux filtres interviennent dans des situations différentes :

* Filtres qui agissent à un moment donné (récupération de nouveaux messages, action de l'utilisateur sur un groupe de messages) et affecte les propriétés de stockage d'un message. Ces filtres manipulent le message de la façon suivante:
    - o Déplacer les messages vers un répertoire
    - o Supprimer un message

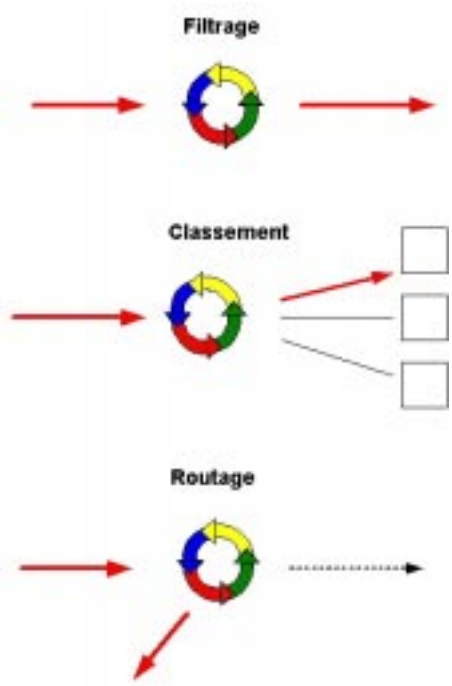


Figure 1.9 Filtrage/classement/routage

- o Changer la priorité d'un message
- o Générer automatiquement un "faire suivre" ou "répondre"

Exemples de filtres de classement:
si *From* **contient** "@liia.u-strasbg.fr", **déplacer** dans le répertoire "travail".
si *Received* **contient** "205.199.212.", **déplacer** dans le répertoire "spammers".
si *To* **contient** "tout_le_monde@liia.u-strasbg.fr ", **déplacer** dans le répertoire "Trash".

* Filtres qui analysent certaines propriétés dynamiques d'un message. Ces filtres sont généralement activés quand on exécute la visualisation d'un message.

Ils ne changent pas du tout le message même. Quelques exemples de propriétés dynamiques traitées peuvent être:

- o le score d'un message déterminé sur la base d'un tri
- o la couleur à utiliser pour afficher les lignes d'informations d'un message

Exemples de filtres d'affichage:
si *From* **contient** "winner@some.host", **rapprocher** son score de la tête de liste d'affichage.
si *Subject* **contient** "foo", **diminuer** son score.
si *taille* > 100 Ko, **diminuer** fortement son score.
si *priorité* = 3, **l'afficher** en rouge.

4. Notre problématique

Nous terminons ce chapitre par la présentation de notre objectif et de nos différentes hypothèses. Ces hypothèses cristallisent les points d'ancrage de notre méthodologie à savoir: l'analyse d'un domaine à partir de textes, l'étude des distributions des termes pour classifier ces termes et l'étude de l'interprétation des résultats d'un processus de classification automatique.

4.1. Analyse d'un domaine à partir de textes

En général le sens d'un terme se présente dans 2 types de ressources : sens consigné dans des ressources stables (i.e thesaurus), et éclosion du sens dans l'usage (i.e. dans les textes). Ces

aspects du sens qui ne sont pas formels présentent un caractère statique et dynamique. Dans le cas d'une réalisation dynamique du sens, nous nous intéressons à cerner le sens grâce à l'analyse des contextes d'utilisation des termes.
[Osipov, 1997] relève 7 problèmes ou buts principaux pour l'acquisition des connaissances à partir des textes:

1- minimisation du rôle des ingénieurs de la connaissance dans la construction d'un système appliqué;
2- minimisation d'interviews subjectives et diminution du temps de l'acquisition des connaissances;
3- résolution des lacunes de la connaissance;
4- introduction des méthodes d'identification de connaissances non verbales dans les systèmes technologiques de bases de données;
5- transformation des résultats du travail des algorithmes d'apprentissage par l'exemple en moyen de présentation, maintenu par les systèmes intelligents;
6- identification d'information explicitement absente concernant les propriétés des liens sémantiques;
7- construction de dictionnaires d'objets d'un domaine dans un processus d'acquisition des connaissances

Hypothèse 1 : Utilisation d'un corpus dont le contenu est représentatif d'un domaine spécifique, et adapté au traitement d'une statistique relationnelle.

Le texte est un ensemble ordonné de symboles dont l'ordre n'est pas soumis à des règles systématiques. Le sens est construit par juxtaposition de morceaux de sens qui ont un rôle particulier dans des morceaux plus grand comme la phrase ou le paragraphe. La dénotation joue un rôle important dans la description du sens. On la retrouve notamment confinée dans des groupes structurels que l'on appelle syntagmes nominaux ou syntagmes verbaux. Ceux-ci n'étant pas toujours très stables d'ailleurs. Leur forme syntaxique peut varier. Cependant, dans une terminologie spécialisée, des groupes se révèlent très accepté et véhiculent un sens très précis.

Hypothèse 2 : Extraction des termes susceptibles de caractériser le domaine concerné par le corpus.

4.2. Approche distributionnelle du sens

Pour ce faire on va étudier la distribution des associations syntaxiques dans les corpus. On utilisera la notion de cooccurrence terminologique pour identifier et analyser ces associations.

Hypothèse 3 : Analyse des cooccurrences exprimant significativement la contextualisation du sens.

L'analyse des données nous offre de telles démarches d'analyse. Les techniques de classification utilisées jusqu'à présent dans l'indexation tiennent très peu compte de la structure des données dans la construction de la représentation matricielle et l'analyse de cette représentation. L'émergence, à partir d'un ensemble de données, d'une structure spécifique doit restituer l'essentiel de l'information tout en condensant la masse des données. La structure résume le contenu des données grâce à ses associations intrinsèques.

Hypothèse 4 : Réalisation d'une classification en adoptant une représentation de type tableau de contingence pour aboutir à une classification automatique efficace.

4.3. Interprétation de la classification

Notre objectif est comprendre comment améliorer la construction d'une représentation des associations entre termes d'une part, et d'autre part de comprendre comment améliorer l'exploitation de cette représentation pour aboutir à des regroupements de termes interprétables et exploitables (figure 1.10).

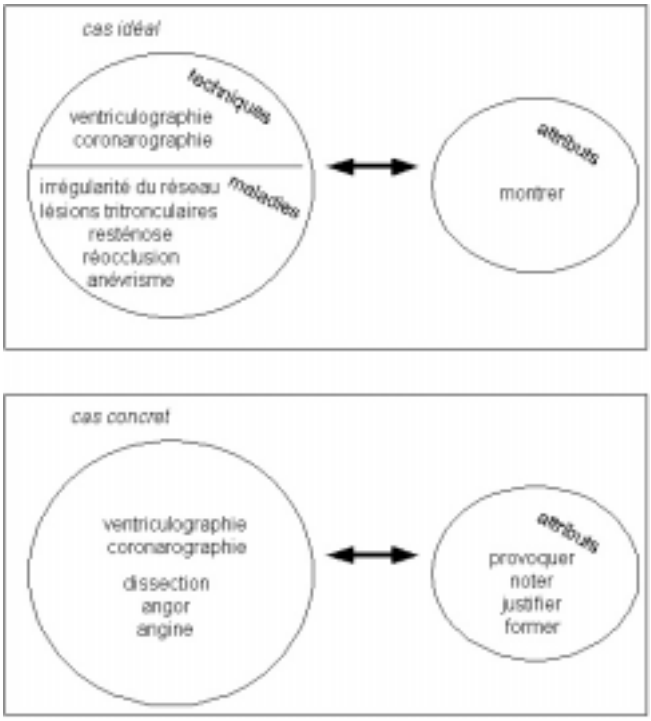


Figure 1.10 Exemple de concept

Hypothèse 5 : Possibilité d'attribuer des étiquettes aux classes et au corpus grâce à une ressource sémantique.

Hypothèse 6 : Possibilité de réaliser un filtrage de document performant en exploitant les classes de termes.

Résumé du chapitre

Le projet a pour objectif d'établir les fondements d'une nouvelle génération de système qui permet un traitement du langage naturel rapide, efficace et robuste, tout en gardant un caractère suffisamment général. Partant de certains aspects de la performance qui sont gouvernés par des règles grammaticales, nous poursuivons une nouvelle approche de sémantique lexicale qui réunit dans son modèle des aspects de performance et de compétence. En particulier nous visons à modéliser l'interaction de la générativité, qui est caractéristique de la compétence, et de la lexicalisation, qui relève du domaine de la performance.
Pour y arriver nous voulons examiner l'interaction de la lexicalisation et de la générativité sur la base du phénomène de cooccurrence. Cette interaction sera modélisée à l'aide de schémas morphosyntaxiques et d'une classification automatique des associations à partir de corpus .
Dans les systèmes d'information, une des difficultés majeures réside dans le fait de cerner le besoin de l'utilisateur. Très tôt l'informatique a été perçue comme un rouage important d'un système d'information avec des fonctionnalités dont l'ambition consistait à simuler l'activité humaine. L'application immédiate du traitement du langage naturel a été la recherche documentaire et par-là l'indexation. C'est un des besoins que l'intelligence artificielle a pu approfondir. Le problème de la recherche documentaire reste partiellement résolu dans la mesure où la formulation du besoin de l'utilisateur dépasse rarement 3 ou 4 mots-clés (2.35 en moyenne d'après une étude de [Silverstein & Henzinger, 1999]). Dans certains cas où l'utilisateur est habitué à condenser sa requête ce type d'approche n'est pas nécessairement un obstacle pour sa résolution mais peut le devenir dans le cas d'une formulation détaillée du besoin. Nous nous attachons au problème inverse c'est-à-dire au filtrage d'information. Au lieu d'attaquer une base documentaire par une requête, c'est la base documentaire qui va transférer les documents utiles. La condition requise est l'apprentissage du besoin de l'utilisateur par ses habitudes de consultation documentaire. Les documents spécifiques à un domaine de consultation serviront de corpus d'apprentissage dont le traitement par une méthode d'apprentissage statistique conduira à la création d'un filtre sémantique. Ce filtre sera ensuite utilisé comme moyen pour sélectionner des documents relatifs au domaine de consultation.

CHAPITRE 2
ETAT DE L'ART
Méthodes et outils

Dans ce chapitre nous présentons les méthodes qui vont nous permettre de dégager des champs sémantiques à partir d'un corpus. Ce chapitre se décompose en 3 sous-parties: les méthodes de classification statistique, les méthodes de traitement linguistique et les outils applicatifs. De prime abord la présentation ressemble à une simple énumération de techniques tel un inventaire qui présente ce qui existe mais sans lien évident avec le contexte. N'ayant pas de parti pris pour une méthode en particulier il nous a semblé opportun de présenter, succintement et d'un point de vue technique, les approches utilisées pour implémenter une classification non supervisée. Les limites de chaque méthode et les hypothèses fixant notre objectif permettent ensuite de mieux comprendre vers quel choix nous avons orienté notre méthodologie de classification. De même pour comprendre quelle influence tel ou tel phénomène linguistique peut avoir sur un processus de classification automatique on se devait de présenter les différents problèmes de la langue que l'on rencontre avec les principales écoles de pensée.

Nous ne disposons d'aucune donnée extérieure sur la définition d'un champ sémantique du corpus. Nous nous appuyons donc sur une approche d'apprentissage non supervisé et plus exactement (hypothèse 4, §1) sur une approche de classification automatique non supervisée.

La première partie de ce chapitre est consacrée aux méthodes de classification non supervisées qui se répartissent en 3 grandes familles (pouvant s'imbriquer): les méthodes hiérarchiques, les méthodes de partitionnement et les méthodes factorielles. Ces familles se subdivisent elles-mêmes en différentes méthodes. Toutes ces méthodes se réfèrent à un calcul matriciel et à une notion de similarité intra-classe/dissimilarité inter-classe. La présentation de chaque méthode est plutôt technique et courte pour ne pas surcharger le chapitre.

La deuxième partie traite des théories concurrentes décrivant les phénomènes syntaxiques. Ensuite nous décrivons les principales notions de sémantique lexicale et les techniques d'extraction automatique de syntagmes nominaux. Nous exposons 5 principales techniques: la méthode du dictionnaire, la méthode des cooccurrences, la méthode des segments répétés, la méthode des schémas morpho-syntaxiques et la méthode des bornes. Enfin nous exposons l'utilisation réservée à l'étude des collocations dans les textes.

Une partie intermédiaire discute des limites et des apports possibles à la mise en œuvre d'une méthodologie de classification automatique de termes.

La troisième partie de ce chapitre présente les catégories d'outils qui mettent en oeuvre des processus de classification automatique de termes dans des solutions de système d'information. Nous mettons l'accent sur des outils qui présentent un intérêt pour la fouille de texte: la recherche documentaire, les agents intelligents, le filtrage/routage, la bibliométrie et la classification de textes.

1. Méthodes de classification non supervisées

1.1 Problème de classer

La statistique textuelle a réellement pris son essor depuis l'avènement de l'informatique après 1945. Avant guerre quelques rares études ont montré l'intérêt de relever et d'analyser la distribution des mots [Estoup, 1916][Zipf, 1935][Rugg, 1941][Yule, 1944][Yardi, 1944] notamment pour l'étude de la sténographie et du style littéraire par des méthodes quantitatives. Les méthodes de classification (taxinomies, réseaux de neurones, analyses factorielles) étaient connues avant même leur mise en forme algorithmique. Les premiers algorithmes de classification automatique prennent forme en 1955 [Sokal & Sneath, 1963] et ne cessent de s'améliorer dans les années 60 en formant de réelles communautés d'études donnant des applications immédiates en indexation de documents. C'est à cette époque que la statistique textuelle prend ses lettres de noblesse dopée par l'informatique lui permettant de valider et d'appliquer les méthodes (figure 2.1).

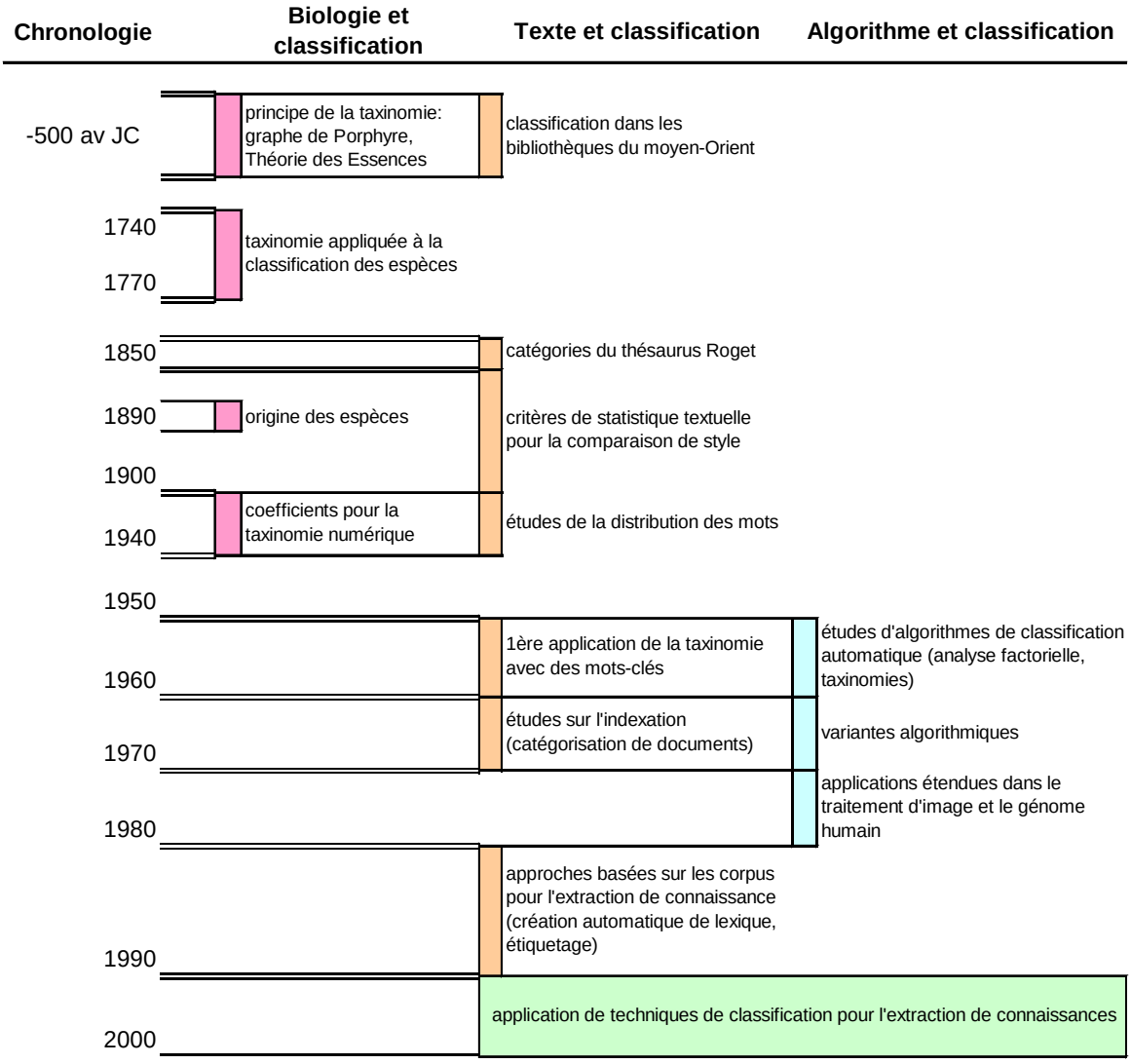


Figure 2.1 Synoptique de l'évolution de la classification automatique

[Sokal & Sneath, 1963] ont écrit le premier ouvrage marquant sur l'intérêt d'utiliser des techniques d'analyse multivariées à base statistique et informatique. D'après [Sokal & Sneath, 1963] une classe est ordinairement définie par référence à un ensemble de propriétés qui sont

à la fois nécessaires et suffisantes (par stipulation) pour l'appartenance à une classe. Il existe un parallèle étroit entre la ressemblance de famille de Wittgenstein (du "sens") et l'affinité taxinomique. La terminologie de la classification a été marquée par l'analogie mécanique d'un nuage de points matériels: axe d'inertie, centre de gravité, nuage de points, masse, poids, déformation du nuage, théorème de Huyghens, dispersion.
Soit I une ensemble de variables, une partition totale est une suite d'ensembles tels que chaque élément de I appartienne à un des ensembles. Une partition partielle est telle qu'un élément de I peut n'appartenir à aucun des ensembles. Une partition recouvrante est telle qu'un élément de I peut appartenir à plusieurs ensembles. Une classification se ramène à rechercher une partition ou des partitions emboîtées (i.e. une hiérarchie).
La classification est un problème hautement combinatoire. Le nombre de partitions de n objets est le nombre de Bell, $B_n=\frac{1}{e}\sum_{k=1}^{\infty}\frac{k^n}{k!}$ (somme de toutes les partitions de 1 éléments, 2 éléments …, n-1 éléments). Par exemple pour *n=4* objets (a, b, c, d) ce nombre[4] vaut 15 et toutes les partitions possibles sont:
partition à 1 classe (abcd),
partitions à 2 classes (ab, cd), (ac, bd), (ad, bc), (a, bcd), (b, acd), (c, bad), (d, abc)
partitions à 3 classes (a, b, cd), (a, c, bd), (b, c, ad), (b, d, ac), (d, c, ab), (d, a, bd),
partition à 4 classes (a, b, c, d) .
En revanche pour *n=26* objets ce nombre vaut $1{,}6.10^{21}$; pour un Pentium Céléron de fréquence d'horloge 500MHz un temps de cycle vaut 2.10^{-9} seconde. Si on admet qu'il faut 1 cycle pour 1 partition, on obtient 100 années de calcul. Les techniques de classification visent donc à optimiser les capacités de calcul tout en convergeant vers une partition acceptable du point de vue interprétatif.

La classification est un processus d'édification d'une partition ou d'une hiérarchie de classes. Le classement est l'opération qui consiste à affecter un élément à un système de classes déjà conçues. Le regroupement d'objets (dans notre cas les termes) en classes traduit certaines propriétés communes. Il fonctionne de la manière suivante [Hansen & Jaumard, 1997]:
1. définir les propriétés dont on s'occupe, et être capable de donner des valeurs pour chaque propriété;
2. créer un vecteur de longueur n avec les n valeurs numériques de chaque objet à classer;
3. visualiser le vecteur de n-dimensions comme un point dans un espace à n-dimensions, aussi visualiser ces points de la classe sont proches les uns des autres.

La procédure conduit aux points suivants :
1. les propriétés, caractérisant les objets à classer, utilisées dans le vecteur;
2. la distance métrique ou le coefficient de similarité utilisée pour décider si 2 points sont « proches »;
3. l'algorithme utilisé pour regrouper.

Un problème de classification soulève les questions suivantes :
1- critère, but de la classification;

[4] Ce calcul correspond à des partitions totales et non recouvrantes. Dans le cas du calcul de partitions partielles il faut d'abord dénombrer l'ensemble des n-uplets de l'ensemble des objets de base : {∅}, {a}, {b}, {c}, {d}, {ab}, …, {abcd} c'est à dire $s=\sum_{p=1}^{k} C_n^p = 2^k$ soit s-{∅}=15 pour k=4 objets. Ensuite on cherche les ensembles de n objets (n<16) de cet ensemble ce qui revient aussi à calculer 2^s . Ce nombre dans notre exemple vaut 32768 partitions. Un exemple de partition à 4 classes : {a}, {ab}, {bd}, {abcd}.

2- axiomatique, justification pour poursuivre le but;
3- choix du type de classification, contraintes à considérer;
4- complexité, difficulté d'implémenter;
5- choix algorithmique, comment peut-on implémenter;
6- interprétation, et signification obtenue.

Les types de classification sont en fait des formes dérivées de partition :
(1) sous-ensemble C de l'ensemble des objets O;
(2) partition totale;
(3) partition non totale;
(4) partition recouvrante;
(5) hiérarchie (partition par coupure).

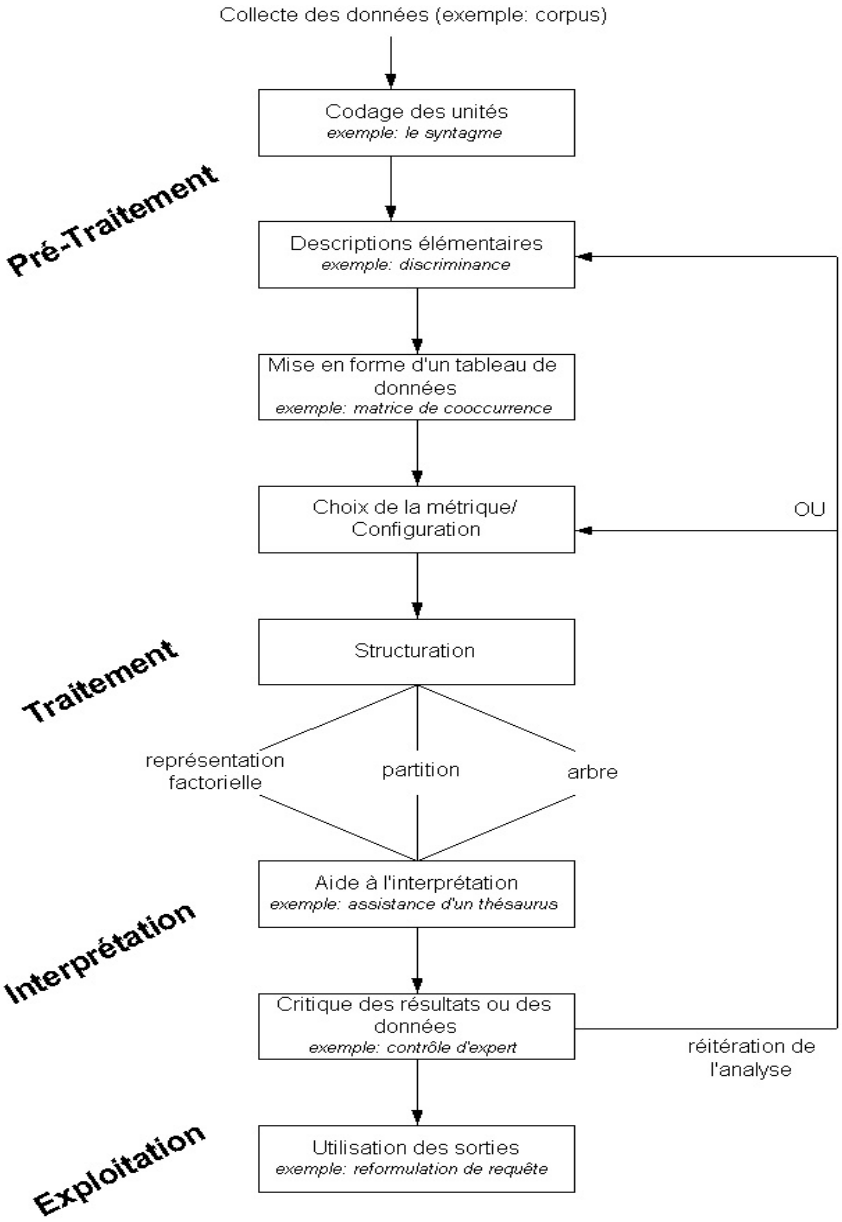


Figure 2.2 Schéma des étapes d'un processus de classification automatique

La programmation mathématique est appliquée avec succès depuis 1970 à la classification (figure 2.2). Il existe aussi depuis 1975 la classification floue dans laquelle les entités ont un

degré d'appartenance dans une ou plusieurs classes, et depuis 1983 la hiérarchie de partitions non totales.
Plusieurs stratégies ou critères d'affectation d'un élément à une classe ont été développés :
1- chaîne (angl. String, on choisit un terme et on adjoint l'élément le plus connexe à ce dernier et ainsi de suite);
2- étoile (angl. Star, on fixe un élément et on affecte tous les éléments adjoints à ce terme modulo un seuil);
3- clique (on teste si un élément donné est lié à tous les membres d'une classe modulo un seuil fixé);
4- d'autres critères d'une classification hiérarchique seront expliqués comme le simple lien, le lien moyen, le diamètre. Dans un espace euclidien on peut comparer avec un centroïde : somme des carrés, variance, rayon continu, étoile continue.

Le début de la classification automatique a été marqué par la construction de dendrogrammes [Mayr et al, 1953][Kelus & Lukaszewicz, 1953). Ceux-ci en fait ne remettent pas en cause le travail interprétatif du biologiste qui choisit lui-même le seuil de similarité pour obtenir ses classes (sa partition). Le travail automatisé facilite grandement le travail lourd et fastidieux du calcul des similarités. Pour cette double raison la classification hiérarchique a été et reste le symbole d'une classification automatique à base statistique. Dans l'algorithme de classification hiérarchique: chaque cycle peut s'accompagner d'un nouveau calcul des similarités de la matrice de similarité; la classe (ou les classes) trouvée(s) au cycle précédent prenant la place d'individus simples.
La classification définit un ordre sur les objets qu'elle classifie. De façon naturelle elle crée une relation d'équivalence. On dit que la relation R est une relation de préordre sur J si elle satisfait aux axiomes suivants:

$\forall j \in J \Rightarrow R(j, j)$ réfléxivité
$\forall j, k, l \in J\ R(j, k)$ et $R(k, l) \Rightarrow R(j, l)$ transitivité
avec l'axiome suivant R est une relation d'ordre sur J
$\forall j, k \in J\ R(j, k)$ et $R(k, j) \Rightarrow j = k$
avec l'axiome suivant R est une relation d'ordre total sur J
$\forall j, k \in J\ R(j, k)$ ou $R(k, j)$.

1.2. Modèle de représentation des données

L'étape fondamentale dans une méthode de classification est la représentation des données sous forme de matrice. La plupart du temps cette matrice sera une matrice de contingence c'est-à-dire un tableau dont les cases sont des entiers définis dans un même système de mesure et d'unité homogène [Zytkow & Zembowicz, 1998]. D'autres types de tableaux comme les tableaux de mesure ou les tableaux logiques (modalités 1/0) peuvent être rencontrés. Par extension un tableau de fréquences sera assimilé à un tableau de contingence. On recueille les k(i,j) observations de l'événement (i,j). Le tableau doit être soumis à 2 critères importants : l'homogénéité et l'exhaustivité.

Dans le cas d'une matrice de cooccurrence entre termes pris dans des données textuelles on évaluera l'élément générique k(i, j) de la matrice de la façon suivante:
$k(i, j) = \text{card}(i \text{ et } j \in I;\ cooc(i, j) \geq 0)$ où cooc(i, j) définit le nombre de fois que i et j apparaissent dans un même voisinage syntaxique (par exemple dans la même phrase).

Il existe quelques variantes quant à l'exploitation de ces données. La plus courante est le stockage de la matrice. Une variante propose de stocker le minimum et le maximum des

colonnes de données. Enfin une autre variante propose de stocker les données inverses (index inversé).

1.2.1. Modèle de vecteur d'objets

Le modèle vectoriel est le modèle qui fait office d'état de l'art en recherche documentaire. Il s'appuie sur une représentation mathématique bien établie et simple à mettre en œuvre du point de vue informatique. Bien que formellement très simple et peu consolidé par des considérations linguistiques, ses performances sont très peu dépassées.
Deux types d'espaces sont généralement considérés: l'espace des documents et l'espace des mots. Dans un espace donné l'unité métrique est arbitraire et l'espace est orthogonal. Les coordonnées sont assignées à chaque élément de l'ensemble des documents ou des mots. Un mot/document représente un vecteur de l'espace. On peut projecter un document dans l'espace des mots donnant lieu au produit scalaire $<\mathbf{m}_i, \mathbf{d}_j>= a_{ij}$ correspondant au nombre a_{ij} de fois que le mot i apparaît dans le document j.
On voit donc que ce croisement d'espace peut facilement être assimilé à une grille où chaque position de l'axe des lignes est un mot/documents et chaque position de l'axe des colonnes est aussi un mot/document. On quantifie la valeur au croisement d'un élément d'un axe avec un élément d'un autre axe. On aboutit à une représentation matricielle du croisement l'espace vectoriel par lui-même ou par un autre.

1.2.2. Modèle des N-grammes

Le modèle des N-grammes est un modèle de représentation du langage particulièrement utilisé pour étudier la prédictibilité d'apparition de suites de chaîne. On qualifie de N-grammes aussi bien des n-uplets de caractères que des n-uples de mots. Dans une matrice de contingence au lieu de croiser des objets entre eux on va croiser des n-uplets entre eux et appliquer ensuite un algorithme de classification. Ainsi pour un alphabet de 26 lettres on obtient 26^2 = 626 bigrammes ou 26^3 = 16 276 trigrammes, pour un dictionnaire de 20 000 mots on obtient $20\ 000^2$ = 400 millions de bigrammes et $20\ 000^3$ = 8 000 milliards de trigrammes. On construit un modèle à partir de données d'apprentissage pour déduire la probabilité d'avoir telle suite de N-grammes [Lelu et al, 1998].

1.2.3. Modèle de l'information syntaxique

La classification utilisant l'information syntaxique exploite la nature du rôle d'une donnée dans le texte source dont il est issu. Ce formalisme est surtout centré sur le schéma verbe-objet [Hindle, 1990][Pereira et al, 1993][Grefenstette & Teufel, 1995]. Ce modèle utilise des composants phrastiques très précis qui doivent être détectés systématiquement dans tout le corpus avec une bonne précision. Le tableau de données va recueillir la distribution des couplages entre noms et verbes via une relation verbe-objet.

Noms \ verbes	acheter	finir	laver	toucher	ouvrir	commencer
Conférence	0	1	0	0	1	1
Meeting	0	1	0	0	1	1
Rideaux	1	0	1	1	1	0
Fenêtre	1	0	1	1	1	0

Une ligne de la matrice donnera le vecteur associé à un nom de l'ensemble des objets à classer V(« Fenêtre »)=<1,0,1,1,1,0>.
Les vecteurs peuvent être normalisés par la fréquence de l'objet et être utilisés comme valeur de probabilité plutôt que fréquence ainsi:
V(objet)=<P(v1|n),P(v2|n),…,P(vk|n)> où P(vk|n) est estimé par (Fréquence de la relation vk)R(n)/(fréquence de l'objet n).

Dans l'exemple, les deux thèmes (conférence, meeting) et (rideau, fenêtre) apparaissent clairement comme ayant des verbes typiques et exclusifs. Dans le cas réel les verbes ont un usage beaucoup plus large que restrictif à une thématique. Les relations entre nom et verbes doivent être explicites et fréquentes dans le corpus, ce qui nécessite un corpus volumineux de plusieurs millions de mots. [Gale et al, 1992] montrent que de l'information peut se trouver à plus de 100 mots d'un autre sans préciser la proportion de cette mesure. L'étude des dépendances courtes occasionne donc une perte d'information sur la sémantique contextuelle.

[Pereira et al, 1993] applique une classification distributionnelle de noms en utilisant une similarité définie par l'entropie relative (annexe 3) D(p,q) où p et q sont 2 distributions de probabilité à comparer ; par exemple $q_n(v)$ représentera la fréquence relative du nom n avec le verbe v dans le cadre d'une approche prédicative. Si q_{pomme}(manger) = 0.2 cela signifie que "pomme" apparaît 2 fois sur 10 en relation syntaxique avec "manger". Des centroïdes de classes sont déterminés comme étant la somme des distributions d'un membre de la classe. Dans le cas où q=0 alors D=0. Cela arrive souvent lorsqu'on étudie des distributions à courte distance syntaxique conduisant à des matrices très creuses (noms-verbes). D'où l'intérêt d'une approche par centroïde évitant le problème des singularités.

1.2.3. Modèle de distance de similarité

Noms \ Attributs	**jaune**	**rouge**	**manger_objet**	**jeter_sujet**	**glisser**
banane	1	0	0	1	1
pomme	0	1	0	1	0

On distingue 2 principales mesures pour dégager un rapprochement entre 2 objets. La mesure la plus immédiate est la notion de distance issue du formalisme de l'espace vectoriel et répondant à ses propres axiomes (annexe 3). La distance va utiliser un vecteur d'attributs de dimension constante égal au cardinal de l'ensemble des attributs. Le coefficient de corrélation sera utilisé aussi avec cette approche de vecteur d'attributs. La deuxième mesure principale est celle du coefficient de similarité qui fait interagir des coefficients de présence commune ou d'absences. Ces coefficients peuvent faire intervenir soit des coefficients de présence uniquement soit des coefficients de présence et d'absence. Ils peuvent être transformés en semi-métrique.

Métrique ou distance inter-objet.
Notons E l'ensemble des n objets à classer. Une distance est une application de E×E dans R+ telle que :

$d(i, j) = d(j, i)$ (1)
$d(i, j) \geq 0$ (2)
$d(i, j) = 0 \Leftrightarrow i = j$ (3)
$d(i, j) \leq d(i, k) + d(k, j)$ (4)

Dans un espace métrique muni d'une distance d on appelle boule fermée de centre $j \in E$ et de rayon r l'ensemble $B(j, r) = \{x \in E \mid d(a,x) \leq r\}$.

Indice ou coefficient de similarité
La similarité est régie par les relations (1) et (2)
Une similarité est une application: s, telle que

$s(i, j) = s(j, i)$ (5)
$s(i, j) \geq 0$ (6)

$s(i, i) \geq s(i, j)$ (7)

Critère de monotonicité: $\forall c, c', c'' \subseteq S : \min(sim(c,c'), sim(c,c'')) \geq sim(c, c' \cup c'')$ quand $sim(c,c' \cup c'')$ est défini; (où S est l'ensemble des classes)
Ce critère permet de garantir que 2 classes dissimilaires dans l'arbre (loin l'un de l'autre) ne soient pas en fait similaires.

indice généralisé
KBG est un système qui sans être algorithmique décrit une méthode de généralisation de la similarité [Bisson, 1996]. Pour KBG chaque objet est uns instanciation de prédicats et de valeurs numériques qui dépasses la logique propositionnelle. On décide d'évaluer la similarité de deux ensembles définie par la conjonction d'objets par la similarité entre chacun de ces objets. Cette similarité favorisera l'appariement d'objets qui en moyenne sont en même relation plus qu'ayant des caractéristiques communes. Pour tout couple d'objets T1 et T2 de même prédicat P d'arité K+L avec X et Y apparaissant au même rang r:
Leur similarité se définit comme suit:

$$T_1 = P(X_1,...,X_r = X,...,X_K, U_1,...,U_L)$$

$$T_2 = P(Y_1,...,Y_r = Y,...,Y_K, V_1,...,V_L)$$

X et Y sont des entités, U et V sont des valeurs

On appelle St la similarité entre objets, Se la similarité entre entités, Sg la similarité entre valeurs.

$$S_t(T_1,T_2) = \omega(P) * \left(\frac{1}{K}\sum_{k=1}^{K} S_e(X_k, Y_k)\right) * \left(\frac{1}{L}\sum_{l=1}^{L} S_g(U_l, V_l)\right)$$

$$S_e(X,Y) = \frac{\sum_{(T_1,T_2)\in M(X,Y)} S_t(T_1,T_2)}{\max\left(\sum_{P(.\cdots.,Y,\cdots.)} \omega(P), \sum_{P(.\cdots.,Y,\cdots.)} \omega(P)\right)},$$

w est un poids arbitraire donnant plus d'importance à certains prédicats qu'à d'autres. M(X,Y) est l'ensemble des couples d'objet de même prédicat dans lesquels X et Y apparaissent au même rang.
Cette solution n'est ni économique en temps de calcul ni incrémentale. Dans le meilleur des cas on aboutit à un système linéaire liant les différentes similarités.

1.3 Aspects probabilistes d'une classification automatique

L'aspect probabiliste d'une classification se rapporte à l'évaluation d'une similarité ou d'une association entre 2 entités comme un objet et un attribut, ou une classe et un paramètre de fonction de distribution. Le calcul est donc souvent indépendant de la représentation des données. En effet, on trouve des coefficients d'association probabiliste aussi bien avec des modèles de vecteurs de caractéristiques qu'avec des descriptions d'attributs symboliques.
Il est presque impossible de présenter un algorithme unique ou générique concernant la description probabiliste d'extraction de classes puisque chaque méthode pose ses propres hypothèses de fonctions de distribution et de dépendances conditionnelles avec son lot d'hypothèses de simplification pour éviter l'explosion exponentielle de recherche des paramètres.

Les modèles probabilistes prennent en compte des fonctions de distribution paramétriques (fonction gaussienne, fonctions multinomiales…) exigeant de trouver les paramètres optimum

(moyenne, écart-type, poids…) pour maximiser la vraisemblance. Par exemple une classe sera représentée par une fonction de probabilité f, le but sera d'estimer des paramètres concernant les classes grâce à la vraisemblance (probabilité d'avoir la totalité d'un système de variables) en estimant les paramètres grâce au données observées:

probabilité de choisir une classe dans la distribution des des g classes $0 \le \omega \le 1$ et $\sum_{i=1}^{g} \omega_i = 1$

$L(Z,\theta) = \prod_{j=1}^{n} f^{M}(j) = \prod_{j=1}^{n} \left[\sum_{i=1}^{g} w_i f(z_j, \theta_i) \right]$ vraisemblence avec $\theta = (\theta_1,...,\theta_g)$,

$\theta = (\mu_i, M_i)$ vecteur repérens tan t la moyenne et matrice de cov ariance qu'on cherche à estimer

$\frac{\partial L(Z,\theta)}{\partial M} = 0$ max imisation de la vraisemblance par rapport au paramètre M

Certaines stratégies utilisent le parcours de l'espace des paramètres pour rechercher les maxima locaux. Le maximum de vraisemblance couple les fonctions de distribution pour chaque cas avec un cas pour chaque classe dans la phase d'apprentissage ce qui rend difficile la découverte d'une classe. La recherche dans l'espace des paramètres pour trouver un maximum absolu est exponentielle en temps de calcul compte tenu du nombre de paramètres et du caractère multiplicatif des probabilités. Nous verrons d'autres défauts à ces approches après avoir décrit le modèle bayésien qui apporte les simplifications nécessaires aux dépendances conditionnelles.

Un modèle bayésien naïf (angl. Naive Bayesian Model) assume que tous les attributs sont conditionnellement indépendants. Un réseau bayésien (angl. Bayesian Belief Network) décrit une distribution de probabilités conjointes sur l'ensemble des variables en spécifiant un ensemble d'hypothèses d'indépendances conditionnelles et un ensemble de probabilités conditionnelles. (X est conditionnellement indépendant de Y signifie que P(X|Y,Z)=P(X|Z).
Un réseau bayésien est un graphe acyclique dirigé dans lequel chaque nœud représente une variable aléatoire, et les arcs entre les nœuds représentent une dépendance probabiliste entre le nœud et ses parents (figure 2.3).

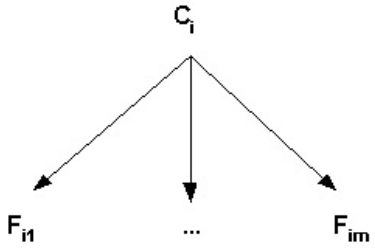


Figure 2.3 Nœud bayésien

C_i représente un nœud "objet lié au concept C_i"
$F_{i1}…F_{im}$ représente les attributs de l'objet
$P(C_i)$ probabilité que d'avoir le concept C_i
$P(F_{ij}/C_i)$ probabilité telle que la caractéristique F_{ij}, où j=1,m, est présente dans le concept C_i
On calcule $P(C_i/F_{i1},…,F_{im})$ probabilité d'avoir C_i sachant la présence de $F_{i1},…,F_{im}$

$$P(C_i / F_{i1},...,F_{im}) = \frac{P(C_i)P(F_{i1},F_{im} / C_i)}{P(F_{i1},...,F_{im})}$$

$P(F_{i1},...,F_{im} / C_i) = P(F_{i1} / C_i)...P(F_{im} / C_i)$, hypothèse bayésienne naive

La règle de décision peut s'écrire:

$$g(C_i / F_{i1},...,F_{im}) = P(C_i)\prod_k P(F_{ik} / C_i), \text{terme constant de normalisation qui peut être ignoré}$$

Cette règle signifie que l'on peut affecter un objet X_i décrit par les attributs $F_{i1},\ldots,F_{im}$ à la classe C_i si $g(C_i/F_{i1},\ldots,F_{im})$ est maximum pour cette classe.
On peut simplifier par une règle de décision linéaire:

$$g(C_i / F_{i1},...,F_{im}) = \sum_R I(F_{ik}).w(F_{ik}, C_i)$$

où $I(F_{ik})$ est un indicateur < 1 si F_{ik} est présent pour l'objet 0 sinon w est un coefficient (poids) correspond à une paire spécifique concept/forme.

AutoClass [Cheeseman et al, 1988] est une application qui développe un modèle bayésien avec modèle mixte fini des fonctions de distributions décrivant les paramètres des fonctions de distribution et des attributs. Au lieu de raisonner sur une maximisation de la vraisemblance, AutoClass parcourt l'espace des paramètres pour trouver des maxima locaux grâce au modèle de fonction de distribution et au nombre de classe. Par exemple une fonction de distribution peut être une fonction de Bernoulli (attributs valués discrets) ou une densité gaussienne (attributs numériques). Un premier niveau de parcours pose comme hypothèse de fixer une distribution pour toutes les classes et une recherche sur toutes les classes. Un deuxième niveau permet de faire varier une distribution individuelle d'une classe à l'autre.

Ces modèles probabilistes basés sur le parcours de l'espace des paramètres et une maximisation de la vraisemblance sont fondés sur une distribution continue de probabilités qui décrivent assez peu le caractère d'événement purement discret avec des attributs mal définis. Les hypothèses de simplification nécessaires à l'optimisation de ces modèles posent l'indépendance conditionnelle comme hypothèse fondamentale de simplification (exemple indépendance des paires de termes (x,y)) ainsi que des distributions uniformes ou des probabilités relatives. Ces hypothèses traduisent assez peu les caractéristiques relationnelles non aléatoires de variables liées. Des modèles métaphoriques comme ceux liés à la physique statistique (entropie, énergie libre, recuit simulé..), s'il peuvent présenter un intérêt calculatoire, n'en sont pas moins difficile à interpréter quand ils ne relèvent pas de leur domaine d'origine c'est-à-dire la matière.

Une description, intéressante et précurseur, des associations par des probabilités est l'information mutuelle provenant de la théorie de Shannon.
L'information mutuelle est un critère qui permet de maximiser les contraintes d'association des objets entre classes [Brown et al, 1992][Li & Abe, 1998][Baker & McCallum, 1998].

On prend l'exemple de mots qui sont contraints par leur apparition syntaxique dans une phrase.
Soit V(x) le vecteur représentant le contexte du mot x, on a
$V(x)=<|m_1|,|m_2|,\ldots,|m_n|>$
m_i est le nombre de fois que le mot m_i suit le mot x dans un corpus
En normalisant par le nombre d'apparition de x on obtient la probabilité conditionnelle $P(m_i|x)$
A partir de ces probabilités conditionnelles on dérive une information mutuelle concernant l'apport d'information d'un mot avec un autre.
Par définition l'information mutuelle entre x et y :

$$I(x;y)=(-\log P(x)) - (-\log P(x| y))=\log \frac{P(x,y)}{P(x)P(y)}$$

Pour illustrer le comportement de I par rapport à la relation entre x et y, prenons deux objets qui n'ont pas de corrélation entre eux on aura P(x, y) = P(x).P(y)
D'où I(x,y) =0
Dans le cas contraire où x et y sont systématiquement liés : P(x,y)=P(x), I(x,y) devient très grand.
Le critère de classification sera de maximiser l'information mutuelle lors de la création des classes d'objets.
Dans le cas d'un ensemble de variables aléatoires le critère sera généralisé au nombre de valeurs possibles à l'information mutuelle moyenne. On considère qu'une variable peut avoir les valeurs (w^1,…w^ω). Cette information moyenne peut aussi être définie avec la notion d'entropie conditionnelle :

$$I(X;Y)=\sum_{y=1}^{\omega}\sum_{x=1}^{\omega}P(w^x,w^y)I(w^x;w^y) \ .$$

Dans la problématique de classification, l'information mutuelle moyenne se ramène à calculer la somme des informations moyennes des classes prises deux à deux. On recherche la perte minimale d'information, la métrique correspond donc à la maximisation de l'information mutuelle moyenne. L'idée est de trouver des groupes dont la perte est minimale. En général la perte est d'autant plus faible que les vecteurs sont similaires.

Du point de vue algorithmique, si le temps de calcul était sans importance il suffirait de traiter toutes les combinaisons possibles et de retenir le taux d'information mutuelle maximal. En pratique il y a trop de groupes à traiter pour que ce soit soluble. Dans ce type de problème on part d'un algorithme "glouton" (angl. *Greedy*) . L'algorithme se démarre avec w classes, un pour chaque mot. Il combine ainsi les deux classes ce qui aboutit à une perte minimale d'information mutuelle, et répète le processus jusqu'à ce que le nombre de classes désiré soit atteint. Il n'est pas garanti de trouver les meilleures classes. Dans le cas d'un nombre de mots assez grand, on se fixe un nombre réduit de classes initiales (1000 par exemple pour 250000mots).

Le processus est dépendant de l'ordre d'évaluation par paire et de l'amorçage (angl. Bootstrap). Brown utilise un modèle de trigramme et mesure une information mutuelle avec un modèle par classes. Le modèle de calcul par classes perd en précision mais utilise moins de paramètres qu'un modèle par mot, et doit permettre de mieux saisir les régularités exprimées par le modèle de trigramme.

L'algorithme suivant est intéressant car il couple les notions de distribution, de probabilité et de classification automatique. [Tishby et al, 1999] présente une méthode hiérarchique qu'il appelle méthode de classification du goulot d'information (angl., agglomerative information bottleneck). Cette méthode correspond beaucoup plus à la définition d'une distance probabiliste basée sur les distributions qu'une stratégie probabiliste globale de classification. La méthode considère une distribution d'un objet x de l'ensemble X (exemple les noms) selon une variable y de l'ensemble Y (exemple les verbes) $p(y|x)=\frac{n(y,x)}{\sum_{y\in Y} n(y,x)}$ où n(y,x) est le nombre de cooccurrences de x et y dans le corpus. L'information mutuelle I(X,Y) est la mesure statistique qui va permettre de regrouper 2 objets en fonction de leurs distributions. La variation δI doit être minimisée pour autoriser un regroupement. Une fonction d'énergie libre dépendant de l'information mutuelle et d'un multiplicateur de Lagrange lié à une fonction de distortion (moyenne des distances d'un élément avec une classe par sa probabilité

d'appartenance). La minimation par analyse variationnelle conduit à une solution de la probabilité d'appartenance de x à une classe qui est proportionnelle à l'exponentielle de la distance de Kullback-Leibler (D_{KL}) entre la distribution de y par rapport à x et la distribution de y par rapport à la classe.

Algorithme

1 On donne la matrice de probabilité p(x,y), N=|X|, M=|Y|
2 On inititalise la partition Z de X avec tous les éléments de X ($z_i=x_i$, $p(z_i)=p(x_i)$, $p(y|z_i)=p(y|x_i)$, $p(z|x_j)=1$ si j=i et 0 sinon) et calculer $\delta I_Y(i,j)=(p(z_i)+p(z_j)).JS_{\Pi 2}[p(y|z_i),p(y|z_j)]$ (chaque $z_{i,j}$ pointe vers un couple de Z)

$$JS_{\Pi 2}[p_i, p_j] = \pi_i D_{KL}[p_i \| p'] + \pi_j D_{KL}[p_j \| p'],$$

$$\text{ou } \pi_i = \frac{p_i}{p'} \text{ probabilité de la classe i par rapport au regoupement de i et j}$$

3 Pour chaque classes t de 1 a N
Trouver(α,β) = $argmin_{i,j}\{d_{i,j}\}$ (si plusieurs minima en choisir un au hasard)
Grouper $\{z_\alpha, z_\beta\}$ en z'
p(z')=p(z_α)+p(z_β)
p(y|z')=(p(z_α,y)+p(z_β,y))/p(z') pour chaque y de Y
p(z'|x)=1 si x $\in$ $z_\alpha \cup z_\beta$ et 0 sinon pour chaque x de X
Mettre à jour Z = {Z-$\{z_\alpha, z_\beta\}$}$\cup${z'} (Z est une partition de N-t classes) et les pointeurs /coûts z' de $d_{i,j}$

1.4. Méthodes des plus proches voisins ("k-moyennes")

[Rogers & Tanimoto, 1960] semblent avoir imaginé une méthode qui jette les premières bases d'une méthode des k-moyennes. Ils sélectionnent un objet cible grâce à deux paramètres qui somment les caractéristiques communes et forment une hiérarchie avec cet objet cible par rapport à un deuxième objet cible (classification nodale). [Forgy, 1965] et [MacQueen, 1967] élaboreront une méthode des k-moyennes indépendamment d'une construction de hiérarchie. Le principe d'une méthode des k-moyennes est de choisir k points de l'espace des individus qui serviront de repère aux futures classes. La classification des plus proches voisins est aussi connue pour être liée à la recherche d'un MST (arbre de couverture minimum ou, angl., "minimum spanning tree"). Dans la suite M est le nombre de classes et N le nombre d'individus à classer.

1.4.1 Méthodes avec simple passe (angl. single-pass)

On désigne k_c centres de groupes prévisionnels $C_1^0, C_2^0, \ldots, C_k^0, \ldots, C_{k_c}^0$. Une règle d'affectation utilisant une distance ou coefficient de similarité permet d'aggréger les objets plus proche à un centre qu'un autre, On aboutit à $P_1^0, P_2^0, \ldots, P_k^0, \ldots, P_{k_c}^0$. On redéfinit ensuite les centres de gravité de chaque classe. Ceux-ci étant redéfinis, on recompose les k_c classes P_k^i pour cette itération i [Forgy, 1965][Salton & McGill, 1983].
Similarité: traitée entre l'entrée et tous les représentants des classes existants
Par exemple --- l'algorithme du coefficient de couverture permet de sélectionner un ensemble d'objets comme centres de classe, d'assigner chaque objet à une classe qui le couvre au maximum.
Temps: O(N log N).
Espace: O(M).
Avantages: simple, demande seulement une passe avec les données, peut être utile comme point de départ pour une méthode de réallocation.

Limites: produit de grosses classes tôt dans le traitement; les classes formées sont dépendantes de l'ordre de traitement des données.

Algorithme

```
1-Désigner M objets comme les représentants de M classe
2-Calculer la similarité entre un objet et chaque classe, garder la
trace de la plus grande, Smax
3-Si Smax est plus grande qu'un seuil, ajouter l'objet à la classe
correspondante, sinon créer une nouvelle classe avec l'objet comme
centroïde.
4-Si un objet subsiste, revenir à l'étape 2
```

1.4.2 Méthode de réallocation (angl. reallocation)

Similarité: permet diverses définitions de similarité de cohésion du classe.
Type de classes: Amélioration d'une classification initiale en entrée.
Temps: O(MN).
Espace: O(M + N).
Avantages: permet le traitement de grands ensembles de données comparé à d'autres méthodes
Limites: peut prendre du temps pour converger vers une solution, dépend du type de critère (approprié ou non à la structure des données)

Algorithme

```
1-Choisir M classes représentants ou centroïdes
2-Affecter chaque objet au centroïde le plus similaire
3-Recalculer le centroïde pour chaque classe
4-Répéter les étapes 2 et 3 jusqu'à que le changement soit minime
d'une passe à l'autre dans l'appartenance des membres aux classes.
```

1.4.3 Méthode des nuées dynamiques et des centres mobiles

Dans la méthode des centres mobiles on définit k représentants des classes, alors que dans les méthodes des nuées dynamiques la classe est définie par un noyau de q objets supposé plus représentatif que le centre de gravité.
On définit la distance $D(i,P) = \sum_{j \in P} d(i,j)$ d'un individu i à une partie P de la partition recherchée. La fonction agrégation-écartement (R1 ou R2) sert à qualifier l'appartenance d'un individu i au noyau m d'une classe C pour l'ensemble des noyaux M

$$R_1(i,C,M) = \frac{D(i,m) * D(i,C)}{\left(\sum_{m \in M} D(i,m) \right)^2} \quad \text{ou} \quad R_2(i,C,M) = D(i,C) .$$

On retient les individus ayant le même comportement pour plusieurs itérations successives de recherche de partition. Ces individus sont appelés formes fortes. Le principe algorithmique reste le même que pour la méthode de réallocation [Diday, 1971].
Une des variantes de l'algorithme des centres mobiles consiste à prendre un individu simple au lieu du centre de gravité pour éviter de tomber dans une zone creuse du nuage. D'autre part on peut aussi réitérer l'algorithme et garder les formes fortes des partitions finales obtenues.

Algorithme

```
1- Définir un noyau pour chaque classe
2- Calcul de la somme de distances d'un point au noyau
3- Déterminer un nouveau noyau
4- S'il reste un objet, revenir à l'étape 2
```

1.5. Méthodes factorielles

1.5.1. Décomposition en valeurs singulières

Les méthodes factorielles ne sont pas à proprement parler des méthodes de classification. Elles constituent un processus interprétatif qui peut mener à des regroupements d'objets par similarité. (angl., Factorial Analysis ou Multidimensional Scaling).

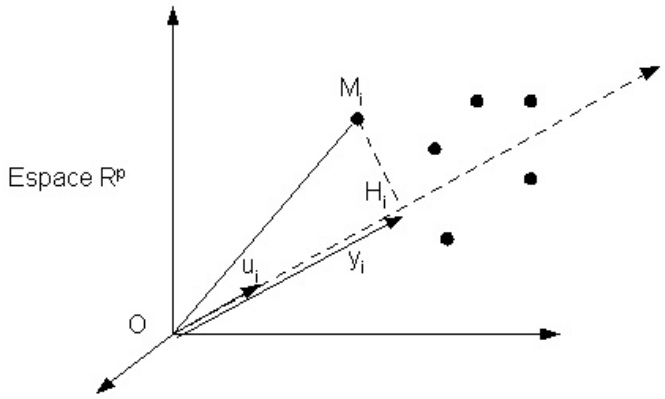


Figure 2.4 Projection d'un nuage sur l'axe d'inertie i

Les méthodes factorielles sont basées sur l'algèbre matricielle et l'obtention de valeurs propres de tableaux de données. Le problème consiste [Saporta, 1988], à partir de points décrits dans une matrice X(n individus*p variables), à trouver la meilleure projection des n "points-lignes" dans l'espace des variables (figure 2.4). Le vecteur résultant est **y** = X**u**. La valeur de la ième composante du vecteur **y** est la longueur OH_i. La liste des coordonnées c_i des individus sur l'axe i forme une nouvelle variable, artificielle, c.

ci = a'M**e**$_i$ = <**a** ;**e**$_j$>M (projection de **e**$_j$ sur l'axe i)

c = XM**a** =X**u** = $\Sigma x^j u_j$

Avec **u**=M**a** et M est la métrique (si la distance est euclidienne M=I)

c est appelée composante principale.

D'une manière générale la notation matricielle permet d'écrire une distance sous la forme :

$$d^2(e_i;e_j) = \sum_i \sum_j (e_i - e_j)^2 = (e_i - e_j)\, M\, (e_{i'} - e_{j'}) .$$

Dans le cas d'une distance euclidienne M=I (1 sur la diagonale, 0 ailleurs), dans une ACP par défaut M vaut l'inverse des variances sur la diagonale, et 0 ailleurs.

La recherche du meilleur vecteur **u** d'après les moindres carrés revient à minimiser

$\sum_i M_i H_i^2$ ou à minimiser $\sum_i O_i H_i^2$ =y'y

c'est-à-dire (X**u**)'X**u** = **u**'X'X**u** sous la contrainte **u'u**=1(nouvelle base normée). En posant :

L(**u**) = **u**'X'X**u** –λ (**u'u**-1) .

On doit avoir $\frac{\partial L}{\partial u}$=0 d'où X'X**u** = λ**u**.

Cela revient à rechercher les valeurs propres λ associées aux vecteurs propres **u** de la matrice de covariance X'X (rappel $\mathrm{cov}(X,Y) = \frac{\sum_i (X_i - \overline{X})(Y_i - \overline{Y})}{n-1}$)

MV **u** = λ**u** (si la distance est euclidienne M=I donc V**u** = λ**u**)

λ est la valeur propre pour le vecteur propre **u**, V est la matrice de covariance

On appelle inertie totale du nuage de points la moyenne pondérée des carrés des distances des points au centre de gravité. L'inertie est la trace de VM, produit de la métrique par la matrice de la variance-covariance. On cherche donc à maximiser l'inertie du nuage projeté, ce qui revient à trouver les valeurs et vecteurs propres de VM.

On aboutit finalement à la représentation graphique des variables (O) dans l'espace des vecteurs propres:
$\mathbf{U}=(\mathbf{u}_1,\ldots,\mathbf{u}_p)$ espace vectoriel des vecteurs propres associés à la matrice de covariance;
$\mathbf{E}=(\mathbf{e}_1,\ldots,\mathbf{e}_p)$ espace des vecteurs initiaux associés aux individus;
$\mathbf{U}=T\mathbf{E}$ relation de changement de base entre U et E;
$\mathbf{E}=T^{-1}\mathbf{U}$ relation de changement de base inverse entre U et E;
$\mathbf{O}=(\mathbf{o}_1,\ldots,\mathbf{o}_p)$ vecteurs associés à chaque individus (profil ligne);
$O=XE$ relation reliant les vecteurs associés aux individus et l'espace vectoriel initial;
$\mathbf{O}=XT^{-1}\mathbf{U}$ relation reliant les vecteurs associés aux individus et l'espace des vecteurs propres.

Une relation d'équivalence permet de passer de l'espace de projection sur les variables à l'espace de projection sur les individus.
On a la même relation concernant la projection des individus dans l'espace des variables dans l'espace des individus R^n $XX'v=\lambda'v$ Pour chaque axe i les valeurs propres sont identiques $\lambda = \lambda'$ on a donc proportionnalité entre v et Xu d'une part, u et X'v d'autre part, d'où la relation d'équivalence:

$$v_i = \frac{1}{\sqrt{\lambda_i}} Xu_i$$

$$u_i = \frac{1}{\sqrt{\lambda_i}} Xv_i$$

La variance d'une composante principale est égale à la valeur propre. Les axes d'inertie les plus représentatifs ont une valeur propre élevée. Un objet $\mathbf{o}_i$ est représenté par une combinaison linéaire de $x_1,\ldots x_p$ de variance maximale.
Facteurs principaux (vecteurs propres) (figure 2.5) :

$F_1=\{F_1(i); i\in I\}$; $G_1=\{G_1(j); j\in J\}$; valeur propre λ_1
$F_2=\{F_2(i); i\in I\}$; $G_2=\{G_2(j); j\in J\}$; valeur propre λ_2
. . . .
$F_\alpha=\{F_\alpha(i); i\in I\}$; $G_\alpha=\{G_\alpha(j); j\in J\}$; valeur propre λ_α
. . . .
$F_m=\{F_m(i); i\in I\}$; $G_n=\{G_n(j); j\in J\}$; valeur propre λ_n
m=card I et n=card J
$F_\alpha(i)$ est la valeur du facteur de rang α au point i de I
$G_\alpha(j)$ est la valeur du facteur de rang α au point j de J

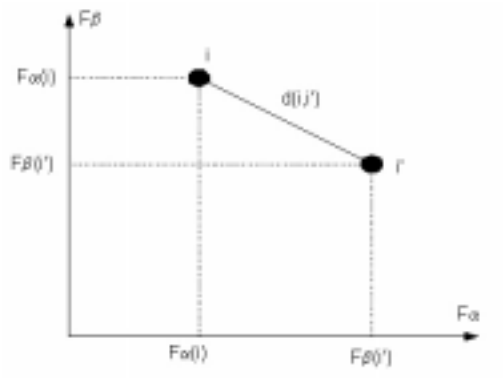


Figure 2.5 Coordonnées de i et i' dans l'espace constitué des axes factoriels α et β

$\lambda_n<...<\lambda_\alpha<...<\lambda_2<\lambda_1$

propriété: $\sum_{i\in I} f_i F_\alpha(i) = 0 \quad \sum_{j\in J} f_j G_\alpha(j) = 0$

$$f_I = \left\{ f_i = \frac{k(i)}{k}; i \in I \right\} \text{avec } k(i) = \sum_{j\in J} k(i,j) \text{ et } k = \sum_{i\in I, j\in J} k(i,j) \text{ ; } k(i,j) \text{ étant l'élément de la matrice de données}$$

$$f_J = \left\{ f_j = \frac{k(j)}{k}; j \in J \right\} \text{avec } k(j) = \sum_{i\in I} k(i,j)$$

1.5.2.Analyse Factorielle des Correspondances (AFC)

L'analyse des correspondances a été établie pour étudier la liaison entre deux ensembles de variables qualitatives. L'origine de la méthode provient de [Guttman, 1941] et l'adaptation aux données textuelles par [Benzécri, 1973]. Cette analyse est très similaire en soi à l'analyse en composantes principales c'est-à-dire à la recherche des axes principaux d'inertie (axes factoriels). Le traitement exploite des matrices carrées dérivées d'une matrice initiale objets/attributs. La plus petite dimension (J) représente, en général, l'ensemble des variables décrivant le phénomène. La méthode s'intéresse donc au croisement J*J et éventuellement à celui des attributs I*I. Les objets sont projetés sur un espace de dimension inférieure (un plan en général) dit plan-factoriel minimisant la différence entre les distances projetées et les distances réelles des objets pris deux à deux.

Les principes méthodologiques se résument à la simultanéité de la projection des variables/individus sur le même diagramme, le principe barycentrique, l'équivalence distributionnelle, l'utilisation de la métrique du chi-2.

Reprenons la méthode :
Le tableau de données est un tableau de contingence N à m1 lignes et m2 colonnes résultant du croisement de 2 variables qualitatives à m1 et m2 catégories respectivement.
On note D1 et D2 les matrices diagonales des effectifs marginaux des deux variables :

$$D1=\begin{pmatrix} n_1 & & 0 \\ & n_2 & \\ & \ddots & \\ 0 & & n_{m1} \end{pmatrix} \quad \text{et} \quad D2=\begin{pmatrix} n_1 & & 0 \\ & n_2 & \\ & \ddots & \\ 0 & & n_{m2} \end{pmatrix}.$$

Le tableau des profils des lignes est alors $\frac{n_{ij}}{n_{i.}}$ est alors $D_1^{-1}N$.

Pour calculer la distance entre 2 profils lignes i et i' on utilise la métrique appelée du chi-2 par similitude avec la distance du test du chi-2 qui compare la distribution aléatoire avec une population espérée grâce à l'écart quadratique pondéré entre les deux populations :

$$d^{2}_{\chi^2}(i,i') = \sum_{j=1}^{m_2} \frac{n}{n_{.j}} \left(\frac{n_{ij}}{ni.} - \frac{n_{i'j}}{ni'.} \right)^2 \quad \text{avec} \quad n = \sum_{i,j} n_{ij}$$

La pondération $\frac{n}{n_{.j}}$ de chaque carré de différence revient à donner des importances comparables aux diverses variables.

Deux AFC (ACP avec distance du chi-2) sont possibles. Avec les *profils lignes* (division d'une ligne par la marginale ligne), on a les éléments suivants :

Le tableau de données $X=D_1^{-1}N$;

Métrique $M=nD_2^{-1}$;

Poids $\frac{D_1}{n}$.

Avec les *profils colonnes* (division d'une ligne par la marginale colonne) on a les éléments suivants :

Le tableau de données $X = D_2^{-1}N'$;

Métrique $M=nD_2^{-1}$;

Poids $\frac{D_1}{n}$.

Les composantes principales correspondent aux vecteurs propres de XMX'D

Pour le profil ligne : $X=D^{-1}{}_1N \Rightarrow XMX'D=D^{-1}{}_1ND^{-1}{}_2N'$

(projection des n variables dans l'espace R^p des individus)

Pour le profil colonne $X=D^{-1}{}_2N \Rightarrow XMX'D=D^{-1}{}_2N'D^{-1}{}_1N$

(projection des p individus dans l'espace R^n des variables).

Considérons 2 éléments j_1 et j_2 de J tels que leur profil sur I soient identiques ($f_I^{j_1} = f_I^{j_2}$).

Si on substitue aux colonnes j_1 et j_2 une colonne j_s telle que $f_{ijs}=f_{ij1}+f_{ij2}$, $f_{js} = f_{j1}+f_{j2,}$

alors la distance distributionnelle entre éléments de I n'est pas modifiée. Il s'agit du *principe d'équivalence distributionnelle*.

Une processus de classification peut s'amorcer grâce à cette méthode de correspondance. Cela présente les deux avantages suivants: description complémentaire des données utilisant différents principes computationnels, deuxièmement, économies de temps de calcul et le cas de grands ensembles de données.

Algorithme

```
1 calcul de l'inverse de la matrice de poids D
      méthode de Cholesky
      on transforme D en produit TT' où T est triangulaire inférieure
```

(1) $t_{1,1} = d_{1,1} \ \ldots \ t_{j,1} = d_{1,j} / t_{1,1} \ \ldots \ t_{n,1} = d_{1,n} / t_{1,1}$

(2) $\forall i = 2,n \ \ t_{i,i} = \sqrt{d_{i,i} - \sum_{k=1}^{i-1} t_{i,k}^2} \quad \forall j = i+1,n \ \ t_{j,i} = \frac{d_{i,i+1} - \sum_{k=1}^{i-1} t_{i,k}.t_{j,k}}{t_{i,i}}$

(3) $\forall i = 1,n \ \ c_{i,i} = 1/t_{i,i}$

(4) $\forall j = 1,i-1 \ \ c_{i,j} = -\sum_{k=j}^{i-1} t_{i,k}.t_{k,j} / t_{i,i}$

c est le coefficient de T^{-1}

reste à calculer le produit matriciel $(T^{-1})'T^{-1} = D^{-1}$

2 calcul de la matrice de covariance (XMX'D)

3 calcul des vecteurs propres

méthode de Jacoby (autres méthodes: Lanczos ou Gens-Householder)

L'algorithme de Jacoby consiste à approcher le vecteur inconnu X de B . U = b par une suite de secteur U(p).

A est la matrice déduite de B par suppression des termes de la diagonale.

V est le vecteur formé des termes diagonaux de B.

Pour réaliser ce calcul, une méthode itérative est utilisée dont l'algorithme est :

répéter

pour i de 0 à N-1 faire

$$(1)\ Y_i \longleftarrow \sum_{j=0}^{N-1} \left(A_{ij} - U_j^{(p-1)}\right)$$

$$(2)\ U_i \longleftarrow \frac{(B_i - Y_i)}{V_i}$$

$$(3)\ Z_i \longleftarrow U_i^{(p)} - U_i^{(p-1)}$$

$$(4)\ U_i^{(p-1)} \longleftarrow U_i^{(p)}$$

jusqu'à ||z|| < epsilon

4 calcul des coordonnées des objets dans l'espace des vecteurs propres

$c_l(i) = N(i).U_l$ l ième coordonnées du l ième objet (N(I) est la ième ligne de la matrice des données).

1.6. Méthodes hiérarchiques descendantes

Une classification hiérarchique descendante réalise un dendrogramme non pas par agrégation, mais par division depuis tous les éléments formant une classe jusqu'à la partition formée de tous les éléments simples. Le processus est généralement dichotomique, bien que tout algorithme fonctionnant par centroïde puisse être appliqué. L'ensemble I est divisé en 2 classes I_1 et I_0; puis ces classes sont divisées en deux I_1 en I_{10} et I_{11} et I_0 en I_{01} et I_{00} et ainsi de suite jusqu'aux classes à un élément.

Pour scinder une classe en deux, une des variantes consiste à prendre le point le plus périphérique d'une classe au lieu de prendre la distance maximale.

Algorithme général

1- Calcul des distances sur I et tri des valeurs par ordre décroissant

2- Evaluation de la distance max $d(i_1, i_0)$ où chaque élément de I est associé à une classe s0 représenté par i_0 et s_i représenté par i_1.

3- Il existe n classes. On considère la classe s_i qui possède le diamètre maximum (distance maximum). On divise s_i en s^a_i et s^b_i en attribuant chaque élément de s_i soit à s^a_i soit à s^b_i

4- Recalcul des diamètres par ordre décroissant, s'arrêtant dès que le nombre de classes = card(I).

Algorithme du chi2 d'Alceste [Reinert, 1986]

1- n1 est le nombre de ECU ("Elementary Context Unit") dans la classe

n2 est le nombre de ECU où le terme est présent

n est le nombre total d'ECU

n12 est le nombre d'ECU dans la classe où le terme est présent

Si j est l'indice courant sur les termes et p l'indice sur les classes, on a:

$chi2=\Sigma_{j=1,m}\Sigma_{p=1,2}(n_{jp} - n_p s_j/N)2/n_p s_j/N$

où $n_1=\Sigma_{j=1,m}\ n_{j1}$ nombre de 1 dans la classe 1

$n_2=\Sigma_{j=1,m}\ n_{j2}$ nombre de 1 dans la classe 2

$N=n_1+n_2$ nombre de 1 dans la matrice ECU*terme

$s_j=n_{j1}+n_{j2}$ nombre de 1 pour le terme j dans la matrice

2- On compare n_{12} à n_1n_2/n par le chi2, on affecte le signe de $n_{12} - (n_1n_2/n)$.

3- On maximise le chi2 avec la table des marges:

	termes en colonne		
	1	j	m
classes en ligne 1		n_{j1}	
2		nj2	

4- Les classes sont générées par dichotomie de la matrice binaire.

5- On réitére le processus n fois.

1.7. Méthodes hiérarchiques ascendantes

1.7.1. Algorithme général de classification hiérarchique ascendant

Les méthodes hiérarchiques ascendantes sont les plus connues des méthodes de classification automatique. Elles aboutissent à un empilement de partitions dont le sommet réuni une partition et la base autant de partition que d'objets à classer ; cet empilement peut se représenter par un arbre composé de branches ayant une bifurcation binaire du sommet vers la base, la raison pour laquelle on les nomme hiérarchique (figure 2.6). L'obtention d'une partition se fait par coupure de l'arbre à un seuil donné. La classification hiérarchique agglomératif conduit à une catégorisation polythétique (plus grand nombre d'attributs en commun).

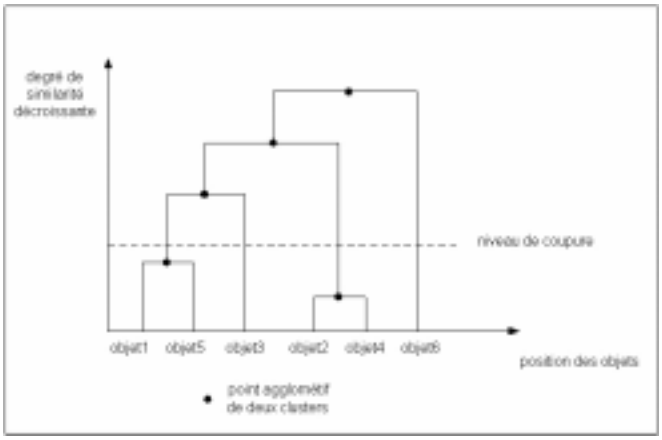


Figure 2.6. Dendrogramme

Toutes les méthodes de classification agglomérative hiérarchique (CAH) peuvent être décrites par l'algorithme général suivant:

Algorithme

1.Identifier les deux points les plus proches et les combiner en une classe

```
2.Identifier et combiner les 2 points les plus proches (considérant
aussi les classes comme des points)
3.Si plus d'une classe subsiste revenir à l'étape 1.
```

Implémentation de l'algorithme général:
L'approche flexible de [Jambu, 1978] propose une formule actualisée de la dissimilarité de Lance-Williams pour calculer les dissimilarités entre une nouvelle classe et des points existants, basée sur les dissimilarités de manière à former la nouvelle classe. Cette formule a 4 paramètres (affectés aux distances), et chaque CAH peut être caractérisée par son propre jeu de paramètres de Lance-Williams [Williams & Lance, 1965]:
$d((a,b);c) = a_1 d(a,c) + a_2 d(b,c) + a_3 d(a,b) + a_4 i(a) + a_5 i(b) + a_6 i(c) + a_7 | d(a,b) - d(b,c) |$
i est l'application indicée sur la hiérarchie si $a \subset b$ alors $i(a) < i(b)$
Pour éviter l'inversion on pose la contrainte a1+a2+a3>1
$a_1, a_2, a_3, a_4, a_5, a_6, a_7 > 0$
$a_7 > -\min(a_1; a_2)$
Par exemple, pour le simple lien $d((a,b);c)=\inf(d(a,c);d(b,c))$ ou $a_1 = a_2 = 1/2$, $a_3 = a_4 = a_5 = a_6 = 0$, $a_7 = 1/2$

Ensuite, l'algorithme ci-dessus peut être appliqué, utilisant les dissimilarités appropriées de Lance-Williams.

Algorithme

```
Approche de la matrice stockée: Utiliser une matrice, et ensuite
appliquer Lance-Williams pour recalculer les dissimilarités entre les
centres des classes. Le stockage est ainsi O(N^2) et le temps de calcul
est au moins O(N^2), mais sera O(N^3) si la matrice est balayée
linéairement.
Approche des données stockées: O(N) d'espace pour les données mais
retraiter les dissimilarités par paires donc le besoin en temps de
calcul deviendra O(N^3)
Approche de la matrice triée: O(N^2) pour calculer la matrice de
dissimilarités, O(N^2logN^2) pour la trier, O(N^2) pour construire la
hiérarchie, mais on n'a pas besoin de stocker cet ensemble de
données, et la matrice peut être stockée linéairement, ce qui réduit
les accès au disque.
```

1.7.2. Méthode du simple lien ou saut minimum

Similarité: unir la paire d'objets la plus similaire qui n'est pas encore dans la même classe. La distance entre 2 classes est la distance entre la paire de points les plus proches, chacun d'eux étant dans un des deux classes. Soient i et i' deux objets déjà réunis en classe, on mesure la distance entre k et iUi' : d(iUi',k)=min(d(i,k),d(i',k)).
Type de classes: classes longues étendues, chaînes, ellipsoïdes
Temps: généralement $O(N^2)$ peut varier de O(N log N) to $O(N^5)$.
Espace: O(N) .
Avantages: propriétés théoriques, implémentations efficaces, largement utilisée. Aucune classe centroïde ou représentative n'est requise, ainsi nul besoin de recalculer la matrice de similarité.
Limites: Pas adapté pour isoler des classes sphériques ou faiblement distinctes .

Algorithme

```
1- Enregistrer les pointeurs sur les classes et l'information sur la
distance dans des tableaux
2- Traiter les objets d'entrée un par un. Pour chacun:
3- Analyser et stocker une ligne de la matrice des distances
4- Trouver l'autre point le plus proche, en utilisant la matrice
```

```
5- Renommer les classes
```

1.7.3. Algorithme de l'arbre de couverture minimum

Un arbre minimum (MST, angl. minimum spanning tree) possède toutes les informations nécessaires pour générer une hiérarchie de simple lien en $O(N^2)$ opérations, ou la hiérarchie de simple lien peut être construite en même temps que l'arbre minimum.

Algorithme [Prim, 1957][Dijkstra, 1960]

```
1- Placer un objet arbitraire dans l'arbre minimum et connecter son
plus proche voisin à lui, pour créer un fragment initial d'arbre
minimum.
2- Trouver l'objet extérieur à l'arbre minimum qui est le plus proche
de n'importe quel point de l'arbre minimum, et l'ajouter au fragment
courant d'arbre minimum.
3- S'il reste un objet qui n'est pas dans un fragment d'arbre
minimum, répéter l'étape précédente.
```

1.7.4. Méthode du diamètre ou lien complet (angl. complete link)

Similarité: unir la paire la moins similaire entre chacun des deux classes
Soient i et i' deux objets déjà réunis en classe, on mesure la distance entre k et iUi' : d(iUi',k)=max(d(i,k),d(i',k)).
Type de classes: toutes les entrées dans une classe sont liées à une autre avec la similarité minimum, ainsi on obtient des petites classes fortement liées.
Temps: algorithme de Voorhees est en $O(N^3)$ dans le pire des cas, mais les matrices creuse demandent moins
Espace: algorithme de Voorhees est en $O(N^2)$ dans le pire des cas, mais les matrices creuse demandent moins
Avantages: Bons résultats parmi les stratégies agglomératives hiérarchiques.
Limites: Difficile de l'appliquer à de gros ensembles de données du fait que l'algorithme le plus efficace est l'algorithme général CAH utilisant des données stockées ou une approche de matrice stockée [Anderberg, 1973].

Algorithme de Voorhees

```
Variation de l'approche CAH de la matrice triée
Basée sur le fait suivant: si les similarités entre les paires
d'objets sont traitées dans un ordre décroissant,2 classes de taille
m_i et m_j peuvent être réunis aussi tôt que la m_i x m_j ième similarité
des objets dans ces classes est atteinte.
Requiert une liste triée des similarités objet-objet
Requiert une façon de compter le nombre de similarités analysées
entre 2 classes actives.
```

1.7.5. Méthode du lien moyen (angl. Group average link)

Similarité: calcul de la valeur moyenne des liens deux à deux avec une classe, et fonction de tous les objets dans la classe. Gagne de l'espace et du temps si on utilise un produit interne de 2 vecteurs (approximativement pondérés).
Soient i et i' deux objets déjà réunis en classe, on mesure la distance entre k et iUi' :

$$d(i \cup i', k) = \frac{p(i)d(i,k) + p(i)'d(i',k)}{p(i) + p(i')}$$

où p désigne le cardinal de i ou de i' .
Type de classes: rigueur intermédiaire entre le lien simple et le lien complet.
Temps: $O(N^2)$
Espace: O(N)

Avantages: se place en bonne position dans les études d'évaluation
Limites: coûteux pour de gros volumes de données

Algorithme

1- Similarité entre le centroïde d'une classe et n'importe quel objet = similarité moyenne entre l'objet et tous les objets dans la classe.
2- Ensuite, utilisation possible des centroïdes pour traiter les similarités classe-classe

1.7.6 Méthode de Ward ou de la variance minimum (minimum variance method)

Similarité: unir la paire de classes dont l'union minimise l'augmentation au total avec la somme générale des déviations au carré, basé sur la distance euclidienne entre les centroïdes (perte d'information) [Ward, 1963].
Type de classes: classes homogènes dans une hiérarchie symétrique; le centroïde de la classe est caractérisé par son centre de gravité. Conduit à des regroupements dans des régions spatiales distinctes.
Temps: $O(N^2)$.
Espace: O(N).
Avantages: bon pour retrouver la structure en classes, conduit à une hiérarchie unique et exacte.
Limites: sensible aux effets de bords, pauvre pour retrouver des classes allongées.

Algorithme

1- Calculer les centres de gravité g et poids p pour chacune des classes courantes
2- Calculer la distance de Ward (basée sur une distance d euclidienne) entre 2 classes $\delta(a, b)=\frac{p_a p_b}{p_a + p_b} d^2(g_a, g_b)$
3- Appliquer l'algorithme général

1.7.7. Méthode du plus proche voisin réciproque

On peut appliquer un algorithme du voisin le plus proche réciproque (RNN ou, angl., reciprocal nearest neighbour), puisque, pour n'importe quel point ou classe, il existe une chaîne de plus proches voisins (NN) terminée par une paire NN des autres [Benzécri, 1973][Murtagh, 1983].
Le plus proche voisin i' d'un objet i est celui pour lequel la distance d(i,i') est la plus petite des distances entre i et tout autre objet. On appelle voisins réciproques deux objets dont l'un est le plus proche voisin de l'autre et vice versa. Propriété: d(iUi',k)>Min(d(i,k), d(i',k))

Algorithme (classe simple)

1- Choisir un objet arbitrairement
2- Suivre la chaîne des NN de celui-ci jusqu'à trouver une paire de NN réciproque.
3- Fusionner les deux points en une paire, en les remplaçant par un point simple.
4- S'il n'y a qu'un point, arrêter. Sinon, s'il existe un point dans la chaîne des NN précédents les points fusionnés, revenir à l'étape 2, sinon revenir à l'étape 1.

1.7.8. Méthode mixte du centroïde et de la médiane

Appelée "hybrid clustering" en anglais, cette méthode combine une méthode des proches voisins avec une méthode hiérarchique [Wong, 1982].

Similarité: a chaque étape unir la paire de classes ayant les centroïdes les plus similaires.
Type de classes: la classe est représentée par les coordonnées du centroïde ou de la médiane du groupe.

Limites: les nouvelles classes formées peuvent différer excessivement des points constituants, c'est-à-dire que les actualisations peuvent causer de grands changements vis à vis de la hiérarchie des classes.

1.7.9. Méthode de la vraisemblance de liens

L'algorithme de vraisemblance de lien (AVL) [Lerman, 1970] utilise un critère qui permet de décider au vu d'une comparaison de ce critère avec des tests statistiques, du niveau informatif de formation des classes de la hiérarchie, en d'autres termes du niveau de coupe optimal de l'arbre pour obtenir une "partition significative" de la population.

On remplace la valeur de l'indice de similarité par la probabilité de trouver une valeur inférieure dans le cadre de l'hypothèse d'indépendance (appelée «absence de lien »).

Algorithme

L'AVL consiste alors à utiliser comme mesure de proximité entre 2 groupes A de m variables et B de l variables la probabilité associée à la plus grande valeur observée de l'indice probabiliste de similarité.

Soit : t_0=sup s(x, y) $\sup_{x\in A, y\in B} s(x,y)$.

1- Dans l'hypothèse d'absence du lien on a : $\text{Prob}(\sup_{x\in A} s(x,y) < t) = t^m$.

D'où $\text{Prob}(\sup_{x\in A, y\in B} s(x,y) < t) = t^{ml}$.

On prendra comme indice de similarité entre A et B : t_0^{ml} .

2- Algorithme de CAH général.

1.7.10. Méthodes ultramétriques

L'objectif des ces méthodes est de trouver l'ultramétrique la plus des distances initiales, puisqu'à tout arbre correspond, de manière bijective, une distance ultramétrique, c'est-à-dire une distance vérifiant les axiomes suivants:
1) d(i,i) = 0;
2) d(i,j) = d(j,i);
3) d(i,j) ≤ min(d(i,k), d(k,j)) ∀k.

Algorithme

Condition de chaîne de J.A Hartigan et arbre minimum de Kruskal.

1.7.11. Méthode d'échange

Soient Nj(I) le nuage associé $\{i \in I; x_j^i; m_i\}$, Q une partition de I et $M^2(N_J(Q))$ le moment centré d'ordre 2 de la partition Q avec $N_J(Q) = \{q \in Q; x_j^q; m_q\}$. On cherche à déterminer une partition Q' à partir d'une partition initiale Q_0 où les partitions Q' essayées se déduisent de la précédente par transfert d'un seul élément en faisant croître $M^2(Q')$ [Régnier, 1965].
Le critère de maximisation est $\Delta = M^2(N_J(Q)) - M^2(N_J(Q_0))$.
C'est à dire

$$\Delta = m_{q_k} M^2(q_k) + m_{q_l} M^2(q_l) - (m_{q_k} - m_i) M^2(q_k - \{i\}) - (m_{q_l} + m_i) M^2(q_l \cup \{i\})$$

avec $M^2(q - \{i\} \cup \{i\}) = M^2\{(q - \{i\}), i\} + M^2(q)$

Moment d'ordre 2 d'une partition:
x_{ij} $j^{ième}$ coordonnée du $i^{ième}$ individu (n individus, p variables)

$$x_{gj} = \frac{x_{1j} + ... + x_{nj}}{m}$$

$m=m_1+\ldots+m_n$ masse totale du nuage
g est le centre de gravité du nuage de points (individus).

Le moment centré d'ordre 2 s'écrit:

$$x_{gj} = \frac{x_{1j} + ... + x_{nj}}{m}$$

$m=m_1+\ldots+m_n$ masse totale du nuage
g est le centre de gravité du nuage de points (individus).

Le moment centré d'ordre 2 s'écrit:

$$M^2(I/g) = m_1 d^2(i=1,g) + ... + m_n d^2(i=n,g)$$

$$\text{où } d^2(i,i') = \sum_{k=1}^{p} (x_{ik} - x_{gk})^2 \text{ est la distance euclidienne usuelle entre i et i'}$$

La variance d'une variable s'écrit:

$$\text{var}(j) = \frac{m_1 (x_{1j} - x_{gj})^2 + ... + m_n (x_{nj} - x_{gj})^2}{m}$$

elle caractérise la dispersion statistique de la variable.

Au facteur n près $M^2(I/g)$ représente la variance généralisée (somme des variances) du nuage de p variables.

Si G est le centre de gravité du nuage et g une centre de gravité local

$$m_i (x_{ij} - x_{Gj})^2 = m_i (x_{ij} - x_{gj})^2 + m_i (x_{gj} - x_{Gj})^2$$

d'après le théorème de Huyghens :

$$M^2(I/a) = M^2(I/g) + m d^2(g,G)$$

Dans le cas d'une partition on décompose sur toutes les classes:

$$M^2(I/g) = \sum_{q \in Q} \left(\sum_{i \in q} m_i d^2(i,g) \right).$$

Le théorème de Huyghens nous donne avec les centres de gravité de chaque clase g_q

$$M^2(I/g) = \sum_{q \in Q} \left[M^2(q/g_q) + m_q d^2(g_q, g) \right]$$

$$M^2(I/g) = \underbrace{\sum_{q \in Q} M^2(q/g_q)}_{I_{\text{intra-classe}}} + \underbrace{M^2(Q/g)}_{I_{\text{inter-classe}}}$$

Plus les distances des points aux centres de gravité sont petites plus $M^2(I/g)$ sera petit. On cherche donc à minimiser le moment d'ordre 2 pour aboutir à une bonne partition.

Algorithme

1- Initialiser la partition initiale Q_0 avec une classe pour chaque élément i.
2- Calculer

$$M^2\{q,i\} = \frac{m_q m_i}{m_q + m_i}\|q - i\|^2 \qquad M^2\{q \cup \{i\}\} = M^2\{q,i\} + M^2\{q\}$$

on maximise $\Delta = M^2(N_J(Q)) - M^2(N_J(Q_0))$
3- Répéter 2 pour tout i.

1.8. Méthodes d'extraction de graphes

Les méthodes d'extraction de graphe sont très anciennes. Une des théories fondamentales de l'informatique, les automates à état finis, représente son formalisme propre à l'aide de graphes. Par contre les méthodes consistant à rechercher une partition dans un hypergraphe sont réputées coûteuse en temps de calcul et np-complet en complexité (annexe 4). C'est pourquoi elles ne sont pas répandues dans la majorité des algorithmes courants et ne constituent pas un axe de recherche majeur. L'exception reste le développement de réseaux bayésien, simples à implémenter et efficaces dans des cas simples.

1.8.1. Partitionnement de graphe

Les méthodes de partionnement de graphes sont encore aujourd'hui étudiées pour 2 problématiques, l'une concernent l'étude de la disposition des composants sur un circuit intégré et l'autre concerne l'optimisation des méthodes des éléments finis [Fjallstrom, 1998]. Le problème s'énonce très synthétiquement:
Soit un graphe G=(N,E) (où N est un ensemble de nœuds pondérés et E un ensemble d'arêtes pondérées) et un entier positif p, trouver p sous-ensembles $N_1,N_2,\ldots,N_p$ de N (i.e. une partition de N) tels que:

1. $\bigcup_{i=1}^{p} N_i = N$ et $N_i \cap N_j = \varnothing$ pour $i \neq j$;
2. $W(i) \approx W/p$, i=1,2,…,p, où W(i) et W sont les sommes des poids des nœuds dans N_i et N respectivement;

La taille de coupure,i.e., la somme des poids des arêtes entre sous-ensembles est minimisée.

On rencontre 2 approches du partitionnement de graphe: l'approche séquentielle et l'approche parallèle. L'approche parallèle consiste, en règle générale, à paralléliser les algorithmes séquentiels sur plusieurs processeurs.

On rencontre 2 types d'algorithmes séquentiels: les méthodes locales et les méthodes globales (annexe 4). Les méthodes locales ont pour but de trouver une solution optimale autour d'une partition obtenue par une méthode globale.

Parmi les principales méthodes locales on trouve : les algorithmes génétiques, l'algorithme du recuit simulé, l'algorithme KL, la méthode Tabu, la méthode des k-ensembles utiles. Parmi les principales méthodes globales on trouve les méthodes géométriques et les méthodes libres de coordonnées. Les méthodes globales les plus efficaces sont les algorithmes multiniveau-RSB basés sur l'étude spectrale des valeurs propres et l'algorithme multiniveau-KL (Kernighan & Lin, 1970) qui évalue une fonction de coût de la somme des poids par bissection.

Les méthodes de partitionnement de graphe présentent des similitudes de traitement dans leur approche de regroupement d'objets. On peut mentionner notamment la méthode d'échange vue au 1.7.11 pour les aspects de permutation de nœuds d'une partie d'une bissection à l'autre, la méthode hiérarchique descendante vue au 1.6 pour les aspects de bissection d'un graphe en

deux parties équilibrées. On retrouve aussi des aspects de décomposition en valeurs singulières vues pour l'AFC au 1.5.1 dans les méthodes spectrales de type multiniveau-RSB, ainsi que la fonction de gain de la méthode KL vue dans l'approche de Cobweb au 1.11.2.

1.8.2. Motifs de graphe

Les cliques et les chaînes sont des modèles de graphes qui se retrouvent dans les méthodes hiérarchiques. L'équivalence de ces modèles de graphes et des modèles hiérarchiques a lieu grâce au critère d'affectation. La similarité par paire agit pour la première paire d'une classe. Dés qu'on cherche à fusionner un objet avec plusieurs objets d'une classe se pose la question de l'affection.

Le critère du lien complet d'une classification hiérarchique exige que la similarité soit totale avec tous les éléments d'une classe. Cette représentation se confond avec la définition d'une clique dont les noeuds sont tous liés deux à deux.

Le critère de la méthode des voisins réciproques est tel qu'un élément est lié à une classe si cet élément constitue un couple de voisins réciproques avec un élément de cette classe. La classe est elle-même formée d'une chaîne de voisins réciproques. Cela coïncide, notamment, avec la représentation d'une chaîne d'objets.

Dans la méthode des k-moyennes on retrouve une représentation significative qui est celle d'un motif en forme d'étoile. Tous les éléments d'une classe donnée sont liés au centroïde de cette classe.

On trouve ces modèles de graphes à travers les stratégies développées dans d'autres méthodes mais aucun algorithme spécifique n'est dédié à la conception de modèle de graphe pour l'extraction de classes à partir d'un graphe.

1.8.3. Graphe connexe

L'algorithme le plus connu et le plus utilisé dans le cadre du repérage d'un graphe connexe est l'arbre de recouvrement minimum ou MST (angl. Minimum Spanning Tree) [Kruskal, 1956] [Dijkstra, 1960].
On appelle (i,i') les arêtes du tableau de données, et un sous-ensemble de l'ensemble des arêtes sur I un polygone. On appelle longueur du polygone la somme des longueurs des arêtes; on appelle arbre de longueur minimal un polygone particulier donnant structure d'arbre et tel que la somme des longueurs des arêtes de ce polygone soit minimum. Un tel arbre se présente comme un ensemble de points répartis dans un plan et dont certains sont reliés par des arêtes.

Algorithme

```
Construit une forêt de recouvrement minimum pour un graphe g non
oriente et value sur les arêtes. Les sommets du graphe sont les
entiers naturels de 0 à n-1.
1- Les arêtes de g sont valuées par des coûts entiers strictement
positifs. On représente la forêt f de la même manière que le graphe
g.
2- On ajoute à f des arêtes de g,  une par une,
en choisissant à chaque étape, parmi les arêtes qui ne sont pas dans
f, une arête de coût minimum ne formant pas de cycle avec les arêtes
de f.
```

1.8.4. Arbres de décision

L'arbre de décision permet de répartir des individus suivant les valeurs d'un certain nombre d'attributs. Cette représentation a été développée dans une finalité de classement beaucoup plus que pour la classification. Elle reste néanmoins une possibilité de classification dans le

cas où les attributs sont définis et en nombre restreint. Nous présentons un algorithme ancêtre de tous les autres ID3, il en existe des plus récents apportants des propriétés d'adaptivité à la construction de l'arbre (incrémentalité,…)

Algorithme ID3 [Quinlan, 1986]

```
Fonction ID3(
R: un ensemble d'attributs  non-cibles,
C: l'attribut cible,
S: un ensemble d'apprentissage)

1- Si S est vide, retourner un nœud simple avec la valeur échec
2- Si S consiste en données ayant toutes la même valeur pour les
attributs cibles
   Retourner un nœud simple avec cette valeur
3- Si R est vide retourner un nœud simple avec comme valeur les plus
   fréquentes des valeurs de l'attribut cible qui sont trouvés dans
   les données de S;
4- Soit D l'attribut avec le plus grand Gain(D,S)
    Parmi les attributs dans R;
    Soit {dj| j=1,2,….,m} les valeurs de l'attribut D;
    Soit   {Sj|   j=1,2,…,m}   le   sous-ensemble   de   S   consistant
    respectivement en données avec la valeur dj pour l'attribut D;
5-Retourner un arbre avec la racine labélisée D et les arcs labélisés
    d1,d2,…,dm allant respectivement aux arbres
    ID3(R-{D},C,S1), ID3(R-{D},C,S2),…,ID3(R-{D},C,Sm)
```

(la fonction gain peut être une fonction de calcul de l'entropie des probabilités des attributs)

1.8.5. Treillis de Galois

A partir d'une matrice objet/attribut on établit des correspondances communes entre objets ou entre attributs [Godin et al, 1995][Carpineto & Romano, 1996]. Cette méthode présente l'avantage de recueillir toutes les dépendances fonctionnelles. Chaque élément du treillis est un couple ou concept formel. La matrice exprime une relation binaire entre l'ensemble des objets E et l'ensemble des attributs E'. Le couple est appelé (X,X')[Godin et al, 1995].

$X'=f(X)$ où $f(X)=\{x' \in E' \mid x \in X,\ x \Re x'\}$

$X=f(X')$ où $f(X')=\{x \in E \mid x' \in X',\ x \Re x'\}$.

Les couples sont dits complets. La complétude vient de la fermeture dont les propriétés sont

$\forall x, \forall y\ x \geq y \Rightarrow h(x) \geq h(y)$;

$\forall x,\ h(x) \geq x$;

$\forall x,\ h(h(x)) = h(x)$.

Un treillis peut s'obtenir par héritage. Pour un couple complet (X,X'), X apparaît dans tous les ancêtres de (X,X'), et symétriquement X' apparaît dans tous ses descendants (figure 2.7). On peut donc éliminer ces éléments redondants sans perdre d'information si la structure du graphe est maintenue. Les éléments de x ainsi éliminés sont retrouvés par héritage (figure 2.8).

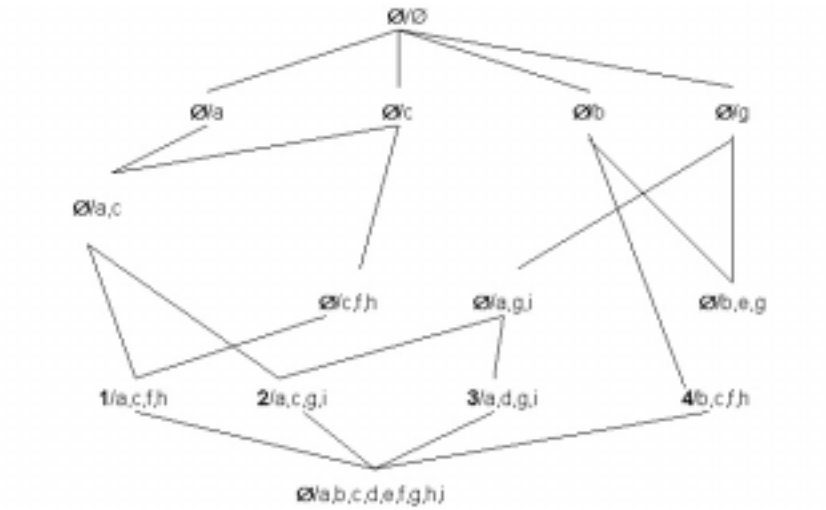


Figure 2.7. Réseau de Galois complet

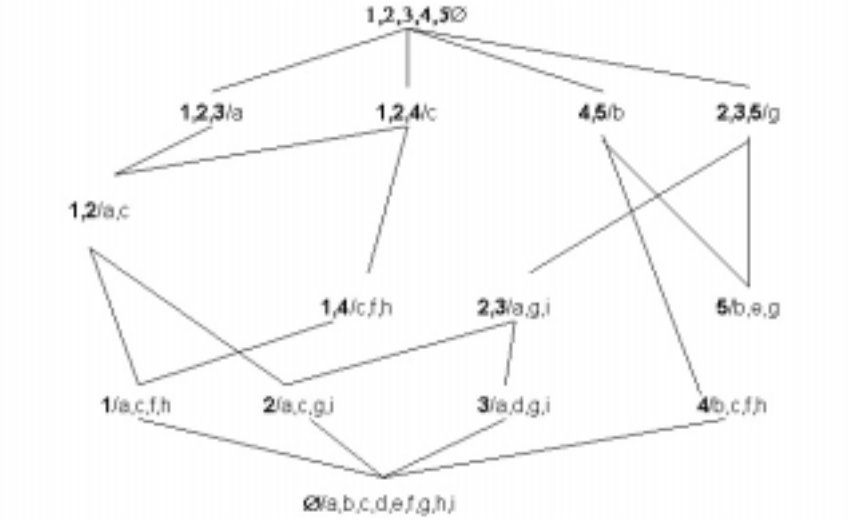


Figure 2.8 Réseau de Galois d'héritage selon X

Un treillis peut être élagué (figure 2.9). Il s'agit de l'ensemble des couples complets pour lesquels $\Re$ (x), $\Re$ (x') $\neq \varnothing,\varnothing$. On pet élaguer selon X ou X'; couples pour lesquels soit $\Re$(x) soit $\Re$(x') est vide.
$\Re(X')=\{x' \in E' \mid f'(\{x'\})=X\}$.

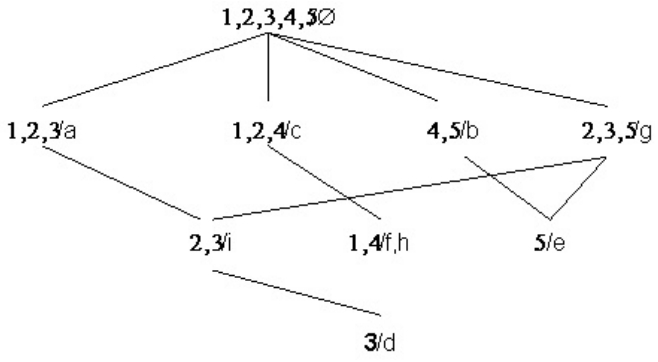


Figure 2.9 Réseau de Galois hérité selon X' et élagué selon X'

La critique que l'on peu émettre vis à vis de l'espace de calcul est sza cpomplexiteé en espace et temps de calcul.
La complexité en espace est $(v+1)^m$ concepts (si m est le nombre d'attributs avec v valeurs chacuns, et N est le nombre d'objets à classer).
La complexité en temps est $O(N.(v+1)^{2m})$.
Certaines heuristiques drastiques doivent être mises en oeuvre pour élaguer le treillis et éviter de parcourir la totalité de l'espace des attributs.

Algorithme (Carpineto, 1996)

```
1- Affecter le réseau au réseau courant
2- Pour chaque concept du réseau
3- Calculer l'intersection du concept avec un nouvel objet.
       Jusqu'à ce que l'intersection soit vide
       Ou l'intersection égale le concept
       Ou le parent du concept est l'intersection
       Ou un parent du concept est inclus dans l'intersection
 4- Créer un nouveau nœud avec l'intension de l'intersection
 5- LIER( nouveau nœud et le réseau courant) et affecter au réseau
 courant
```

La fonction LIER agit sur le réseau qui existe au moment où le nouveau nœud est créé. Elle détermine pour chaque nouveau nœud 2 ensembles de frontières: l'ensemble de frontière inférieur, S, contient les concepts les plus généraux qui sont plus spécifiques que le nouveau noeud, et l'ensemble de frontière supérieur, G, contenant les concepts les plus spécifiques qui sont plus généraux que le nouveau nœud, il lie le nouveau nœud avec S et G et supprime les liens entre S et G s'ils existent. Cette fonction retourne le réseau mis à jour.

1.9. Méthodes de sériation

1.9.1. Analyse relationnelle

La prise en compte des données est basée sur des principes logiques de comparaison par paires: on s'intéresse aux relations qu'entretiennent, deux à deux, les objets à classifier. La méthode prend son origine dans la résolution du problème des votes de Condorcet au XVIII ième siècle énonçant le paradoxe selon lequel si A est majoritaire à B, B est majoritaire à C mais A n'est pas forcément majoritaire a C. La modélisation s'effectue à l'aide d'un programme linéaire composé d'une fonction linéaire à optimiser et un système de contraintes linéaires [Marcotorchino, 1991][Bédécarrax & Huot, 1991].
Voici la modélisation linéaire dans le cas du critère de Condorcet.
Soit à partir de m variables, représentées par leur tableau relationnel A, la partition X (tableau relationnel binaire) qui maximise l'ajustement à chaque A pour le critère de Condorcet.

$$\max \sum_{k=1}^{m} \mathrm{Cond}(X,A) \text{ avec } \mathrm{Cond}(X,A) = \sum_{i}\sum_{i'} (c_{ii'}x_{ii'} + \overline{c}_{ii'}\overline{x}_{ii'})$$

$$c_{ii'} = \sum_{k=1}^{m} k_{ij}k_{i'j} \quad \text{avec } k_{ij} = \begin{cases} 1 \text{ si } A(i,j) \neq 0 \\ 0 \text{ sinon} \end{cases}$$

$c_{ii'}$ est donc la somme des variables en commun entre les objets i et i'. $\overline{c_{ii'}}$ est le complémentaire de $c_{ii'}$ (somme des variables qui ne sont pas en commun).
$x_{ii'} \in \{0,1\}$ signifie que les objets i et i' sont dans la même classe ou pas;
$x_{ii'} - x_{i'i} = 0 \; \forall i \neq i'$ (symétrie);
$x_{ii'} + x_{i'i''} - x_{ii''} \leq 1 \quad \forall (i \neq i' \neq i'')$ (transitivité).

Les contraintes caractérisent le fait que la solution X cherchée (partition) doive être une partition. Le critère est linéaire en $x_{ii'}$.

Algorithme

```
1- Mettre tous les objets de A dans la liste O et les attributs de A dans
   la liste L;
2- On choisit l'élément i de O dont la valeur A(i,j) est la plus forte
   sinon on prend une valeur au hasard si le tableau est binaire; stocker i
   dans Ofinale;
3- Pour tout i' de O calculer c(i,i') et Cond(X,A) pour x(i,i')=1 et
   x(i,i')=0;
   Garder la valeur de x(i,i') qui maximise D;
   Eliminer i' de O et éliminer les attributs de L communs à i et i'
   (l_{i∩i'}).
   Stocker l_{i∩i'} dans Lfinale et stocker i' dans Ofinale;
4- Répéter 2- tant qu'il y a un élément dans O .
```

1.9.2. Méthode par permutation

Cette méthode travaille directement sur la matrice des données. Une sériation par blocs sur une matrice M revient à une restructuration de la matrice selon deux partitions simultanées et compatibles (X et Y) des ensembles I et J, qui après permutation lignes et colonnes de I et I' (associé à X) et de J en J' (associé à Y) revient à une structure quasi-diagonale en blocs denses mettant en correspondances les classes de X(lignes) et de Y(colonnes) deux à deux. Ceci permet de lire directement les éléments de J qui caractérisent la formation des classes de I. Un des intérêts de cette méthode est de proposer une classification visuelle de la matrice avec les données permutées.

Algorithme des mots associés [Michelet, 1988]

On définit le coefficient d'association (E) $E_{ij}=C_{ij}^2/f_if_j$, carré du nombre de cooccurrences entre les 2 termes i et j (C_{ij}) divisé par le produit de leur fréquence (f_i and f_j).

On fixe un seuil d'association Epsilon et un nombre N maximal de termes par classe.

```
1- Pour un terme donné i calculer les coefficient Eij
2- Les trier par ordre décroissant les coefficients Eij
3- Garder les N termes dont Eij > Epsilon
Supprimer les termes j (retenus dans la classe de I) de la liste des
   termes à classer
4- Répéter 1- tant qu'il reste un terme
```

1.10. Méthodes neuronales

Les travaux sur les réseaux de neurones font l'objet de recherche de communautés entières. Il semble difficile de résumer brièvement le détail du fonctionnement des tous les réseaux de neurones étudiés. Dans un premier temps nous décrivons le principe général du réseau de neurones et nous présentons 3 réseaux de neurones ayant des propriétés de non supervision.

Un réseau de neurones est un système statistique de résolution d'un système d'équations non linéaires. Il s'inspire métaphoriquement d'un vrai réseau de neurone dans la mesure ou des nœuds sont affectés d'un poids, sont répartis sous forme de couches et transfèrent un signal d'une couche à une autre avec une couche d'entrée (données d'entrée) et une couche de sortie (résultat final). Un réseau de neurones opère en 2 temps. Les poids sont définis par une étape d'apprentissage grâce à des données d'apprentissage et grâce à des paramètres fixés par

l'utilisateur. Une fois l'architecture du réseau de neurones en place il est confronté aux données réelles. Il existe plusieurs types de réseaux de neurone dont l'objectif est de classer des objets de manière supervisée. Cependant certains réseaux peuvent apprendre à classer de manière non supervisée. Leur fonctionnement s'inspire largement d'une approche de type k-moyennes.

1.10.1 Cartes auto-organisantes de Kohonen

Le réseau de Kohonen est à la fois une méthode de classification et de visualisation sous forme de carte hypertextuelle [Kohonen, 1989][Lingras, 1994].
Le réseau se répartit en deux couches: une couche d'entrée formée par les vecteurs d'entrée et une couche de sortie qui est la couche de Kohonen (figure 2.10). Une distance euclidienne est utilisée comme fonction de transfert.
Les noeuds de la couche de Kohonen sont répartis uniformément sur une grille en deux dimensions qui sera représentative d'une carte de navigation sur les documents. La carte positionne les classes sur chaque nœud mais associe chaque classe à un groupe de documents.

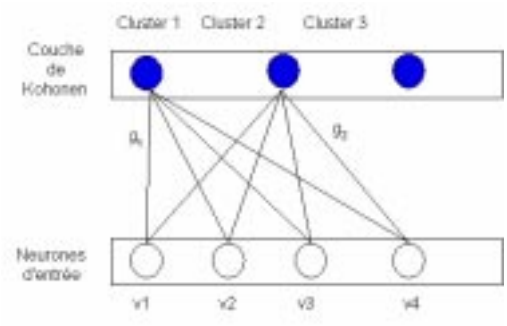


Figure 2.10 Réseau de Kohonen

La carte est graphique et permet donc une navigation documentaire à partir des champs sémantiques identifiés par les classes de termes. La grille initiale est grossière et comporte peu de nœuds, chaque nœud est ensuite affiné en une carte qui répartit les termes du nœud sur cette sous-carte. Une couleur est associée au nœud de la grille convoluée sur les pixels voisins par une fonction gaussienne dont la largeur à mi-hauteur vaut 2 unités de la grille.

Algorithme

1- Initialisation

$g_{c_j}(t=0) = \frac{\sum_{i=1}^{n} v_i}{n}$ est le poids du kième nœud d'entrée pour la classe c_j avec $0 \le j \le m-1$

2- Activation et itération

$\sum_j \left(v_{kj} - g_{c_j}^j(t)\right)^2 \ge \lambda$ est la règle d'affectation du noeud k à la classe c_j.

3- Mise à jour du vecteur poids g si affectation:

$g_{c_j}(t+1) = g_{c_j}(t) + \alpha(t) \cdot v_k$ où α vaut à peu près 0.01 et décroit en fonction de t .

4- Convergence si $\sum_{j=1}^{m} \sum_{k=1}^{n} \left[g_{c_j}^k(t+1) - g_{c_j}^k(t) \right] < \varepsilon$.

1.10.2 Réseau de Hopfield

Le réseau se compose de nœuds reliés entre eux par des liens asymétriques (figure 2.11).

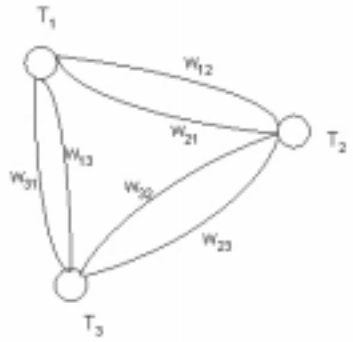


Figure 2.11 Réseau de Hopfield

Ce réseau peut, entre autres, être généré par une matrice termes/documents à partir de laquelle les similarités probabilistes des couples terme/terme seraient évaluées [Chen et al, 1994].
Chaque terme est donc traité comme un neurone et la relation asymétrique entre 2 termes est prise comme une connexion pondérée unidirectionnelle.
On considère chaque terme comme un vecteur d'entrée dont le traitement active ses voisins, en combinant les poids de ses voisins associés. Le processus est répété jusqu'à la convergence du vecteur de sortie.

Algorithme

1- Assignation des poids et des nœuds du réseau initial en fonction des relations terme/terme.
2- Initilisation des vecteurs de sortie avec les vecteurs d'entrée $\mu_i^k(t=0) = x_i$, avec $0 \le i \le n\text{-}1$ et $0 \le k \le n\text{-}1$ (k est l'indice du vecteur de sortie et x_i vaut 1 pour un nœud et vaut 0 pour les autres à t=0) .
3- Activation et itération

$$\mu_i^k(t+1) = f_s\left[\sum_{j=0}^{n-1} t_{ij}\mu_i^k(t)\right],\ 0 \le j \le n-1$$

$$f_s(net_j) = \frac{1}{1+\exp\left(\frac{-(net_j - \theta_j)}{\theta_0}\right)}$$ est la fonction sigmoïde où $$net_j = \sum_{i=0}^{n-1} t_{ij}\mu_i^k(t)$$

θ_j sert de seuil et θ_0 modifie la forme de la sigmoïde.

4- Convergence $\sum_{j=0}^{n-1} \mu_i^k(t+1) - \mu_i^k(t) \le \varepsilon$.

1.10.3 ART (angl. Adaptive Resonance Theory)

L'étape de construction du réseau est une phase d'apprentissage qui consiste à opérer de multiples itérations avec les échantillons de départ pour apprendre les différents voisinages des couches cachées c'est-à-dire identifier les classes des données d'entrée [Carpenter, 1997] .
Des tests vérifient qu'un voisinage existant est modifié seulement si l'individu courant en entrée est suffisamment similaire à l' individu moyen de ce voisinage. Si le vecteur d'entrée courant passe le test, le nœud le plus proche du voisinage est activé, ses poids sont mis à jour pour réduire le décalage, et l' individu moyen pour ce voisinage est mis à jour. Sinon le nœud le plus proche à l'extérieur des voisinages est activé et est utilisé comme noyau de nouveau voisinage (création de classe) (figure 2.12).

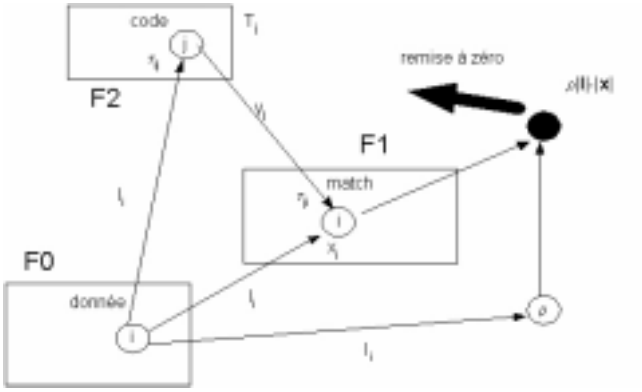


Figure 2.12 Réseau ART

Le rapprochement entre un voisinage et un individu courant se fait grâce à une distance euclidienne. Les paramètres du réseau sont: le nombre de classes, le nombre maximum d'itérations et la distance minimum entre classes.
La méthode est incrémentale et recouvrante. Elle est utilisée pour regrouper des segments de textes de taille équivalente selon les mots qu'ils contiennent (matrice initiale segments*mots). Le traitement s'avère coûteux en temps de calcul et en consommation de mémoire [Memmi et al, 1998]. [Torres-Moreno et al, 2000] compare une méthode des réseaux ART avec une méthode de l'application *Classphères* qui associe des voisins réciproques basée sur une distance de Hamming. Les résultats donnent l'avantage à la distance de Hamming.

Algorithme

1- On définit les variables i=1…M, j=1…N
LTM (angl., long-term memory): F_0 (entrée, x_i), F_1 (matching, x_i) F_2 (codage, y_i)
STM (angl., short-term memory): Δ_{ij} – phasique et δ_{ij} - tonique
MTM (angl., medium-term memory): τ_{ij}: F_0->F_2 et τ_{ji}: F_2->F_1
Signal F_0->F_2: S_j phasique, Θ_j tonique, T_j total; signal F_2->F_1 : σ_i total
2- Règle du signal; on définit la fonction T_j du signal F_0->F_2 $T_j=g_j(S_j,\Theta_j)$, où $g_j(0,0)=0$ $\partial g_j/\partial S_j>\partial g_j/\partial \Theta_j >0$ pour $S_j>0$ et $\Theta_j>0$, (exemple $T_j=S_j+(1-\alpha)\Theta_j$ avec $\alpha \in (0\text{-}1)$)
Règle CAM (angl., content-addressable memory) on définit la fonction à état stable $y_j=f_j(T_1,\ldots,T_N)$ où $\partial f_j/\partial T_j>0$
3- 1ière itération, n=1, diminution MTM (seuils de F_0->F_2 et F_2->F_1, vecteurs d'entrée)
4- Remise à zéro (angl., reset): nouvel état stable STM à F_2 et F_1
5- Diminution MTM: les sites F_2 sont réfractaires sur l'échelle de temps de la recherche
Remise à zéro ou résonance: vérifier le critère de matching de F_1

si $\sum_{i=1}^{M} x_i \geq \sum_{i=1}^{M} \rho I_i$, alors on a résonance ($\rho$ est un paramètre de vigilance $\in [0,1]$)

passer à 7- sinon revenir à 4-
7- Résonance: nouveaux seuils LTM et récupération MTM
8- Nouvelle itération: incrémenter n (nouvelle entrée I_i) et nouvelle activation F_1 (x_i) revenir à 4-

1.11 Méthodes symboliques

Les systèmes utilisant des méthodes symboliques (annexe 2) visent à dépasser l'usage d'un critère statistique et du modèle de vecteurs pour caractériser des classes d'objets par un ensemble de sélecteurs caractéristiques de la classe (donnée numérique d'un ensemble de

cardinal fini, donnée numérique d'un ensemble de cardinal infini, présence ou absence de symbole). La classification se base sur une matrice objet/attribut mais ne dépend pas seulement des valeurs d'un élément matriciel mais aussi et surtout de la nature et combinaison des attributs qui caractériseront les classes à rechercher [Ketterlin, 1995].

1.11.1 Approche des plus proche voisins

Cluster [Michalski, 1980] est des premiers systèmes modernes d'apprentissage symbolique non supervisé. Il mélange harmonieusement apprentissage supervisé et non supervisé en considérant le deuxième comme un sous-problème du premier. Le système s'inspire de la méthode précédemment décrite des k-moyennes en fixant K objets arbitrairement comme concepts de bases par les n objets à classer.
Star est un ensemble de couples "critère-seuil" à atteindre et définis par l'utilisateur, *Lef* est une fonction équivalente à un critère global création de classe et défini par l'utilisateur

Algorithme

```
1- Sélectionner une partition initiale de K noyaux.
2- Pour chaque noyau
 calculer la règle recouvrante pour un noyau avec une fonction Star;
 sélectionner les objets concordant avec la définition de la classe
tant qu'une fonction Lef < seuil .
```

AutoClass (Cheeseman, 1988) admet une fonction de coût associée aux objets et aux classes: la longueur minimale de description (MDL, i.e. minimum length description). Les objets et les classes sont codés grâce à des mots dont la longueur est mesurée en bits. Plus il y a de classes moins on a besoin de bits pour encoder les objets. La minimisation de la fonction de coût détermine à la fois le nombre de classes et comment assigner les objets aux classes.

1.11.2 Approche descendante

Unimem/*Researcher* [Lebowitz, 1983] et *Cyrus* [Kolodner, 1983] relèvent d'une classification hiérarchique à caractère descendante mais non strictement monothétique. Il intègre également la notion d'incrémentalité dans le cadre de son mécanisme apprentissage.
L'algorithme utilise plus de 10 paramètres qui sont ajustés dans la fonction "Mise à jour", la fonction "Généraliser" teste la création possible d'un concept réunissant tous les attributs de l'objet et de O_i en retirant O_i de la hiérarchie. Les règles caractérisant les concepts ne sont pas disjointes puisqu'un objet peut appartenir à plusieurs concepts mais se ramènent à une conjonction de sélecteurs.

Algorithme

```
1- Partir de la racine C de la hiérarchie.
2- Faire la Mise à jour d'un nouvel objet avec C.
3- Faire pour chaque sous-concept de c si O et Si sont compatibles ;
     répéter étape 2 avec S
4- Sinon pour chaque objet Oi de C Généraliser (O, Oi), si possible .
5- Sinon créer un sous concept de C avec O.
```

Cobweb [Fisher, 1997[1986]] (et ses variantes *Classit* [Gennari, 1994], *Adeclu* [Decaestecker, 1992], *Arachne* [McKusick & Langley, 1980]) sont des algorithmes de parcours de hiérarchie par approche descendante fonctionnant par opérateur. Trois opérateurs permettent de modifier une hiérarchie d'objets: fusion, scission et création. Une fonction de gain probabiliste permet de décider si tel opérateur doit être utilisé et à quel niveau de la hiérarchie. L'algorithme a l'avantage d'être incrémental et d'agir localement dans une hiérarchie mais dépend de l'ordre et dans le pire des cas peut générer deux fois plus de classes que d'objets. Les classes sont

décrites par un ensemble d'attributs qui sont utilisés par la fonction de gain (angl. Global Utility).
La fonction de gain est le produit simple de la probabilité d'avoir la valeur d'un attribut connaissant un concept (similarité intra-classe) par la probabilité d'avoir un concept connaissant la valeur d'un attribut (similarité inter-classe). Ce produit est sommé sur toutes les valeurs posibles des attributs et des concepts.
Ce produit sera pondéré par la probabilité d'avoir la valeur de l'attribut:

$$PU(C,\{C_1,...,C_k\}) = \sum_i^I \sum_j^{J(i)} \sum_k^K P(A_i = V_{ij}|C_k) P(C_k|A_i = V_{ij}) P(A_i = V_{ij})$$

$$= \sum_k^K P(C_k|A_i = V_{ij}) \sum_i^I \sum_j^{J(i)} P(A_i = V_{ij}|C_k)^2$$

ou $$PU(C,\{C_1,...,C_k\}) = \frac{1}{K}\left(\sum_k^K P(C_k|A_i = V_{ij}) \sum_i^I \sum_j^{J(i)} P(A_i = V_{ij}|C_k)^2 - \sum_i^I \sum_j^{J(i)} P(A_i = V_{ij}|C_k)^2 \right)$$

après correction des biais liés aux concepts évidents (un élément par exemple).
Prédictivité d'un attribut A_i au sein d'un concept C où J(I) représente le nombre de valeurs de A_i :

$$\Pi(A_i, C) = \sum_{j=1}^{J(i)} P(A_i = V_{ij}|C)^2$$

Prédictivité globale d'un concept (normalisée):

$$\Pi(C) = \frac{1}{I} \sum_{j=1}^{I} \Pi(A_i, C)$$

$$PU(C,\{C_1,...,C_k\}) = \frac{1}{K} \sum_{k=1}^{K} P(C_k)(\Pi(C_k) - \Pi(C))$$

(angl. Probability Utility ou Global Utility).

Une opération de renumérotation permet d'évaluer la façon dont ils accueillent un nouvel objet: i<j si et seulement si:
$PU(C,\{\ldots,C_i+O,\ldots\}) > PU(C,\{\ldots,C_j+O,\ldots\})$
L'opérateur de scission élimine un concept et place ses sous-concepts à sa place dont le nouvel objet à classer fusionné avec l'un deux.
L'opérateur de fusion réunit deux sous-concepts d'un concept avec le nouvel objet à classer.
L'opérateur de création ajoute un sous-concept à un concept donné formé du nouvel objet à classer.

Algorithme

```
1- Soit un nouvel objet O à classer.
2- Si le concept courant C est une feuille
      si Π(O_i,C) < seuil alors création et mise à jour.
3- Sinon renuméroter les sous-concepts de C par rapport à O.
4- Pour chaque opérateur évaluer le score de
      PU(C, Opérateur(o,{sous-concepts}).
5- Appliquer l'opérateur dont le score est maximal.
6- Répéter 2 pour un sous-concept de C.
```

L'algorithme de Cobweb a été utilisé par [Basili et al, 1996] pour catégoriser les verbes d'un corpus. Le processus se déroule en trois étapes. La première est une étape d'étiquetage syntaxique où des schémas morphosyntaxiques sont détectés de type Nom-Prép-Nom, Objet-

Verbe… Dans une seconde étape ces schémas sont étiquetés sémantiquement à la main en utilisant une série d'étiquettes de haut niveau de façon à faire ressortir de façon automatique les relation sémantiques de plus grande importance comme:
[acte]->(bénéficiaire)->[être_humain].
A partir des relations principales on prélève des structures attributs/valeurs pour chaque occurrence des verbes comme:
Produire:/ (manière: propriété, thème: contenu_cognitif, instrument: instrumentalité). A chaque occurrence (considérée distincte même pour un même verbe) on obtient donc des couples rôle sémantique/ catégories sémantiques (exemples de rôle: agentif, affecté, figuratif, manière, référence, cause, lieu…; exemple d'étiquette: acte, être humain, document, quantité, statut, état, entité temporelle, qualité,…). Une classe au sens Cobweb sera décrite par le vecteur suivant: $C=(c_c,[x]_{ij}, V_c, S_c)$
où $[x]_{ij}$ représente la matrice rôle/catégorie contenant la distribution des probabilités parmi les relations pour la classe, cc représente toutes les instances de la classes (un verbe pouvant avoir plusieurs occurrences dans la classe mais pas toutes), V_c est le nombre de verbes contenus dans la classe et S_c est l'ensemble des sous-types. La classification élabore une hiérarchie dont le vecteur rassemble toutes les occurrences de tous les verbes en une seule classe pour aboutir à des classes singleton. Les meilleures classes sont caractérisées par 2 facteurs :
- le pouvoir de généralisation ω qui signifie que toutes les instances d'un verbe sont décrites par la classe $\tau_c = \frac{\sum_{i,j \in T_c} x_{ij}}{card(T_c)}$,

T_c est la typicalité du concept, c'est-à-dire les couples (i, j) tels que $x_{ij}>\alpha$

- et la typicalité t. qui signifie que les valeurs d'attributs sont uniquement présents dans les rôles de la classe $\omega = \frac{card(V_c)}{c_c}$.

Les classes qui vérifient $\omega > \gamma$ et $\tau_c > \delta$ seront retenues comme catégories de verbe de niveau basique et donc plus stables.

1.11.3 Approche ascendante

Witt [Hanson & Bauer, 1990] combine une similarité intra-classe et interclasse basé sur les corrélations entre attributs avec une stratégie de classification identique à une approche descendante $H_k = \frac{W_k}{O_k}$.

W_k représente la cohésion interne du concept C_k (à maximiser) et O_k représente la cohésion du concept C_k avec tous les autres concepts (à minimiser). Dans chaque concept, pour chaque couple d'attributs on analyse une table de contingence qui contiendra le nombre d'occurrences simultanées des couples de valeurs des attributs correspondants. On aboutit à une estimation de la similarité entre deux concepts pour développer une construction hiérarchique d'un arbre de concepts décrits par leurs attributs. Cette estimation probabiliste de la similarité est considérée comme une distance classique présentée en 1.2.

$$W_k = \frac{\sum_{i}^{I-1} \sum_{j=i+1}^{I} D_{kij}}{N(N-1)/2} \quad \text{et} \quad D_{ij} = \frac{\sum_{m=1}^{J(i)} \sum_{n=1}^{J(j)} f_{[kij]mn} \log f_{[kij]mn}}{\left(\sum_{m=1}^{J(i)} \sum_{n=1}^{J(j)} f_{[kij]mn} \right) \left(\log \left(\sum_{m=1}^{J(i)} \sum_{n=1}^{J(j)} f_{[kij]mn} \right) \right)}$$

avec J(I) est la taille du domaine de l'attribut Ai, I est la taille des attributs,
$f_{[kij]mn} = P(A_i=V_{im} \wedge A_j=V_{jn} \mid C_k)$
est la cellule de la table de contingence associée aux attributs A_i et A_j. La cohésion interne W est la moyenne des valeurs contenues dans les tables de contingences des attributs caractérisant au mieux le concept.

$$O_k = \frac{\sum_{l=1,l\neq k}^{K} B_{kl}}{(K-1)} \text{ et } B_{kl} = \frac{1}{W_k + W_l - 2W_{k\cup l}}$$

B s'apparente à un coefficient d'association classique et O fait la somme de tous les coefficients associant le concept C_k avec un autre (O utilise W pour évaluer la différence d'un concept avec un autre, W(kUl) sera calculé comme si les concepts étaient réunis et avaient leurs attributs en commun).

Algorithme

```
1- Phase préliminaire d'agrégation par paire par un indice classique
(voir 1.8) pour n cycles
2- Calcul des tables de contingence
3- Pour chaque concept calculer Hk et fusionner le concept avec le
concept de la hiérarchie qui maximise la Hk.
```

1.12 Amorçage et l'algorithme EM

Des méthodes comme les k-moyennes ou une classification hiérarchique peuvent servir de point de départ pour définir un modèle distributionnel paramétrique. On parle d'amorçage (angl. Bootstrapping). On peut ensuite utiliser un algorithme dit EM (Estimation-Maximisation) pour optimiser les paramètres du modèle [Manning & Schütze, 1999]. Dans cette perspective les modèles probabilistes tiennent une place importante avec notamment les modèles gaussien ou bayésien.
Prenons l'exemple d'une distribution gaussienne. Une famille multivariée m-dimensionnelle gaussienne est paramétrée par une moyenne μ_j et par une matrice symétrique positive inversible m×m (la matrice de covariance S_j):

$$n_j(\vec{x_i}; m_j, S_j) = \frac{1}{\sqrt{(2\pi)^m |S_j|}} \exp\left[-\frac{1}{2}(\vec{x}-\vec{\mu_j})^T S_j^{-1}(\vec{x}-\vec{\mu_j})\right]$$

Alors $P(w_i|c_j) = n_j(x_i; \mu_j S_j)$ où P désigne la probabilité que w_i soit un mot du concept c_j et n_j désigne la distribution gaussienne liée au jième concept.
Le mot w_i doit se trouver dans un des concepts c_j, pour cela on maximise:

$$\sum_{j=1}^{k} [\pi_j \cdot n_j(x_i; \mu_j S_j)]$$

L'opération de maximisation (M) se calcule par le maximum de vraisemblance pour les n mots à catégoriser ce qui permet d'évaluer l'ensemble Θ des paramètres du meilleur modèle du mélange des distributions gaussiennes:

$$l(X|\Theta) = \log \prod_{i=1}^{n} P(x_i) = \log \prod_{i=1}^{n} \sum_{j=1}^{k} [\pi_j \cdot n_j(x_i; \mu_j S_j)] = \sum_{i=1}^{n} \log \sum_{j=1}^{k} [\pi_j \cdot n_j(x_i; \mu_j S_j)]$$

La somme de logarithme est difficile à optimiser . On passe donc par une méthodologie itérative à 2 étapes:
1- l'étape d'estimation (E) sert à estimer un paramètre d'appartenance moyenne à une catégorie:

$$h_{ij} = \frac{P\left(x_i \middle| n_j; \Theta\right)}{\sum_{j=1}^{k} P\left(x_i \middle| n_j; \Theta\right)};$$

2- l'étape de maximisation (M) permet de calculer la moyenne et la matrice de covariance de chaque distribution de l'ensemble Θ:

$$\vec{\mu_j} = \frac{\sum_{i=1}^{n} h_{ij} \vec{x_i}}{\sum_{i=1}^{n} h_{ij}}$$

$$S_j^{'} = \frac{\sum_{i=1}^{n} h_{ij} \left(\vec{x_i} - \vec{\mu_j^{'}}\right)\left(\vec{x_i} - \vec{\mu_j^{'}}\right)^T}{\sum_{i=1}^{n} h_{ij}},$$

les poids sont aussi évalués par $\pi_j^{'} = \frac{\sum_{i=1}^{n} h_{ij}}{n}$.

Les étapes EM ne sont pas un algorithme en soi. Elles dépendent du modèle de contrainte choisi qui est probabiliste (distribution paramétrique) dans le cas général. L'approche EM participe à une phase finale de classification pour la détermination de paramètres.

1.13 Aide à l'interprétation/Evaluation

Les aides à l'interprétation existent en Analyse des Données et sont surtout focalisées sur la participation des variables dans la construction des classes (partitions ou arbres). On établit un tableau pour évaluer la contribution des variables. On calcule:
- soit la variation de gravité
$m_q(g_{qj}-g_j)$
où g_j est la jième coordonnée du centre de gravité de tous les objets, g_{qj} celle de la classe q et m_j l'effectif de q.
- soit un coefficient de type X^2 suivant qu'il s'agisse de données quantitatives ou qualitatives.

Le coefficient de Cramer est couramment utilisé:

$$\varphi^2(j,q) = \sum_{l \in L_j} \left(\frac{e_{ql}}{m_q} - \frac{e_q e_l}{m_q^2} \right)^2 \frac{m_q^2}{e_q e_l}$$

e_{ql} nombre d'objets ayant la modalité l dans q,
$e_{q.}$ effectif de la classe q,
$e_{.l}$ effectif de la modalité l de J,
l parcourt l'ensemble des modalités de la variable j,

on réalise $\varphi = \frac{\varphi^2}{\min(e_{.l} - 1, e_{q.} - 1)}$.

1.14 Visualisation

Plusieurs types de visualisation on été mis en œuvre depuis l'origine de la classification. La première fut l'arbre taxinomique (figure 2.6).

Les méthodes factorielles ont adopté une représentation de type cartésien avec une projection des points en 2D ou 3D sur les axes propres. Les points sont représentés par rotation des axes (1,2,3), (2,3,4) ….

Les catégories elles-mêmes peuvent faire office d'axes de projection (échelle multidimensionnelle ou, angl., multidimensional scaling). On peut par exemple représenter l'acuité visuelle (1/10,2/10…,10/10) par tranche d'âge (20-30,…,70-80) et aussi obtenir le diagramme pour chaque couleur des yeux (bleu, marron, vert).

Une représentation peut aussi consister en un réseau de type ontologique décrit par des noeuds et des relations. On peut visualiser une zone locale du réseau pour un nœud particulier et constater ses propriétés relationnelles propres.

2. Connaissances terminologiques et collocations

Après avoir parcouru les approches de la classification automatique par des moyens statistiques passons en revue les principaux problèmes et phénomènes linguistiques pour mieux comprendre l'interaction envisageable entre des traitements statistiques et des traitements linguistiques.

2.1. Linguistique générale

La linguistique est une science qui traite des langues naturelles. L'observation des phénomènes se fait grâce à un échantillonnage appelé corpus. Il s'agit d'un ensemble fini de données recueillies par écrit et qui sert à la description linguistique. L'observation est jugée adéquate si le corpus est exhaustif et représentatif des faits de communication, c'est à dire capable de couvrir ou d'illustrer toute la gamme des caractéristiques structurelles de la langue à analyser. Un modèle ou une théorie linguistique est un ensemble organisé de règles destinées à expliquer la compétence linguistique d'un locuteur et permettant d'effectuer des opérations d'encodage et de décodage. Une théorie est un moule qui sert à fabriquer des grammaires. Le rôle du linguiste est de décrire le fonctionnement d'une langue naturelle [Essono, 1998].
La linguistique générale peut se décomposer en plusieurs branches:

- La linguistique comparée qui sert à rapprocher deux ou plusieurs langues en vue d'établir une filiation entre elles.
- La linguistique typologique qui vise à classer les langues en groupes de même caractéristiques.
- La linguistique évolutive qui constate les mutations au sein de la langue.
- Et la linguistique descriptive qui cherche à isoler les unités constitutives d'une langue pour en déterminer les fonctions.

C'est cette dernière branche qui nous intéresse plus particulièrement. La linguistique descriptive se décompose elle-même en plusieurs domaines d'étude:

- La phonétique qui étudie les sons en tant que réalité physique et acoustique.
- La phonologie qui cherche à dégager les principes qui régissent l'apparition et la fonction des phonèmes dans une langue particulière où ils constituent un système.
- ***La morphologie*** traite des mots en fonction de leurs variations: flexion, dérivation, composition.
- ***La syntaxe*** traite de la combinaison et de l'ordre des mots dans la phrase.
- ***La lexicologie*** traite de l'ensemble des signifiés et acceptions des mots d'une langue. Le lexique a pour objectif l'analyse du vocabulaire ou l'élaboration d'un dictionnaire.

- ***La sémantique*** est l'étude de la valeur, du sens ou de la signification des mots ainsi que du rapport entre la forme et le sens, entre signifiant et signifié.

2.2. Morphologie et syntaxe

La morphologie est étymologiquement l'étude des formes. Il existe des liens étroits liant morphologie et syntaxe conduisant à la détection des entités responsables des objets, sujets, ... d'une phrase. L'étude de la morphologie repose sur le concept de mot qui peut être défini de façon multiple: entités séparées entre elles par des blancs, entités formant une suite de sons ayant une fonction dans la phrase, ou première unité significative dans une phrase. Plusieurs dénominations ont été mises sur la scène pour palier ce problème de définition dont morphème ou lexème. On distingue deux types de mots: les mots simples et les mots liés. Les mots simples agissent de façon autonome ("boule", "maison", "fruit") et peuvent s'assembler par composition se comportant alors comme un mot simple ("pomme de terre", "vaisseau spatial"). Le mot composé peut éventuellement être réduit à un de ses éléments ("dactylo" pour "dactylographe", "télé" pour "télévision"). Le mot lié est une entité non autonome dépendant d'un mot simple (" –iste " dans "trapéziste"). Le mot lié est aussi appelé affixe.
On trouve les *infixes* (comme le pluriel), les *suprafixes* (indicateur de ton ou intonations), les préfixes (comme " pré- ") et les *suffixes* (" –iste ").

2.2.1 Dérivation

La dérivation est l'agglutination d'éléments lexicaux en une forme unique. Une dérivation est réalisée par adjonction affixale à un radical (livre → livresque ou pardon → pardonnable). On appelle base dérivationnelle la forme à laquelle on adjoint un affixe.
On peut distinguer 3 classes de dérivations
- la dérivation dénominative (courage → courageux)
- la dérivation déverbative (laver → lavage)
- la dérivation déadjectivale (aride → aridité).

On appelle dérivation explicite ou propre une dérivation réalisée avec affixe. Dans le cas contraire il s'agit d'une dérivation implicite ou impropre :
transformation d'un *nom propre en nom commun* : une "poubelle",
d'un *infinitif en nom*: le "devoir",
d'un *pronom en nom*: le "moi" est inconscient,
d'un *participe présent en adjectif* : "étudiant".

La dérivation régressive forme un dérivé par *perte d'une syllabe* comme agrafer → agrafe.

2.2.2 Flexion

On trouve des flexions rattachées soit au nom soit au verbe. Les flexions transforment les mots par adjonction de suffixes mais peuvent changer le radical.
Dans le cas de la flexion nominale on trouve le pluriel (les patients), le genre (la patiente) et les cas (plus visibles dans des langues germaniques ou slaves):
Nominatif → le patient (sujet);
Accusatif → le patient (objet);
Génitif → du patient (oblique);
Datif → au patient (destinataire);
Ablatif → par le patient (agent).

La flexion verbale comprend les modes, les temps,les personnes… Les flexions verbales sont en nombre fini. Prenons l'exemple intéressant du verbe irrégulier du verbe *aller*:
nous allons, je suis allé,…, ils iront, j'irai…, je vais, tu vas…

Dans cet exemple on obtient 3 racines: *all-*, *ir-*, *v-*. Dans le cas du verbe régulier comme *finir* on n'a qu'une racine *fin-*.

2.2.3 Racinisation, lemmatisation, étiquetage, correction et réaccentuation

La racinisation consiste à extraire la racine d'un mot par troncature de suffixe. Par exemple le verbe *aller* possède trois racines qui sont *ir-*, *all-* et *v-*, toutes ses flexions dérivent de ces racines. L'extraction d'un seul suffixe ne suffit pas toujours pour aboutir à la racine, par exemple l'extraction du suffixe *–eur* à *travailleur* demande à effectuer une troncature de la double consonne pour tomber sur *travail* et ensuite de renvoyer sur l'extraction du suffixe *–ail* (*-aux*) pour aboutir au préfixe *trav-*. On obtient finalement la classe d'équivalence suivante (trav**ail**, trav**aux**, trav**ailleur**, trav**ailleurs**, trav**ailleuse**, trav**ailleuses**). L'extraction d'un suffixe ne coïncide pas systématiquement avec une même classe d'équivalence, par exemple un protestant et une protestation. L'algorithme de racinisation doit donc seulement considérer les mots techniques ou les mots nouveaux qui suivent des constructions souvent régulières. [Lovins, 1968][Porter, 1980].

La lemmatisation consiste à trouver l'entrée du dictionnaire pour une forme fléchie: par exemple *aller* est la forme lemmatisée de *iront*. Le processus semble simple a priori mais bute sur des problèmes classiques des langues à savoir les ambiguïtés comme *avions* qui peut être lemmatisé en *avoir* et *avion*. Pour éliminer le problème d'ambiguïté on essaye d'attribuer des étiquettes syntaxiques. Deux types d'étiqueteur fonctionnent par apprentissage et arrivent à des taux d'environ 2% d'erreur pour l'attribution d'une étiquette syntaxique.

Le premier type utilise des modèles probabilistes (HMM ou chaînes de Markov cachées) [Markov, 1916] (annexe 5). Dans les modèles de Markov observables, chaque état est équivalent à un état observable. L'état présent dépend seulement de l'état précédent. Dans notre cas, une étiquette grammaticale (verbe, adverbe, nom..) représente un état. Un HMM aide à étiqueter les phrases et résoudre les ambiguïtés. Une ambiguïté est modélisée par différents chemins dans un graphe ou une suite de séquence d'états [Chanod & Tapanainen, 1995]. Cependant, pour un HMM la sortie n'est pas la séquence d'état interne, mais une fonction probabiliste de cette séquence interne. Quelques auteurs font des symboles de sortie d'un modèle de Markov une fonction de chaque état interne, tandis que d'autres font de la sortie une fonction des transitions. A chaque état donné, il y a un choix de symboles, chacun avec une certaine probabilité d'être sélectionné. Une chaîne de Markov cachée est un processus doublement stochastique. Elle consiste en un processus stochastique sous-jascent qui ne peut pas être observé, décrit par les probabilités de transition entre une paire d'états et calculées à partir d'un corpus d'apprentissage. Deuxièmement, un processus stochastique gère les symboles de sortie qui peuvent être observés à partir des données d'entrée au processus, et représentés par les probabilités de sortie du système. Les principaux paramètres du HMM peuvent être résumés par l'ensemble des probabilités de transition, l'ensemble des probabilités de sortie et l'état initial du modèle. L'utilisation de modèles de Markov cachés dans la résolution d'un problème d'étiquetage implique 3 problèmes algorithmiques : l'apprentissage, l'évaluation, l'estimation. Pendant l'apprentissage, les paramètres initiaux du modèle sont ajustés pour maximiser la probabilité d'observer une séquence de symboles. Cela rendra le modèle actif pour prédire de futures séquences de symboles. L'apprentissage implémente l'algorithme de re-estimation de Baum-Welch. A ce stade un corpus étiqueté à la main permet de calculer les paramètres du modèle (probabilité de transition d'être dans l'état i à la position p et de suivre un état j à la position p+1). Le problème de l'évaluation est de calculer la probabilité qu'une séquence observée de symboles apparaisse comme résultat d'un modèle donné, et est résolu en utilisant un algorithme *forward-backward*. Dans le problème d'estimation, nous observons une séquence de symboles produite par une chaîne de Markov cachée. Il s'agit d'estimer la séquence d'états la plus probable que le modèle permet d'obtenir

pour produire cette séquence de symboles, et une solution est d'utiliser l'algorithme de Viterbi.
Le deuxième type est l'étiqueteur de [Brill, 1993]. Il s'agit d'une architecture d'application de séquences de règles. Un étiquetage initial d'un texte brut est produit grâce à un dictionnaire. Les lexèmes sont étiquetés avec l'étiquette la plus commune associée au léxème observé dans un corpus d'apprentissage. Cet étiquetage initial est affiné par deux types de règles de transformations. Les transformations morphologiques reformulent l'étiquetage par défaut des mots qui n'ont pas été trouvés dans le dictionnaire. Les règles morphologiques sont suivies par des règles de transformations contextuelles: ces règles inspectent le contexte lexical pour renommer leur étiquette due à leur ambiguïté (par exemple: changer l'étiquette "Verbe" pour "montre" en "Nom" si un article (la, cette, une) est trouvé devant "montre"). L'application des règles de contexte s'arrête dès qu'un critère de performance descend en dessous d'un seuil ou qu'un nombre défini de règles a été appliqué.

La classification est parfois utilisée pour apparier des formes morphologiques et permettre une correction orthographique (angl. string matching). La réaccentuation s'apparente à un travail de lemmatisation pour lequel toutes variantes accentuées sont analysées pour évaluer celle qui est la plus probable en fonction du contexte.

2.2.4 Grammaire distributionnelle

La syntaxe traditionnelle place le verbe en position centrale dans l'analyse syntaxique. Mais le sujet n'étant pas toujours repérable ou présent d'autres approches ont vu le jour: la syntaxe dépendentielle de Tesnière, la syntaxe psychomécanique de Guillaume, la tagmémique de Pike, le fonctionnalisme de Martinet, le distributionnalisme de Bloomfield et la syntaxe générative de Chomsky.

Dans la syntaxe fonctionnelle, on identifie un énoncé grâce à un noyau irréductible. Ce noyau est de type *prédicat* et ordonne autour de lui des fonctions dépendantes. La fonction *sujet* contrôle le prédicat auquel s'ajoute une expansion additive mais pas nécessaire.
Dans l'exemple: "l'étoile géante brille dans le ciel", "l'étoile" sera considéré comme sujet du prédicat "brille", et "dans le ciel" sera l'expansion. Les fonctions sont aussi appelées *monèmes* et classées en trois catégories: les monèmes autonomes (comme les adverbes), les monèmes fonctionnels (comme les prépositions) et les monèmes dépendants.

La syntaxe distributionnelle ou structuraliste se base sur la répartition des unités linguistiques au sein de la phrase. Chaque unité constitutive de la phrase a des occurrences bien précises. La distribution d'un élément est l'ensemble des positions que cet élément peut occuper. Les unités pouvant figurer dans les mêmes contextes appartiennent à la même classe distributionnelle, à la même catégorie morpho-syntaxique. Ces éléments cooccurrents forment un paradigme. Le procédé qui consiste à remplacer une unité dans une position donnée par d'autres unités de même type est appelé *commutation*. Cette opération de substitution est à la base du distributionnalisme. Le distributionnalisme conduit à l'analyse en constituants immédiats des éléments d'une phrase. "L'étoile géante" représente un syntagme nominal et "brille dans le ciel" un syntagme verbal. Les "angles de Fries" regroupent les constituants en classes syntagmatiques qui décomposent la phrase en plus petits éléments et emboîtent les formations pour former des blocs plus gros jusqu'au syntagmes et à la phrase. Ce développement est repris par Hockett avec une numérotation des éléments emboîtés. La "boîte de Bloch" et Z.Harris décomposent les éléments avec des symboles de catégorie ("ART" pour article, "GV" pour groupe verbal…). Toute phase est la concaténation de syntagmes. Chomsky définit un arbre diagramme symbolisant une taxinomie des groupes décomposés.

Cette théorie a des défauts pour représenter le type de phase (interrogative, affirmative,…), les discontinuités (ne…pas), et désambiguer. Le distributionnalisme décrit des propriétés que l'on peut exploiter à travers des méthodes comme la classification. Ces propriétés sont bien décrites par [Harris, 1968] mais sans présentation formelle.
"The only distance between any two words of a sentence is the sequence of other words between them. There is nothing in language corresponding to the bars in music, which make it possible, for example, to distinguish rests of different time-lengths. Hence, the only elementary relation between two words in a word sequence is that of being next neighbours. Any well-formedness for sentence structures must therefore require a contiguous sequence of objects, the only property that makes this sequence a format of the grammar being that the objects are not arbitrary words but words of particular classes (or particular classes of words). But the sequence has to be contiguous; it cannot be spread out with spaces in between, because there is no way of identifying or measuring the spaces." (p16).
"We have seen that language can be described simply as sequences of classified words. The words (or at least subsequences of them) have semantic interpretation, and we can say that they designate classes of objects and relations and events in the real world. Words do not in general designate uniquely each individual object in the world, although a particular occurrence of a word may. Therefore ask how the semantic effect of sameness of individuals is obtained from sequences of words which by themselves do not have such a meaning" (p142).
"In describing the sentences of biochemistry we can define particular subclasses of words, such as names of proteins or various classes of molecules, and names of solutions, reagents, etc., and verbs for classes of reaction or laboratory activities that are carried out on the molecules. Of these specially defined subclasses, only particular sequences will be found in (true or false) well-formed sentences in biochemistry discourses. These sentences are also in the language as a whole, but other sentences in the language do not keep to the particular sequence of these particular words, so that the biochemical word subclasses and their well-formed sequences do not exist as such for the language as a whole. The axiomatic view of grammars is that a grammar constructed for a language (a set of sentences) consists of a set of word and morpheme classes (and subclasses), a set of well-formed sequences of these (elementary sentence structures), and a set of transformational rules which derive one sentences structure from another. In this sense, the grammar of the sentences in a particular science contains items additional to those of the grammar of the language as a whole." (p152).

La théorie distributionnelle s'affranchit donc de catégories syntaxiques prédéfinies et se propose de détecter des blocs syntagmatiques ayant le même comportement par rapport à un autre bloc (syntagme verbal, par exemple), d'où la notion d'analyse en constituants immédiats.

2.2.5 Grammaire transformationnelle

N. Chomsky a donc développé une théorie se basant sur l'idée que les compositions sont fixes et universelles pour toutes les langues. On peut donc, à partir d'universaux finis et un nombre fini de règles de composition, générer et analyser toutes les phrases possibles existantes et futures. La grammaire générative consiste à déterminer les liens entre 3 composantes : *sémantique* (mécanisme grammatical d'assignation, à base de règles, d'un ou plusieurs sens aux énoncés), *phonologique* (confère une forme phonétique aux unités de la phrase), *syntaxique* composée elle-même de composantes catégorielle, lexicale et de règles de transformations. La *composante catégorielle* est un ensemble de règles permettant d'analyser la structure profonde grâce à des symboles de catégories:

Par exemple:
P $\rightarrow$ SN + SV (phrase est un syntagme nominal et un syntagme verbal);

SN → Dét + N (un syntagme nominal est un déterminant et un nom);
SV → V + SN (un syntagme verbal est un verbe et un syntagme nominal).

On aboutit à un arbre des différents constituants appelé *dérivation*. La composante lexicale recense des unités de la langue avec des traits sémantiques valués. L'ensemble des traits est fini et chaque unité dispose d'un vecteur avec + si le trait est présent et – sinon. Exemple: étoile [+ N][+ commun][+ dénombrable][- humain][- adulte]…. A l'issue de l'analyse catégorielle et lexicale des ambiguïtés peuvent subsister. Les règles sont complétées par des règles de transformation. Elles peuvent rendre compte des modifications de structure que subissent les constituants.

Exemple:
X+Y → Z (réduction);
X+Y → Y+X (permutation).
…

Les règles de transformations sont la charnière du passage d'une structure profonde en une structure de surface. Par exemple, si la base des constituants est *l'+ étoile + s'effondre* et *la +masse +de + l' +étoile +fait que + quelque chose* conduit à "la masse de l'étoile fait qu'elle s'effondre". Des améliorations récentes perpétuent la pratique de l'approche transformationnelle, telles que la théorie des principes et des paramètres, la théorie minimaliste et la théorie x-barre.

2.2.6 Grammaire d'unification

Les grammaires statiques ou transformationnelles ont permis de progresser dans l'analyse syntaxique, mais assez peu en analyse sémantique. Des grammaires dites d'unification ont vu le jour pour essayer d'approfondir les aspects sémantiques. La grammaire lexicale fonctionnelle (angl. Lexical Functional Grammar, LFG) présente une adaptation aux universaux linguistiques plus que des arbres. La grammaire syntagmatique généralisée (angl. Generalized Phrase Structure Grammar, GPSG) présente une analyse grammaticale hors contexte. La grammaire syntagmatique guidée par les têtes (angl. Head-driven Phrase Structure Grammar, HPSG) essaie de creuser les niveaux d'analyse pour une meilleur interprétation. Les grammaires d'arbre adjoint (angl. Tree Adjoining Grammar, TAG) présente une extension des grammaires en chaîne pour formaliser l'approche distributionnaliste harrissienne.

2.3. Syntagmes nominal et verbal

Le recensement d'unités typiques d'un domaine s'appelle la lexicographie. La lexicographie vise à constituer des dictionnaires. La plus petite unité porteuse de sens dans la terminologie d'un domaine doit être univoque et monosémique. La lexicographie existe du fait des *néologies* c'est-à-dire de la création de nouveaux mots ou termes caractéristiques d'une langue vivante. Le syntagme est le représentant d'une terminologie. Il existe différents types de syntagmes:

- le syntagme nominal équivalent au mot simple: "proton", "fenêtre"…
- le syntagme nominal à trait d'union: "pompe-hélice"…
- les syntagmes nominaux sont soit épithétiques sans joncteur: "énergie solaire", soit synapsiques, avec joncteurs prépositionnels : "véhicule sur rail".

[Mari & Saint-Dizier, 2000] trouvent que les verbes ont très peu de comportement en classes d'équivalence, en moyenne 2.5 verbes ont un comportement équivalent par leurs arguments (méthode des Qualia du lexique génératif de J.Pustejovsky).

2.4. Champ sémantique

Le problème du sens a toujours fait l'objet de travaux en logique ou philosophie ou en psychologie. Il existe plusieurs formes de sémantique:

- la sémantique *linguistique* qui vise à produire une description du sens grâce aux composants linguistiques (mot, syntaxe, relations…);
- la sémantique philosophique (issue de Wittgenstein);
- la sémantique logique ou théorie logique des signes;
- la sémantique générale est l'application de la sémantique philosophique aux communications sociales.

2.4.1 Référence et extension

La *référence* est le lien entre le signe et le monde réel. Le *référent* est l'objet évoqué par le signe. L'extension d'un signe est la classe d'objets à laquelle ce signe se rapporte (l'extension de "fruit" est l'ensemble de tous les "fruits").

2.4.2 Dénotation et connotation

On oppose *dénotation*, élément stable de la définition d'un signe, avec sa *connotation*, c'est-à-dire ce qu'il peut représenter dans un contexte (L' âne est un animal de ferme mais peut représenter la bêtise)

2.4.3 Sens et signification

Le sens d'un signe représente l'aspect *intensionnel* et la signification l'aspect *extensionnel*. Le sens de "ville" a pour signification Paris, Strasbourg,… Pour le grammairien F. de Saussure le sens est la composition d'une valeur différentielle par rapport aux autres signes et d'une valeur référentielle qui se rapporte à un signifié.

2.4.4 Sèmes

Certains ont voulu aboutir à un catalogage des unités de sens minimum pour décrire des sens plus complexes liés aux mots. On appelle ces unités des *primitives*, des *sèmes* ou des mots pivots.

2.4.5 Champs sémantiques

Déterminer un champ sémantique c'est dégager la structure d'un domaine de significations. Traditionnellement cette structure est symbolisée par des relations d'ensemble hiérarchiques ou non hiérarchiques bien particulières. Il existe 4 types de relations : l'existence d'un ou plusieurs sens (*polysémie*/*monosémie*), l'existence d'un sens proche (*synonymie*/*antonymie*), la relation de spécificité (*hyponymie*/*hyperonymie*), la relation partie-tout (*méronymie*):

- les relations polysémiques traduisent la multiplicité des sens autour d'un mot ("bleu" peut vouloir dire "couleur" mais aussi un "nouveau recruté" ou un "personnel du service sanitaire");
- les relations monosémiques traduisent l'unicité du sens d'un mot ("gris" est la couleur résultant du mélange du noir et du blanc);
- les relations hyperonymiques traduisent une relation de généricité ("couleur" par rapport à "bleu");
- les relations hyponymiques traduisent la spécificité d'un mot par rapport à un autre ("bleu" par rapport à "couleur");
- les relations méronymiques traduisent un phénomène d'inclusion ("rouge" est inclus dans le "rose");
- les relations antonymiques traduisent un phénomène de symétrie ("blanc" et "noir");

- les relations synonymiques traduisent la possibilité pour deux mots de commuter parce que leur sens est voisin ("rouge" et "rougeâtre").

2.5. Extraction de termes

On distingue les méthodes linguistiques basées sur des règles syntaxiques, les méthodes statistiques basées sur les répétitions de séquences et les méthodes mixtes . Dans cette partie nous présentons 5 modèles issus de ces 3 approches.

2.5.1 Méthode du dictionnaire

Cette méthode s'appuie sur une ressource externe qui consigne les mots et expressions figées voire semi-figées susceptibles d'être rencontrées dans un texte du domaine. La terminologie a donc été répertoriée. Les formes rencontrées successivement sont donc comparées avec la ressource pour être identifiées.

2.5.2 Méthode des cooccurrences

La méthode des cooccurrences permet de créer un lexique par la répétition des formes présentes dans un texte. La base théorique constitue aussi le défaut de la méthode c'est-à-dire disposer d'une quantité statistique significative d'information.
[Smadja & McKeown, 1990] présentent un outil appelé *Xtract* qui extrait des syntagmes grâce aux répétition de n-grammes. Dans un premier temps des bigrammes sont extraits. Pour chaque entité un histogramme est établi dans une fenêtre de 5 mots avant et après l'entité donnée. Un test effectué grâce au z-score permet d'identifier les formes les plus discriminantes: $z = \frac{(N - E)}{\sqrt{E * q}}$. Où N est le nombre de mots dans le texte, E est le nombre d'occurrence espéré et q est la probabilité pour B d'apparaître où A apparaît. Une fois les bigrammes recueillis, on cherche à obtenir les n-grammes à partir des bigrammes et à leurs concordances de façon à aboutir au lexique final. Par exemple, dans la forme N2 N1 telles que "ammonium nitrate" on recherche les fréquences d'apparition du mot N2 à une distance -1 du mot N1. D'après l'auteur, cette méthode de traitement de distance permet de détecter des noms composés ou des termes. Mais c'est à l'utilisateur (un linguiste), de valider les cooccurrences obtenues.

2.5.3 Méthode des segments répétés

La méthode des segments répétés s'appuie sur la détection de chaînes constituées de morceaux existant plusieurs fois dans le même texte [Oueslati, 1996][Oueslati, 1997] [Lebart & et al 1998]. On symbolise les morceaux par des lettres A B C D équivalent par exemple à "château de la Loire", c'est-à-dire A="château", B="de", C="la", et D="Loire".

On fixe une fréquence minimale d'apparition dans le texte f_{min}. Cette fréquence permet aussi d'accélérer le processus de détection. On stocke tous les mots du texte dans une table *tnum*(i) dont la valeur correspond soit à une occurrence, soit à une ponctuation, soit à un symbole de structure du texte (saut de paragraphe, chapitre …).
Pour chaque forme A du texte, n'étant pas ni un article, ni une conjonction, ni un verbe, ni une préposition, ni un nombre, on répertorie toutes ses occurrences dans un tableau *list*(j) (jième occurrence de A). La forme A sera appelée *tête de syntagme*. On cherche donc toutes les expansions se rapportant à cette tête.

Exemple:
pour la tête "artère" on pourra trouver les expansions: "gauche", "droite", "coronaire droite"…
On trie par ordre alphabétique toutes les formes suivant la tête A:

S1 A B
S2 A B
S3 A B C
S4 A B C D E F G
S6 A B C D E F G
S7 A C B A

On n'a seulement besoin de *tnum* et *list* pour identifier les suites. Chaque suite sera conservée seulement si sa fréquence est supérieur à f_{min}. On parcourt l'ensemble des suites et on s'arrête de compter dès qu'une forme de la suite diffère de la forme précédente.
Exemple:
S1=S2
S2≠S3 car C∉ S1 on réinitialise le compteur pour S3 et on garde S1 si son compteur est supérieur à f_{min} .
On itère pour toutes les suites de la tête A.
On réitère le processus pour chaque tête du corpus.

2.5.4 Méthode des schémas syntaxiques

Le repérage de la structure interne d'un syntagme consiste à apparier des séquences lexicales dans un corpus avec des schémas syntaxiques préétablis. En ce qui concerne les syntagmes nominaux par exemple, on part de la constatation qu'ils sont construits sur un petit nombre de schémas, tels que :

- N N ("pomme Golden")

- N PREP N "(pomme de terre", "moulin à vent")

- N ADJ ("champ magnétique")

- N PREP N ADJ ("angine de poitrine instable")

- N ADJ ADJ ("champ magnétique intense")

Si à chaque mot du corpus est associée une étiquette représentant sa catégorie grammaticale, alors il est aisé d'extraire tous les syntagmes qui correspondent à un schéma donné.

L'assignation de catégories (étiquetage, ou angl., tagging) se déroule en deux phases : la reconnaissance des mots du texte et l'assignation de la catégorie. Dans le cas où plusieurs étiquettes sont possibles, le système en choisit l'une ou l'autre selon le contexte. Par exemple pour décider de l'étiquette de *la* dans *il la voit*, le système examine *il* et *voit*. *Il* n'est pas un déterminant, on peut donc exclure l'interprétation de *la* comme nom commun (comme dans *donner le la*); *voit* est un verbe, on peut donc retenir l'interprétation de *la* comme pronom.

La reconnaissance se fait à l'aide d'un dictionnaire et/ou d'un analyseur morphologique. Le choix de l'assignation se fait selon des règles préétablies, ou selon des procédures stochastiques. Il semble que ces dernières soient plus performantes. On procède ainsi. On étiquette manuellement quelques milliers de mots d'un corpus d'apprentissage. Le système infère des règles à partir de mesures de cooccurrences des catégories, par exemple avant un nom, il y a telle probabilité pour que l'on trouve un adjectif, ou un déterminant, ou un adverbe, etc. Ces règles sont ensuite appliquées à un corpus plus volumineux, et le résultat est corrigé à la main. Le système est ensuite en mesure de traiter le reste du corpus. Parmi les méthodes par règles, citons [Brill, 1995]; parmi les méthodes stochastiques, citons [Karttunen et al, 1997]. Le taux d'erreur de ces assignateurs est de l'ordre de 3 à 5 % selon les méthodes,

mais la désambiguation lexicale doit se faire à l'aide de grammaires locales que l'utilisateur doit définir lui-même.

2.5.5 Méthode des bornes

Deux variantes algorithmiques sont opérationnelles.

La première consiste à établir une liste de mots comme bornes d'expressions à extraire. Si une expression se trouve entre deux mots de la liste, alors on la retient comme terme.

La deuxième option est d'étiqueter un texte par des catégories syntaxiques aux mots de la phrase et de découper les fragments grâces à des frontières bien spécifiques. [Bourigault, 1994] utilise des marqueurs de frontières complétés par un étiquetage grammatical des mots du corpus afin d'acquérir des syntagmes susceptibles d'être des termes. Son outil d'extraction terminologique, *LEXTER*, utilise des bornes de syntagmes nominaux (SN) qui sont des mots appartenant, pour la plupart, aux catégories grammaticales suivantes : les verbes, les pronoms, les déterminants, les adverbes. Voici un exemple de bornes de syntagmes utilisés par *LEXTER*:
contexte : "...une douleur thoracique nocturne probablement..."
SN : douleur thoracique nocturne
bornes de syntagmes : une, probablement

LEXTER utilise différentes étapes pour accomplir ce travail de repérage des SN, on peut les résumer comme suit :
1) l'étape de catégorisation : elle consiste à assigner automatiquement une étiquette grammaticale aux mots de la phrase.
2) L'étape de découpage : elle consiste à repérer les groupes nominaux en utilisant la notion de frontière décrite ci-dessus.
3) l'étape de décomposition : elle consiste à analyser les SN en têtes (T) et expansions (E). Par exemple le SN "douleur thoracique nocturne" est décomposé en :T : douleur et E : thoracique nocturne
4) l'étape de structuration : elle consiste à organiser les termes en réseaux (pour chaque terme on établit des liens avec les autres termes dans lesquels il est en position de tête ou d'expansion). Cette organisation permet par exemple de naviguer dans le réseau à l'aide d'une interface utilisateur. D'après l'auteur, LEXTER repère presque la totalité des SN d'un texte (mais tous ses SN ne sont pas des termes). Pour permettre la confirmation des groupes nominaux en candidats termes, l'auteur utilise une validation qui consiste à observer dans le corpus les paires candidates les plus fréquentes qui sont utilisées pour désambiguer les structures obtenues par l'analyse de surface.

2.6. Modèle des classes d'objets

Le modèle des classes d'objets a des origines purement linguistiques. Le modèle répond aux lacunes du modèle des traits sémantiques caractérisant l'usage d'un mot dans un de ses sens [Gross, 1994]. Le principe de base s'inspire d'une relation de type prédicat-argument et de l'organisation syntaxique distributionnelle des groupes nominaux et verbaux d'une phrase pour établir des sous-catégories sémantiques explicites. On affecte des termes qui ont le même comportement distributionnel par rapport à d'autres termes, principalement des verbes. Par exemple: "emprunter" peut avoir deux sens. Dans un premier sens on obtient:

Emprunter de l'argent;
Emprunter une somme d'argent;
Emprunter 10 mille francs;

Dans un second sens on aura :

Emprunter un chemin;

Emprunter un passage obscur.

Deux classes seront donc constituées, dont l'une avec la sous-catégorie <transport> et les instances {passage obscur, chemin}, et l'autre avec la sous-catégorie <finance> et les instances {argent, somme d'argent, n mille francs}. Pour différencier les usages présents dans le même texte on utilise d'autres prédicats ("marcher sur un chemin" et non "marcher sur de l'argent"), ou bien on utilise des règles de transformations ("emprunt d'une somme d'argent" et non "emprunt d'un chemin"). [Gross, 1994] a constitué, d'après cette méthodologie et un certain nombre de corpus, un dictionnaire de 50000 substantifs regroupés en 100 sous-catégories.

2.7. Collocations et thésaurus

2.7.1 Collocation et cooccurrence

Il n'existe pas de différence très nette entre les notions de collocation et cooccurrence. Deux nuances pourraient les séparer; premièrement l'usage selon lequel les collocations seraient plus utilisées par des praticiens de méthodes linguistiques et cooccurrence plus usité par des statisticiens; deuxièmement, la notion de collocation fait plutôt appel à des relations syntaxiques très proches au sein d'unités syntagmatiques alors que la cooccurrence porte sur une relation plus élastique [Oakes, 1998]. Un des premiers à avoir travaillé sur le comportement statistique des cooccurrences dans les textes (avec un impact encore aujourd'hui dans les outils du TAL) est le mathématicien [Markov, 1916].
Les linguistes [Firth, 1968[1953]] [Trier, 1931] émettent l'hypothèse que l'on peut catégoriser un mot par les mots qui lui sont cooccurrents. Le repérage et l'extraction automatique des collocations ont été la base de toutes les études de construction automatique de dictionnaire et de thésaurus. Le projet européen DECIDE (Designing and Evaluating Extraction Tools for Collocations in Dictionaries and Corpora) [Fontenelle et al, 1994] qui a eu lieu entre 1993 et 1996 fait un très bon état de l'art des outils d'extraction de collocation et de leur utilisation possible. On peut noter l'utilisation de l'outil Sextant [Grefenstette, 1994] qui permet de considérer des collocations n'apparaissant pas forcément dans le même contexte syntaxique. L'outil se base sur des cooccurrences entre des noms et leurs prédicats pouvant être des adjectifs et des verbes. Un étiquetage permet de repérer les futures entités à classer et les relations prédicatives, notamment de types: Nom-Adjectif, Sujet-Verbe et Verbe-Objet. Les individus à classer sont donc des noms simples et les variables des unités formant des relations prédicatives syntaxiques. Pour chaque paire d'entités on calcule un indice de similarité dépendant de combien les entités sont différentes de la paire analysée avec laquelle les attributs s'associent et combien de fois l'attribut apparaît avec telle entité.
Coefficient de similarité :

$$C(\text{objet}_A, \text{objet}_B) = \frac{\sum_i \min_i \left(w(\text{objet}_A, \text{attribut}_i), w(\text{objet}_B, \text{attribut}_i)\right)}{\sum_i \max_i \left(w(\text{objet}_A, \text{attribut}_i), w(\text{objet}_B, \text{attribut}_i)\right)} \quad ,$$

$$w(\text{objet}_A, \text{attribut}_i) = \log(F(\text{objet}_A, \text{attribut}_i) + 1) * \frac{p(A,i) * \log p(A,i)}{\text{nrel}}$$

poids de la relation où F est la fréquence de l'objet A et de l'attribut I et nrel est le nombre total de relations dans le corpus;

$$p(A,i) = \frac{F(objet_A, attribut_i)}{N^i_A}$$

est la probabilité d'avoir une association (A,I) où N^i_A est le nombre d'attributs de A.

Le coefficient d'association peut s'interpréter comme un coefficient de Tanimoto, ce dernier considère le rapport des attributs communs et des attributs non partagés alors que C teste l'intensité d'association d'un couple objet-attribut grâce à la fréquence et à l'information mutuelle et pas simplement avec le nombre d'attributs. Les cooccurrents sont retenus si le coefficient C dépasse un seuil fixé. Pour chaque nom on obtient ainsi les cooccurrents les plus représentatifs. La fiabilité est d'autant plus élevée qu'il y a plus de 50 contextes par entité. On atteint des performances de l'ordre 75% dans l'exactitude des paires obtenues. Cette approche peut presque s'apparenter à une méthode des k-moyennes dans laquelle chaque entité joue le rôle de classe.

2.7.2 Désambiguation et structuration du sens dans les textes

[Zernik, 1991] utilise une méthode d'étiquetage des sens d'un mot par classification hiérarchique (lien complet). Il utilise des vecteurs de mots trouvés dans les concordances (5 mots avant et 5 mots après) pondérés par des probabilités de présence. Il arrive à désambiguer *to train* et *a train* mais pas office dont la catégorie syntaxique est la même avec des sens différents. [Phillips, 1989] utilise la classification de terme avec le sens relatif pour découvrir la structure lexicale des textes techniques, c'est-à-dire la macrostructure lexicale d'un texte. Il utilise une méthode de Ward avec une matrice de collocations normalisée par un coefficient de normalisation (annexe 7). Les collocations sont détectées dans une fenêtre de 12 mots maximum à droite et à gauche.

2.7.3 Exploitation d'un thésaurus

[Loukachevitch & Dobrov, 2000] utilise un thesaurus avec 3 types de liens sémantiques: *hypéronymie*, association et *partie-de*. Elle projette un texte sur le thesaurus pour extraire le réseau de concept relatif au texte. Les concepts principaux sont identifiés et leurs relations sont associées à la *cohésion lexicale* du texte.

[Grefenstette, 1996] utilise des thésaurus pour tester la cohésion lexicale d'une paire de mots (mot cible/mot contextuel le plus caractéristique) avec un corpus de 4 Mo et 3000 mots cibles de fréquence > 10. 3 thésaurus sont utilisés: Roget's, MacQuarie et Webster. Le thésaurus Roget classe 30000 mots en 1000 catégories, Webster est un dictionnaire de définition ramené en définition séquentielle de mots nettoyés par une liste de mots vides (angl. "stop list") (434 mots). MacQuarie a la même structure que Roget. Les résultats visent à comparer la cohésion du mot le plus caractéristique extrait soit par une méthode syntaxique (relation adj-nom, nom-verbe…), soit par une méthode de cooccurrence basée sur une fenêtre de 10 mots (à gauche et à droite), le mot le plus caractéristique étant décidé par un indice de Jaccard. L'expérience montre 2 choses: que l'appartenance de la paire à une même catégorie n'intervient pas à plus de 50% des cas et ce pour les mots cibles les plus fréquents, ensuite il est difficile de départager la méthode syntaxique de la méthode des cooccurrences. Comme la probabilité que 2 termes appartiennent à la même catégorie <1%, l'utilisation d'un thésaurus semble être utile mais pas suffisante en soi pour caractériser la sémantique d'une paire.

3. Apports possibles aux approches de classification

3.1. Limitations des méthodes

On constate d'une manière générale que le problème de montée en charge (angl., "scalability") des données conduit à repousser un certain nombre de méthodes classiques; ce n'est pas la

même chose de travailler avec un tableau de 100000*5000 qu'avec un tableau de 200*50. La nature des données joue aussi sur le processus dans son aspect interprétatif; ce n'est pas la même chose de travailler sur une image que sur du texte.

Les méthodes de regroupement par paires proposent d'aboutir à un schéma d'arbre dont la coupure fixerait la nature de la partition. L'arbre présente donc les similarités entre objets de manière pseudo-continue. Le critère s'avère souvent arbitraire (10% de la base). De plus une classification hiérarchique ascendante crée des chaînes ("escalier") si les individus ont des écarts faibles deux à deux dans le cas d'un lien par saut minimum. Dans le cas d'un lien complet ou des voisins réciproques des boules sont crées là où les distances sont proches, le lien moyen semble être un compromis entre les deux défauts précédemment expliqués. La classification descendante demande à ce qu'on fixe le nombre de classes tout comme la méthode des centres mobiles (en général entre 1 et 20). La construction de l'arbre est en plus contrainte par la nature des attributs et se prête peu à une construction incrémentale par ajout d'attributs, l'ajout d'attributs remettant en cause la construction précédente. La classification hiérarchique doit être considérée stable pour être modifiée. Son mode de construction ne permet pas des modifications de type incrémental. On parcourt l'arbre pour réaliser une opération élémentaire. Il s'agit davantage de classement ou de complétion plutôt que de classification dans ce cas. Des algorithmes de complétion existent permettant de parcourir un arbre et de le modifier localement. On envisage les opérations suivantes: *élimination* d'un élément perturbant, *illustration* par des centres de gravité fictifs, *édification* avec des individus complémentaires, *placement* d'individus ayant une description incomplète, *vérification* de l'homogénéité (calcul d'erreur).

D'après [Roux, 1985], la faiblesse des algorithmes de classification est due uniquement aux lacunes théoriques des méthodes qui fait qu'aucune des méthodes ne se démarque par la qualité de ses résultats. On ne peut palier ce défaut qu'en combinant des méthodes pour optimiser la classification et améliorer la qualité des résultats. Par exemple, l'analyse factorielle peut servir à sélectionner les meilleures variables grâce aux axes de plus grande inertie, ou à sélectionner des groupes susceptibles de conduire à de bonnes classes avec un nombre réduit d'individus. A la suite de cela, une classification ascendante hiérarchique permet d'aboutir à une bonne classification. Sur les 6 méthodes classiques (centres mobiles, classification hiérarchique descendante, classification hiérarchique ascendante de lien simple, de lien moyen, de moment centré ou de diamètre) la meilleure, selon [Roux, 1985], semble être la classification hiérarchique de lien moyen. Mais elle ne peut être utilisée comme boîte noire et doit être validée par d'autres méthodes ou un expert qui valide sa cohérence. Le problème vient du fait que même dans le cas ou tout ou partie des données n'est pas classifiable l'algorithme va aboutir à une partition. Comment caractériser mathématiquement des données non classifiables?

Les réseaux de Kohonen ou ART ont tendance à produire une classe fourre-tout et des petites classes tout autour.

3.2. Contraintes techniques

Dans la plupart des méthodologies de classification certaines approximations liées à la nature des classes sont souvent prises en compte. Ces approximations apportent du confort au traitement plus qu'une connaissance sur les données traitées. Dans le processus de classification de termes que nous souhaitons développer on s'affranchit des limites concernant le nombre de classes et la taille maximale des classes recherchées. On suppose que ces paramètres ne dépendent que du domaine et de la nature des données. Ils sont donc

transparents à l'utilisateur dans le cadre d'une méthodologie robuste et sans connaissances a priori sur les classes recherchées. De plus nous critiquons l'utilisation d'un seuil de similarité pour décider l'affectation d'un élément à une classe. Notre critique se situe sur la question du voisinage autour du seuil S et la justification théorique associée à la variation de la distance dans ce voisinage. Pourquoi affecter à S+ε et ne pas affecter à S-ε avec ε <<S ?
Notre objectif sera donc de s'affranchir d'un seuil de similarité.

3.3. Connaissances utiles

La collecte des données est un facteur de réussite de n'importe quel traitement statistique. Nous cherchons à obtenir une bon dénombrement statistique des entités textuelles avec le moins de biais possible. Pour ce faire, nous devons considérer des processus de réduction canonique de formes (la variation des formes étant très fréquente dans le langage naturel). Deux processus sont envidagés:
- la lemmatisation;
- la réduction des formes composées variantes.
De plus nous considérons que des connaissances statiques sont utiles pour qualifier l'état d'une entité textuelle (terme, ou groupe de termes). Des connaissances sémantiques générales sous forme de relations simples (i.e. thesaurus) devraient aider l'interprétation de groupes de termes du même contexte. Nous envisageons donc la constitution de telles ressources pour des aides à l'interprétation et la qualification des classes de termes obtenues.

3.4. Interaction avec le besoin utilisateur

L'application est sensible à l'exploitation des résultats. Dans le cas d'un module de filtrage/routage l'utilisateur ne doit pas être sollicité en permanence pour évaluer les résultats. On peut donc s'affranchir d'une interface de transcription dont la syntaxe est normalisée. Les résultats peuvent se comporter comme une boîte noire. Cela n'est pas forcément possible avec d'autres types d'applications où l'utilisateur est sollicité comme une interface de recherche documentaire ou un outil d'aide à la traduction.

4. Outils de fouille de texte

4.1. Recherche documentaire vs agent de recherche d'information

Il est quasiment impossible de parler de la fouille de texte sans évoquer la recherche documentaire. La recherche documentaire est un domaine d'application aussi ancien que la recherche de données en informatique [Bush, 1945]. Les applications liées ont buté sur les problèmes de la langue naturelle. C'est pourquoi les systèmes n'ont cessé d'être améliorés encore aujourd'hui.

Au début, des méthodes théoriques et expérimentales ont été mises en place pour élaborer des architectures complètes du traitement d'une requête utilisateur, du stockage des données en passant par les différents types de traitement ainsi que l'évaluation (modèle de Cranfield). Un système d'extraction et de stockage d'information est composé de 4 fonctionnalités principales [Kowalski, 1997]:
* normalisation de documents,
* diffusion sélective d'information (par exemple par messagerie),
* recherche de documents dans une base archivée,
* processus d'indexation.

On distingue 2 types de moteurs de recherche d'information: la recherche exacte de type annuaire (l'utilisateur connaît le format de sa réponse et formule sa requête en adéquation, en général ce type de recherche est souvent lié à des requêtes par champs textuels, exemple:

publication dont l'auteur est Dupont); le 2ième type concerne la recherche floue (l'utilisateur sait ce qu'il cherche sans connaître le format de la réponse avec une requête à formulation variable, c'est à dire floue elle-même, exemple: publication parlant de galaxie).

Des traitements automatiques de la langue se sont mis en place, quoique plus rudimentaires que de nos jours. Des optimisations se sont développées pour combler le déficit de performance du matériel. Les méthodes d'indexation de l'époque étaient très souvent basées sur des mots simples présents ou non dans des résumés. Des efforts de normalisation du vocabulaire ont été réalisés grâce à des thésaurus ou aux méthodes de classification. Mais les résumés présentaient peu de consistance en matière de fréquences d'apparition simultanée entre plusieurs termes. Les approches syntaxiques considérant des relations dans une unité inférieure au résumé n'ont pas été non plus approfondies. Le renouveau des méthodes basées sur les corpus (plein texte) à partir de 1980 avec les performances accrues du matériel ont permis de reconsidérer les méthodes statistiques avec un meilleur degré d'application. Les améliorations des outils TAL, notamment des traitements morpho-syntaxiques, ont permis d'envisager une indexation à base linguistique. Des approches mixtes statistico-linguistiques apparaissent pour exploiter des considérations propres et simultanées aux données: leur fréquence et leurs relations [Sparck-Jones, 2000].

[Salton & McGill, 1983] recommande d'utiliser une matrice termes*termes en passant par une matrice termes*documents et d'appliquer une méthode hiérarchique avec simple lien pour réaliser une constitution automatique de thésaurus.

La différence entre un système de recherche d'information (SRI) et un système de gestion de base de données (SGBD) tient à la sémantique des données à gérer. Un SGBD stocke les données avec une sémantique définie et non ambiguë sous forme de table et d'attributs qui évite toute confusion dans le traitement de requête pour l'interrogation ou la mise à jour. Un SRI gère des données non structurées ayant un format plein texte. Les thèmes et les relations décrits dans les documents sont explicités par des formes non normalisées et souvent ambiguës.

Les systèmes robustes d'indexation automatique prennent leur source dans l'automatisation des sources de données bibliographiques (résumés) ordonné par le gouvernement américain vers 1965 pour éviter l'indexation manuelle (catalogage). Trois types d'indexation se sont imposés:

- *l'indexation statistique* utilise la fréquence des occurrences d'objets pour calculer le pourcentage de pertinence (approche vectorielle, ou angl., "vector indexing").
- *l'indexation probabiliste* calcule une probabilité qui devrait être invariante dans la méthode et dans un corpus (approche bayésienne)
- *l'indexation linguistique* qui traite des termes principaux (simples ou complexes) et de leurs relations en différentes classes sémantiques (synonymie, cause/effet, temporelle, actions, hyperonymie…) (approche NLI, ou angl., "natural language indexing").

Les tendances actuelles visent à fusionner les approches pour profiter des avantages manifestes de chacune des approches . Cependant l'usage le plus répandu focalise sur l'approche statistique qui construit des vecteurs pondérés de mots.

On citera 2 exemples de pondération:
Pondération probabiliste par régression logistique [Gey, 1994]:

$$P(R/(Q_i, D_j) = \frac{1}{1 - e^{-\log O(R|Q_i, D_j)}}$$

Log O est le log des chances de pertinence pour les termes présents dans la requête Q_i et le document D_j

Pondération statistique par la fréquence inverse du document [Salton & McGill, 1983].

Fréquence corrigée du terme I dans le document j:

$$\tilde{F}_i = \frac{\frac{1+\log F_i}{1+\log \overline{F}}}{(1-K).P + K.N}$$

Poids du terme i pour le document j:

$$w_{ij} = \tilde{F}_i . \log\left(\frac{n}{IF_j} + 1\right)$$

Pour un terme i de requête on a:

$$QTerm_i = (0.5 + (0.5\frac{F_i}{MaxF_i})).IFi$$

F_i: fréquence du terme i,
$MaxF_i$: maximum des fréquences des termes dans les documents et appartenant à la requête,
N: nombre de mots uniques,
F: fréquence moyenne des termes,
K: 0.2,
P: nombre moyenne de termes uniques apparaissant dans la collection,
N: nombre total de documents,
IF_j: nombre de documents qui possèdent terme i.

La correspondance entre une requête et un document se calcule comme une similarité. On pourrait utiliser un simple produit vectoriel comme:

$$SIM(Q_i, D_j) = \sum_k Term_{ik} . Term_{jk} \ ,$$

mais cette mesure ne tient pas compte de la variabilité de présence (normalisation) des termes dans les documents. Il faut donc pondérer la distribution des termes en fonction de la collection de documents. D'autres indices, classiques, sont utilisés comme le cosinus, l'indice de Dice ou de Jaccard (annexe 3).

Une alternative au modèle de l'espace vectoriel qui représente un document et une requête par des occurrences et des fréquences est l'indexation sémantique "latente" (angl., Latent Semantic Indexing ou LSI) [Deerwester et al, 1990] qui utilise une projection dans un espace propre identique à une analyse en composantes principales. Une décomposition en valeurs singulières réduit les dimensions de l'espace et conserve les facteurs les plus représentatifs. La requête est ensuite projetée dans le même espace pour y être comparée document par document par mesure de similarité de type cosinus. Cette solution semble attractive pour sélectionner les bons représentants de chaque document pour économiser l'espace de recherche mais s'avère coûteuse en temps de calcul:

$O((\textit{dim espace propre})^3 * (\textit{matrice terme*document})^4)$.

L'évaluation a été assez tôt normalisée par des paramètres qui s'appliquent surtout à de faibles quantités de données et pour un utilisateur, il s'agit du paramètre de précision p et du paramètre de rappel. Le paradoxe de la recherche d'information est de ne jamais avoir simultanément une bonne précision et un bon rappel (figure 2.13).

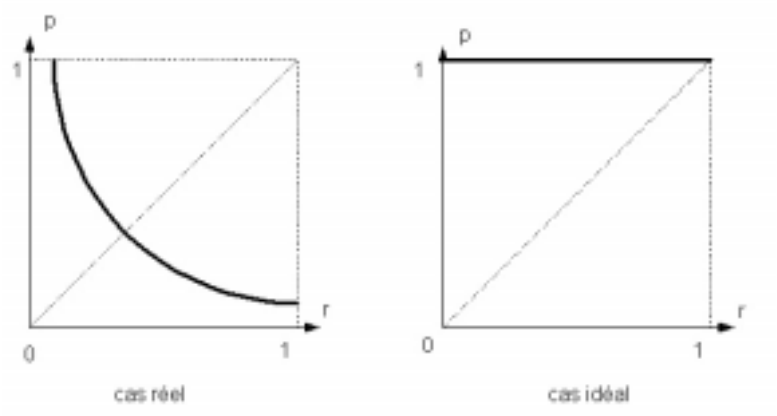


Figure 2.13 Diagramme précision-rappel

On trouve 2 fonctionnalités principales dans un SRI: la formulation de requête booléenne et la navigation. Trois opérateurs booléens fondamentaux sont utilisés: *AND*, *OR* et *NOT*, plus un opérateur de proximité soit "…" (contigu) ou *NEAR* (non contigu). Le navigateur permet de se retrouver dans les résultats. Les documents peuvent être synthétisés par des résumés (angl. zoning) constitués par le début de document, sa date de modification, son titre… La liste des documents résultats est souvent importante (>10), on établit alors un ordre d'affichage (angl. ranking) calculée par une méthode donnant un pourcentage de pertinence (angl. scoring). Les mots de la requête peuvent être mis en surbrillance pour identifier la cause de l'extraction du document (angl. highlighting).

La requête est souvent réduite (2 à 4 mots en moyenne). Elle peut être étendue de plusieurs manières: l'expansion (grâce à un thésaurus qui fournit des termes synonymes), la reformulation (grâce à des groupes de termes associées souvent rencontrés dans le même contexte), le "relevance feedback"(angl.) (par des mots clés issus des meilleurs documents résultant d'une première requête).
De nombreux outils de recherche d'information et de traitement arrivent sur le marché, depuis les moteurs de recherche jusqu'aux logiciels les plus sophistiqués (réseaux neuronaux, analyseurs sémantiques, automates de réseau), en passant par les agents intelligents sur Internet.

Sur Internet, des indexeurs permettent aux utilisateurs d'accéder aux pages du web par leur portail (angl. Gateway): un des premiers à s'être mis à indexer le web a été *WAIS* (Wide Area Information Server), aujourd'hui disponible librement (*FreeWais*). Il a participé à la mise au point d'un standard d'indexation sur le traitement et les opérations à réaliser notamment avec le protocole Z39.50 qui définit 8 opérations de base pour un outil de recherche d'informations (initialisation, présentation, recherche, suppression, identification, tri, report, services étendus). On trouve plusieurs types de moteurs (annexe 1):

- les métamoteurs, qui permettent de traduire une requête dans la syntaxe propre à plusieurs moteurs et de l'expédier à chacun d'entre eux, en éliminant en retour les doublons. L'intérêt est d'augmenter son pourcentage de couverture du web.

- les moteurs mêmes couvrant partiellement l'ensemble du web.

- les agents qui voyagent dans le réseau. Les " PC Seekers " (angl.), spécialisés dans la collecte d'informations, prolongent le fonctionnement des métamoteurs en reformulant la requête avec des opérateurs booléens.

DR-LINK (angl. Document Retrieval through Linguistic Knowledge) traite les documents avec une approche morphologique, syntaxique, sémantique et des niveaux du discours en utilisant des modèles prédéfinis de relations de faits. Une abstraction de chaque phrase est faite où chaque analyse (TAL) tire partie de l'analyse précédente pour aboutir à l'analyse du discours. Ainsi une analyse temporelle peut être dégagée et différencier "la compagnie vend" de "la compagnie a vendu".

4.2. La fouille, un outil du système d'information

La fouille de texte est l'idée de la fouille de données (angl. data mining) transposée aux données non structurées ou moins structurées comme les fichiers au format textuel. A ce titre la classification est utilisée aussi bien pour classifier des documents que des termes. La fouille permet au propriétaire ou à l'utilisateur des données de trouver des nouveaux aspects et connaissances par l'application de:

- Schémas dans les données, étant identifiables par l'usage de requêtes traditionnelles et de tableaux de bords.
- Techniques, permettant des comparaisons à réaliser à travers les données émanant de plusieurs sources de différents types, par extraction d'information ne pouvant pas être perceptible par l'utilisateur, et d'organiser les documents et l'information par leur sujet et leur thème.

La fouille de données structurées a l'avantage de l'infrastructure des données stockées, (e.g labels et relations) afin d'extraire des informations utiles additionnelles. Par exemple en fouillant une base de données clients on pourra découvrir chaque personne qui achète un produit A mais aussi un produit B et C mais seulement 6 mois plus tard. Un approfondissement de l'analyse montrerait s'il s'agit d'une progression nécessaire ou d'un retard causé par une information commerciale inadaptée. Dans ce dernier cas des techniques marketing pourraient être appliquées pour éduquer les utilisateurs et écourter le cycle de vente.

La fouille de texte doit opérer dans un monde moins structuré. Les documents possèdent rarement une infrastructure interne rigide (et dans ce cas cela concerne la nature du document beaucoup plus que le contenu du document même). Les données clés de documents que représentent les termes du domaine et les relations typiques du domaine sont stockées comme les données factuelles et temporelles dans une base de données pour être ensuite recherchées. La fouille de texte répond à une nécessité de disposer de nouvelles méthodologies de traitement des volumes croissant de données textuelles à notre disposition et sous lesquelles on peut se retrouver très vite dépassé. La fouille de texte est puissante en soi dans la mesure où le désarroi de l'utilisateur face à cette information massive peut s'effacer au profit d'une mise en valeur de données précises utiles dans son quotidien professionnel.

La fouille de texte aide à (annexe 2) [Kohonen, 1997] [Hearst, 1999]:

- capter le contenu "caché" des documents induisant des relations utiles;
- informer sur l'existence de documents émanant de divisions non référencées; découvre que les clients dans une autre division de produit ont les caractéristiques recherchées;
- grouper des documents en thèmes communs, e.g. tous les clients qui terminent leur contrat ou ceux qui réclament quelque chose.

Cela peut induire une société à:
- identifier et résoudre un problème ;
- éliminer des défauts en trouvant des schémas répétitifs ;
- trouver de nouvelles opportunités, notamment avec les clients actuels.

L'analyse de contenu, d'après [Berelson, 1952], est une technique qui a pour objet une description objective, systématique et quantitative du contenu manifeste. Ainsi par analyses successives on définit un certain nombre de thèmes et dans chaque thème un certain nombre d'attitudes.

La stratégie "par requête", adoptée par la quasi-totalité des moteurs de recherche ou agents intelligents actuellement disponibles, utilise de nombreux palliatifs, qui se combinent également entre eux, pour augmenter la pertinence.
On peut citer :
Les technologies linguistiques permettent, par un traitement du langage (lemmatisation, analyse syntaxique), de lever la majorité des ambiguïtés de la langue et de réduire à ce titre le nombre de réponses. Cependant, le sens d'un texte n'est en aucun cas la somme des sens des phrases qui le composent, et l'extension d'une phrase à l'autre par gestion des anaphores n'est pas suffisante pour reconstituer le sens global d'un texte .
Les technologies conceptuelles (graphes conceptuels) : elles s'appuient sur une analyse conceptuelle utilisant une nomenclature de structure syntactico-sémantique, un ensemble de méta-connaissances, et un ensemble de règles d'inférence. Le goulot d'étranglement de cette technologie est la mise au point d'un dictionnaire conceptuel par domaine, qui prend plusieurs mois de travail.

Intelligent Miner d'IBM permet d'identifier le langage d'un document c'est-à-dire construire un dictionnaire de noms propres, d'organisations, de lieux, d'abréviations, de termes composés. Les formes variantes des mots seront associées à une forme unique. Ce traitement n'est pas effectué par un dictionnaire mais de façon automatique. Les documents sont regroupés par similarité en utilisant une technique de classification. De cette manière la catégorisation est rendue autonome. Le moteur assure une recherche par requête dans un mode booléen, flou, texte libre, phonétique ou combiné pour des langues européennes et asiatiques. Un navigateur lexical (interface graphique utilisateur) permet de visualiser le réseau de relations et de concepts et d'accéder aux documents. Une aide à la reformulation de requête est proposée pour assister l'utilisateur à ajuster sa requête et converger vers la solution recherchée. L'outil peut par exemple trouver des documents grâce aux exemples ou aux dates butoirs (angl. Deadlines) dans des messages électroniques.

Une solution intéressante pour traiter des corpus hétérogènes, volumineux et instables est fondée sur les techniques d'apprentissage, ce que l'on qualifie de démarche "par apprentissage". Contrairement aux approches documentaires décrites ci-dessus, elle présente directement à l'utilisateur des événements et associations contenus dans sa base d'information traitée (dans le cas de *ETAT PARTENAIRE*, qui a été développé en 1997 sur financement du Secrétariat Général de la Défense Nationale (SGDN)).

[Waterman, 1996] essaie d'extraire des schémas sémantiques grâce à la classification en utilisant une distance de Levenshtein (comptage de suppression, substitution) et une classification hiérarchique ascendante par lien moyen. Il trouve très peu de schémas se ramenant à un paradigme comme S_a="without admitting T1" T1 étant une connotation

négative ou S_b ="joint venture of T1 corp and T2 inc" où T1 et T2 sont des entreprises. S_a et S_b représentent des expressions idiomatiques utilisant une catégorie paradigmatique.

[Nédellec, 2000; Faure & Nédellec, 1998] crée une ontologie grâce aux compléments verbaux extraits d'un corpus. Un prétraitement étiquète syntaxiquement un corpus d'apprentissage pour extraire des propositions *Verbe + noms + complément*. Les verbes sont considérés comme des prédicats à deux arguments. Les noms dont la position syntaxique est identique par rapport au même prédicat et aux mêmes compléments sont considérés comme des paradigmes ou classes de base. Ces classes de base peuvent être scindées en plusieurs classes complémentaires. Si C12(= C1∪C2) est une classe de base alors (C1∩C2) ∪ (C2-C1) ∪ (C1-C2) =C12 sont considérées aussi comme classes de base. Une classification ascendante hiérarchique permet de fusionner les concepts (classes ou prédicats) qui ont des objets en communs.

$$d(C_1, C_2) = 1 - C(C_1, C_2)$$

$$C(C_1, C_2) = \frac{\log \sum_{i=1}^{|C_1 \cap C_2|} F_1(\cap) * \omega_1 + \log \sum_{i=1}^{|C_1 \cap C_2|} F_2(\cap) * \omega_2}{\log \sum_{i=1}^{|C_1 \cap C_2|} F_1 * \omega_1 + \log \sum_{i=1}^{|C_1 \cap C_2|} F_2 * \omega_2}$$

$F_1(\cap)$ est la fréquence d'un des objets de C_1 aussi objet de C_2;
F_1 est la fréquence d'un des objets de C_1;
ω_1 est le nombre d'objets en commun entre C_1 et C_2 sur le cardinal de C_1.

Cette distance rapproche des classes qui ont des objets en commun et ayant des fréquences voisines dans chaque classe. Par exemple, la classe [cuire dans](sauteuse,4; cocotter,12; poêle,2; four,4) et [faire revenir dans](sauteuse,4; cocotter,12, poêle,2) auront une distance de 0.04. L'utilisateur fixe un seuil pour autoriser les agrégations. L'utilisateur nettoie aussi les classes impropres et établit les étiquettes des concepts pour chaque nœud de la hiérarchie. Le résultat est une description des entités du domaine et de leurs relations (ontologie) dont l'évaluation d'après les auteurs reste difficile à entreprendre.

Dans le même esprit, qui consiste à modéliser un domaine pour en exploiter les mot-clés les plus importants et les relations qui les unissent, [Assadi, 1998] tente de construire une ontologie de façon semi-automatique. Il procède en trois étapes. La première assure l'extraction de termes du domaine à partir de documents techniques. La deuxième étape, analyse macroscopique des champs sémantiques, réalise une classification hiérarchique ascendante des termes extraits grâce au module Lexiclass. La troisième étape qu'il qualifie d'analyse microscopique fait intervenir un expert du domaine pour décrire les concepts centraux grâce aux sèmes (unités de description sémantique) émanant des regroupements. L'analyse globale exploite le format des termes *tête+expansion* créant ainsi une matrice de données utilisant les têtes comme prédicats à classer, en fonction de leurs arguments les expansions. Dans ce cadre d'étude l'intervention de l'expert n'est pas négligeable, car il valide, tout au long du processus, les termes, les regroupements, les labels de concepts, les relations essentielles à conserver. Les mises à jour et le stockage sont ensuite automatisés par des requêtes de type SQL.

[Feldman & Dagan, 1997] établit une hiérarchie conceptuelle à la main se rapportant au thème à partir d'un corpus de documents d'un thème donné. Il compare les distributions de termes pour un nœud donné de la hiérarchie grâce à un calcul d'entropie relative. Le système *KDT* (angl. Knowledge Discovery from Texts) s'inspire de cette approche pour naviguer dans les

textes en comparant les distributions de mots. Le système *FACT* [Feldman, 1996] est quant à lui basé sur les cooccurrences. Il génère un langage de requête grâce à une interface visuelle. La requête va agir sur les mots-clés qui sont écrits dans les documents (champ mot-clé d'une dépêche). Les mots clés représentent 5 catégories: pays, sujet, personne, organisation et marché. La gestion de la requête se base sur une connaissance a priori tirée d'un livre (1995 CIA World Textbook) divisé en 6 sections: géographie, état, personnes, économie, communication, défense. Le but du système est de trouver des associations de termes catégorisés et d'extraire des documents. Deux types d'associations sont en vue: association gauche et association droite. Deux paramètres sont utilisés: un seuil de nombre de documents pour les associations et un seuil de confiance (probabiliste) du pourcentage d'apparition d'une association. Finalement, deux types de contraintes (utilisant l'ontologie) décrivent les associations: une prédication unaire sur le mot-clé de l'association mot-clé/membre de la classe de la contrainte, et une prédication binaire définit des relations entre mot-clés donnant un argument pour le prédicat, FACT retourne une série de valeurs pour le 2^{ième} argument qui doit être vrai.

Les systèmes *Zellig* et *Syclade* [Habert et al., 1996; Nazarenko et al, 1997] tentent de décrire un domaine grâce aux mots contenus dans un corpus. Cette description du domaine (ou ontologie) est supposée ensuite assister une recherche d'information. La structuration est basée sur l'information syntaxique de termes complexes qui sont décomposés en sous expressions par contiguïté ou par suppression de mots intermédiaires. Par exemple, "sténose serrée du tronc commun gauche" sera décomposé en "sténose serrée", "sténose du tronc", "tronc gauche" et "tronc commun". Ces expressions ont une structure syntaxique décomposable en *tête+expansion*. Les expressions sont ensuite reliées en réseau par le fait qu'elles disposent d'un nombre commun d'expressions dans lesquelles elles figurent à la même place comme tête ou expansion. Par exemple "malade" et "athérosclérose" sont reliées grâce à (- coronarien; diagnostic de - ; fréquence de -). Un seuil fixe le nombre minimal d'expressions communes pour conserver une relation. Le réseau obtenu donne une vue globale sur un corpus avec une exploration alternative des aspects syntagmatiques et paradigmatiques des contextes d'un mot. Deux types de sous-graphes sont aussi extraits : les sous-graphes fortement connexes présentant les mots du voisinage et les cliques présentant des classes de similarité. D'après les auteurs les cliques apportent une vue paradigmatique de l'ensemble des formes qui, dans un premier temps, s'interprètent comme des classes ontologiques reflétant des concepts.

Le système *Syndikate* de [Hahn & Schnattinger, 1997] propose de choisir le concept le plus crédible d'une ontologie (subsumant) établie avec un processus de raisonnement terminologique. Le système exploite des connaissances qualitatives sur les phénomènes linguistiques de textes libres et des configurations structurelles dans les bases de connaissances d'un domaine. Un réseau de 345 concepts et 347 relations sont ainsi prédéfinis et décrits dans un formalisme de logique de description. Les logiques de descriptions relèvent de la logique des prédicats et sont composées de concepts (objets) primitifs et de relations (rôles) primitives à partir desquels on dérive d'autres concepts et relations. Pour un objet donné d'un texte du domaine, à partir de concepts repérés dans son contexte, des hypothèses de concepts sont générées pour rechercher son plus proche subsumant. Par exemple, pour l'objet inconnu *itoh-Ci-8* les phrases "la mise en marche de *itoh-Ci-8* ..." donnera l'interprétation (Groupe nominal, Itoh-Ci-8, a_pour_fonctionnement, fonctionnement.1) ou (Itoh-Ci-8 avec la mémoire principale" donnera (Groupe prépositionnel,Itoh-Ci-8,, a_pour_mémoire, memoire.1) avec, d'une manière générale, (nature linguistique, objet, rôle,

argument). Un taux d'apprentissage sera évalué pour chaque hypothèse de concept:

$$LA_i := \begin{cases} \frac{CP_i}{SP_i} & \text{si } FPi = 0 \\ \frac{CP_i}{FP_i + DP_i} & \text{sin on,} \end{cases}$$

où SP_i spécifie la longueur du chemin le plus court (nombre de nœuds traversés) depuis le nœud TOP du concept de la hiérarchie au concept spécifique subsumant l'instance de l'hypothèse i. CP_i spécifie la longueur du chemin du TOP au concept prédit. FP_i spécifie la longueur du chemin depuis le TOP au concept prédit et DP_i spécifie le nœud entre le concept prédit (faux) et le concept commun le plus spécifique (sur le chemin entre le TOP et le nœud prédit faux). Grâce aux relations déduites des concepts extraits des textes et des concepts qui en découlent, on parcourt la hiérarchie pour explorer les branches de concepts les plus favorables (score élevé de LA= moyenne des LA_i) pour arriver au subsumant. Pour reprendre l'exemple pour Itoh-Ci-8 on part d'*ObjetPhysique* pour arriver à *Produit* (grâce à la relation a_pour_taille) et ensuite a *informatique* (grâce à la relation a_pour-fonctionnement) pour arriver à *imprimante* et finalement à *LaserPrint* qui est le dernier concept à être validé; il s'agit donc du subsumant.

Le projet *SEEK* permet de détecter des relations sémantiques entre termes identifiés dans un texte comme renvoyant vers un concept. Il permet de complexifier un réseau terminologique en donnant des étiquettes sémantiques aux relations entre concepts. Une base 1500 marqueurs indiciels permet de trouver les relations sémantiques. Ce projet se fonde sur le modèle de Grammaire Applicative et Cognitive et la méthode d'exploration contextuelle. La méthodologie d'exploration contextuelle consiste à rechercher dans les textes des indices pertinents qui jouent le rôle de déclencheur à partir desquels on effectue une exploration contextuelle en recherchant dans le contexte du déclencheur des indices complémentaires. Lorsqu'un déclencheur est identifié dans un texte avec des indices linguistiques complémentaires, il est possible de prendre certaines décisions comme caractériser une relation sémantique entre deux termes qui expriment deux concepts.

4.3. Outils de filtrage/routage

Un processus de diffusion sélective ou ciblée d'information (DSI) est capable de comparer dynamiquement de nouveaux documents reçus dans le système d'information par rapport aux centres d'intérêt d'utilisateur et transmet le document aux utilisateurs dont le centre d'intérêt correspond avec le contenu du document. Les techniques " push " sont des applications dans lesquelles l'émetteur (le serveur) appelle le terminal et lui expédie des informations, se développent rapidement. Il en existe une grande variété. Le courrier électronique déjà très répandu est une forme de technique " push ". C'est le cas *d'APERTO LIBRO* développé par INFORAMA qui correspond à une diversité d'usages. Il se caractérise comme suit: envoi en temps réel de toutes les informations nouvelles correspondant à un profil déterminé ; réception de données pertinentes en mode " push " ; outil d'édition et de création de thesaurus. Des chercheurs de l'Ecole Nationale Supérieure des Télécommunications ont développé un outil de routage des messages, *MESSIE*, pour le compte du ministère de l'intérieur. Très peu de recherches ont porté sur l'étude de la diffusion de messages électroniques. La plupart des études portent sur une adaptation des techniques liées aux documents pour les messages électroniques.

[Pincemin, 1999] détecte des unités linguistiques simples, que l'on peut assimiler à des termes. Ces termes sont ensuite classés par une méthode des nuées dynamiques donnant lieu à

des "communautés" (ou isotopies) de façon à générer des profils de routage de documents. Sa méthodologie s'inscrit dans un processus de routage du logiciel *DECID* (diffusion électronique ciblée de documents) dont le logiciel *ADOC* (association de documents) participe à la création de profils. Elle apporte 2 améliorations à la méthode des nuées dynamiques :

- la détermination automatique du cardinal de la partition. Le principal reproche attribué à la méthode des nuées dynamiques ou ses variantes, était la détermination a priori du nombre de classes à obtenir. L'hypothèse est d'avoir des groupes dont les éléments sont séparés par une distance nulle à l'intérieur d'un groupe et par une distance maximale pour une paire d'élément pris dans deux groupes différents. Le diamètre H de l'ensemble est alors le produit du nombre de paires de groupes k(k-1) par le nombre d'éléments dans un groupe $\left(\frac{n}{k}\right)^2$ par la distance maximale d'où $k = \frac{n^2 M}{n^2 M - H}$. L'hypothèse se base sur une heuristique de répartition uniformément répartie des distances.

- le recouvrement des classes finales, un terme peut appartenir à plusieurs classes et certains termes peuvent être considérés inclassables. Un terme peut aussi être considéré comme inclassable. Une distance maximale (seuil) est fixée pour décider de l'appariement d'un objet avec une partie de la partition en cours de traitement. Si la distance entre l' objet et la partie est supérieure au seuil, l'objet est placé dans une classe Z d'objets inclassables. Ces objets inclassables peuvent quand même être promus au rang de noyau d'une nouvelle classe. L'algorithme de Diday considère des formes fortes pour éviter le biais de l'ordre, et de la convergence vers la solution. Un algorithme classique partant de "k noyaux" formant les "k futures classes" opère n tirages ou n itérations. Une forme forte est un groupe d'objets qui suivent la même trajectoire, c'est–à-dire un comportement similaire. [Pincemin, 1999] propose de lier des objets qui participeraient à plusieurs trajectoires voisines. L'algorithme est similaire à un appariement hiérarchique par un saut minimum. Les formes fortes sont pré-déterminées par l'algorithme classique de Diday formant une partition non recouvrante. Un degré de variation est défini (entre 0 et 9). Le degré de variation définit l'écart maximum entre la trajectoire d'une forme forte et une autre. L'algorithme considère un écart variant de 1 à degré de variation. Toutes les classes dont une forme forte s'écarte de moins de l'écart sont regroupées si une des formes d'une classe ne s'écarte pas de plus du degré de variation (écart maximum autorisé entre 2 trajectoires). Ceci est répété pour toutes les classes et en incrémentant l'écart jusqu'au degré de variation;

- La confrontation d'un document avec un profil ("communautés") se fait grâce au nombre de mots en commun (hors mots grammaticaux) qui sont classés par ordre de fréquence.

4.4. Outils de bibliométrie

L'analyse bibliométrique fait appel à de nombreuses techniques pour traiter l'information brute. Ce sont notamment les techniques d'élaboration (tris de toutes sortes, élimination du bruit ou des doublons, reformatage des données), et les techniques d'analyse de données (méthodes classificatoires, analyse en composantes principales, analyse factorielle des correspondances, analyse relationnelle des données, analyse en composantes locales angulaires, analyse discriminante, analyse sphérique).

L'ingénierie documentaire progresse aussi rapidement. Elle porte sur le traitement de textes, la structuration de documents, la mise au point de systèmes relationnels de gestion de bases de données, la sélection de l'information spécialisée dans les bases documentaires (indexation

automatique de bases de données recourant à l'analyse linguistique plein texte, outils de veille pour l'industrie, confection de bases de connaissances sectorielles, construction automatique de thésaurus, routage et diffusion de l'information spécialisée vers des utilisateurs ciblés). Elle fait appel à la technologie des agents intelligents et aux techniques d'infométrie, de bibliométrie ou de scientométrie.

Parmi les centres de recherche qui travaillent à un titre ou à un autre dans ce domaine, on peut mentionner:
- le centre de recherche rétrospective de Marseille (CRRM) sous la direction du professeur H. Dou qui a conçu *Dataview*, logiciel de transformation de l'information textuelle (structurée en champs en données numériques ;
- le centre européen de mathématiques appliquées (CEMAP) d'IBM qui a développé le logiciel *Tewat* qui permet d'automatiser l'analyse d'information extraite de bases de données internationales ;
- le centre de sociologie et d'innovation du CNRS qui, en collaboration avec l'INIST, a développé un outil d'analyse des mots associés, *Leximappe* ;
- le CEDOCAR (Centre de Documentation de l'armement au ministère de la Défense), qui a développé, en liaison avec le Centre de Recherche en informatique de Nancy ou encore l'Institut de Recherche en Informatique de Toulouse, la plate-forme ATLAS (outil *Tétralogie*).

TechOptimizer est un logiciel d'analyse de brevet original. Commercialisé par une société américaine il s'appuie sur une étude empirique de 40 années entreprise par l'ingénieur soviétique H.Altschuller entre 1950 et 1990. L'idée est simple mais ambitieuse : l'innovation s'inspire de ce qui a été déjà mis au point. L.Altschuller a donc essayé de rechercher les principes fondamentaux, scientifiques et techniques, qui sont à la base de la création d'un brevet: c'est la méthode TRIZ. Des équipes de plusieurs centaines d'analystes se sont ensuite relayées pour intégrer de nouveaux brevets dans leur base et dégager les lois et propriétés de ces brevets. Le résutat est une matrice de relations fondamentales. Ces relations sont ensuite analysées en fonction d'une requête pour associer des brevets existants pouvant apporter des éléments de solution à un nouveau problème.

4.5. Outils de classification de textes

La classification ou catégorisation de documents consiste à apprendre, à partir d'exemples caractérisant des classes thématiques, un ensemble de descripteurs discriminants pour permettre de ranger un document donné dans la (ou les) classe(s) thématique(s) correspondant à son contenu [Brooks, 1995]. Les descripteurs sont intégrés dans une fonction qui servira à établir un score de pertinence pour affecter le document à classer dans la catégorie ayant obtenue le meilleur score. Deux grandes tendances font office de référence : les systèmes issus de l'analyse discriminante c'est à dire établissent des règles de régression avec les descripteurs et les systèmes probabilistes ou classifieurs bayésiens naïfs ($P(C)\prod_{i=1}^{n} P(\omega_i|C)$ sert à évaluer le score, P(C) est la probabilité associée à la catégorie C, $P(\omega|C)$ est la probabilité d'avoir le descripteur w sachant la catégorie C). Un classifieur bayésien semble donner de bons résultats. Pour une terminologie des documents normalisée manuellement, le classifieur bayésien présente une efficacité supérieure à 95% même pour des documents très courts (dépêches). Pour des documents non normalisés manuellement l'efficacité est variable selon la nature thématique du document, la langue traitée et la qualité rédactionnelle. Cette efficacité fluctue entre 70% et 95%.

TREVI projet européen, classification de documents (dépêche de Reuters). Les techniques d'analyse grossière (angl. Shallow parsing) et vectorielle probabiliste (angl. Probabilistic Vector-Model) sont utilisées pour affecter une dépêche à une ou plusieurs catégories parmi un total de 20 catégories.

CLASTER (Institut d'intelligence artificielle de Pereslavl-Zalessky) projet russe de classification de documents fonctionne par apprentissage de vecteur typique d'une catégorie.

FACILE (Université de Venise et Société Quinary) permet d'extraire des structures attributs/valeurs (angl. Frames) et de faire une classification de textes, catégorisations rapide et exacte de l'information à l'aide de l'ingénierie linguistique. Le système s'appuie sur deux stratégies complémentaires:
1- une analyse de surface à base de reconnaissance de formes, qui doit accomplir la catégorisation et en même temps livrer des informations de base sur les textes et qui sera réalisée pour des textes anglais, allemands, italiens, et espagnols
2- une analyse approfondie à base de linguistique informatique qui devrait permettre (moyennant un coût plus important) une meilleure interprétation des textes, et qui sera réalisée pour les langues anglaise et italienne.

Résumé du chapitre

Ce chapitre présente les principales techniques de classification automatique non supervisée, utilisées pour classer des objets en groupes homogènes. On y trouve : les méthodes: des plus proches voisins, factorielles, hiérarchiques ascendantes, hiérarchiques descendantes, d'extraction de graphes, de sériation, neuronales et symboliques. Ces méthodes issues de l'Analyse des Données travaillent à partir d'une matrice individus/variables. Les objets sont associés grâce à une distance ou similarité et grâce à un critère d'affectation. Les modèles de représentation des objets se réfèrent quasiment toujours à un modèle vectoriel à la fois pratique et efficace. Les méthodes sont génériques et ne s'inspirent pas du type des données à classer même si elles ont été appliquées, pour certaines dès leur origine, à des données textuelles.

Cette thèse se veut pivot entre la classification automatique et le traitement du langage naturel, les principaux courants du TAL y ont donc leur place. Nous discutons de la rivalité entre l'école distributionnaliste de Z.Harris et l'école générativiste de N.Chomsky. Cette dernière prétend pouvoir contrôler l'expressivité du langage naturel à partir de règles de transformations définies a priori. L'approche harrissienne propose une analyse plus empirique par rapport aux aspects répétitif et non aléatoire des suites de constituants phrastiques. Sans rentrer dans le détail des théories de la sémantique, en général très informelles ou alors s'appuyant sur le formalisme logique, nous exposons les traits définitionnels d'un champ sémantique lexical. On peut effectivement penser que la notion de champ sémantique draine l'équivalence du point de vue linguistique des caractéristiques recherchées d'un résultat de classification automatique. Finalement nous présentons les méthodes d'extraction automatique de syntagmes nominaux que nous assimilons à des termes représentatifs de la lexicalisation des idées développées dans un texte.

Nous terminons le chapitre par un aperçu des applications finales possibles utilisant une classification automatique de termes d'un domaine. Nous ne pouvons passer à côté de la recherche documentaire, champ d'investigation du TAL depuis l'origine de l'intelligence artificielle. Ce champ constitue encore un terrain d'expérimentation pour les nouveaux formats de documents électroniques. Nous présentons ensuite les caractéristiques de la fouille de texte, domaine nouveau mais s'appuyant sur des techniques éprouvées: règles inductives, classifications, arbres de décision. Enfin le filtrage et la classification de documents, cible de notre application finale, viennent clore ce chapitre.

CHAPITRE 3

ARCHITECTURE DU CLASSIFIEUR GALEX

Dans ce chapitre nous exposons la méthode de classification en 2 phases: la première phase consiste à extraire les unités qui seront classées par la suite, c'est la phase de collecte des données ou collecte des individus et des variables. Il est clair qu'un mauvais échantillon de données induira une interprétation biaisée quel que soit le traitement réalisé sur ces données. La deuxième phase consiste à classer les unités linguistiques extraites qui sont données en fichier d'entrée au classifieur.

Nous présentons d'abord quelles caractéristiques à considérer dans un texte avec les différents type de cooccurrences que l'on peut rencontrer dans un document.

Une étude empirique sur un corpus médical et un choix méthodologique conduisent à privilégier une approche de classification par modèle de graphe dont les relations sont des attributs choisis préalablement. Le modèle de graphe est centré autour d'un terme pôle auquel on agrège d'autres termes associés dans une configuration de clique d'ordre 3. Dans un deuxième temps on agrège des cliques d'ordre 3 ayant le même terme pôle pour former des cliques d'ordre 4. Les cliques d'ordre 4 sont fusionnées modulo 2 termes pivots en commun pour former une classe. Enfin les termes, ayant la même distribution d'hapax que le terme pôle, sont agrégés à la classe correspondante. Les classes qui n'ont pas plus de n termes en commun sont finalement retenues.

Nous exposons finalement l'architecture de notre classifieur qui se décompose en 2 sous-module distincts:
1- le module d''extaction et de réduction des termes, qui va permettre d'établir les fichiers d'entrée du module de classification
2- le module de classification qui va implémenter la méthodologie décrite précédemment.
Nous détaillons l'algorithme tout au long du traitement. En illustrant chaque partie par des exemples issus du traitement réel.

Les classes ainsi formées relèvent d'une classification monothétique due au rôle du terme pôle. L'agrégation autour de ces termes pôles inscrit le modèle de classification dans la famille des k voisins au même titre que le modèle des réseaux de Kohonen ou des nuées dynamiques. Les classes obtenues possèdent plusieurs propriétés. Les classes sont recouvrantes, en effet un terme peut appartenir à plusieurs classes. Les classes sont hiérarchiques parce qu'un terme pôle est à la base de plusieurs classes (le prochain chapitre abordera un type de généralisation supplémentaire). Une classe peut être interprétée comme un concept. L'intension du concept est décrite par les relations symbolisées par les attributs et le contexte pragmatique d'une partie des termes de la classe. L'extension du concept est décrite par les termes de la classe.

Ces classes, symbolisant une suite de concepts et émanant d'un domaine, seront utilisés ultérieurement dans une fonctionnalité d'analyse de l'information grâce à l'identification d'une partie de ces concepts dans des documents inconnus.

1. Choix du modèle de classification

1.1 Ingrédients d'une analyse automatique de texte

Dans le langage nous disposons d'un certain nombre d'ingrédients qui sont présents quel que soit le texte en entrée. On considère une séquence de mots comme variable d'analyse.
Soit X l'ensemble de ces variables définies par:

$$X = \{x_1, x_2, \ldots, x_n\}.$$

Une variable d'analyse possède 3 paramètres d'observations qui sont les observables du système:

- *la fréquence dans le texte :*

$$f_i = \sum_j \delta_{ji} \quad \text{où } \delta \text{ est le symbole de Kronecker } \begin{cases} \delta_{ij} = 0 \text{ si } x_j \neq x_i \\ \delta_{ij} = 1 \text{ si } x_j = x_i \end{cases},$$

i est l'indice sur toutes les occurrences du texte.

- *les différentes positions de x_i dans le texte :*

$$P_i = \{p_1, p_2, \ldots, p_k\} .$$

- *les cooccurrents* pour chaque position de x_I :

$$C_i = \left\{ x_j / R = (r_{ij}) \text{ et } r_{ij} \neq 0, \text{ avec } r_{ij} \neq 0 \text{ si } \exists k,l \quad 0 < (p_{il} - w) < p_{jk} < (p_{il} + w) \right\}$$

, où R est la matrice de cooccurrence admettant une relation pour chaque couple (x_i, x_j), avec w étant une fenêtre de mots .

La figure 3.1 nous montre les différents types de coccurrence que l'on peut rencontrer dans un document. Les cases grisées correspondent aux types que le classifieur Galex peut traiter.

Bien que les cooccurences traitées par Galex soient non syntaxiques il est envisageable de considérer les cooccurrences à l'intérieur d'un syntagmes. Le traitement peut aussi s'appliquer à une classification des têtes de syntagme en fonction des expansions les têtes forment le fichier des variables et les expansions le fichier des attributs.
La méthode d'extraction des variables et les heuristiques de classification devront donc uniquement tenir compte de ces facteurs.

1.2 Choix général de l'approche de classification

Compte tenu des différentes approches existantes exposées au chapitre 2, de notre choix de pouvoir interpréter les classes en fonction de leurs attributs et de pouvoir revenir aux données sources contribuant à leur construction nous décidons de nous attacher plus particulièrement aux approches de graphes. Ces approches semblent mieux adaptées à l'interprétation avec retour aux données. Les graphes sont effectivement des associations liant des nœuds qui sont pour nous des associations de termes issus du texte. On appelle GR=(X,R) le graphe G pour la relation R. L'ensemble des arêtes définies pour ce graphe est:
$E(GR)=\{(x_i,x_j)|r_{ij}=R(x_i, x_j),\ x_i, x_j \in X\}$ où
(a) l'ensemble des sommets (vertex) est constitué par les éléments $x_i \in X$ tels que $r_{ii} \neq 0$
(b) l'ensemble des arêtes est défini par la relation binaire R rassemblant des paires non ordonnées (x_i, x_j) telles que $x_i, x_j \in X$ et $r_{ij} \neq 0$.
La fusion de relations pour former une classe devrait donc permettre de retrouver les relations du texte qui ont permis de créer cette classe. Le problème qui se pose maintenant est de choisir un modèle de graphe. Pour cela nous allons étudier le corpus à la main pour extraire les relations.

cooccurrence syntaxique		**cooccurrence non syntaxique**		**unité textuelle**
cooccurence sujet/verbe/objet exemple: "Le noyau d'uranium a été déformé dans un champ de 1Gev." "Le noyau de Césium se déforme avec une symétrie SO2." "noyau de Césium" et "noyau d'uranium" sont liés comme sujets de "déformer".	*cooccurence syntagmatique* *exemple:* *"ligne haute tension "* *"pylône haute tension".* *"pylône" et "ligne" sont liés par "haute tension".*	*cooccurence* avec *attribut* *exemple:* *"Le satellite s'appuie sur la gavité pour changer sa trajectoire."* *"satellite" et "gravité " sont liés par l'attribut "changer".*	*cooccurence sans attribut* *exemple:* *"Le neutron et le proton forment le noyau d'un atome".* *"neutron" et "proton" sont liés par voisinage.*	**phrase**
./	**./**	*cooccurence* avec *attribut* *exemple:* *"Le satellite s'appuie sur la gavité pour changer sa trajectoire. Cela lui permet de s'orienter vers une autre planète".* *"trajectoire" et "planète" sont liés par l'attribut "orienter" .*	*cooccurence sans attribut* *exemple:* *"Le neutron et le proton forment le noyau d'un atome. Leur nombre varie et modifie les propriétés de l'atome".* *"noyau" et "propriétés de l'atome" sont liés par voisinage.*	**paragraphe**
./	**./**	cooccurence avec attribut exemple: auteur: M. Martin "Le satellite s'appuie sur la gavité pour changer sa trajectoire." "Martin" et "satellite" sont liés par l'attribut "auteur" .	cooccurence sans attribut exemple: "Le neutron et le proton forment le noyau d'un atome." ... "Le numéro atomique varie et modifie les propriétés de l'atome". "noyau" et "propriétés de l'atome" sont liés par voisinage.	**document**

Figure 3.1 Types de cooccurrences.

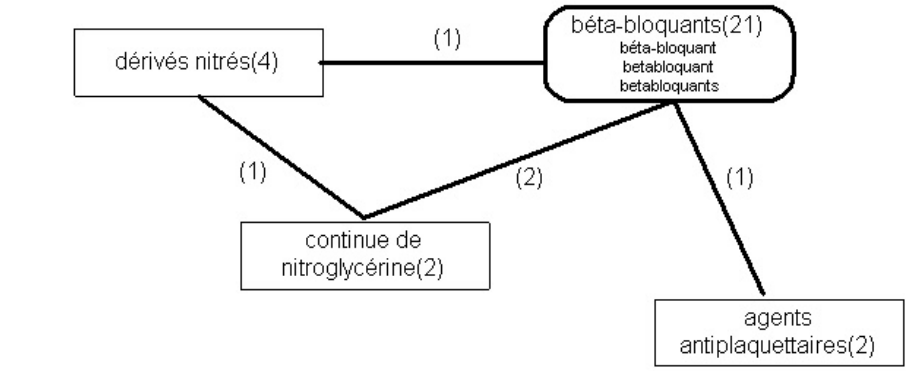
1.3 Etude empirique d'un corpus

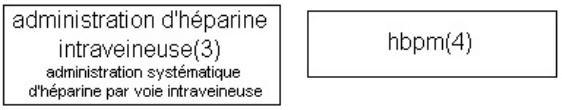
Le corpus est un corpus de bilans médicaux portant sur la coronarographie et les maladies du cœur. Il a 30000 mots simples dont 2900 sont distincts. Un extracteur de termes par segments répétés nous donne 350 termes ou groupes nominaux. Certains sont simples et la plupart sont des termes composés. Les fréquences vont de 260 pour le terme "coronarographie" à 2 pour une grande majorité des termes de l'échantillon. Dans un premier temps nous établissons une classification conceptuelle de ces termes pour les classer en champs sémantiques homogènes (annexe 7). Des dictionnaires et un expert du domaine assistent la construction de cette hiérarchie de référence.

1.4 Description des classes obtenues

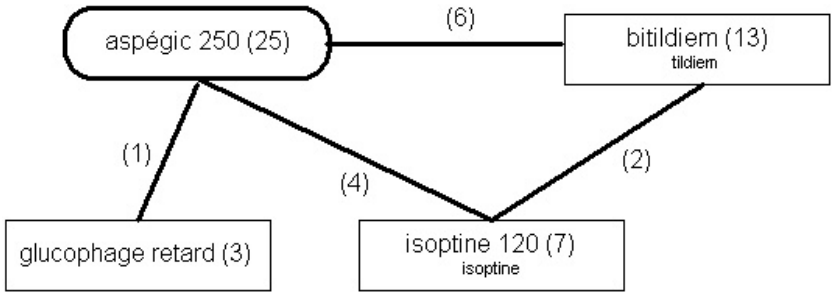
A partir de la hiérarchie de concepts établie à la main nous nous intéressons plus particulièrement à 6 classes de termes sémantiquement bien délimités. Dans un premier temps nous cherchons toutes les cooccurrences directes d'un terme avec un autre terme de la même classe et ce pour une fenêtre de termes d'environ 20 mots à droite et à gauche. On aboutit aux résultats de la figure 3.2. Cette figure nous montre que les associations sont en faible nombre. Il est difficile d'extraire un schéma de graphe à partir de ces 6 graphes. Par contre on remarque qu'un terme génère systématiquement un plus grand nombre d'associations avec les autres membres de la classe. On se trouve d'ores et déjà confronté à un des problèmes de langue qui est la morphogenèse ou la création de formes variantes. Une expression peut se décliner en plusieurs formulations possibles. Certaines variantes sont bien connues et maîtrisables comme le genre et le nombre. D'autres peuvent poser de sérieux problèmes de traitement automatique comme la nominalisation (exemples: "réalisation d'une double angioplastie" / "réaliser une double angioplastie"; "visualisable au niveau de l'interventriculaire" / "visualiser au niveau de l'interventriculaire"). D'autres encore font appel à des paradigmes sémantiques pour qualifier l'état d'une expression (exemple: "altération de la fonction ventriculaire" / "mauvaise fonction ventriculaire"). Nous ne considérerons pas les variations paradigmatiques et les variations par nominalisation qui demandent des traitements sémantiques poussés et peu robustes. Après l'observation de ces premiers graphes nous décidons de traiter des cooccurrences indirectes c'est-à-dire de se baser sur la notion de prédicat et plus spécialement de prédicat verbal. Par exemple dans les deux exemples qui suivent "coronarographie montre…" et "ventriculographie montre…" les deux items "ventriculographie" et "coronarographie" seront liés par cooccurrence indirecte. Cette relation dite indirecte peut bien sûr devenir directe si les items se trouvent dans la même phrase. Nous ne nous attachons pas à respecter des contraintes de dépendance syntaxique de type sujet/objet/propositionnelle…. Le comptage de cooccurrence se base toujours sur la notion de fenêtre d'observation. L'ordre de la cooccurrence entre deux termes n'est pas pris en compte. Les graphes de la figure 3.3 nous présentent les résultats avec les verbes associés aux relations. L'hypothèse d'un terme plus associatif avec une certaine fréquence est de nouveau validée. Nous observons de plus que le nombre d'associations est supérieur aux graphes précédemment analysés. Les verbes recueillis ne font pas systématiquement intervenir le terme comme argument direct, indirect ou dans une phrase prépositionnelle, mais sont des verbes présents dans une fenêtre contextuelle de plus ou moins 15 mots autour du terme. Les verbes ayant le terme comme argument sont soulignés sur la figure 3.3. On constate que les verbes ayant un rôle avec le terme sont tous recueillis grâce à la fenêtre contextuelle de repérage mais aussi que ces verbes ne sont pas à eux seuls porteurs de la majorité des relations.

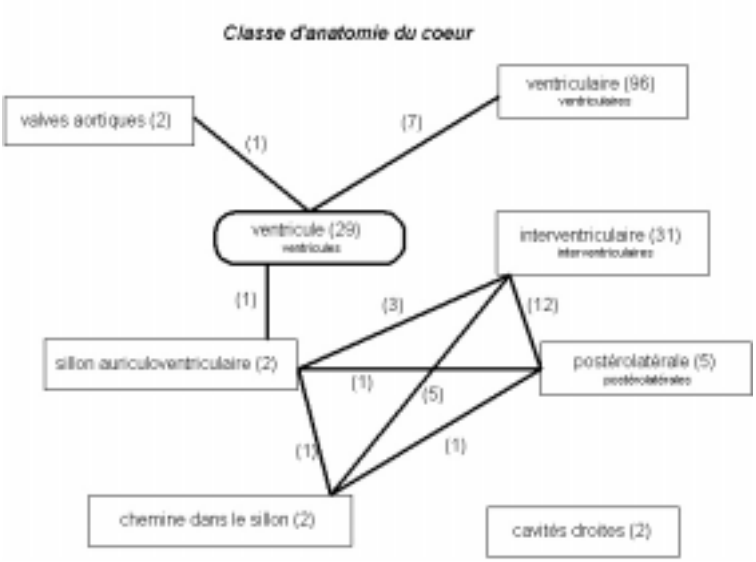
Classe des agents traitants non commerciaux





Classe des agents traitants commerciaux





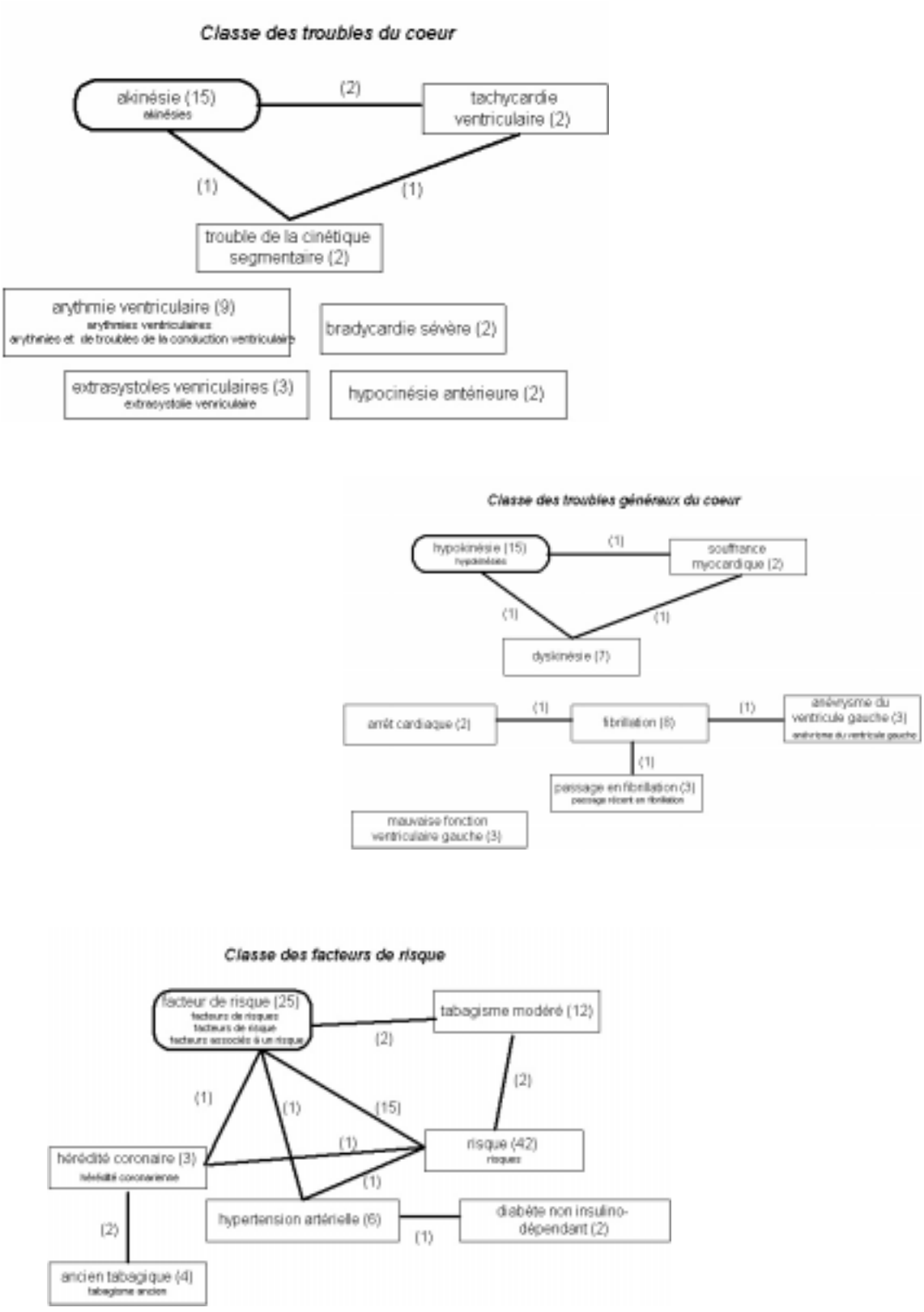
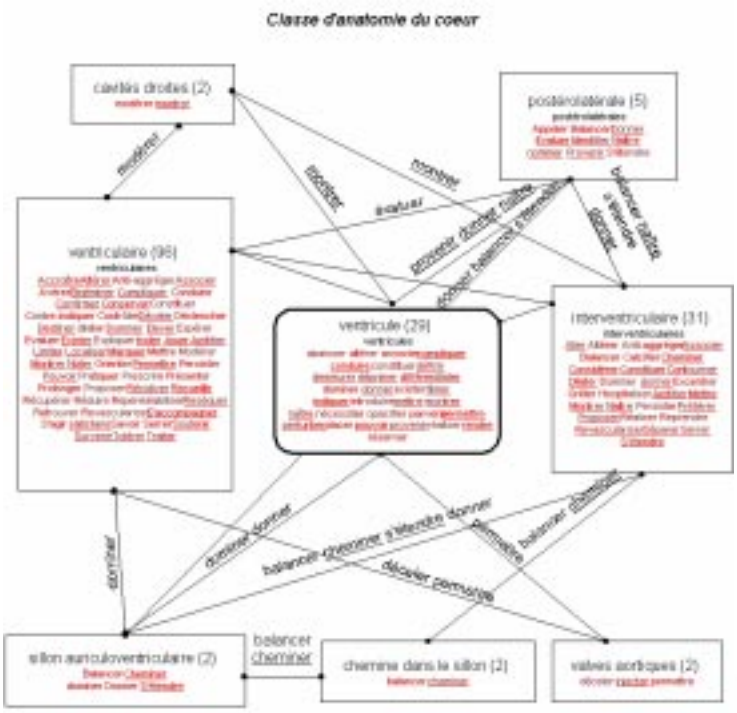
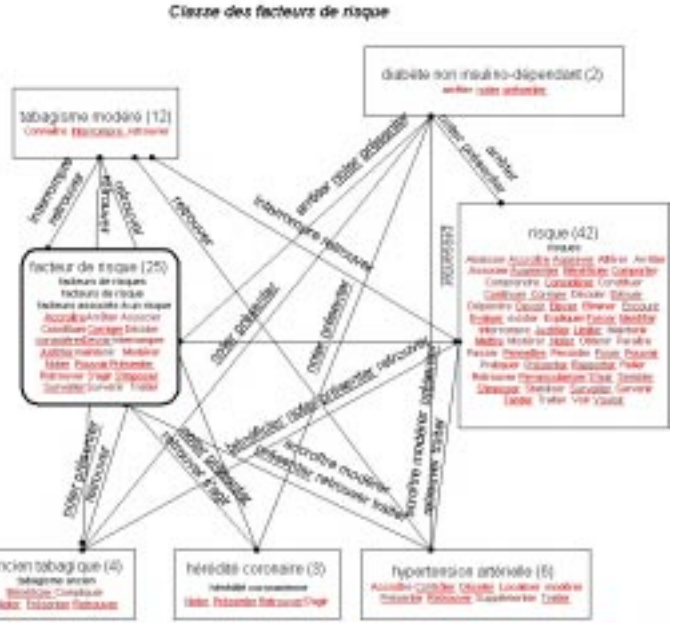
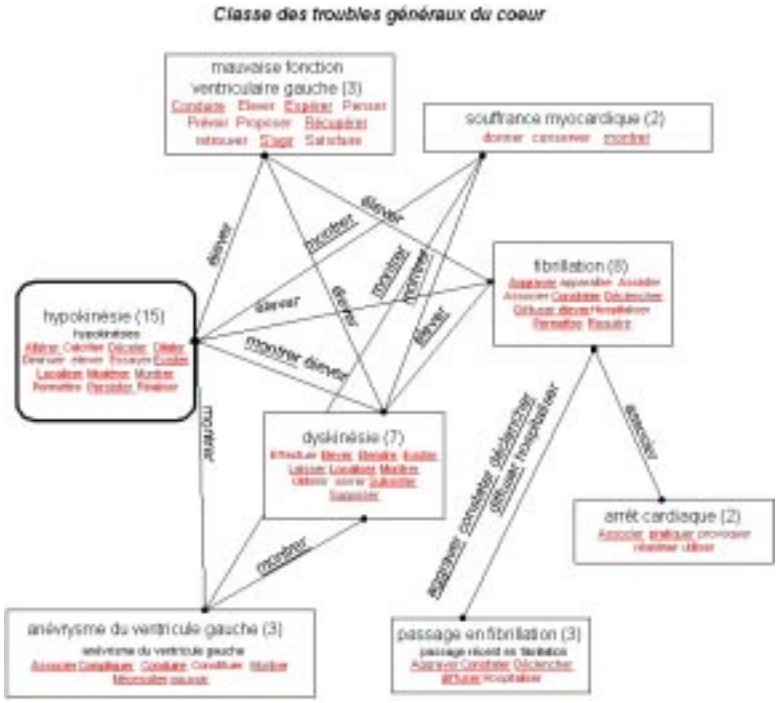


Figure 3.2 Graphes de cooccurrences terme-terme







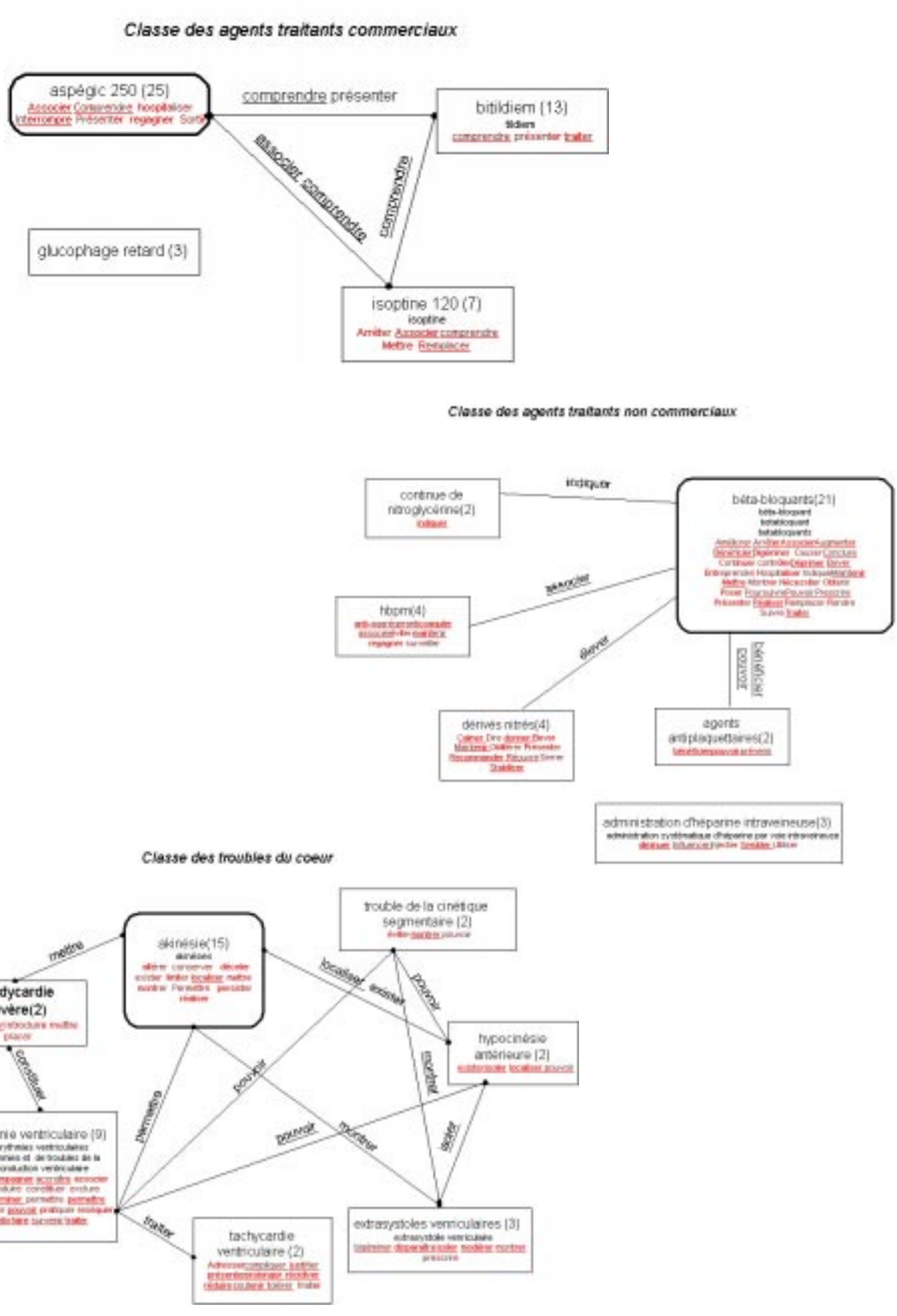


Figure 3.3 Graphes de cooccurrences terme-verbe

Lors de l'étude empirique et manuelles des classes de la figure 3.2 nous avons remarqué qu'un terme central apparaît constamment. Nous qualifions ce terme de **terme pôle**. Il se comporte comme un terme central dans la structure de graphe de la classe et possède une fréquence médium par rapport aux termes les plus fréquents. Cet intervalle correspond dans le cas du corpus médical à 30% et 10% de la fréquence maximale soit entre 70 et 20 sachant que la fréquence maximale (230) est celle du terme "coronarographie".
D'autre part nous avons mené une étude d'extraction automatique des relations et notamment des **cliques d'ordre 3** (3 termes tous associés). La contrainte d'extraction était d'avoir au moins un terme pôle inclus dans le graphe sachant que l'extraction devait être exhaustive. Nous avons constaté une quantité importante de ces graphes bien que l'ensemble des termes pôle soit relativement réduit (environ 50 termes pôles). En moyenne on observe 10000 à 100000 cliques d'ordre 3 pour des corpus d'environ 50000 mots. Cette observation nous a amené inévitablement à contraindre les associations sous forme de cliques d'ordre 4. L'extraction des **cliques d'ordre 4** est encore exhaustive et conduit aussi à un nombre important de graphe centré autour d'un terme pôle. Ce nombre, très variable, est compris en général entre 10000 et 200000 toujours pour un corpus d'environ 50000 mots. Cette troisième observation nous a amené à faire intervenir la notion de **terme pivot** pour contninuer le processus d'agrégation. Un terme pivot est tel qu'il assure l'affectation d'un nouveau terme dans une classe si ce terme est à la fois lié à ce terme pivot et au terme pôle. Le nombre de termes pivots nécessaire à la création de classes est fixé à 2. Considérer un seul terme pivot reviendrait à créer une classe en étoile autour du terme pivot. Il est envisageable d'avoir plus de termes pivots; on modifierait ainsi le motif de graphe. Par conséquent dans notre processus 2 termes pivots permettent d'associer d'autres termes à une clique d'ordre 4. Le résultat est une classe temporaire qui n'est pas une clique. Les classes résultant du processus ne seront donc pas des grpahes maximaux complets et n'induisent pas de problème d'extraction de type np-complet. Le détail de l'algorithme est expliqué au §4.

1.5 Méthodologie générale

Pour plusieurs raisons nous décidons d'orienter notre approche de classification vers une méthode d'extraction de graphes:
a) Les graphes de la figure 3.3 nous montrent la potentialité d'extraire des graphes avec suffisamment d'associations pour inférer un graphe à partir d'un modèle (terme central ou pôle, attributs verbaux,…).
b) Les graphes sont le reflet le plus direct des associations qui sont à l'origine de la génération des textes. Ces associations étant contraintes sémantiquement.
c) Les graphes permettent compte tenu de leur création de mieux remonter aux données sources qui ont permis de créer les associations du graphe (ce qui n'est pas le cas pour les autres types de méthodes basées sur des coefficients de similarité ou la recherche d'axes propres).
d) Les graphes ne dépendent pas de seuils (de similarité) numériques fixés, souvent arbitraires et indépendants de la nature des données.
e) Enfin les graphes peuvent assurer des propriétés de relations de terme intéressantes: recouvrement des classes, hiérarchisation, appartenance non systématique d'un terme à une classe.

Compte tenu de ces remarques, notre approche de classification va comporter 2 volets. Ces volets sont en fait distincts mais se chevauchent. Le premier volet est la sélection des variables à classifier et le deuxième est le résultat de la classification en groupes de termes non exclusifs décrits par des attributs. Dans cette thèse nous développons l'extraction de

concepts à partir d'un modèle de graphe se basant sur la notion de terme pôle et de termes pivots:

étape 1 : identification d'échantillons de termes,
étape 2 : inférence de concepts par application du motif de graphe (graphe d'induction).

La structure générale des groupes de termes a pour objectif de refléter la sémantique ou les champs sémantiques issus du texte d'origine.

Pour aboutir à une bonne homogénéité sémantique ou interprétabilité des classes on doit sélectionner les meilleures variables interprétables. Les présence des termes dans des classes après classification est conditionnée par les occurrences et les relations. Il en résulte une sélection des associations bénéfiques à la constitution des classes d'où sélection implicite des termes intéressants. On a donc une sélection explicite des variables en amont par une extraction de termes et une sélection implicite en aval après le processus de classification. La classification opère donc une deuxième vague de sélection dans la mesure où des termes inclassables sont considérés.

2. Architecture du classifieur GALEX

2.1 Architecture générale

La figure 3.4, ci-dessous, présente l'interaction entre les différents sous-modules de Galex (Graph Analyzer for LEXicometry) qui sont en fait développés en terme d'objets mais que l'on qualifiera de modules [Turenne, 1999].
Un corpus de textes au format ASCII est donné en entrée (1) du module d'extraction de termes (haut de la figure 3.4). Ce corpus est traité par un extracteur[5] de groupes nominaux lesquels ne sont ni triés par fréquence ni lemmatisés (2). Un filtre trie ces groupes nominaux par fréquence (3) et sont ensuite lemmatisés (5). Le corpus est parallèlement traité par le lemmatiseur (4) qu'un générateur de position (7) transforme en fichier de positions (8). Le lemmatiseur traite le fichier de groupes nominaux et fournit 2 fichiers: le fichier de termes (groupes nominaux complétés par des mots simple discriminants) (5) et le fichier d'attributs (verbes, …) (6). Le fichier des positions, de termes et d'attributs permettent de construire une matrice (9) qui rassemble les comptages d'associations entre paires de termes. On extrait une liste de termes pôle qui seront la cheville des classes à rechercher (11). A partir des termes pôle et de la matrice on recherche tous les graphes triangulaires contenant un terme pôle (cliques d'ordre 3) (12). A partir des cliques d'ordre 3 on crée des cliques d'ordre 4 qui sont regroupées grâce à deux termes pivots pour former des classes (13). On évalue la distribution de termes hapax (terme de fréquence un) d'un terme lié à la classe par association pour l'agréger à cette classe. L'agrégation a lieu si la distribution des valeurs de cooccurrence d'un terme et celle d'un terme pôle d'une classe sont proportionnelles (la distribution pour un terme donné est la suite de ses coefficients de cooccurrence). Cette dernière étape conduit au résultat final produisant 2 fichiers, l'un répertoriant les relations binaires de chaque classe avec leurs attributs respectifs (14) et un fichier consignant les éléments de chaque classe (15).

[5] L'extracteur nous a été gracieusement prété par Inxigth (une filiale de Xérox), cet extracteur ne lemmatisant pas les syntagmes il nous a fallu développer une opération de lemmatisation a posteriori.

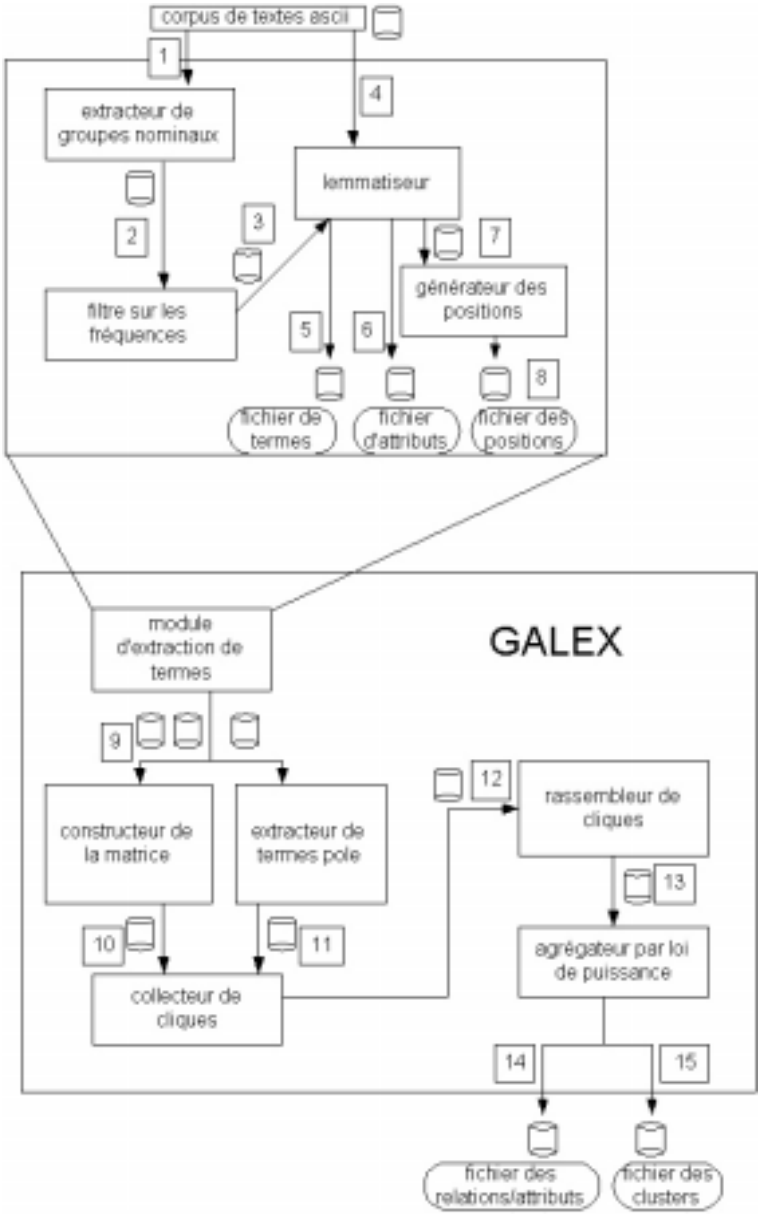


Figure 3.4 architecture générale du processus de classification

2.2 Module d'extraction de termes

Ce module a pour but d'extraire les syntagmes, du corps et du sujet d'un message envoyé par l'interface, avec leur position. Un terme composé ou multiterme sera considéré comme une forme variante d'une autre terme et équivalent à ce dernier si les mots de début et de fin du multiterme sont les mêmes quel que soit l'ordre ; leurs positions seront rassemblées pour un seul terme. (exemple: "méthode de classification" et "méthode efficace de classification" seront considérés comme deux expressions identiques). On suppose que les monotermes (i.e. terme simple comme "chat") et les multitermes (i.e. terme composé comme "chat gris"), seront normalisés au sens d'unités lexicales (exemple : 'lisible' et 'lisibilité' sont normalisés en 'lire'). La base de donnée termes/positions renvoyée est propre à un seul message.

2.3 Module de classification

Ce module assure la classification des syntagmes. Il s'inspire d'une méthode graphe-statistique pour créer des groupes de termes. La construction d'une matrice de dénombrement utilise la base de termes/positions provenant du module linguistique. Deux possibilités sont offertes pour calculer la matrice. On croise le fichier de termes avec lui-même pour les lignes et pour les colonnes. Ou bien on croise deux fichiers différents l'un (termes) en ligne et l'autre (attributs) en colonne. Cette dernière solution sera privilégiée pour faire ressortir des attributs aux relations entre termes au sein de leur classe. Cette matrice est donc exploitée pour produire une base de classes de termes.

2.3.1 Constructeur de la matrice

Ce module permet de créer/modifier une matrice M qui dénombre les associations entre 2 ensembles d'unités textuelles. La matrice prend une $1^{ère}$ suite d'unités en lignes et une $2^{ième}$ suite d'unités en colonne. Dans le cas où les termes en ligne et sont croisés avec les attributs en colonne, la matrice n'est pas symétrique. Une opération de transposition est effectuée sur la matrice. Le produit de la matrice par la transposée ($M_* {}^tM$) donnera une matrice symétrique par rapport aux données en ligne de la matrice M. On ne se sert que de la moitié de la matrice symétrique on la rend donc triangulaire droite.

Matrice de cooccurrence

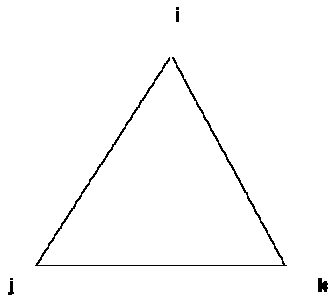
5	0	1	0	2
0	3	0	0	0
0	0	5	0	2
0	0	0	2	0
0	0	0	0	2

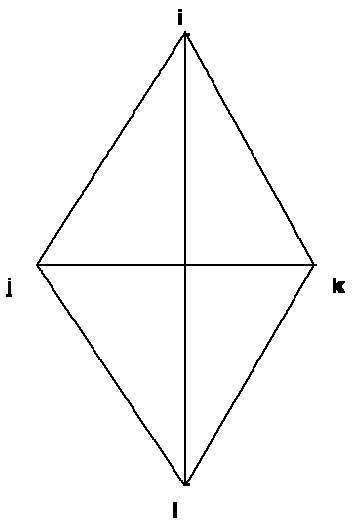
2.3.2 Extracteur de termes pôles

On se sert de la base de syntagmes pour extraire les termes pôle. L'extraction est conditionnée par un intervalle de fréquence fixé par la fréquence maximale des termes du fichier de syntagme. Le terme pôle est l'objet central dans la construction d'une classe de terme.

2.3.3 Collecteur de cliques-3

Des associations mutuelles d'ensemble de trois termes dont un terme pôle seront extraites. Ces ensembles sont appelés clique d'ordre trois, ce sont les briques de base pour classer les termes. On passe par cette étape pour des facilités d'implémentation pour passer à l'étape suivante.

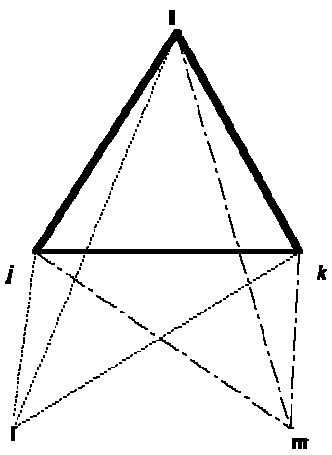




2.3.4 Rassembleur de cliques-4

Les cliques d'ordre 3 sont assemblées pour construire des cliques d'ordre 4.

La réunion de 3 cliques d'ordre 3 est effectuée tout en gardant un même terme pôle. Les cliques d'ordre 4 obtenues sont assemblées en fonction de deux critères : avoir



un terme pôle en commun et deux termes pivots. On obtient ainsi une série de classes temporaires de termes.

On trie ces classes en ne gardant que celles qui ont plus de n termes en commun avec ($n \leq 2$); Il s'agit d'un critère de discrimination pour retenir les classes qui deviendront les classes finales.

2.3.5 Agrégateur par loi de puissance

A partir des classes précédentes l'agrégateur va traiter un terme (T1) lié au terme pôle (T2) d'une classe et qui possède une distribution de liens de cooccurrence proportionnelle à celle de T2. On teste l'écart entre les deux distributions pour décider de l'appartenance de T1 avec la classe de T2. Cette heuristique n'a pas été implémentée pour des raisons de coût de calcul.

3. Extraction des termes

3.1 Définition d'un terme

Un ***terme*** est syntaxiquement une suite consécutive de mots. Sémantiquement il doit représenter une entité interprétable par rapport à un domaine (par opposition à une locution).
Soit ***t*** un *terme* composé des mots mot_1 à mot_n, $t = mot_1\ mot_2 \ldots mot_n$.
Pour tout terme de *n* mots, on ne considère que le premier et le dernier, soit mot_1 et mot_n repérés par leur position absolue en nombre de caractères dans le fichier d'entrée.
Tout terme du texte construit avec l'expression rationnelle $mot_1\ \{\ mot_i\ \}^*\ mot_n$ dont la distance entre le premier et le dernier mot est inférieure à une constante donnée, est considéré comme appartenant à la classe d'équivalence du terme *t* défini plus haut.

Soit un terme *t* d'un fichier de termes, $t = mot_1 \ldots mot_n$
$\forall$ terme *e* du texte, tel que $e = mot_1\ \{\ mot_i\ \}^*\ mot_n$ et
distance (mot_1, mot_n) < *longueur_max_terme*
alors e $\in$ classe_d'équivalence(*t*)

Exemple : *pomme de terre, assemblée, …etc*

Dans certains programmes intermédiaires les termes ne seront codés au plus que sur 64 octets par souci d'économie, s'ils dépassent cette taille ils sont négligés. Ceci afin aussi d'éviter de traiter des chaînes, d'un texte de langue non générale, qui ne correspondrait pas à un syntagme nominal.

3.2 Extracteur des groupes nominaux

3.2.1 Modèle de Markov

Dans un modèle de Markov observable, chaque état est équivalent à un état observable. L'état du modèle de Markov est une fonction de chaque état interne, tandis que l'état présent dépend seulement de l'état précédent. Dans notre cas, une étiquette grammaticale (verbe, adverbe, nom,…) représente un état. Un HMM (angl., Hidden Markov Model) aide à l'étiqueter les phrases et résoudre les ambiguïtés. Une ambiguïté est modélisable par différents chemins dans un graphe ou une suite de séquence d'états [Chanod, 1994] (annexe 5).
Cependant, pour un HMM, la sortie n'est pas la séquence d'état interne, mais une fonction probabiliste de cette séquence interne. Quelques auteurs font des symboles de sortie d'un modèle de Markov une fonction de chaque état interne, tandis que d'autres font de la sortie une fonction des transitions. A chaque état donné, il y a un choix de symboles, chacun avec une certaine probabilité d'être sélectionné. Une chaîne de Markov cachée est un processus

doublement stochastique. Elle consiste en un processus stochastique qui ne peut pas être observé, décrit par les probabilités de transition entre paires d'états et calculées à partir d'un corpus d'apprentissage. Deuxièmement, un processus stochastique gère les symboles de sortie qui peuvent être observés à partir des données d'entrée au processus, et représentés par les probabilités de sortie du système. Les principaux paramètres du HMM peuvent être résumés par l'ensemble des probabilités de transition, l'ensemble des probabilités de sortie et l'état initial du modèle.
L'utilisation de modèles de Markov cachés dans la résolution d'un problème d'étiquetage implique 3 problèmes algorithmiques: l'apprentissage, l'évaluation, l'estimation. Pendant l'apprentissage, les paramètres initiaux du modèle sont ajustés et maximisés afin d'observer une séquence de symboles. L'apprentissage implémente l'algorithme de réestimation de Baum-Welch. A ce stade un corpus étiqueté à la main permet de calculer les paramètres du modèle(probabilité de transition d'être dans l'état i à la position p et de suivre un état j à la position j+1). Le problème de l'évaluation est celui de calculer la probabilité qu'une séquence observée de symboles apparaisse comme résultat d'un modèle donné. Il est résolu en utilisant un algorithme forward-backward. Dans un problème d'estimation, nous observons une séquence de symboles produite par une chaîne de Markov cachée. Il s'agit d'estimer la séquence d'états la plus probable que le modèle permet d'obtenir pour produire cette séquence de symbole. L'algorithme de Viterbi permet de calculer une telle séquence.

3.2.2 Patrons syntaxiques

Après avoir obtenu un corpus proprement étiqueté et "désambigué", l'étape suivante d'extraction de groupes nominaux consiste en quelques règles de grammaire. Certaines études sur des corpus français montrent que les groupes nominaux fréquents apparaissant dans les textes sont seulement au nombre de 4: Nom-Adjectif, Adjectif-Nom, Nom-Préposition-Nom et Nom-Nom. Ces séquences représentent des syntagmes nominaux parmi les plus nombreux dans les textes. Bien entendu il est possible d'enrichir la grammaire en ajoutant un adverbe ou un adjectif dans un groupe nominal pour obtenir des formes variantes comme Nom-Adjectif-Préposition-Nom ou Nom-Adverbe-Adjectif…etc. Cette action étend la collecte des GN intéressants en réduisant des formes variantes à leur forme la plus fréquente. Cette réduction est particulièrement importante pour une approche par classification que nous verrons au paragraphe 4. Les règles de grammaire sont spécifiées en tant qu'expressions régulières récursives. Le texte d'entrée est transformé en un texte de sortie étiqueté traité par un compilateur d'expressions régulières. Toutes les étapes du traitement (étiquetage et extraction des GN) sont implémentées grâce à des automates à états finis. Cela signifie que les chaînes de caractères sont converties en arbre avec des nœuds simples et des nœuds finaux. Les automates à états finis sont largement utilisés par les compilateurs de langage. Fortement optimisés, ils permettent d'accélérer les temps de traitement et peuvent aussi réduire le stockage de données (dictionnaire…).
Nous utilisons l'extracteur de groupes nominaux de Xérox qui prend le corpus en entrée (figure 3.5) et renvoie la liste des groupes nominaux non lemmatisés et non triés (figure 3.6).

Usage : **npr –french corpus.txt > out.txt**

Exploration de la circulation coronarienne par la coronarographie

M.G.BOURASSA

Avant l'ere de la coronarographie, nos connaissances de la circulation coronarienne etaient limitees aux donnees d'examens anatomopathologiques.

Les rayons X furent decouverts par Roentgen en 1895 et peu apres, en 1906, on opacifiait pour la 1ere fois, au moyen de substance de contraste,les arteres coronaires chez le cadavre.

Figure 3.5 Extrait du corpus médical (corpus.txt)

```
  0 np   Exploration de la circulation coronarienne
  7 np   coronarographie
  9 np   M.G.BOURASSA
 13 np   ere de la coronarographie
 19 np   connaissances de la circulation coronarienne
 25 np   limitees
 27 np   donnees d' examens anatomopathologiques
 34 np   rayons X
 37 np   decouverts
 39 np   Roentgen
 44 np   apres
 53 np   1ere fois
 57 np   moyen de substance de contraste,les arteres coronaires
 66 np   cadavre
 83 np   arteres coronaires
```

Figure 3.6 Extrait des groupes nominaux (out.txt)

3.3 Lemmatiseur

Etant donné notre servitude à un extracteur étranger à nos programmes, nous devons implémenter un processus de lemmatisation en prévision de l'utilisation d'un extracteur de segments répétés s'affranchissant de tout étiquetage morphosyntaxiques réutilisable.
Ce processus de lemmatisation ne s'appuie pas sur des règles de dépendances syntaxiques mais sur des règles morphologiques. Nous disposons de 2 ressources : une base de suffixes et un dictionnaire de lemmes. Nous testons si un mot appartient au dictionnaire sinon nous opérons une troncature de suffixe.

Cutrac.exe traite un fichier de terme : les termes ont le même traitement que Suf.exe. En effet, il est évident que les mots composants les termes et le corpus doivent avoir le même traitement.

Usage : **cutrac dictionnaire fichier_terme fichier_terme_destination**

Cutrac prend mot par mot le fichier des termes à l'aide d'un programme en **Lex**. L'unité lexicale est identique à Suf.exe (***M_suf.l***). L'action lancée à chaque mot est légérement différent; le lemme ou la concaténation est enregistrer dans un fichier. Je rappelle que le chargement en mémoire du dictionnaire est identique à celui de suf.exe.

3.3.1 Le dictionnaire

Le dictionnaire de lemmes est un fichier texte ASCII dont chaque entrée correspond à une ligne du dictionnaire. Il contient environ 161000 mots différents de la langue générale reliés

chacun à un des 56300 lemmes c'est-à-dire entrées uniques d'un dictionnaire classique (figure 3.7).
Pour des raisons de commodités il nous a fallu retravailler le dictionnaire pour traiter les mots accentués et non accentués dans le cas ou un texte était écrit avec des mots accentués sans diacritiques (exemple: "ce critere s'interprete comme indice de similarite" au lieu de "ce critère s'interprète comme indice de similarité"). Nous avons donc sélectionner les mots accentués pour les désaccentuer et les reclasser par ordre des codes ascii dans le dictionnaire d'origine. Une deuxième variante à l'étude est le dictionnaire admettant les deux formes d'un mot accentué : dans sa forme avec et sans diacritiques au sein du même dictionnaire. Les règles gérant la lemmatisation (§3.3.5) peuvent s'appliquer de la même façon avec un dictionnaire avec les formes normales et sans diacritiques ensembles, qu'avec un dictionnaire avec formes diacritiques seules.

```
161400
a,avoir.V1:P3s
abaissa,abaisser.V:J3s
abaissaient,abaisser.V:I3p
abaissais,abaisser.V:I1s:I2s
abaissait,abaisser.V:I3s
abaissant,abaissant.A:ms
abaissant,abaisser.V:G
abaisse,abaisser.V:P1s:P3s:S1s:S3s:Y2s
abaisse,abaisse.N:fs
```

Figure 3.7 Extrait du dictionnaire de lemme

Le dictionnaire commence par un nombre représentant le nombre de lignes contenues dans le dictionnaire (sert à l'allocation des tableaux de pointeurs sur les mots, lemmes et tags). Chaque ligne est constituée du ***mot***, du ***lemme*** associé et du ***tag (étiquette syntaxique)***.

Voici les étiquettes que l'on rencontre dans ce dictionnaire dont la plus fréquente est V à cause des formes conjuguées:

V:	verbe	(98700 occurrences)
N:	nom	(37500 occurrences)
A:	adjectif	(23400 occurrences)
ADV:	adverbe	(1000 occurrences)
XI ou XINC:	inconnu	(200 occurrences)
PREP:	préposition	(140 occurrences)
PFX:	préfixe	(100 occurrences)
DET:	déterminant	(100 occurrences)
INTJ:	interjection	(70 occurrences)

Un programme utilitaire ("**tri_dico.exe** ") trie les entrées du dictionnaire par ordre de code ASCII croissant et ajoute le nombre de mots contenus en en-tête.
La première étape de "**cutrac.exe**" est donc de mettre en mémoire ce dictionnaire (Annexe 12). Trois tableaux de pointeurs sur des chaînes de caractères sont alloués (la taille des tableaux est connue par la première lecture du dictionnaire). A chaque mot est donc allouer le mot, l'étiquette et le lemme si celui-ci est différent du précédent (sinon il pointe sur le précédent).

Cette représentation peut être optimisée par le stockage des chaînes à l'aide d'automates.

3.3.2 Algorithme de Cutrac

Algorithme

```
pour toute chaîne lue du corpus
appliquer Dicho(chaîne) pour trouver la forme lemmatisée
si retour vide alors appliquer une troncature par suffixe Action(n,x)

//routine de recherche de code dans le dictionnaire
int Dicho(chaîne)
initialiser haut = nombre de mot du dictionnaire, bas =0
      tant que bas < haut
      milieu = (haut+bas)/2
      si chaine < mot_dictionnaire[milieu])    haut = milieu - 1
      sinon si chaine > mot_dictionnaire       bas = milieu + 1
      sinon renvoyer TrouveIndice(milieu)
si pas trouve renvoyer -1

//routine de désambiguation
int TrouveIndice(numero mot)
S'il n'existe qu'un mot ayant le même code
Renvoyer ce code (mot unique dans le dictionnaire)
Sinon si ambiguïté nom-verbe
            Si le mot précédent est un déterminant
            renvoyer le code du nom
            sinon renvoyer le code du verbe
      sinon si le code du verbe existe
            le renvoyer
            sinon si le code du nom existe
                  le renvoyer
                  sinon si le code de l'adverbe existe
                        le renvoyer
                        sinon renvoyer le premier code de la liste

Si Lex détecte que la forme analysée correspond à la ligne action(n) cela
signifie que la n^ième action possède le suffixe le plus long de la grammaire
correspondant à la forme
Action(n,x)
identifier la longueur de la chaîne = L
Renvoyer la chaîne de longueur L-x
```

Exploration de la circulation coronarienne par la coronarographie
M.G.BOURASSA

Avant l'ere de la coronarographie, nos connaissances de la circulation coronarienne etaient limitees aux donnees d'examens anatomopathologiques.
Les rayons X furent decouverts par Roentgen en 1895 et peu apres, en 1906, on opacifiait pour la 1ere fois, au moyen de substance de contraste,les arteres coronaires chez le cadavre.
Plus tard, en 1933, on parvenait a opacifier et a filmer les arteres coronaires et le ventricule gauche chez l'animal au cours de la vie.
En 1945, la premiere coronarographie non selective etait realisee chez l'homme.
Suite a la description par Seldinger d'une methode percutanee d'introduction des catheters dans le systeme musculaire, plusieurs techniques semi-selectives d'angiographie coronaire ont ete pratiquees.
D'abord, la substance de contraste etait injectee a la racine de l'aorte au-dessus des valves aortiques.

Figure 3.8 Fichier corpus.txt

exploration de le circulation coronar par le coronarograph
m g bourassa
avant le ere de le coronarograph notre connaissance de le circulation coronar etaient limite au donne de examen anatomopatholog
le rayer x être decouvert par roentgen en et peu apr en on opacifiait pour le ere foi au moyen de substance de contraster le arter coronaire chez le cadavre
plus tard en on parvenir avoir opacifier et avoir filmer le arter coronaire et le ventricule gauche chez le animal au courir de le vie
en le premier coronarograph non select etait realise chez le homme
suite avoir le description par selding de un method percutane de introduction du catheter dans le system musculaire plusieurs technique semi-selectives de angiograph coronaire avoir ete pratique
de abord le substance de contraster etait injecte avoir le racine de le aorte au-dessus du valve aort

Figure 3.9 Fichier résultat.txt

3.4 Filtreur des GN par fréquence

Algorithme

```
initialiser n
lire le fichier d'entrée
conserver les chaînes qui n'apparaissent plus de n fois dans le fichier lu
et les stocker dans le fichier de sortie
```

Le fichier des groupes nominaux est trié par un filtre qui va sélectionner les unités présentes au moins n fois dans le fichier.
Nous choisissons n=2 pour assurer la présence de plusieurs contextes.

Usage : **essai sortie.txt test.txt**

1 np	fichier attachmate
5 np	proposition
11 np	propal " type
16 np	termes
18 np	vérification d' aptitude
26 np	recette provisoire
29 np	admission
31 np	recette définitive
36 np	recette définitive
43 np	semaines
46 np	recette principale
52 np	Absence Autre Réunion Commerciale facturation

Figure 3.10 Fichier sortie.txt

proposition
recette définitive
semaines
but
RDV Client CONSEIL DE L' EUROPE
prix

David
Arjo Wiggins
fax
Bon de commande
problème
bon de commande
date d' intervention
travail
mostafa
roth frere
jour

Figure 3.11 Fichier test.txt

De manière identique le fichier d'attributs sera trié par un filtre qui va sélectionner les unités présentes au moins n fois dans le fichier.
Dans notre cas précis nous choisissons toujours n=2. Ce programme est différent du précédent par la grammaire du fichier d'entrée. Le principe algorithmique reste le même.

Algorithme

```
initialiser n
lire le fichier d'entrée
conserver les chaînes qui n'apparaissent plus de n fois dans le fichier lu
et les stocker dans le fichier de sortie
```

Usage : **triverb verbaz.txt verb.txt**

persister
suspendre
retrouver
adresser
rendre
charger
bruire
presser
dilater
sortir
rendre
remanier
prolonger
rythmer
exister
remercier
confirmer

Figure 3.12 Fichier verbaz.txt

persister
suspendre
rendre
charger

bruire
remanier
rythmer
opter
augmenter
suivre
parer
produire
reconnaître
entreprendre

Figure 3.13 Fichier verb.txt

3.5 Générateur des positions

Le programme "suf.exe" analyse un corpus mot par mot et produit (ou complète) un fichier de position. Le corpus est parcouru mot à mot. Pour chaque mot, la position courante est incrémentée et ajoutée dans une table des symboles. Si le mot se trouve dans le dictionnaire, on enregistre son lemme, sinon le suffixe est supprimé pour enregistrer le mot sous sa forme canonique.

Usage : **suf dictionnaire fichier_position_precedent corpus.txt fichier_position**

Le fichier de position précédent n'est pas indispensable. Lorsque celui-ci est absent, la position initiale est initialisée à zéro. Le résultat du programme consiste en 3 fichiers:
deviant.txt (fichier des mots simples déviants); la notion de mot déviant sera définie au §3.6.
fi_pos.txt (fichiers des positions pour chaque token)
verbaz.txt (fichiers des attributs)

3.5.1 Fichier des positions

Celui-ci est représenté ainsi :

Mot ; fréquence ; index_suffixe [, index_suffixe] ; position_occurrence [; position_occurrence]

	Type	Définition
Mot	Chaîne de caractères	représente le mot analysé
Fréquence	Entier	nombre d'occurrence du mot dans le courrier
Index_suffixe	Entier	entier associé au suffixe du mot
Position_occurrence	Entier	position du mot dans le courrier

En considérant le texte présenté à la figure 3.14 on obtient le fichier des positions de la figure 3.15.

Désormais, on gère au niveau de chaque client / personne, une fiche type.

Figure 3.14 Fichier corpus.txt

désormais;1;0;1
on;1;0;3
gérer;1;0;4
au;1;0;5

```
niveau;1;0;6
de;3;0;7;15;22
chaque;1;0;8
client;1;0;9
personne;1;0;11
un;1;0;13
ficher;1;0;14
type;1;0;16
```

Figure 3.15 Fichier fi_pos.txt

Le *mot* est soit le lemme associé au mot si celui-ci existe dans le dictionnaire du lemme ou la forme canonique (s'il n'existe pas dans le dictionnaire).

3.5.2 Algorithme

En ce qui concerne la position, celle-ci incrémentée de 50 pour chaque nouveau paragraphe si le booléen **PARAGRAPHE** est mis à 1 dans le fichier Cooc.ini.
Un nouveau paragraphe est identifié par :
une double étoile dans le fichier corpus (**),
un double saut de ligne.

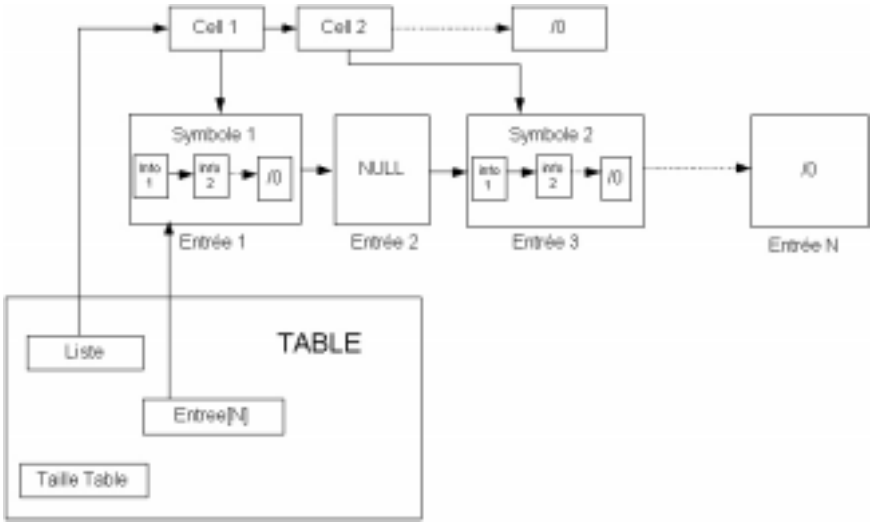


Figure 3.16 Table des symboles

Les mots et leurs positions sont stockés dans une table de symboles hachés (figure 3.16). Cette table contient un tableau d'entrées de taille fixe et une liste chaînée dont les éléments contiennent les éléments non nuls du tableau. Le hashage de la liste chaînée Entree permet de faire un accès plus rapide sur la liste des symboles à partir de la chaîne correspondant à un mot lu du texte. La table s'assimile donc à une liste hashée et doublement chaînée (symbole et info). Pour chaque mot on dispose donc d'une liste de positions pour chaque suffixe.

Algorithme d'ajout d'un élément à une table

```
LongListe   liste chaînée d'objet entiers
Info        structure contenant une liste chaînée des positions (LongListe)

Cellule     liste chaînée de type quelconque
Liste       liste chaînée d'objet Cellule
Symbole     structure contenant la chaîne de caractères d'un mot et un
            objet Liste. La taille de la chaîne est fixée a maximum 64
            caractères
```

```
TableEntree liste chaînée sur des objets de type Symbole
Table       structure contenant un objet de type TableEntree dont la taille
            est fixée a 300000 symboles et un objet de type Liste

Une Info est créée avec un objet de type Info, une initialisation de
            paramètres (déviance, fréquence) une nouvelle LongListe et un
            ajout de la première position trouvée dans LongListe, et la
            valeur numérique du suffixe. L'objet Info est retourné
Un Symbole est créée avec un objet de type Symbole, du nom d'un mot, de sa
            fréquence et d'une liste contenant des Infos (un suffixe, des
            positions) pour chaque suffixe. L'objet symbole est retourné
Une Table est créée avec un objet de type Table,  une initialisation de
            tous les éléments (TableEntree) de Table par un null, une
            taille 0, et un objet de type Liste. L'objet table est
            retourné.
1-Calcul du hashcode d'un  nouvel élément. On associe l'élément de la table
des entrées d'indice le hashcode à un pointeur de type TableEntrée.
2-On teste si le nom du nouvel élément est dans les éléments (symbole) de
TableEntree de la table.
3-Si l'élément est trouve
      On teste si le suffixe est déjà dans la liste des infos pour le
      symbole (liste des Entrée)
      S'il n'y est pas on ajoute une Nouvelle Info avec les informations
      (suffixe, position)
      Sinon on ajoute à l'info contenant le suffixe une position à la liste
      des positions de ce suffixe. On incrémente la fréquence de l'Info
      On incrémente la fréquence du symbole.
4-sinon
      on crée une Nouvelle Entrée (dans TableEntree)
      on affecte un Nouveau Symbole (nom,  position, suffixe) à l'élément
      de cette entrée et un Null au suivant
      on affecte la Nouvelle Entrée à l'entrée de la table avec le hashcode
      comme indice
      on ajoute en fin de la liste de la table l'élément de la nouvelle
      entrée (cette liste sera manipulée,  avec plus de facilité, lors
      d'une sortie fichier par exemple)
```

Algorithme général

```
créer une table
1-appliquer le même algorithme que Cutrac pour lire chaque entrée et
lemmatiser
2- si le mot lemmatisé n'existe pas dans la table créer une entrée dans la
table des symboles (i.e. le lexique)
sinon pour l'élément existant de la table compléter la liste des position,
incrémenter sa fréquence, incrémenter le nombre de paragraphe dans lequel
le mot apparaît s'il n'a pas déjà été détecté dans ce paragraphe
si le mot est un verbe le sauvegarder dans un fichier à la volée
(verbaz.txt)
3-lister toutes les entrées de la table dans un fichier texte (fi_pos.txt)
4-lister les mots où D/Nd (D nombre de paragraphes où le mot est présent,
Nd nombre total de paragraphes) < seuil fixé lu dans le fichier cooc.ini
(deviant.txt)
```

3.6 Ajout de mots déviants

On appelle mot déviant un mot simple dont la distribution dans le texte n'est pas uniforme. Pour cela on calcule sa participation dans le texte en fonction des paragraphes. Pour être sûr de récupérer des mots simples, utiles et interprétables on trie la liste des résultats avec une suite de mots simples les plus ambigus correspondant généralement à des mots vides (prépositions, articles, conjonctions …). Il est souvent risqué de vouloir classer des termes

avec des mots simples dans la mesure où ceux ci sont bruités (peu interprétables, polysémiques). Les mots simples sont souvent très fréquents tissent des relations en plus grand nombre et difficile à discriminer. La déviance peut donc être un moyen de filtrer les mots simples en ne retenant que ceux qui sont significatifs et dont les relations sont restrictives. Dans cette mesure ils servent de relateurs entre mots composés qui ne seraient pas reliés ensemble sans l'exploitation des relations de type terme composé/mot simple.

Le calcul de la déviance se fait en même temps que l'analyse du texte pour élaborer le fichier position (voir §3.7). A la suite de ce calcul on obtient un fichier *deviant.txt* qui énumère les mots déviants susceptibles de nous intéresser.

Le module *devi* agit en fonction d'un seuil ε_p qui permet de retenir les mots simples. Le module intègre les mots simples dans le fichier de termes si $D_p < \varepsilon_p$. Ce seuil est tributaire de la méthode numérique de repérage des mots déviants:

$$D_p = \frac{N_t}{N_{tot}} \quad ,$$

où N_t est le nombre de paragraphes dans lesquels le terme t participe, et N_{tot} est le nombre total de paragraphes du corpus. Pour un seuil donné (ε_p = 0.03 , participation dans 30 paragraphes sur 1000) on aura plus de termes deviants si la taille du corpus est grande jusqu'à convergence dès que le corpus atteint une taille critique très élevée à cause de la diminution des mots nouveaux. Si la taille d'un corpus est faible le nombre de mots déviants est quasi nul mais insuffisant pour pouvoir profiter des relations entre multitermes. Les termes déviants sont nécessaires. On doit initialiser ε_p avec une valeur élevée (par exemple 200 paragraphes sur 1000). Mais au fur et à mesure que la taille d'un corpus augmente ce seuil induit une présence élevée de mots simples et donc la formation de classes avec une forte proportion de mots simples fréquents. Pour avoir un résultat équivalent de celui d'un corpus de petite taille et ne pas aboutir à un nombre explosif de classes constituées de mots simples fréquents le seuil doit baisser. Le seuil n'est donc pas fixe et inversement proportionnel à la taille du corpus.

L'extracteur peut aussi considérer des locutions non normalisées comme mots simples (exemple "tel-que") ou des mots vides au sein de groupes nominaux. L'utilitaire devi.exe va aussi nettoyer les termes du fichier recueilli pour enlever des bouts de terme qui ont été considérés à cause d'un défaut algorithmique pour l'extraction de groupes nominaux ou des règles linguistiques mal respectées dans le texte d'origine.

Environ 140 mots vides sont utilisés pour cette opération. Ces mots vides sont intégrés sous leur formes brutes et lemmatisés par cutrac. Par exemple:
"élaboration d", "connaissance ne", "le ingénierie du connaissance" …etc
deviendront après nettoyage :
"élaboration", "connaissance", "ingénierie du connaissance"…etc

Une opération de dédoublonnage a lieu par la suite pour éviter de traiter des termes identiques
Et la liste est ensuite soumise à la comparaison des mots déviants du fichier deviant.txt pour finalement créer le fichier final de termes fi_term.txt.

Usage : **devi.exe deviant.txt fi_term.txt**

```
marcher 0.00383 0.00239 ; 261 ; 419
noël 0.00383 0.00239 ; 261 ; 419
que-ce 0.00383 0.00239 ; 261 ; 419
magasin 0.00383 0.00239 ; 261 ; 419
hifi 0.00383 0.00239 ; 261 ; 419
vidéo 0.00383 0.00239 ; 261 ; 419
demain 0.00383 0.00239 ; 261 ; 419
solde 0.00766 0.00477 ; 261 ; 419
attendu 0.00383 0.00239 ; 261 ; 419
armen 0.00383 0.00239 ; 261 ; 419
```

Figure 3.17 Fichier des mots simples déviants (deviant.txt)

```
cert-renater
certsvp@renater.fr
keith@liia.u-strasbg.fr
info
message
peu
trinoo
stacheldraht
pirate
victime
control un
agent
zombie
port
tcp
communication
mot de passe
port tcp
flood icmp
```

Figure 3.18 Fichier final de termes (fi_term.txt)

4. Méthode de classification[6]

4.1 Caractéristiques de la méthode

4.1.1 Réduction canonique de termes

La densité des cooccurrences extraites à partir d'un texte pourrait s'atténuer à cause de la variété des formes. Durant des siècles les langues ont su créer des familles morphologiques de mots et d'expressions avec approximativement le même sens. Pour nous le phénomène linguistique n'est pas négligeable. Ce phénomène linguistique est partiellement traité dans les produits du marché en recherche d'information et connu sous le nom anglo-saxon de stemming. Pour l'implémenter nous avons besoin de connaître deux types de connaissance linguistique. La première est l'équivalence entre les formes usuelles et leur lemme associé. Nous appelons l'action utilisant cette première connaissance : lemmatisation. Cette première connaissance doit être appliquée aux mots courants du fait de leur forme irrégulière. En effet les mots utilisés dans le dialogue et les documents écrits ont un comportement morphologique

[6] Les exemples donnés en guise d'illustration des traitement proviennent directement du système; Les mots sont donc tronqués et lemmatisés sans retouche de correction grammaticale et orthographique. (On trouvera par exemple "restenos" à la place de "resténose")

irrégulier particulièrement en français par opposition aux mots nouveaux qui suivent des règles de construction restreintes. La seconde connaissance est une liste de suffixes standard. Elle sera utilisée pour les mots spécifiques provenant d'un domaine technique ou d'un jargon. Nous appelons l'action utilisant cette connaissance : troncature. Ainsi ces deux actions sont effectives sur les mots simples : lemmatisation et troncature. Mais ces deux processus concernent seulement les variations de monotermes et non les variations de multitermes. Un autre phénomène linguistique complexe apparaît avec les formes variantes composées [Polanco et al, 1995]. Les groupes nominaux composés ou multitermes peuvent être déclinés en différentes structures ayant des similitudes sémantiques tels que « accélération d'un électron libre » et « accélération d'un électron ». Nous distinguons trois principales variations : l'insertion, l'expansion et la permutation. Ces variations prennent leur origine dans des propriétés géométriques mais pour certaines, comme par permutation, les facteurs sémantiques sont utiles pour corréler « électron accéléré » et « accélération d'un électron » en rapprochant le verbe « accélérer » et le nom « accélération » dans une même famille sémantique. Une des variations les plus simples à traiter est l'insertion. Notre hypothèse de base est la suivante : dans une langue deux formes différentes expriment un sens différent, même si la différence est faible, mais certaines expressions sont plus à proprement parler corrélées par leur sens comparées aux autres. Malheureusement les théories de la linguistique moderne ne nous apportent pas de formalisme pour différencier quantitativement un couple de termes avec leurs traits sémantiques qui leur sont propres.

4.1.2 Echantillons de termes

Pour établir notre méthode de classification, dans un premier temps nous sélectionnons les termes les plus pertinents du fichier de sortie donné par l'extracteur de groupes nominaux. L'extracteur de GN nous donne une liste non triée de groupes nominaux trouvés dans le corpus. Un tel résultat n'est pas directement exploitable. Nous soumettons deux contraintes pour obtenir une entrée adaptée à notre système. La première contrainte est le filtrage de fréquence. La fréquence est le nombre d'occurrences d'un groupe nominal dans un corpus. Nous choisissons 2 comme seuil de sélection d'un groupe nominal ceci afin de collecter plusieurs contextes. Ainsi nous obtenons l'équivalent de segments répétés sur la base des fréquences des séquences de mots. Nous pensons que les expressions fréquentes sont plus représentatives de la terminologie du domaine que les expressions non fréquentes. Nous devons prendre garde que les expressions fréquentes ne sont pas majoritaires dans un corpus. Ainsi la quantité d'information résultante n'est pas susceptible de montrer des signaux faibles d'information. Mais en utilisant des méthodes statistiques et comme nous l'avons expliqué au paragraphe 2.1 nous décidons de traiter les corpus avec des méthodes faibles pour gagner en robustesse. Nous rappelons qu'un hapax est un mot qui a une fréquence unité. La proportion des hapax dans un corpus est majoritaire. Le second paramètre de filtrage permet d'obtenir un fichier de terme final. Il s'agit d'un paramètre de discrimination qui équivaut à une fréquence en paragraphes. En fait le paramètre est double : il concerne la structure du corpus avec les documents et avec les paragraphes. Nous définissons un corpus comme une collection de documents séparés. Nous définissons un paragraphe comme une unité textuelle séparée d'une autre par un saut de ligne multiple ou un couple d'astérisque (placé à la main pour les tests) et un saut de ligne. Le paramètre de discrimination par paragraphe s'écrit $D_p = Nw_p/Nt_p$ où Nw_p est le nombre de paragraphes contenant le mot, Nt_p est le nombre total de paragraphes dans un corpus. Le paramètre de discrimination par document $D_d = Nw_d/Nt_d$ où Nw_d est le nombre de documents contenant le mot, Nt_d est le nombre total de documents dans le corpus. Nous utilisons plus couramment le paramètre de discrimination par paragraphe en coupant la sélection au seuil supérieur de 0.03 . Le second échantillon approprié dans notre méthode est un fichier de tous les verbes exprimés dans le corpus. Les verbes sont essentiellement

communs et bien répertoriés dans les dictionnaires avec leurs flexions. Nous pouvons facilement les détecter dans un corpus et les stocker dans un fichier spécifique. Le troisième échantillon de termes et très important consiste à sélectionner un sous-échantillon du fichier de termes. Nous appelons les éléments de cet échantillon les termes pôles. Des études sur Internet ont aussi fait émerger cette notion [Albert et al, 1999,2000; Barabasi et Albert, 1999]. Nous avons conduit une étude empirique sur un corpus médical nous ayant amené à construire des classes à la main sur la base du contenu médical conceptuel. Les résultat nous a permis d'observer une répartition des classes autour d'un mot spécifique dans un intervalle de fréquences medium par rapport à la fréquence maximale de l'échantillon de termes. Cela correspond à notre idée de construire des classes avec une structure monothétique. Après l'étape du préclassification nous rentrons dans le cœur du processus.

4.1.3 Utilisation de schémas linguistiques

Nous dégageons notre approche de la voie structuraliste de description du langage. Une recherche de fouille dans un corpus peut révéler des relations non-aléatoires [Harris, 1968][Habert et al, 1996]). Quelques relations peuvent être appelées schéma du fait de leur composition. Nous nous intéressons notamment aux structures relationnelles de schéma verbe-GN. D'autres types de schémas pourraient servir dans le repérage de relations mais nous disposons d'un fichier de verbes, le schéma verbe-GN s'impose naturellement dans la pratique dans le traitement matriciel. Nous pouvons espérer que les verbes spécifiques soient utilisés syntaxiquement devant/après un élément d'une terminologie [Rousselot et al, 1996]. Ce n'est pas ce que nous observons. Mais comme les verbes représentent une typologie d'état et d'action ils impliquent une utilisation spécifique d'attributs. Nous exploitons le rôle des verbes qui font figure de relateurs quasi-systématiques dans les phrases entre GN. La linguistique computationnelle pure permettrait de trouver des schémas typiques de la forme [terme A][verbe V][terme B] plusieurs fois. Ainsi une règle d'inférence permettrait de grouper le terme B et le terme C de par leur relation [terme A][verbe V][terme C]. Dans notre méthode distributionnelle nous collectons toutes les relations liant un terme A et un terme B du fait qu'il ont un verbe ou plusieurs en commun dans leurs contextes. Les liens un terme et un verbe peuvent avoir ou ne pas avoir de lien syntaxique. Ces relations de cooccurrence entre termes par le biais des verbes (relateurs) seront mises en évidence grâce au produit par la transposée de la table de contingence termes*verbes. Le résultat sera la matrice de cooccurrences qui n'est pas normalisée et sera traitée comme une matrice de présence/absence (0/1) pour effectuer des extractions de graphes. Des corrélations similaires ont été développées en informatique documentaire pour exprimer des relations entre termes et documents. Une matrice termes*documents est construite et multipliée par sa transposée pour obtenir des ensembles lexicaux par classification.

4.1.4 Modèle de graphe

J.L Kuhns en 1959 déclare que la classification basée sur des graphes peut être bénéfique à l'indexation terminologique. Mais à cette époque aucune application informatique n'a validé cette hypothèse. [Sparck-Jones, 1987[1967]][Augustson & Minker, 1970] ont optimisé des algorithmes de recherche de cliques pour l'appliquer à une matrice termes*documents. Ils ont pu extraire un certain nombre de classes intéressantes à partir d'un ensemble de 4000 termes. Comme nous le savons, l'extraction de sous-graphes maximaux à partir d'un graphe est un problème NP-complet. C'est pourquoi depuis les années 70 aucune application n'a vraiment utilisé d'extraction de graphes de façon significative. Nous pensons que la classification par graphe pourrait répondre à notre postulat du fait qu'il s'implémente par association, et les liens entre variables sont traités séparément.

Soit l'ensemble d'items I dénotant l'ensemble des vertex ou sommets de graphes (les termes dans notre cas). Un hypergraphe sur I est une famille $H=\{E_1, E_2,\ldots,E_n\}$ de côtés ou sous-ensembles de I, tels que $E_j \neq \emptyset$, et $\cup^n_{i=1}E_i=I$. Un hypergraphe simple est un hypergraphe tel que, $E_i \subset E_j \Rightarrow i=j$. Un simple graphe est un hypergraphe simple pour lequel les côtés ont une cardinalité 2. La cardinalité maximum de côté est appelé le rang, $r(H) = \max_j |E_j|$. Si tous les côtés ont la même cardinalité, alors H est appelé hypergraphe uniforme. Un hypergraphe uniforme simple de rang r est appelé hypergraphe r-uniforme. Pour un sous-ensemble $X \subset I$, le sous-hypergraphe induit par X est ainsi défini, $H_x = \{ E_j \cap X \neq \emptyset | \ 1 <= j \leq n\}$. Un hypergraphe complet r-uniforme avec m sommets, dénoté par K^r_m, consiste en tous les r sous-ensemble de I. Un sous-hypergraphe complet r-uniforme est appelé clique d'hypergraphe r-uniforme. Une clique d'hypergraphe est maximale si elle n'est pas contenue dans n'importe quelle autre clique. Pour les hypergraphes de rang 2, cela correspond au concept familier de clique maximale dans un graphe. Dans la suite du chapitre nous appelons une clique un sous-hypergraphe complet 2-uniforme (i.e. un graphe dont tous les sommets sont reliés). Nous définissons l'ordre o d'une clique C comme la cardinalité de son ensemble d'arêtes N distinctes, o = card(N (C)). Voici l'ensemble K_3 des graphes de base (cliques d'ordre 3):

$$K_3 = \left\{ C = (i, j, l) \text{ avec } i \in P \text{ et } j, l \in I = (1,\ldots,n) \ \middle| \ o = 3 \right\}$$

où P est l'ensemble des termes pôle (figure 3.19)

akines(Po),	examen,	angioplast	(a)
akines(Po),	examen,	trois	(b)
akines(Po),	angioplast,	trois	(c)
akines(Po),	examen,	tritronculair sever	
akines(Po),	angioplast,	tritronculair sever	
akines(Po),	examen,	stenos	
akines(Po),	angioplast,	stenos	
akines(Po),	examen,	sever du lesion	
akines(Po),	angioplast,	sever du lesion	
akines(Po),	examen,	serre	
akines(Po),	angioplast,	serre	
akines(Po),	examen,	scintigraph	
akines(Po),	angioplast,	scintigraph	
akines(Po),	examen,	risque	
akines(Po),	angioplast,	risque	
akines(Po),	examen,	restenos	
akines(Po),	angioplast,	restenos	
akines(Po),	examen,	risque	
akines(Po),	examen,	restenos	

Figure 3.19 Cliques d'ordre 3 autour du pôle "akinésie"

$$K_4 = \left\{ C = (i, j, l, m) \text{ avec } i \in P \text{ et } j, l, m \in I = (1,\ldots,n) \ \middle| \ o = 4 \right\}$$

(figure 3.20)

(A)	akines(Po),	examen,	angioplast,	trois
	akines(Po),	examen,	angioplast,	tritronculair sever
	akines(Po),	examen,	angioplast,	territoire
	akines(Po),	examen,	angioplast,	stenos
	akines(Po),	examen,	angioplast,	sever du lesion
	akines(Po),	examen,	angioplast,	serre

akines(Po),	examen,	angioplast,	scintigraph
akines(Po),	examen,	angioplast,	risque
akines(Po),	examen,	angioplast,	restenos

Figure 3.20 Cliques d'ordre4 formée avec les clique-3 (exemple: a,b,c donnent A)

$$K_{agg} = \left\{ \bigcup_{k=1}^{\mu} C_k \text{ avec } C_k = (i, j, l, x_k) \text{ et } C_{k'} = (i, j, l, x_{k'}) \; \forall k, k' \in \{1,...,\mu\} \text{ et } C_k, C_{k'} \in K_4 \right\}$$

(figure 3.21)

akines(Po), examen(Pi), angioplast(Pi), trois, tritronculair sever, territoire, stenos, sever du lesion, serre, scintigraph, risque, restenos

Figure 3.21 Aggrégation des cliques-4 autour de 2 pivots:"examen" et "angioplastie"

La figure 3.21 ne montre qu'une classe temporaire mais le résultat réel est davantage de l'ordre de quelques centaines de classes temporaires. Beaucoup de ces classes ont un recouvrement important et tendent à abaisser considérablement la dissimlarité entre les classes. Un critère de discrimination va permettre de retenir les classes qui sont interprétables les unes par rapport aux autres. On utilise un critère de recouvrement minimum c'est-à-dire un nombre maximum de termes en commun. Ce nombre se calcule grâce au cardinal minimum de toutes les classes temporaires que l'on divise ensuite par 2. Par exemple si parmi les classes temporaires la plus petite classe contient 6 termes alors toutes les classes ne devront pas contenir plus de 3 termes en commun vis-à-vis des autres. Ce critère, utile et efficace, a néanmoins le défaut de dépendre de l'ordre de passage des classes temporaires; on pourrait envisager de calculer une fonction de gain qui dépende des attributs des classes.

$$K_{clas} = \left\{ C_k \;\middle|\; C_i \cap C_j < \frac{\min(\text{card}(C_k))}{2} \quad \forall i, j, k \in \{1,...,\mu\} \right\}$$ (figure 3.22)

akines(Po), examen(Pi), angioplast(Pi), trois, tritronculair sever, territoire, stenos, sever du lesion, serre, scintigraph, risque, restenos

Figure 3.22 Discrimination des classes partielles de la figure 3.21 (ici on n'a qu'une seule classe)

Propriétés géométriques:
La configuration géométrique des graphes (annexe 8) répond a des propriétés de symétrie que l'on peut associer à des propriétés des groupes de symétrie SO. Ces symétries sont des

- symétrie de rotation: C_n (axe passant par l'origine) $(C_n)^n=1$;

- symétrie des réflexion: σ_h(plan horizontal), σ_v(plan vertical) (passant par l'origine) $\sigma^2=1$

- symétrie de rotation réfexion: $S^n=\sigma C^n=C^n\sigma$

renversement d'espace: parité $P(=\sigma C^2)$

4.2 Calcul de la matrice

4.2.2 Calcul d'une cooccurrence

Dans un graphe, 2 sommets x et y sont dits adjacents s'il existe une arête E qui les relie; deux arêtes sont dites adjacentes si leur intersection est non vide. La matrice d'incidence du graphe H=(X,E), dont les vecteurs colonnes représentent les arêtes (coefficients matriciels a_{ij}) et les vecteurs lignes représentent les sommets. La matrice de cooccurrences peut ainsi être interprétée comme une matrice d'incidence du graphe H.

Cette considération s'avère utile en considérant la matrice comme des dépendances entre objets par le biais des arêtes. On peut s'en servir pour former des associations de graphes en traitant les coefficients matriciels comme des arêtes d'un graphe d'arité E=2. Les graphes seront donc constitués de relations binaires.

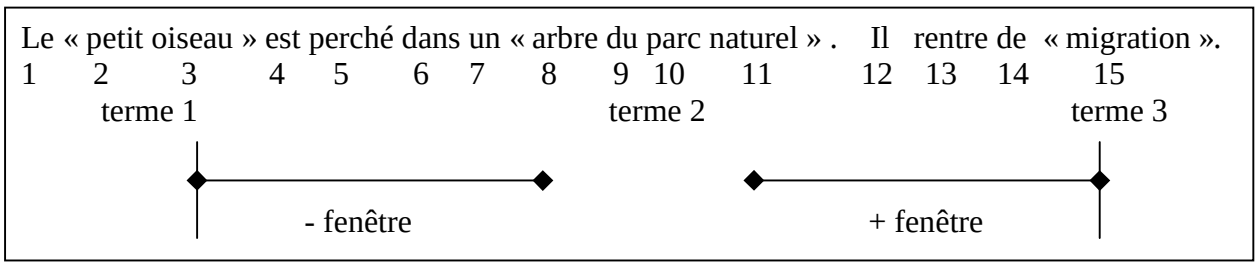
Définition d'une cooccurrence

Une cooccurrence est l'apparition commune d'une expression E1 avec une autre E2 dans une unité textuelle T.

$$\mathrm{Cooc}(E1,E2)=\begin{cases}1 \text{ si } \exists T / E1 \in T \text{ et } E2 \in T \\ 0\end{cases}$$

La définition précédente est à la fois précise et vague en même temps à cause de lacunes formelles. On peut aussi bien considérer une cooccurrence d'un mot avec un autre (cooccurrence de "chat" avec "lait" dans "le chat gris boit du lait"), qu'une étiquette syntaxique avec une autre étiquette (cooccurrence du sujet "le chat gris" et du verbe "boire" dans (le chat gris)/SUB boit/V (du lait)/COMP). La cooccurrence peut aussi bien se faire dans le contexte d'un syntagme, d'une phrase, d'un paragraphe ou d'un texte. Cette définition est à géométrie variable et forge un caractère spécifique de chaque algorithme qui traite des cooccurrences. A ce jour il n'existe pas d'algorithme qui puisse traiter tous les types de cooccurrences peut être à cause du codage algorithmique que nécessite la nature même du type de cooccurrence considéré.

Nous choisissons de traiter une cooccurrence à l'aide d'une fenêtre de mots entourant une expression. Voici la représentation graphique dans le texte d'une fenêtre de cooccurrence :



Les termes du fichier de syntagmes sont "petit oiseau" (terme 1), "arbre du parc naturel" (terme2) et "migration" (terme 3). On pose la largeur de la fenêtre de cooccurrence égale à 10 mots. Entre terme 1 et terme 2 la distance est de 5 mots. Elle est inférieure à 10 donc cooc(terme 1,terme 2) est incrémenté de 1 (idem pour cooc(terme 2, terme 3)).

Nous expliquons dans ce qui suit notre calcul des coccurrence rélaisé à partir des mots simples et des séquences de mots. C e calcul n'est pas optimal puisque les chaînes peuvent être considérées comme des codes avec une information de longueur. Il est ensuite plus aisé d'appliquer le calcul de coccurrence comme s'il s'agissait de mots simples.

Cooccurrence entre 2 monotermes

Dans le texte 2 cas possibles :
Mot A situé avant Mot B et Mot B situé avant Mot A
Exemple : l'« oiseau » se cache dans la « forêt » et la « forêt » cache l' « oiseau »

Algorithme

```
Variables :        Position_Mot A, Position_Mot B,
                   Position_Mot C, Position_Mot D,   différence, m(i,
                   j), pos_inf_T1, pos_sup_T1 pos_inf_T2, pos_sup_T2,
                   coeff_tempo(i, j)
Paramètres :       fenêtre_terme (10 par défaut)
                   fenêtre_variante (6 par défaut)

Pour une valeur de position_Mot A on calcule
      Différence= (Position_Mot A - Position_Mot B)

Si ABS(Différence) < fenêtre
Alors l'élément de la matrice est incrémenté :
m(Mot A, Mot B)++
```

Cooccurrence entre 2 multitermes

Terme 1 (T1) : début du terme (Mot A), fin du terme (Mot C)
et
Terme 2 (T2) : début du terme (Mot B), fin du terme (Mot D)

Exemple : Le « petit oiseau » est perché dans un « arbre du parc naturel »
Mot A=petit Mot C=oiseau
Mot B=arbre Mot D=naturel
On identifie T1 ou T2 par leur premier mot et leur dernier mot.
On recherche d'abord les positions du multiterme :

D'abord il faut identifier les positions des multitermes .
définition d'une forme variante :
deux formes sont variantes l'une de l'autre par insertion (mots de début et de fin de syntagmes identiques)

Exemple de deux formes variantes par insertion :

	sans lemmatisation	avec lemmatisation
multiterme 1	classification de termes	classification de terme
multiterme 2	classification efficace de terme	classification efficace de terme

Algorithme

```
début de T1 Mot A, fin de T1 Mot C
      et
début de T1 Mot C, fin de T1 Mot A

T1 :  Pour une valeur des position de A et C on calcule
  Différence= (Position_Mot A - Position_Mot C)

Si ABS(Différence)  < fenêtre
 Si Diff>0 alors pos_inf_T1=pos_MotA et pos_sup_T1=pos_MotC
 Si Diff<0 alors pos_inf_T1=pos_MotC et     pos_sup_T1=pos_MotA

On entend par _inf la partie inférieure du terme (dernier      mot
appelé expansion)
On entend par _pos la partie supérieure du terme (premier   mot
appelé tête)
```

```
De même pour T2 :
début de T1 Mot A, fin de T1 Mot C
      et
début de T1 Mot C, fin de T1 Mot A

Si Différence>0 alors pos_inf_T2=pos_MotB et pos_sup_T2=pos_MotD
Si Différence<0 alors pos_inf_T2=pos_MotD et pos_sup_T2=pos_MotB

Résultat  liste  de couples (pos_inf_T1, pos_sup_T1)
 et       de couples (pos_inf_T2, pos_sup_T2)
```

Identification des cooccurrences entre 2 multitermes :

Algorithme

```
    2 cas possibles :
Terme1 situé avant Terme2
et
Terme2 situé avant Terme1

Différence= (Pos_inf_T1 - Pos_sup_T2)
ou
Différence= (Pos_inf_T2 - Pos_sup_T1)

Si ABS(Différence)  < fenêtre (exemple 10 mots )
Alors l'élément de la matrice est incrémenté :
m(Terme1,  Terme2)++
```

Problème de collision de 2 cooccurrences

exemple 1 :
si on a : T1= « pomme de terre » et T2= « terre cuite »
alors « pomme de terre cuite » sera considéré comme cooccurrence de « pomme de terre » et « terre cuite »

Pour le cas précédent il faut réunir les 2 conditions :
Que le mot inférieur de T1 soit égal au mot supérieur de T2
Qu'une expression réunissant les trois mots soit dans le corpus

exemple 2 :
si T1= « méthode d'analyse » et T2=« méthode de classification »
dans une phrase du type «méthode de classification et méthode d'analyse »
« classification et méthode » sera considéré comme une occurrence de « méthode de classification » par inversion (traitement des formes variantes par inversion) ce qui est faux.

On peut résoudre ce problème par la condition suivante sur la recherche des couples de position d'un terme T1 :
`si pos_inf_T1 de la position du couple courant = pos_sup_T1 de la position du couple précédent alors passer à la recherche suivante d'un couple de T1.`

Etat de la matrice
Deux possibilités sont offertes pour calculer la matrice. On présente un fichier de termes pour les lignes et un pour les colonnes.
Cas d'une matrice symétrique:
m(Terme1, Terme2) = m(Terme2, Terme1)
La diagonale de la matrice est formée par les fréquences des termes.

Cas d'une matrice non-symétrique:
Notre méthode s'attache plus à ce cas qui permet de croiser des objets avec des attributs. Pour pouvoir extraire des graphes on doit obligatoirement disposer d'une matrice symétrique d'où la nécessité de transposer la matrice et de multiplier la matrice par sa transposée. La matrice résultant de la symétrisation croise ainsi les termes du premier fichier c'est-à-dire les GN.

Algorithme

```
Pour i  allant de 0 au nombre de terme du premier fichier
      Pour j allant de i au nombre de terme du premier fichier
            Pour k allant de 0 au nombre de termes du deuxième
            fichier
Coeff_tempo (i,j) = coeff_tempo (i,j) + m(i,k)* m(j,k)
Finpour

Nouveau coefficient m(i,j)= Coeff_tempo (i,j)
On fixe à 0 les coefficients pour lesquels i<j .
```

4.2.2 Algorithme de stockage de la matrice

Le calcul des cooccurrences produit une demi-matrice creuse c'est à dire qu'elle est composée majoritairement de zéros. De plus, la matrice peut facilement atteindre une taille importante. Elle sera donc compactée. La technique utilisée est proposée dans [Wilheim & Maurer, 1990]. Nous utilisons donc une technique de compression pour gagner en espace, au prix d'un temps d'accès un peu plus élevé. Cependant les entrées vides du tableau (c'est-à-dire à 0), sont aussi significatives pour l'analyse et les traitements : il faut donc pouvoir disposer de l'information contenue dans ces entrées (Annexe 12).

4.3 Extraction de termes pôles

On se sert de la base de syntagmes pour extraire les termes pôles. Un terme est pôle si sa fréquence appartient à un intervalle calculé. L'extraction est conditionnée par un intervalle de fréquence fixé par la fréquence maximale des termes du fichier de syntagme.

fichier des syntagmes

Numéro de terme	terme	fréquence
1	méthode de classification	100
2	sémantique	25
3	terme	15
4	classe	10
5	clustering conceptuel	2

fichier des termes pôles

Numéro de terme	terme pôle
2	sémantique
3	terme
4	classe

Ce module se propose d'extraire un sous-ensemble de l'ensemble des termes caractérisés par. un intervalle de fréquence et de la fréquence maximum de l'ensemble des termes.

Algorithme

```
Paramètres born_inf (0.1 par défaut) permet de fixer la borne
inférieure de l'intervalle de fréquence, born_sup (0.3 par défaut)
permet de fixer la borne supérieure de l'intervalle de fréquence
Variable     FreqMax (fréquence maximale du fichier de syntagme)
```

La fréquence maximale (F_{max}) des termes est recherchée dans le fichier termes/positions (champ fréquence).
On distingue deux cas :
Un cas de traitement particulier d'un texte de très petite taille

```
      Si FreqMax <= 10
      On fixe l'intervalle des fréquences des termes pôles F_p :
      2 <= F_p  <= 4
```

Un cas de traitement régulier d'un texte volumineux

```
      Si FreqMax > 10
      On fixe l'intervalle des fréquences des termes pôles F_p :
E(FreqMax*born_inf/100) +1< F_p  < E(FreqMax*born_sup/100  +1)

E(x) : partie entière de x

tout terme est pôle si sa fréquence appartient à l'intervalle précédent.
```

Modification incrémentale

La construction incrémentale de la matrice implique une modification de la liste des pôles car les fréquences sont modifiées :

Algorithme

```
      Si(FreqMax modifiée)  { modifier intervalle ;
                      Recalculer la liste des pôles ;
                       vérifier si (terme est nouveau pôle) calculer
                       clique_3 de ce nouveau pôle
                       si (terme ancien pôle) éliminer clique_3 de cet
                       ancien pôle
                       }
      Sinon si (terme ancien pôle) éliminer clique_3 de cet ancien pôle
```

4.4 Extraction de cliques d'ordre-3

Des associations mutuelles d'ensemble de trois termes dont un terme pôle seront extraites. Ces ensembles sont appelés clique d'ordre trois, ce sont les briques de base pour classer les termes.

Une clique d'ordre 3 est un groupe de 3 termes différents formé d'au moins un terme pôle. Les conditions d'acquisition d'une clique sont des contraintes sur les coefficients matriciels de la matrice de cooccurrence (figure 3.23).

Les liens i-j, j-k et i-k sont valués par l'élément de la matrice correspondant (respectivement m(i, j), m(j, k), m(i, k)).
On recherche des cliques dont les trois liens sont supérieurs ou égaux à un seuil (Link1, Link2, Link3). Comme ce triplet de seuil n'est pas ordonné par rapport aux arêtes on teste le seuil par permutation circulaire. Les valeurs de link1, link2 ou link3 sont en principe égales à 1. Elles servent essentiellement à tester le comportement du classifieur en fonction de la quantité de cliques d'ordre 3.

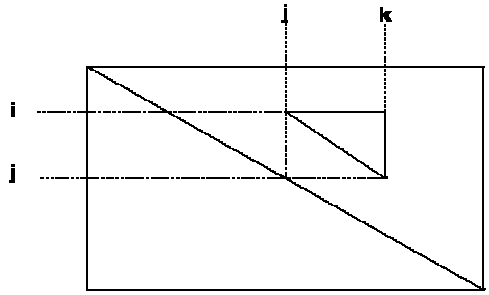


Figure 3.23 Matrice de cooccurrence représentant les termes i, j et k formant une clique-3

Une clique de 3 peut contenir plusieurs termes pôles. Seul le terme pôle courant est considéré, il est d'ailleurs placé en tête de liste des termes constituant la clique de 3, les autres termes pôle pouvant constituer la clique sont traités comme de simples termes. On trie les cliques par ordre croissant de numéro de terme pôle pour gagner en temps de calcul.

Algorithme

```
            variables : m[i][j] élément de la matrice de cooccurrence
            Paramètres :  Link1 (1 par défaut) Link2(1 par défaut) Link3(1
            par défaut)

Pour chaque paire [i, j] modifiée
      Si  (i ou j)  est  terme pôle
          Si [i, j]>=Link1,  pour tout k, de la liste des termes, différent
          de i et j
            si ([j, k]>=Link2 et [i, k]>=Link3) ou
                              ([j, k]>=Link3 et [i, k]>=Link2)
            alors stocker (i, j, k) comme clique de pôle (i ou j)
      même chose par permutation circulaire de Link1, Link2 et Link3

      sinon pour tout pôle k
            si [i, k] >=Link1 et [i, j]>=Link2 et [k, j]>=Link3
      ou    si [i, k] >=Link1 et [i, j]>=Link3 et [k, j]>=Link2
            alors stocker (i, j, k) comme clique de pôle k
      même chose par permutation circulaire de Link1, Link2 et Link3

      Comparer (i, j, k) aux cliques existantes et la garder si elle est
      nouvelle
Trier les cliques par ordre croissant de numéro de terme pôle
```

4.5 Agrégation de cliques

L'agrégation s'effectue en 3 étapes.
Tout d'abord les cliques d'ordre 3 sont assemblées pour construire des cliques d'ordre 4 qui sont des ensembles de 4 termes mutuellement associés. La réunion de 3 cliques d'ordre 3 est effectuée tout en gardant un même terme pôle.
Ensuite les cliques d'ordre 4 obtenues sont assemblées en fonction de deux critères : avoir un terme pôle en commun et deux autres termes.
Finalement on obtient ainsi une série de classes de termes que l'on trie si elles ont plus de n termes en commun ($n \leq 2$).

Recherche de cliques d'ordre 4

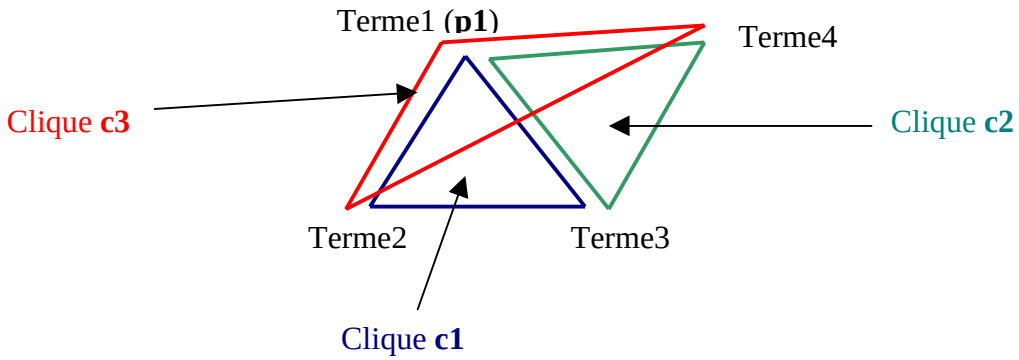
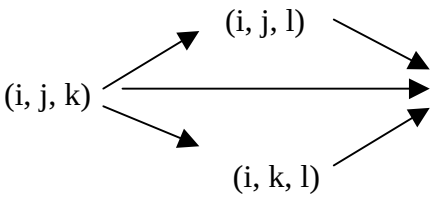


Figure 3.24 Clique-4 représentées par composition de cliques-3

Pour une clique d'ordre 3 donnée composée de (i, j, k) dont le terme pôle est i, on cherche à construire une clique d'ordre 4 de terme pôle i s'il existe un terme l tel que les deux autres cliques d'ordre 3 suivantes existent : (i, j, l) et (i, k, l) (figure 3.24).



(i, j, k, l) clique d'ordre-4 de pôle i

Une clique de 4 peut contenir plusieurs termes pôles. Seul le terme pôle courant est considéré, il est d'ailleurs placé en tête de liste des termes constituant la clique de 4, les autres termes pôle pouvant constituer la clique sont traités comme de simples termes. On trie les cliques par ordre croissant de numéro de terme pôle pour gagner en temps de calcul.

Algorithme

```
Variable: clique_3

pour toute clique_3 de terme pôle i1 (i1, edge1, edge2, edge3)
pour toute clique_3 de terme pôle i2 (i2, edgeA, edgeB, edgeC)
    si i1 = i2
            si  (edge1) = (edgeA)
            pour toute clique_3 (i3,edge*,edge+, edge@)
            si i1=i3
                  si(edge2) = (edge+) et (edgeB)  = (edge*)
            ou si(edge2) = (edge*) et (edgeB) = (edge+)
                 si (i1,edge2, edgeB, edge3) n'existe pas
                  alors l'enregistrer
            si (edge1) = (edgeB)
            pour toute clique_3 (i3,edge*,edge+, edge@)
            si i1=i3
                  si  (edge2) = (edge*) et (edgeA) = (edge+)
                 ou si (edge2) = (edge+) et (edgeA) = (edge*)
                  si (i1,edge2, edgeA, edge3) n'existe pas
                  alors l'enregistrer
            si (edge2) = (edgeB)
            pour toute clique_3 (i3,edge*,edge+, edge@)
            si i1=i3
                  si  (edge1) = (edge*) et (edgeA)  = (edge+)
                  ou si (edge1) = (edge+) et (edgeA) = (edge*)
                  si (i1,edge1, edgeA, edge3) n'existe pas
                  alors l'enregistrer
            si (edge2) = (edgeA)
            pour toute clique_3 (i3,edge*,edge+, edge@)
            si i1=i3
                  si  (edge1) = (edge*) et (edgeB)  = (edge+)
                  ou si  (edge2) = (edge+) et (edgeB)  = (edge*)
                  si (i1,edge1, edgeB, edge3) n'existe pas
                  alors l'enregistrer
```

Regroupement des cliques d'ordre 4

Deux cliques d'ordre 4 sont regroupés si et seulement si
- un terme pôle est commun
- deux autres termes sont communs (termes pivots). (figure 3.25)

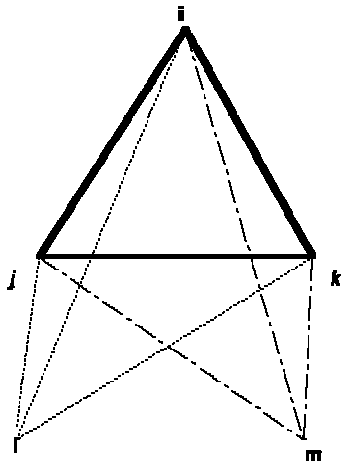


Figure 3.25 **i** est terme pôle, *j* et *k* sont termes pivots.

Un regroupement peux concerner plus de 2 cliques d'ordre 4 à condition qu'elles aient le même terme pôle et les mêmes termes pivots. On appelle le résultat de ce regroupement : une classe.

Exemple : (**i**, *j*, *k*, l), (**i**, *j*, *k*, m) et (**i**, *j*, *k*, n) vont donner la classe (**i**, *j*, *k*, l, m, n).

On limite le nombre de termes communs entre deux classes (term_id). Ce paramètre ne doit pas dépasser 3 sinon cela reviendrait à retenir toutes les cliques d'ordre 4 plus les groupes de plus de 4 éléments (plusieurs milliers de groupes). Empiriquement on observe que le nombre de classes doit être inférieur à $5\sqrt{N}$, N étant le nombre de termes.

Dans le cas d'un corpus conduisant à l'extraction de 10 000 syntagmes la méthode employée permet d'obtenir au maximum 500 classes.

Algorithme

```
variable :   clique_4
Paramètre :  term_id (2 par défaut)

pour toute clique_4 de terme pôle i1
      (i1, edge1, edge2, edge3)
      pour toute clique_4 de terme pôle i2
      (i2, edgeA, edgeB, edgeC)
      si i1 = i2 et (edge1) = (edgeA) et (edge2) = (edgeB)
      alors   créer   la   classe   (i,j,k,l,m)   des   termes   composants
      edge1,edge2,edge3 et edgeC
      ou si i1 = i2 et (edge1) = (edgeB) et (edge2) = (edgeA)  alors créer
      la  classe  (i,j,k,l,m)  des  termes  composants  edge1,edge2,edge3  et
      edgeC
      vérifier que la classe n'a pas plus de term_id termes en commun avec
      les classes existantes sinon garder la plus grande
      sinon si (i1, i2, i3, i4) n'a pas plus de term_id termes en commun
      avec les classes existantes la garder sinon garder la plus grande

      on  trie  la  nouvelle  classe  obtenue  parmi  les  autres  par  ordre
      croissant de numéro de terme pôle
```

4.6 Agrégation par équivalence distributionnelle

A partir des classes précédentes une opération d'agrégation va traiter un terme (T1) lié au terme pôle (T2) d'une classe et qui possède une distribution de liens de cooccurrence proportionnelle à celle de T2. On teste l'écart entre les deux distributions pour décider de l'appartenance de T1 avec la classe de T2. C'est l'étape finale de la classification.

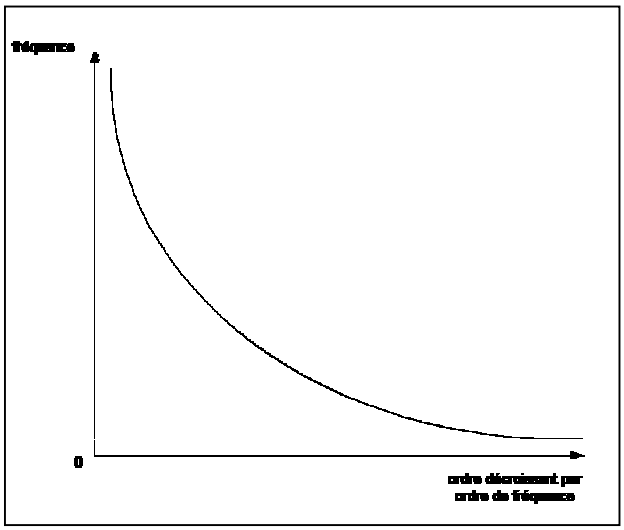


Figure 3.26 Distribution de Zipf

L'agrégation utilise une règle sur la distribution des relations de cooccurrence entre deux termes pour compléter les classes obtenues par le module assembleur de cliques. Cette règle est effective pour un corpus de courriers électroniques relativement volumineux (>10 000 mots). On émet l'hypothèse que le contexte lexical entre deux termes liés sémantiquement est similaire. Il s'agit d'une hypothèse forte sur le contexte sémantique. Ainsi on agrège un terme à une classe si pour tout terme pôle ce terme possède un volume de termes (dans son contexte) proportionnel au volume des mêmes termes dans le contexte du terme pôle.

On modélise la distribution des termes par une loi de puissance de type loi de Zipf (figure 3.26) [Zipf, 1935].
A partir de chaque terme pôle (T2) et de chaque terme (T1) de la liste de syntagmes on teste si le coefficient de la matrice de cooccurrence est supérieur à un certain seuil. Dans le cas favorable T2 et T1 sont ensuite comparés par le nombre d'hapax qu'ils ont en commun. Une matrice de cooccurrence est calculée avec le T2 et T1 dans le fichier des lignes et les termes hapax dans le fichier des colonnes. Ci-dessus est représentée la distribution des termes dans un corpus; l'allure est celle d'une loi de puissance.

Si le nombre d'hapax en commun est supérieur à un certain seuil on compare la distribution des termes non hapax entre T1 et T2 en calculant de la même façon une matrice de cooccurrence entre un fichier des lignes composé de T2 et de T1, et comme fichier des colonnes les termes de fréquence supérieure à 2. Une fonction d'erreur est calculée pour évaluer la proximité distributionnelle. Si cette fonction est inférieure à un certain seuil T1 est ajouté à la classe correspondant à T2. Le processus se déroule en quatre étapes.

4.6.1 Etape de comparaison

La première étape consiste à comparer tous les termes du fichier de syntagmes avec un terme pôle et à ne retenir que ceux qui ont une coefficient matriciel de cooccurrence supérieur à un seuil. Soit T_p un terme pôle et T un terme si cooc(T_p,T) > seuil_cooc alors passer à l'étape 2.

Nous n'avons pas de valeur précise pour ce seuil qui vaut 1 par défaut. C'est la première condition ; elle permet de sélectionner un terme candidat.

4.6.2 Etape de relation avec hapax

La deuxième étape consiste pour le terme candidat à calculer les relations de cooccurrence des termes hapax avec ce terme et avec le terme pôle. On teste si le nombre est supérieur ou non à une valeur.
Si nb_hapax_communs(T_p,T) > seuil_hapax alors passer à l'étape 3.
Nous n'avons pas de valeur précise pour ce seuil qui vaut 1 par défaut. C'est la deuxième condition ; elle permet de passer à l'étape 3.

	Hapax 1	Hapax 2	…	Hapax n
Terme candidat	a	b	…	c
Terme pôle	a'	b'	…	c'

4.6.3 Etape de relation avec non-hapax

La troisième étape consiste pour le terme candidat à calculer les relations de cooccurrence des termes fréquents avec ce terme et avec le terme pôle.

	Mot 1	Mot 2	…	Mot n
Terme candidat	aa	bb	…	cc
Terme pôle	aa'	bb'	…	cc'

f_1 = aa f_2 = bb … f_n = cc $\quad S_f = \Sigma f_i$ = aa + bb +… + cc
g_1 = aa' g_2 = bb' … g_n = cc' $\quad S_g = \Sigma g_i$ = aa + bb +… + cc'

4.6.4 Etape de calcul de la déviation

Dans la quatrième étape on calcule une fonction de déviation. On enlève 10% des mots les plus fréquents et 10% les moins fréquents.
On appelle ε cette fonction de déviation. ε évalue la différence distributionnelle entre la distribution des mots fréquents du voisinage du terme pôle et celle du terme candidat. Si les distributions sont équivalentes (identiques ou proportionnelles) ε doit tendre vers 0. Le calcul de ε correspond à la dérivée nulle du rapport des surfaces de la courbe représentant les cooccurrences (en ordonnée) par rapport aux termes (en abscisse).

M(i,j) : coefficient matriciel de cooccurrence
S_g : somme des M(i, j) pour le terme pôle
S_f : somme des M(i, j) pour le terme candidat
Δ_g : écart entre M(i+1, j) et M(i-1,j) pour le terme pôle (j)
Δ_f : écart entre M(i+1, j) et M(i-1,j) pour le terme candidat (j)

$$\varepsilon = \left| \frac{\sum_{i=2}^{p_{max}-1} (S_g \cdot \left|\Delta f_{i-1}^{i+1}\right| - S_f \cdot \left|\Delta g_{i-1}^{i+1}\right|)}{2S_g^2} \right|$$

La valeur de ε est déterminante pour décider si le terme candidat peut être ajouté à la classe du terme pôle. Si cette valeur est inférieure à un seuil l'ajout est effectif.

Algorithme

```
Variables :  matrice m[][], terme[i], hapax[m],
                          terme_pole[j], classe[k], non_hapax[n]
Paramètres : seuil_cooc (2 par défaut), seuil_hapax (2 par défaut),
seuil_nonhapax (2 par défaut), seuil_e (1 par défaut)
```

```
Pour chaque terme_pole[j]
      Pour chaque terme[i] de la liste des termes
      si m[ terme[i] ][ terme_pole[j] ] > seuil_cooc
              alors  calculer une matrice de cooccurrence         entre la
                       paire terme[i],terme_pole[j] et l'ensemble des termes
                       hapax[m]
                     faire le produit scalaire des deux lignes
                    si ce produit > seuil_hapax
                     alors calculer la matrice de cooccurrence entre la
                             paire terme[i],terme_pole[j] et l'ensemble des
                             termes non_hapax[m]
                            calculer e
                            si e < seuil_e
                            alors ajouter terme[i] dans la classe pour
                                   avoir le centre d'intérêt
```

4.7 Complexité

Temps: Soient N_0 le nombre de termes, N_{3t} le nombre de cliques d'ordre 3 et N_{4t} étant le nombre de cliques d'ordre 4.
Termes pôles: est $O(N_0)$ en temps de calcul.
Cliques d'ordre 3: fait parcourir 1 boucle sur les termes pôles ($N_0/3$) et 2 boucles sur les termes (N_0) pour chaque terme pôle, soit $O(N_0^2/3+N_0^2/3+N_0^2)$ ou $O(4/3*N_0^2)$. Cette estimation concerne le pire des cas puisque en fait on parcours les éléménts non nuls de la matrice, soit environ 10% des relations entre termes.
Clique 4 pour chaque pole on teste le nombre moyen de clique-3 par pôle:
$N_0/3 * (N_0/3 / (N_0/3)) = 3N_{3t}^2/N_0$
Classes temporaires: pour chaque pole pour une clique on parcours les cliques pour trouver clique ayant une paire de terme en commun et on prcours de noveau les cliue pour trouver les autres claiques ayant cette paire de terme soit $N_0/3 * \ln N_{4t} * (N_{4t} / (N_0/3)) = N_{4t} \ln N_{4t}$, dans l'hypothese ou on a en moyenne le meme nombre de classes par pole $N_{4t} / (N_0/3)$
Discrimination: on parcours pour chaque pôle les classes temporaires et on parcours de noueavu les classe tempo pour trouver une classe qui a plus de n termes en commun on élimine cette classe $N_0/3 * \ln Nc * (Nc / (N_0/3))$; dans l'hypothese ou on a en moyenne le meme nombre de classes par pole $(Nc / (N_0/3))$.
L'ordre general est $O(4/3*N_0^2 + Nc \ln Nc + N_{4t} \ln N_{4t} + 3N_{3t}^2/ N_0 + N_0)$.
Le processus le plus gourmand en temps de caclul est la recherche de clique-4 qui prend pres de 70% du temps global de classification.

Espace: Clique3 $O(N_{3t})$, Clique 4 $O(N_{4t})$, Classe $(O(N_{cfinales})$, Classe tempo (N_c).
L'ordre général est $O(N_c+ N_{cfinales}+N_{4t} +N_{3t})$.

5 Perspectives

5.1 Plan technique

Pour augmenter la clarté du code et sa simplicité il serait intéressant de disposer de 4 classes fondamentales: terme, position, matrice et classe.

Ces classes pourraient être écrites en utilisant des API d'un environnement RAD de langage objet comme Java ou C++ profitant ainsi de la gestion mémoire des objets par les librairies de l'environnement. Les méthodes reprendraient la structure présentée en Annexe 12.

Le traitement est relativement lourd notamment pour l'extraction de clique d'ordre 4. De plus si le module doit traiter des flux continus pouvant accumuler une grande quantité de données le temps de calcul augmenterait au fur et à mesure parce qu'il nécessiterait une réinitialisation

du calcul. Pour palier ce problème une solution est l'incrémentalité. La méthode présentée dans ce chapitre possède une propriété d'incrémentalité. Elle n'est pas exploitée à cause de la représentation de la matrice qui permet d'être gérée en mémoire en totalité avec un minimum de ressources. Il est possible d'envisager de stocker la matrice à l'aide d'une base de donnée relationnelle comprenant une table et 3 champs: ligne, colonne, valeur. Seules seraient contenues les valeurs non nulles. Un index serait générer par le gestionnaire de la base pour accélérer les temps d'accès.
Les autres données (termes, pôles, positions, cliques, classes) seraient également stockées dans des tables de base de donnée relationnelle pour assurer cette incrémentalité.
L'incrémentalité ferait diminuer l'extraction des cliques d'ordre 4 d'un ordre la faisant passer de $O(n^3)$ à $O(n^2)$.

5.2 Plan théorique

Un approfondissement théorique serait l'automatisation de l'extraction à partir d'un modèle quelconque admettant un pôle. Nous présentons, dans cette thèse, un seul modèle permettant l'inférence de concept. Comme nous l'avons présenté idéalement en 1.1, des concepts présentés comme données de références pourraient donner lieu à des modèles de graphe utilisés comme patrons d'inférence. (se reporter au chapitre 2 pour l'état de l'art des méthodes de classification).

Une autre orientation serait le couplage de la classification avec une autre méthode d'apprentissage comme l'extraction de règles inductives ou d'arbres de décision permettant de relier des classes entre eux ou de confirmer la présence de termes au sein d'un même classe.

Enfin la compréhension de la nature des collocations/cooccurrence n'est pas encore complètement établie dans les processus de classification. En effet si un terme possédait les même cooccurrents et de façon exclusive on en déduirait des paradigmes de manière évidente ainsi que les relations entre ces paradigmes. Or cette exclusivité d'apparition de cooccurrents est loin d'être vérifier. Il faut donc savoir quels types d'associations se trouvent dans les textes, quels sont ceux qui favorisent la classification et ceux qui seraient les plus utiles. Cette analyse pourrait déboucher sur une meilleure interprétation des attributs à utiliser lors d'une analyse multivariée.

6 Travaux antérieurs

Il n'existe pas de méthode proposant de réaliser une inférence de concepts ou de classes par génération de motifs de graphe. Il existe certaines méthodes basées sur une structure de graphe, notamment les cliques, mais l'approche ne vise pas l'induction de classes à partir d'une famille de modèles de graphe voire même de modèles de graphe déduits des données pour permettre une inférence.

[Needham, 1961][Sparck-Jones, 1987[1967]] développent la théorie des bouquets (angl., Clumps) pour l'appliquer à la génération de thésaurus. [Sparck-Jones, 1987[1967]] utilise 180 lignes de termes composés du thésaurus Roget pour les reclasser automatiquement et les confronter aux têtes de champs sémantiques du Roget. Elle étudie 3 définitions d'associations et trouve que les bouquets non exclusifs donnent de meilleurs résultats. Aucun protocole d'évaluation n'est appliqué à la discussion des résultats. Voici la définition d'une classe d'après [Sparck-Jones, 1987[1967]]:

ce critère est celui de Tanimoto (annexe 3):

$$C_{ij} = \frac{i \cap j}{\sum_{i \notin i \cap j} Ai + \sum_{j \notin i \cap j} A_j}$$

pour lequel les attributs sont les éléments d'un terme (étant donné qu'un terme est composé de mots).

1- Un ensemble C est une clique si

a) pour tout $i \in C$, $j \in C$ où $C_{ij} \geq \varepsilon$ où ε est un seuil adapté, et

b) il n'existe pas d'ensemble $C' \supset C$, tel que C' satisfait a)

Cela signifie qu'un membre une clique est associée à tous les autres membres.

2- Un ensemble C est un B-bouquet si

a) pour tout i, j où $i \in C$ et $y \notin C$, $C_{ij} < \varepsilon$ où ε est un seuil adapté, et

b) C est maximal pour cette propriété (l'ensemble entier étant exclu)

Cela signifie qu'un membre fait partie d'un B-bouquet et un seul.

3- Un ensemble C est un GR-bouquet si chaque membre de C possède un total d'associations avec les autres membres plus qu'avec les non-membres, et vice-versa pour les non-membres

Cela signifie qu'un membre d'un groupe est associé plus fortement aux membres de ce groupe dans son ensemble sans critère d'exclusivité.

[Sparck-Jones, 1987[1967]] juge en fonction de ses résultats et de façon intuitive que la 3ème définition est plus appropriée pour comparer les classes aux têtes des champs sémantique du Roget.

[Augustson & Minker, 1970] développe un module qui extrait des cliques maximales et des classes par composants connexes à partir d'une matrice termes*termes.

D'après Sparck-Jones les résultats d'une classification automatique ne sont pas forcément comparables à des têtes de champs sémantique connues d'un thésaurus. La classification est basée sur le nombre de parties de termes composés en commun.

Cette tendance est reprise dans de nombreuse études actuellement [Enguehard, 1992][Habert et al, 1996][Ibekwe-San Juan, 1997][Assadi, 1998].

Plus récemment:

[Enguehard, 1992] tisse un réseau de termes à partir des schémas lexicaux construisant les termes; par exemple "miel de" est déduit de "miel de Provence" et "miel de campagne" il va permettre d'être le nœud commun à plusieurs termes de même morphologie. De cette façon après un amorçage avec un certain nombre de termes, le réseau se construit donnant lieu à la découverte de nouveaux concepts. La méthode ne s'inspire pas d'une méthode de classification mais de la structure des termes. Des classes sont déduites ensuite à la main par navigation sur le réseau.

[Agarwal, 1995] utilise une variante de Cobweb appelée Cobweb/3. Un étiquetage syntaxique est suivi d'une analyse grammaticale (recherche de sujet, verbe, adjectif, préposition..). Ensuite une classification sémantique classe les termes qu'une ultime étape transforme en patron lexico-sémantiques. Les vecteurs sont formés de 2 façons: pour un nom il s'agit du verbe (si le nom est objet), du verbe si le nom est sujet et de préposition, pour un verbe il s'agit du sujet et de l'objet. Les arguments sont recherchés dans une fenêtre de +/- 10 mots autour d'un terme cible. Les 2 ou 3 valeurs les plus fréquentes sont retenues. La classification aboutit à un dendrogramme qui est coupé à une certaine hauteur par un expert pour obtenir les classes. Wordnet est utilisé pour épurer les classes de mots qui n'appartiendraient pas à une

même classe. En matière d'évaluation un expert valide une classification manuelle de référence. Des paramètres de précision et de rappel ensemble permettent d'estimer la qualité des classes établies automatiquement par rapport à la classification de référence.

[Habert et al, 1996] développe un outil qui réduit des expansions de termes ou des têtes dans un réseau et isole les composantes connexes chacune ayant 50 à 200 termes. Sur cet ensemble une interprétation manuelle assure l'extraction de cliques ou de composantes connexes sous formes de classes composées d'environ 10 termes.

[Ibekwe-San Juan, 1997] utilise un réseau de type Lexter grâce à la structure morphologique des termes en fonction de leur équivalence. Les variations sont étudiées et reliées par un poids mettant en jeu le nombre de constituants morphologiques en commun. Ensuite un algorithme de détection de composantes connexes, et coupure de l'arbre à un certain poids minimum, permet de repérer les classes de termes qui seront assimilées à des tendances thématiques.

[Ploux & Victorri, 1998] relie les synonymes entre eux pour former un graphe. Elle extrait les cliques d'ordre 3 et 4. Par exemple pour "maison" 130 cliques sont extraites. 2 cliques qui n'ont qu'un élément en commun signifie une homonymie.2cliques qui ont plusieurs éléments en commun signifie une polysémie. Une clique unique est une monosémie.

[Tishby et al, 1999] utilise la classification dite du goulot d'information, différente d'une approche de graphe, mais considère les distributions syntaxiques (nom/verbe) en minimisant l'écart de ces distributions avec une mesure probabiliste d'information mututelle généralisée (Jensen-Shannon et Kullback-Leibler) (§chapitre I, 1.3).

Résumé du chapitre

Dans ce chapitre nous avons exposé dans un premier temps l'extraction des unités linguistiques pouvant être classées avec une interprétation aisée, dans un second temps la méthode de classification automatique assurant l'émergence de classes en regroupant les unités linguistiques extraites préalablement.
Le syntagme nominal est considéré comme unité linguistique. Les verbes sont considérés comme attributs. Plusieurs modules assurent l'extraction donnant lieu aux 2 fichiers traités par le module de classification. La première phase est l'utilisation d'un extracteur de syntagme à la volée qui va extraire les syntagmes sans tri ni lemmatisation. La deuxième phase consiste en un processus de lemmatisation des syntagmes. Le lemmatiseur fonctionne de la même façon sur du texte accentué ou non accentué (dans le cas on utilise un dictionnaire accentué et dans le second le même dictionnaire mais non accentué). Le lemmatiseur analyse les mots l'un après l'autre et les lemmatise de 2 manières différentes. Si le mot fait partie du dictionnaire de lemmes, le lemme considéré est celui répondant à des règles simples de désambiguation. L'étiquette syntaxique favorisée est celle du verbe, sinon celle du nom sinon de l'article ou de la préposition et enfin de l'adverbe, si les règles ne décident pas alors la première étiquette est prise par défaut. Si le mot n'est pas membre du dictionnaire alors on réalise une opération de troncature de suffixe en conservant la racine comme forme réduite du mot. Un module de tri prélève les termes au-delà d'une certaine fréquence (2 par défaut). Enfin la troisième phase est un module d'analyse qui transforme le corpus en fichier de position après lemmatisation du mot courant. Pour chaque mot on associe ses positions dans le corpus. Un deuxième résultat de cet analyseur est la recherche des mots simples les plus discriminants, que l'on appelle aussi déviants. La règle qualifiant un mot de déviant est sa participation réduite dans les paragraphes du corpus.
Le processus de classification prend en compte une matrice de cooccurrence qui croisent les termes (éléments du fichier de termes) et les attributs (éléments du fichier de verbes). La matrice est creuse à 90% ce qui nous permet de la traiter en des temps de calcul raisonnables. Un ensemble de termes pôles est extrait du fichier de position grâce aux fréquences des termes. A partir de ces termes pôles on dérive des graphes de base contenant un terme pôle et deux autres termes tous associés: ce sont les cliques d'ordre 3. Ces cliques sont extraites de façon exhaustive. Elles sont ensuite agrégées par groupe de 3 cliques contenant le même terme pôle pour former une clique d'ordre 4: un terme pôle et 3 autres termes tous associés. Ces cliques sont extraites de façon exhaustive. Ces cliques sont agrégées en ayant un terme pôle et deux autres termes (termes pivots) en commun. On aboutit aux classes auxquelles on associe des termes dont la distribution des hapax cooccurrents est la même que le terme pôle. Finalement les classes retenues sont celles qui n'ont pas plus de n termes en commun (n=1 en général).

Apports du prétraitement :
- utilisation de syntagmes nominaux
- utilisation de termes complexes
- utilisation de mots simples déviants
- réduction des formes variantes et des termes par lemmatisation

Apports du traitement :
- utilisation de schémas morpho-syntaxiques comme type de cooccurrence
- utilisation d'attributs comme symboles de la classe (intension)
- utilisation d'un modèle de graphe comme similarité d'association
- utilisation de l'intersection inter-classe comme critère d'affectation
- utilisation d'un terme pôle comme centre d'association d'une classe

CHAPITRE 4
COHESION LEXICALE ET EVALUATION

Dans ce chapitre nous présentons la notion de cohésion lexicale d'une classe de termes et, à travers cette notion, l'évaluation d'une classe par rapport au domaine propre au corpus traité.

L'évaluation reste une tâche complexe à mettre en oeuvre à cause de l'interprétation subjective du résultat. Cette subjectivité provient de la nature des données textuelles qui sont utilisées par tout un chacun sans technicité préalable si ce n'est le fait de se faire comprendre. La compréhension passe par l'assimilation de règles, grammaticales et lexicales, d'un usage confiné et ne nécessitant pas de formalisme pointu si ce n'est la juxtaposition des mots respectant les règles. L'expert ou le non expert cherche cette compréhension à travers ces règles qui sont inhérentes à la structure des données et à leur utilisation. Un paquet de mots pourra, à tort, être perçu comme des données textuelles habituelles, d'où la difficulté d'interpréter. A cela s'ajoute la notion de point de vue duquel dépend le rôle et l'action des constituants.

On essaie d'établir une notion de cohésion lexicale qui vise à interpréter les relations entre termes de façon restreinte par le biais d'une hiérarchie conceptuelle de référence établie en 3 niveaux.

Un premier traitement utilise un thésaurus général externe construit sur le modèle du thésaurus de Roget pour étiqueter les classes parmi les 900 étiquettes proposées par le thésaurus. On utilise un modèle de consensus pour attribuer une étiquette. Ce modèle est basé sur les codes sémantiques émanant de chaque terme d'une classe pour sélectionner l'étiquette la plus probable. Ensuite les étiquettes de l'ensemble des classes sont analysées pour attribuer les 3 étiquettes les plus probables se rapportant au domaine du corpus.

Un second traitement utilise une hiérarchie de référence dépendant cette fois du corpus. Les termes qui seront classés sont manuellement disséminés dans des classes définies par un expert du domaine. Des paramètres de mesure vont permettre de comparer l'ensemble des classes obtenues automatiquement et la hiérarchie de référence. Ces paramètres font appel aux paramètres d'évaluation de la recherche documentaire: le rappel et la précision.

Enfin on essaie de qualifier les résultats par un retour aux données sources pour valider la correspondance entre les relations de cooccurrence, participant à la construction d'une classe, et les relations intra-classe. Ce processus de validation s'effectue par une interface graphique de visualisation des classes avec navigation dans l'espace des classes.

1. La hiérarchie comme référence de cohésion lexicale

1.1 Démarche d'évaluation

A l'époque de [Sokal & Sneath, 1963] il n'existait pas d'évaluation et de stratégie d'évaluation des méthodes de classification automatique. On regardait surtout en quoi les résultats variaient d'une méthode à l'autre.

A l'heure actuelle l'évaluation est considérée comme un problème aussi difficile que l'optimisation dans tous les sujets concernant la recherche d'information. La raison en est simple, elle s'appuie en bout de chaîne sur la satisfaction de l'utilisateur qui juge les résultats pertinents ou pas. Cette satisfaction est rarement unanime et peut varier d'un panel d'utilisateurs à l'autre même quand il s'agit d'experts du sujet concerné par les données. De plus la qualité des résultats peut changer suivant la nature textuelle homogène ou pas des données. Par exemple un texte écrit dans les forums de discussion sur Internet ne suivra pas forcément toutes les règles de base de la grammaire et de ponctuation. Ainsi un étiqueteur syntaxique n'aura peut-être pas la même portée sur un texte encyclopédique que sur un texte de forum de discussion.

Il apparaît donc clairement que l'évaluation constitue une démarche difficile à mettre en œuvre. Cette démarche est néanmoins nécessaire pour se convaincre de l'utilité et de l'exploitation possible des résultats de la classification. C'est une démarche aussi nécessaire pour dégager la fiabilité d'un système de classification par rapport aux autres systèmes ou algorithmes.

Beaucoup de systèmes s'appuient sur des principes de désorganisation d'une structure artificiellement ordonnée, pour retrouver ensuite cette structure et confirmer l'efficacité de l'algorithme de classification automatique. En ce qui concerne les données textuelles ces principes demanderaient à ce qu'on génère un texte artificiellement avec entre autres des termes donnés sachant que ces termes sont reliés en classes de champs sémantiques. Comme il n'existe aucune définition formelle d'un champ sémantique, nous butons sur un problème circulaire: constituer un texte artificiel donnant lieu à une classe idéale, qu'on ne connaît pas, pour évaluer cette même classe.
Certains préconisent le besoin d'un thésaurus de référence, "en or" [Yarowsky, 1992][Resnik, 1992][Grefenstette, 1996].

Nous préférons donc une approche basée sur l'expertise extérieure au système: la cohésion lexicale.

1.2 Notion de cohésion lexicale

Définition
Nous définissons la *cohésion lexicale* d'un ensemble E, de N termes reliés ensemble, de la manière suivante:
Soit $T1 \in E$ et $T2 \in E$, $T1 \Re T2$ si cette relation $\Re$ correspond à une réalité R du monde réel. R est équivalente à un ensemble de contextes d'usage {C1,C2,...,Cp} définis dans une hiérarchie H.
H possède 3 niveaux au plus dans lesquels un fils peut être père de lui-même et peut avoir plusieurs pères distincts.

On peut qualifier cette relation, cohésion lexicale, à caractère sémantique, de relation "parle de" ("about").

Cette définition n'est donc pas restrictive et la relation d'un terme avec la contextualité d'usage est de cardinalité (1,m) c'est-à-dire pour un terme on peut associer m contextes.

1.3 Hiérarchie de base

Cette contextualité peut être décrite par une hiérarchie simple des objets classés. Deux options s'offrent à cette description:

(a) une hiérarchie intuitive et empirique connue par l'expert que l'on peut qualifier de *hiérarchie concrète*

(b) une hiérarchie théorique consignée dans des ensembles prédéfinis de classes sous forme d'arbres que l'on peut qualifier de *hiérarchie abstraite*. Cette hiérarchie provient souvent de nomenclatures à caractère encyclopédique.

La hiérarchie concrète détenue par un expert est évolutive en fonction de son expérience. Elle est très intéressante en ce sens qu'elle s'appuie sur cette géométrie variable dans les formulations des notions rencontrées dans les textes qu'il est difficile de synthétiser automatiquement.
Le défaut de cette évaluation par la hiérarchie concrète d'un expert est double:
Est-on sûr de la couverture technique de l'expert sur un domaine, souvent vaste et en perpétuelle mutation?
Est-on sûr de la fiabilité universelle de l'avis de l'expert par rapport à d'autres experts?
D'autres contraintes peuvent s'ajouter suivant le type d'application. Est-ce qu'une application finale doit nécessiter un expert souvent coûteux pour une entreprise. Quel temps maximum doit-t-on réserver au travail de l'expert dans l'évaluation?

Les 2 types de hiérarchie de référence (concrète et abstraite) sont plus ou moins liés car ils nécessitent des connaissances expertes distinctes et extérieures au système que seul un expert du domaine peut apporter. La différence existe et s'avère substantielle dans la mesure où la hiérarchie théorique se trouve consignée et exploitable automatiquement. Quand elle est disponible se pose encore une fois le problème de maintenance et d'évolutivité.

2. Etiquetage des classes avec un thésaurus général

2.1 Structure du thésaurus

Le thésaurus que l'on utilise est celui de Larousse: "thésaurus des idées", disponible sous format électronique dans le logiciel Microsoft Word™. Ce thésaurus possède la même structure que le Roget's en anglais. Les résultats suivants sont donc portables pour l'anglais.

Le thésaurus contient environ 120 000 formes réparties dans 873 sous-catégories lesquelles sont réparties en 26 catégories:
niveau 0 : les termes, environ 100'000.
niveau 1 : les sous-catégories, 873 en tout;
niveau 2 : les catégorie (thèmes), 26 en tout;
Nous l'avons lemmatisé avec l'utilitaire *cutrac* pour être conforme à la morphologie des termes appartenant aux classes obtenues par le classifieur. Cette lemmatisation réduit le thésaurus à environ 100000 formes.

Partie 1 listes des catégories
La première liste est celle des catégories simples. Sa structure est de la forme:

catégorie	numéro (873 en tout)
exemple :	
chimie	75

photographie	775

La deuxième liste est celle des macro-catégories. Sa structure est de la forme:

intervalle nœud 1 code catégorie nœud 1 (26 en tout)

exemple :	
773-791	10 (l'art)

Partie 2 corps des catégories

Dans cette partie la structure est plus irrégulière que dans la précédente. La citation des termes peut être une liste en colonne, des infos peuvent être entre parenthèses ou entre crochets, des termes peuvent être séparés par une virgule ou un point et un tiret, un terme peut être suivi par un code qui renvoie à une autre catégorie. Un signe peut précéder une locution (comme 'fam. :', 'ou', 'mus.-', ' -.mil. :'…) qui est ignorée.

Structure :

numéro de la catégorie
numéro de paragraphe, terme en gras* ; termes non principaux+
(* signifie 0 ou plus, + signifie 1 ou plus)

	exemple : 775
1	appareil ; appareil photo ; photomaton ; photorama ; fusil photographique
2	laboratoire ; studio ; bain ; affaiblisseur ; développeur ; fixateur ; révélateur ; déclencheur ; nitrate d'argent

On va réduire cette paire en index dont la structure sera identique à celle arbre qui sera stocké dans 2 tables de base de données relationnelle:

terme	étiquette syntaxique	code de catégorie1

Une seconde table raccordera le code de catégorie du 1er niveau (parmi les 873) à un code du 2nd niveau (parmi les 26)

code de catégorie1	code de catégorie2

2.2 Stratégie du consensus

L'objectif maintenant est d'exploiter le thésaurus pour attribuer une étiquette, la plus appropriée, pour généraliser une classe. Nous utilisons la notion de consensus généralement employé dans la classification non supervisée pour la construction d'arbre [Leclerc, 1996]. Le problème du consensus consiste en l'agrégation de plusieurs objets en un objet unique du même type. Cette notion de consensus a été largement exploitée dans la construction d'arbres (graphes acycliques et connectés). On trouve un certain nombre d'heuristiques de consensus dont les plus connues sont: le consensus strict, le consensus par quota, le consensus par intersection et le consensus de la médiane.

Définition du consensus:
Soit N individus qui ont à décider d'une valeur binaire (0,1). Chaque individu à sa propre valeur initiale.
- si tous les individus ont la valeur initiale 0, alors ils décident 0;
- si tous les individus ont la valeur initiale 1, alors ils décident 1
  (cette condition est en général affaiblie en renforçant la prémisse);
- tous les individus décident d'une même valeur (des heuristiques qui s'enchaînent sont à mettre en œuvre).

Nous développerons par la suite une méthode de consensus s'inspirant de la méthode par quota permettant d'inférer une décision par quota sans avoir forcément la majorité.

L'étiquetage se fait en 2 étapes:
* étiquetage d'une classe:
  calcul des codes de niveau 1 associés aux termes de la classe et
  affectation du code mojoritaire;
  affectation des codes de niveau 2 associés aux codes de niveau 1 trouvés.

* étiquetage d'un corpus:
  calcul des codes de niveau 1 associés aux classes extraites et affectation
  des 3 plus fréquents.

La même chose peut être faite en se restreignant aux termes pôles et pivots.
exemple de généralisation étape1 (corpus aéronautique 28000 formes, 3400 mots,430 termes):

Code 243 (turbine à gaz) **combustibilité**
<1> turbine gaz, transport international, transport international

code 98 (98 et 260) **mélange**
<2> alliage, grand vitesse, procédé

code 47 (47, 50, 51, 406, 538, 753, 752, 592, 693) **organisation**
<3> procédé, travaux, disquer

exemple de généralisation étape2 :

catégorie les plus présentes :
fréquence 4, numéro 870 sports
fréquence 4, numéro 820 transports par air
fréquence 4, numéro 656 défense

l'ordre et la mesure
47
98
<2> alliage, grand vitesse, procédé
<3> procédé, travaux, disquer

le temps
185
<7> dernier année, études, consommation carburant

le mouvement
221
<12> forces propulsion, défense antiaérien, cible

la matière
233
243 243
<1> turbine gaz, transport international, transport international
<11> accélération, nombreux pays, compagnie aérien
<9> combustion interne, appel, système antiaérien

Algorithme Etape 1 (étiquetage de 1er niveau)

```
1- Rassember les codes de tous les termes composant une classe.

2- Si un terme est composé
     on cherche le code du terme complet s'il existe
     sinon on rassemble les codes du premier et du dernier mot.
          sauf si c'est un adjectif dans ce cas on ne traite pas
          les codes de l'adjectif.

3- S'il existe choisir le code le plus fréquent comme étiquette
     si plusieurs codes ont la même fréquence max
          choisir le plus petit code
     sinon chercher les codes du terme pôle
           si un code (du terme pôle) est plus fréquent
             le considérer comme étiquette
           sinon considérer le plus petit comme étiquette.
```

(l'heuristique de prendre le plus petit code par défaut vient du fait que l'étiquette de code de poids faible est plus générale)

Algorithme Etape 2 (étiquetage 2ième niveau et global)

```
1- Affecter chaque code de centre d'intérêt à un nœud de niveau 1 et
    compris dans l'intervalle du nœud de niveau 2. exemple : code
    248 est dans l'intervalle de 230 à 267 correspondant à la
    matière ; donc le nœud 2 est « la matière » .

2- Sélectionner les thèmes du corpus qui sont prédominants:
    - Rassembler tous les codes de chaque classe,
    - Calculer la fréquence de chaque code,
    - Ordonner les codes par ordre décroissant de fréquence.

3- Considérer les 3 premiers ayant une fréquence ≥ 3.
```

La figure 4.1 présente l'objet thésaurus avec les différentes méthodes considérées dans les algorithmes d'étiquetage.

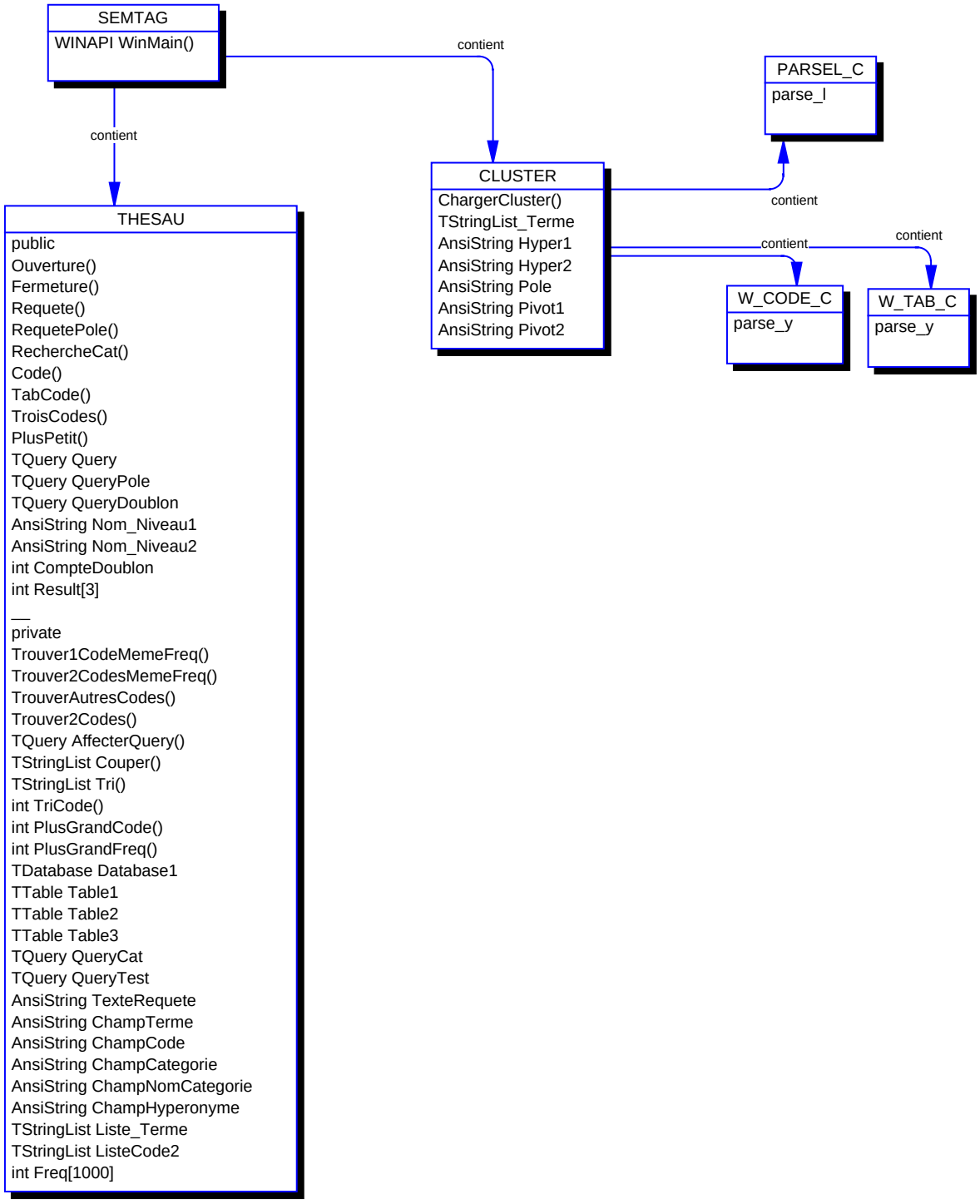


Figure 4.1 Schéma conceptuel du gestionnaire de thésaurus

2.3 Résultats

De la même manière que nous avons décrit le traitement de la généralisation grâce à un thésaurus nous collectons tous les codes de catégorie pour une classe donnée et ceux des termes pôles. Nous déduisons le code de la classe pour une classe, celui correspondant au nom

de catégorie de 1er niveau, et le nom de catégorie de 2nd niveau qui généralise la catégorie de 1er niveau. (figure 4.2).

classe	Catégorie (niv. 1)	Catégorie (niv. 2)
substance de contraste, évolution, obliquer antérieur, technique, test, traitement, ventricule, ventriculograph	Médecine	médecine
seul incidence, diamètre de le obstruction, possible de dilater, angioplast	Dimension	dimensions
valve aort, coronaire, coronar, aorte, circulation	Cœur et vaisseaux	corps
tension artériel, douleur, fraction de éjection, ventriculaire, artère, altération	Cœur et vaisseaux	corps
technique du catheter cardiaque, lésion, examen, angor, cathéter	Méthode	ordre

Figure 4.2 Exemple d'étiquetage pour certains classes.

Le second processus de généralisation consiste à réunir tous les codes liés aux superclasses. Ensuite nous les classons par ordre décroissant pour attribuer les plus fréquents (3) à l'ensemble des classes (figure 4.3) (Annexe 10).

Nombre d'occurrence	Code de catégorie	Nom de catégorie (niveau inférieur)
31	*383*	*Maladie*
23	*331*	*Coeur*
21	*391*	*Médecine*
11	792	Travail
10	185	Période
10	392	Chirurgie
10	393	Soin du corps

Figure 4.3 Catégories les plus fréquentes.

Concernant le corpus médical un aperçu général de l'attribution des catégories montre les résultats suivants : 21 classes ont été catégorisés dans une catégorie médicale et 43 ont été classés dans une catégorie d'état, finalement 85% ont été logiquement classés dans un thème relatif au contenu sémantique du corpus.

Nous avons, de plus, testé l'étiquetage à partir d'un corpus thématique mixte. Nous avons constitué un corpus à partir de textes de l'Encyclopédie Universalis portant sur l'aéronautique d'une part et l'histoire russe d'autre part. La taille du corpus est d'environ 70000 mots. Le traitement du corpus montre une discrimination des sujets à travers les classes de termes extraites par Galex. En effet des 61 classes obtenues par Galex 27 se rapportent à l'histoire russe, 19 se rapportent à l'aéronautique et 15 sont ambigus. De la totalité des classes environ 75 % peuvent être attribuées à leur thème respectif.

3. Evaluation par rappel et précision

3.1 Hiérarchie de référence

Des données artificielles non triées sont mal adaptées pour observer une relation entre les résultats et les données. Nous procédons de manière à utiliser une série de classes de termes prise comme référence d'un domaine thématique. La déviation par rapport à ce cadre sera effective en combinant 3 paramètres. Les 9 catégories sont : physiologie cardiovasculaire, symptomatologie, anatomie coronarienne, pathologie générale, pathologie coronarienne, facteur de risque, diagnostic, thérapie, information patient (annexe 7).

3.2 Paramètres d'évaluation

L'application d'évaluation (figure 4.4) tient compte de *p*, *r* et α, résultant de:

$$T = \alpha \cdot p \cdot \sqrt[3]{r}$$ [Turenne & Rousselot, 1998a].

Ce paramètre T est très similaire à l'évaluation d'un système de recherche documentaire (angl., information retrieval) avec les paramètres de rappel et de précision. T est un indicateur, justifiant, pour une classe donnée, de se rapporter à une catégorie (précision) et de s'y rapprocher (rappel) en considérant le nombre de termes qu'elle contient de cette catégorie. La précision joue un rôle de corrélation interne de la classe en testant l'homogénité (i.e. le nombre de termes de même catégorie). En revanche le rappel joue un rôle de corrélation externe en comprarant le nombre de termes d'une même catégorie par rapport au nombre total des termes existants (dans l'échantillon de termes à classifier) de cette catégorie. Une classe est d'autant plus intéressante que le paramètre T se rapproche de 1, et dans ce cas la classe à évaluer est égale à une classe de référence. Par contre, dans le pire des cas, T vaut 0 et ne contient aucun terme d'une des catégories de référence.

Dans notre expérience r représente le nombre de termes de la même catégorie (parmi 9 catégories) divisé par le nombre de termes de la taxinomie des sous-classes identifiées dans la classe:

$$r = \frac{N_c}{N_{SC}}$$

où :

N_c = {card(T_i) / où $T_i \in$ classe analysée et $i \in I_c$ ensemble des indices sur les classes de référence}

et N_{sc} = { Σcard(SC_i) / où $T_i \in SC_i$ et $SC_i \in C_i$ classe de référence }.

L'autre paramètre p représente le nombre de termes de la même catégorie (parmi les 9 catégories) mais divisé par le nombre d'items de la classe:

$$p = \frac{N_c}{\text{card(Classe analysée)}}.$$

Le paramètre T reflète de bonnes corrélations quand p et r sont ensemble élevés ou dans le cas où p est petit et r est élevé ou vice versa. Pour éviter les bonnes classes de petite taille nous choisissons de réduire T par un facteur de taille α. On modélise α par une croissance exponentielle en fonction du nombre de termes qui converge vers 1 au-delà de 4 termes:

$$\alpha = 1 - e^{-\left(\frac{2 \cdot n}{5}\right)^2},$$

n étant le nombre de termes d'une classe. Ainsi pour une taille de 4 termes α=0.92, pour les autres tailles α=1.

La méthode d'évaluation que nous utilisons peut être qualifiée de semi-empirique puisque nous utilisons une classification manuelle liée au domaine du corpus.

On peut remarquer que T se rapporte à une seule classe et non à l'ensemble des classes. L'évaluation d'une classification nécessite en effet l'analyse globale des résultats. On va donc constituer un histogramme regroupant les valeurs de T par intervalle entre 0 et 1 et représenter le nombre de classes possédant leur valeur de T dans un des intervalles. Une seconde approche exposée au paragraphe 3.6 développera le calcul d'une mesure globale par rapport à la hiérarchie de référence.

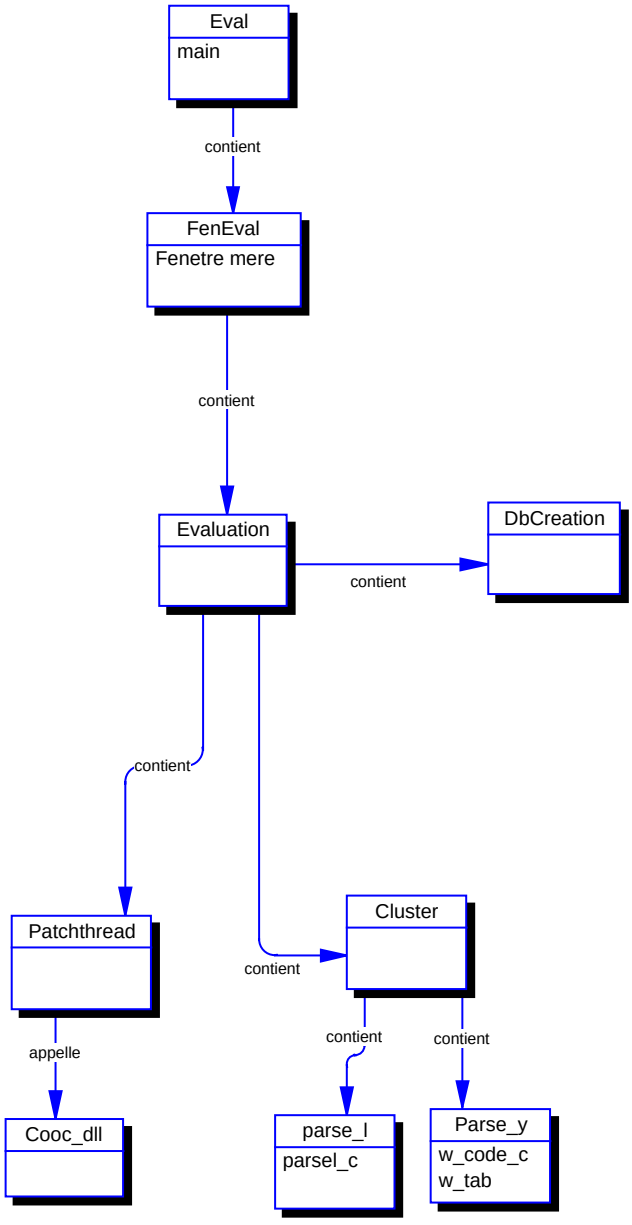


Figure 4.4 Schéma conceptuel de l'application d'évaluation.

3.3 Optimisation du paramétrage du classifieur

Le classifieur Galex admet 2 paramètres numériques essentiels: la *taille de la fenêtre de cooccurrence* (WinSize) et le *nombre maximum de termes en commun entre 2 classes* (TermId). L'intervalle de variation de WinSize est [1-50], celui de TermId est [0-2].
D'autres paramètres, quant à eux secondaires, peuvent être modulés. Il s'agit du *nombre de termes maximum d'une classe* (TermMax), de la *borne inférieure* de l'intervalle de fréquence des termes pôle (BornInf) et la *borne supérieure* (BornSup). Leurs intervalles sont pour TermMax [5-50], pour BornInf [1%-30%] et pour BornSup [10%-50%] avec BornInf<BornSup.

2 paramètres de pré-traitements ne sont pas considérés: la *fréquence des termes et des verbes* (attributs) retenus dans les fichiers d'entrée du classifieur ainsi que le *paramètre de discrimination des termes déviants* (seuil qui permet de retenir le nombre de terme simple dans le fichier de termes à classer).

Nous voyons donc que l'espace des 5 paramètres peut s'avérer important:
$40*3*45*(30*40 - \sum_{n=1}^{31} n) = 3\ 802\ 000$ configurations, soit en admettant un temps de calcul moyen de 1 mn par configuration, le temps d'évaluation s'élève à 7 ans et 40 jours !
Pour optimiser cette évaluation de configuration on découpe le protocole en 3 temps:

1- détermination de WinSize et TermId (120 configurations soit 2 heures de calcul)
2- utilisation du meilleur couple (WinSize, TermId) pour déterminer le meilleur candidat TermMax (45 configurations soit 45 mn de calcul)
3- utilisation du triplet (WinSize, TermId, TermMax) pour déterminer les meilleurs candidats BornInf et BornSup (704 configurations soit 12 heures de calcul)

Ce découpage nous ramène à environ 15 heures d'évaluation au lieu de 4 ans. La figure 4.5 résume les différentes données analysées pour aboutir au calcul de Tmax et pmax pour une classe. Cette procédure est itérée pour l'ensemble les classes. On dispose finalement des intervalles de valeurs de pmax ou Tmax comprenant le nombre de classes qui ont leur pmax ou Tmax dans un intervalle.

coronarograph	C6	ecg		
catheter	C6	ecg		
effort	C6	ecg		
	C7	null		
épreuve de effort	C6	ecg		
nbterme	4			
C6	4 éléments (classe Galex)	28 éléments (classe Ref)	p = 1.000000	T = 0.482378
C7	1 éléments (classe Galex)	14 éléments (classe Ref)	p = 0.250000	T = 0.095718
Tmax	0.482378	Diagnostic		
Pmax	1.000000	Diagnostic		

Figure 4.5 Tableau d'analyse d'une classe pour calculer pmax et Tmax.

Les figure 4.6, 4.7 et 4.8 présentent les résultats des paramètres pmax et Tmax pour les 3 valeurs possibles de TermId (0, 1 et 2).

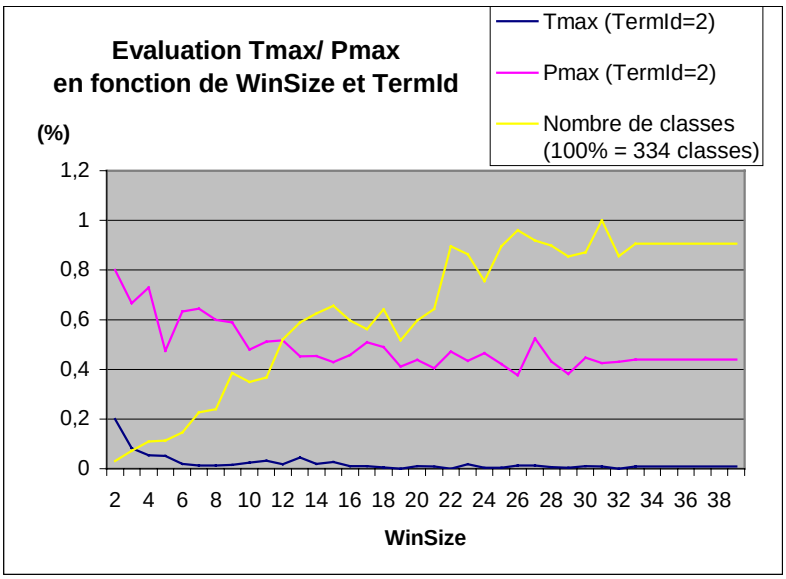


Figure 4.6 Tableau de variation: nombre de clusters (croissant), pmax, Tmax (en bas).

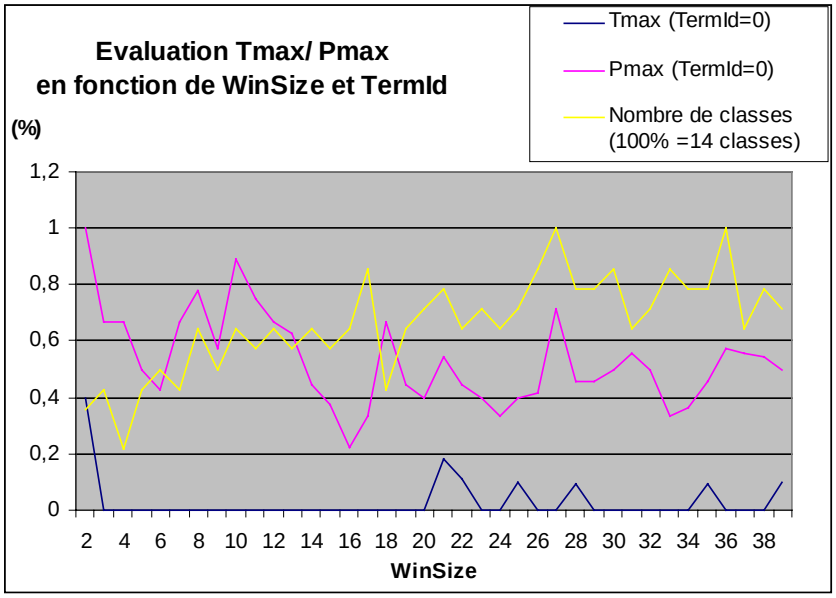


Figure 4.7 Tableau de variation: nombre de clusters (croissant), pmax, Tmax (en bas).

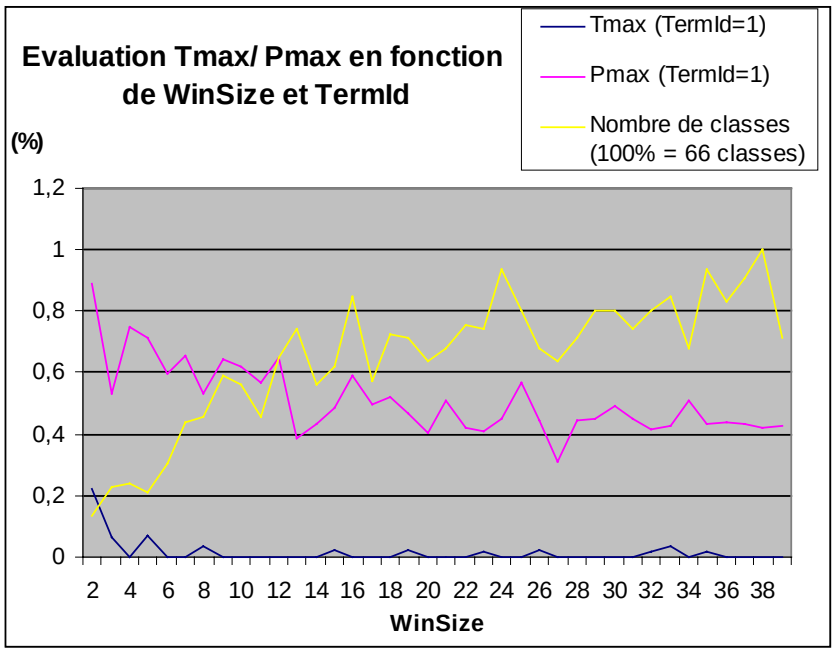


Figure 4.8 Tableau de variation: nombre de clusters (croissant), pmax, Tmax (en bas).

Nous choisissons WinSize entre 9 et 15 termes correspondant à un compromis (rencontre des courbes) du nombre de classes et de Pmax. Ce compromis consiste à avoir un Pmax suffisament élevé tout en obtenant un nombre de classes raisonnable. Cela rejoint l'expérience de [Niwa & Nitta, 1994] qui ont comparé la capacité de désambiguer le sens d'un mot grâce à la définition d'un vecteur de caractéristiques à partir des définitions d'un dictionnaire et grâce à un vecteur de cooccurrences défini à partir d'un corpus. Les vecteurs sont déterminés modulo une taille de fenêtre contextuelle. En deçà d'une taille d'environ 15 mots les deux approches sont aussi efficace. Au delà de 15 mots la désambiguation devient irrégulière et moins bonne dans le cas d'un vecteur de cooccurences.

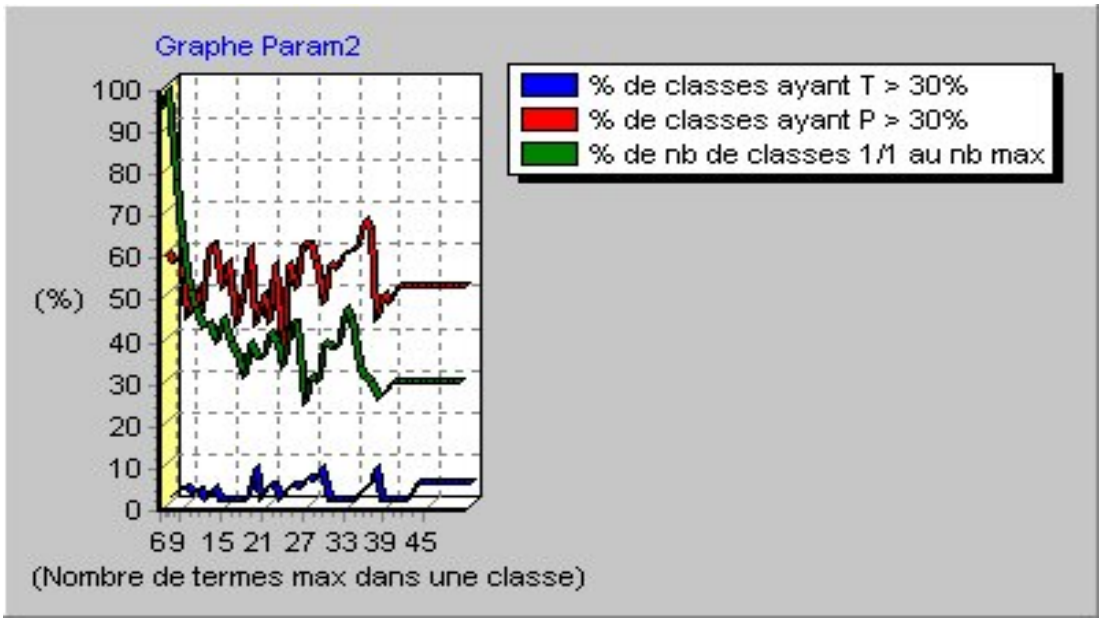


Figure 4.9 Tableau de variation: nombre de clusters (décroissant), pmax, Tmax (en bas).

La figure 4.9 nous présente l'évolution du nombre de clusters, de pmax et de Tmax en fonction du nombre de termes maximum par classe avec TermId=1 et WinSize =15. On remarque que le nombre de clusters décroît avec l'augmentation du nombre maximum de termes par classe. Il se stabilise pour atteindre un plateau minimum après MaxTerm > 25 termes /classes. Au niveau de ce plateau on a une grosse classe fourre-tout et un petit nombre de petites classes. On s'intéresse aux intervalles Tmax prolifiques. On en observe 2 : avec l'intervalle [9-15] et l'intervalle [19-25]. Pour MaxTerm = 17 on obtient 2 classes a 17 termes et les autres (26 classes) <10 termes. Pour MaxTerm = 25 1 classe a 25 termes et les autres (32) < 12 termes. On en vient à conserver l'intervalle [10-12] comme intervalle optimal pour MaxTerm qui permet d'avoir un nombre convenable des classes, un équilibre de leur cardinal et un taux de pmax intéressant (croisement des courbes).

3.4 Echantillonnage par analyse de MonteCarlo

Puisque nous ne pouvons générer un texte réaliste avec une structure qui nous permette d'obtenir des classes "idéales" on peut toutefois désordonner un texte initial en texte aléatoire, évaluer les résultats de classification sur ce texte aléatoire et comparer les résultats avec les résultats de classification du texte initial. C'est ce qu'on appelle l'échantillonnage par analyse de MonteCarlo [Turenne, 2000].

On clone le corpus en 3 échantillons. L'analyse de MonteCarlo s'opère en analysant et comparant 1 corpus initial et 3 corpus désordonnés de même composition lexicale que le corpus initial. On précise que notre échantillons de termes à classifier reste le même et que ces termes (simples ou composés) ne sont pas détruits par la génération du désordre du corpus. En fait on détruit les contextes des termes à classifier.

Echantillon du corpus initial

> Avant l'ere de la **coronarographie**, nos connaissances de la circulation coronarienne etaient limitees aux donnees d'examens anatomopathologiques.
>
> Les rayons X furent decouverts par Roentgen en 1895 et peu apres, en 1906, on opacifiait pour la 1ere fois, au moyen de substance de contraste,les **arteres coronaires** chez le cadavre.
>
> Plus tard, en 1933, on parvenait a opacifier et a filmer les **arteres coronaires** et le **ventricule** gauche chez l'animal au cours de la vie.

Echantillon du corpus désordonné

> droit mopero majorit entier disposition comme je traitement medic gauche douleur avoir devoir au thallium egale qui dire qu **infarctus du myocarde** faire de echograph **coronarograph** present **ventriculograph** avoir intra depuis de
>
> de effet par être sans adresser un mortalit de style plus **traitement** falloir sous-epicardique therapeut le pouvoir plus arythm occlusion coronaire droit bien revascularise singer coronar mais **coronarograph** fonctionnement

Tout d'abord présentons les résultats concernant les 4 corpus.

Voici les résultats du corpus initial. On y observe 89 termes pôles et 146 classes. 9% des termes pôles n'appartiennent à aucune classe. L'ensemble des classes couvrent toutes les

classes de référence. La probabilité d'avoir un terme de la catégorie D (diagnostic) parmi les catégories de la hiérarchie est de : P=53/262=20%. La probabilité d'avoir un terme de la catégorie I (information) parmi les catégories de la hiérarchie est de: P=42/262= 16%. La probabilité d'avoir un terme de la catégorie T (thérapeutique) parmi les catégories de la hiérarchie est de: P=49/262= 19%

Voici les résultats du premier corpus désordonné. On y observe 42 termes pôles et 57 classes. 24% des termes pôles n'appartiennent à aucunes classes. L'ensemble des classes ne contient pas 3 classes de référence: Fr (facteurs de risque), PHC (physiologie cardiovasculaire), PG (pathologie générale). La probabilité d'avoir un terme de la catégorie T: P=82/364=23%. La probabilité d'avoir un terme de la catégorie I: P=57/364= 16%. 43% des classes ont un paramètre de précision maximal entre 10 et 30 %. 12% des classes ont un paramètre de précision maximal supérieur à 50% des catégories T et I. Sur les 7 classes pertinentes 3 classes contiennent une forme variante d'un terme de la classe, 2 classes ont seulement 4 termes.

Voici les résultats du deuxième corpus désordonné. On y observe 53 termes pôles et 74 classes. 17% des termes pôles n'appartiennent à aucunes classes. L'ensemble des classes ne contient pas: PHC. La probabilité d'avoir un terme de la catégorie D: P=117/479=25%. La probabilité d'avoir un terme de la catégorie I: P=64/479= 13%. La probabilité d'avoir un terme de la catégorie T: P=64/479= 13%. 42% des classes ont un paramètre de précision maximal entre 10 et 30 %. 11% des classes ont un paramètre de précision maximal supérieur à 50% des catégories D, I, et T. Sur les 8 classes pertinentes 4 classes contiennent une forme variante d'un terme de la classe, 3 classes ont seulement 4 termes.

Voici les résultats du troisième corpus désordonné. On y observe 44 termes pôles et 56 classes 23% des termes pôles n'appartiennent à aucunes classes. L'ensemble des classes ne contient pas 2 classes: Fr, S (symptomatologie). La probabilité d'avoir un terme de la catégorie T: P=60/369=16%. La probabilité d'avoir un terme de la catégorie D: P=89/369= 24%. 43% des classes ont un paramètre de précision maximal entre 10 et 30 %. 16% des classes ont un paramètre de précision maximal supérieur à 50% des catégories D et T. Sur les 9 classes pertinentes 4 classes contiennent une forme variante d'un terme de la classe, 2 classes ont seulement 4 termes

En guise de commentaire, premièrement on observe que le traitement du corpus initial induit 2 fois plus de termes pôles différents (parmi les classes obtenues) qu'avec les corpus désordonnés. De ce fait la richesse est plus grande et la couverture est totale en ce sens que chaque catégorie de référence possède un représentant parmi les termes pôles. La figure 4.10 montre que les 3 corpus désordonnés donnent à peu près les mêmes résultats. On en déduit que l'ordre aléatoire des termes et des mots n'induit pas de différences dans les résultats entre corpus désordonnés. Ainsi une valeur moyenne peut couvrir les valeurs de chacun des corpus désordonnés vis-à-vis des intervalles de pmax. La proportion des classes de précision élevée (pmax > 50%) est réellement discriminante en faveur du traitement du corpus initial. La proportion des classes de valeur faible de pmax est sensiblement moindre pour le corpus initial.

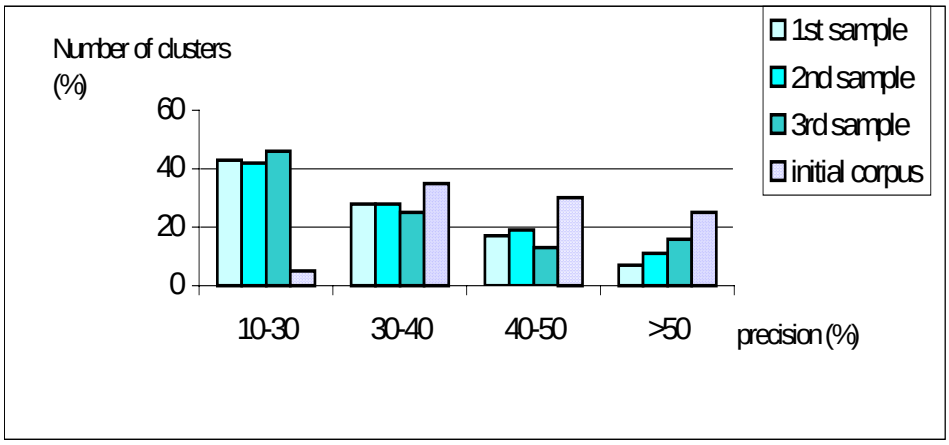


Figure 4.10 Tableau du nombre de classes par intervalle de pmax (corpus initial en pointillé).

On peut observer une corrélation entre la distribution des termes dans la hiérarchie de référence et la distribution des termes dans les classes du corpus désordonné. En effet, par exemple, la probabilité d'observer un terme de la catégorie "information du patient" dans la hiérarchie est de 16% et la probabilité moyenne d'observer le même type de terme dans les classes (du corpus désordonné) est de 14.5%. Ainsi une classe réunissant des termes à partir d'un corpus désordonné se comporte comme un objet qui extrait des termes d'une hiérarchie de référence avec une probabilité associée à la catégorie de hiérarchie.

3.5 Comparaison avec d'autres classifieurs

Il serait présomptueux d'évaluer un nouveau type de système sans faire intervenir des systèmes reposant sur des méthodes qui ont fait leur preuve. Ainsi 3 méthodes ont été comparées au vue de l'état de l'art des méthodes de classification automatique non supervisées du chapitre 2: la méthode des réseaux de neurones de Kohonen, la méthode de sériation ou des mots associés et la méthode hiérarchique agglomérative (distance euclidienne pondérée et lien complet). Ces méthodes ont été utilisées respectivement grâce aux logiciels: Neurotext™, Sampler™ et Tétralogie™.

Le principe d'évaluation est toujours le même [Turenne & Rousselot, 1998a]: nombre de classes ayant un pmax et un Tmax dans un intervalle d'histogramme. Nous avons synthétisé les résultats sur deux diagrammes, l'un représentant les valeurs de pmax (figure 4.11) et l'autre les valeurs de Tmax (figure 4.12). On représente aussi la somme en % (pour chaque méthode) des classes dont le résultat est > à 20%.

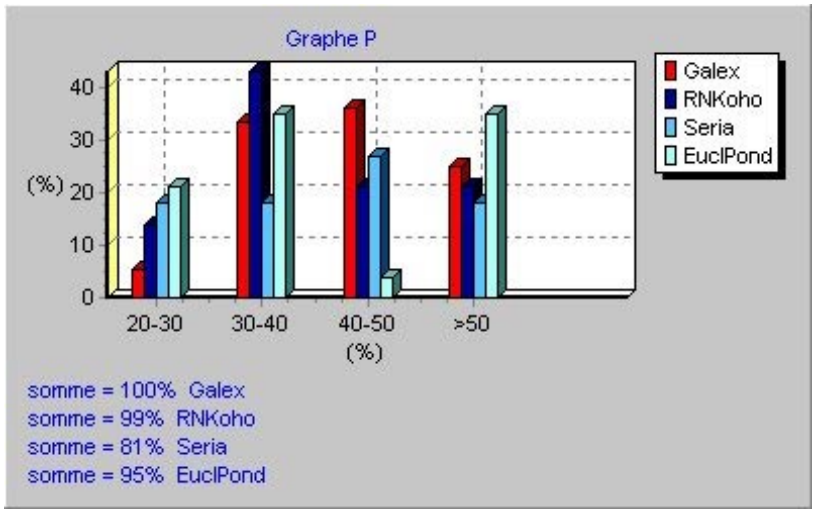


Figure 4.11 Nombre de classes par intervalle de pmax.

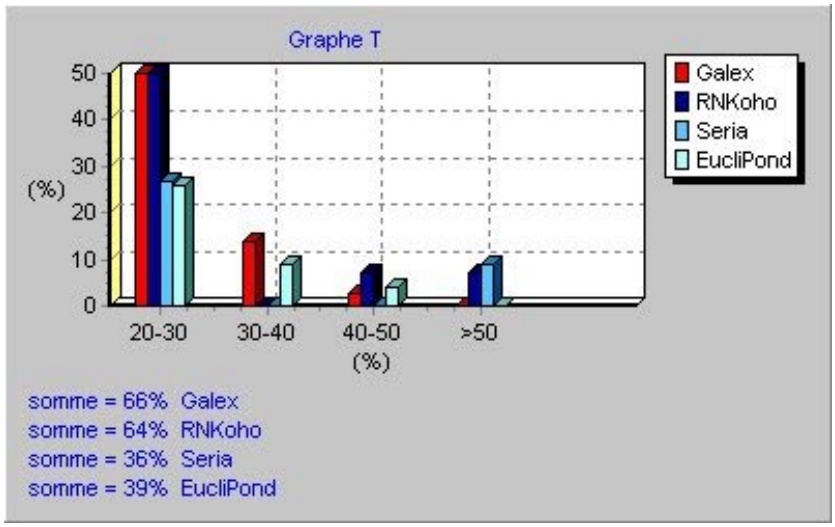


Figure 4.12 Nombre de classes par intervalle de Tmax.

L'interprétation doit être méticuleuse car les figures 4.13 et 4.14 présentent des résultats encore meilleurs avec un taux de bruit (pmax <20%) pour pmax réduit à 0 et des valeurs de Tmax à 40 ou plus de 50 %.

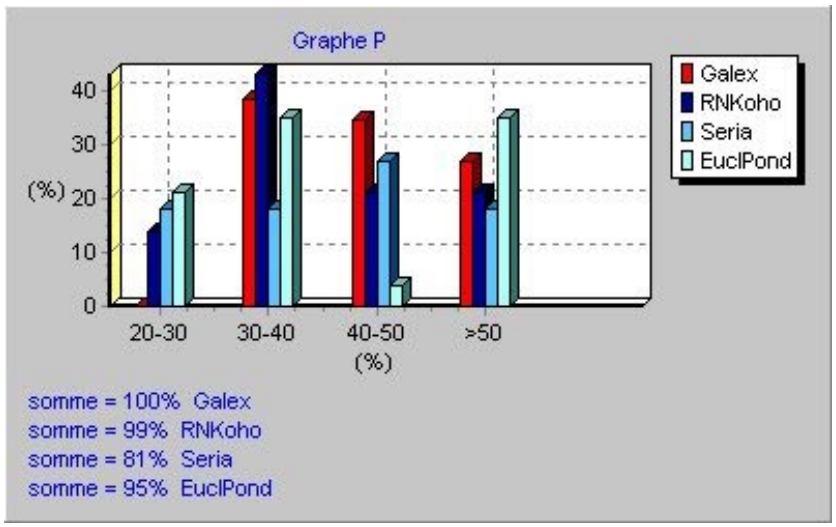


Figure 4.13 Nombre de classes par intervalle de Pmax.

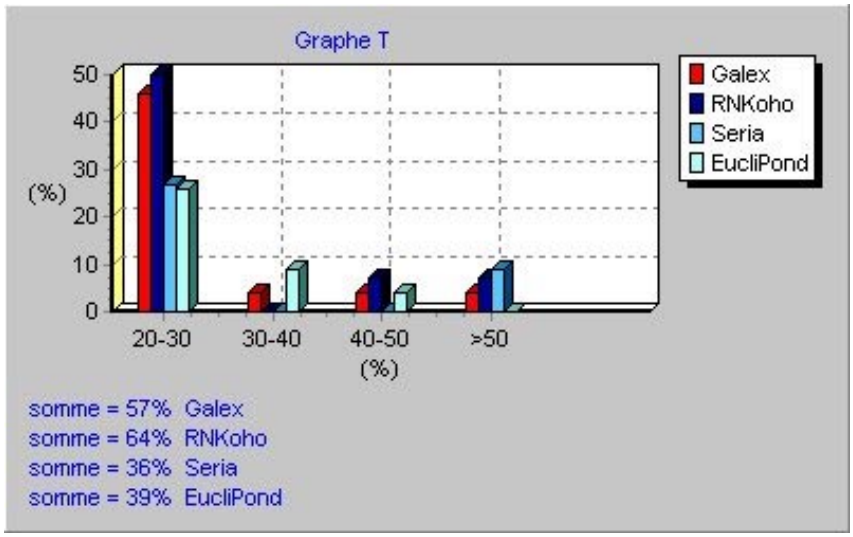


Figure 4.14 Nombre de classes par intervalle de Tmax.

La différence est que les résultats (en pourcentage) pour les figures 4.11 et 4.12 sont relatifs à plus de 40 classes alors que les résultats des figures 4.13 et 4.14 sont relatifs à 1 classe. Un lecteur non averti pourrait se méprendre quant à l'interprétation du pourcentage. Un pourcentage sur un nombre élevé de classes a davantage d'intérêt. A noter que pour les figures 4.12 et 4.13 le nombre de classes de Galex est 2 à 3 fois plus important que pour les autres méthodes. La couverture thématique sur les différentes catégories de référence est d'autant plus large et souhaitable.

Nous observons qu'une majorité de classes contient plus de 30% de termes d'une même catégorie mais rarement plus de 80%. Cela peut être expliqué par la construction de la classe qui regroupe deux termes pertinents de la même catégorie grâce à des termes typiques du contexte ; par exemple « lésion » et « anomalie » seront agrégés par « coronarographie » mais ensuite « coronarographie » sera gardé à l'intérieur du classe. A l'heure actuelle nous n'avons pas d'heuristique pour discriminer « coronarographie » et seulement garder « lésion » et « anomalie ». Les résultats de cette étude montrent la présence d'une très faible quantité de classes satisfaisant la contrainte T>40% . Seulement 1% des classes (1 classe) pouvaient valider cette contrainte, celui-ci étant lié à trois médicaments cités dans la même phrase plusieurs fois (comme une prescription médicale). Nos résultats semblent manifester un comportement équivalent à ceux des autres systèmes, de qualité décroissante quand T augmente. Mais la quantité de bons classes augmente. D'après notre méthode nous obtenons 15% de classes satisfaisant la contrainte T>30% .
Voici certains des meilleures classes :

T=0.43, thérapie/médicaments	Ergométrine, mg, voie intraveineuse, bolus.
T=0.42, anatomie cardiovasculaire	artère circumflexe, branche, circumflexe, distal, bas, marginal, artère gauche, paroi gauche, ventricule gauche.
T=0.41, thérapie/médicaments	Injection, atropine, mg, nitroglycérine.
T=0.40, anatomie cardiovasculaire	Inflation, territoire, territoire.Coronarographie (*artéfact de l'extraction de termes*), territoire latéral.
T=0.38, diagnostic	cine-ventriculographie gauche, cathéter, électro-cardiogramme, dérivation.

Nous avons observé qu'une majorité de classes contient moins de 80% de termes d'une même catégorie. Cette observation montre que la classe se décompose en deux composantes : la 1st

composante réunie des termes d'un catégorie principale et la 2nde composante réunie des termes d'autres catégories liées au contexte de la 1ère composante. Ainsi on compte 75 superclasses. Le terme pôle d'une superclasse a une certaine signification attachée à une certaine catégorie. Cela n'implique pas qu'une superclasse est liée au même thème mais se trouve décliné en une ou plusieurs catégories différentes ; ce phénomène caractérise une structure fine. Par exemple la superclasse suivante traite d'anatomie coronarienne (AC) et de pathologie coronarienne (PC) et possède son terme pôle liée à l'anatomie. On dégage ainsi la structure fine AC ↔ AC/PC :

artère coronaire, fréquence cardiaque, manifestation, myocarde, récent infarctus, infarctus du myocarde, maladies coronariennes, clinique, centre, degré;

artère coronaire, artère gauche, fait, cathéter;

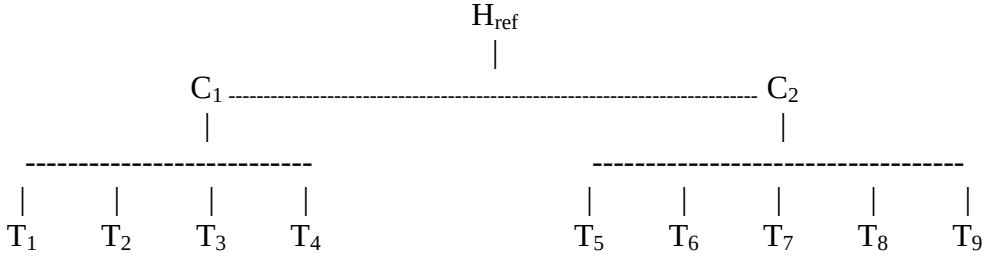
artère coronaire, artère gauche, branche postérolatérale, coronaire gauche, artère droite.

Nous observons une large présence d'instances liées à la pathologie dans 60% classes. Par conséquent certaines classes peuvent être bien définies par rapport à leur thématique comme dans l'exemple mentionné ci-dessus, une vue générale du contenu de ces classes est pathologie cardiovasculaire et plus particulièrement la pathologie analysable par coronarographie. En effet en on trouve 15 techniques utilisées en cardiologie, la coronarographie est une de celles-ci spécialement discutée dans le corpus médical.

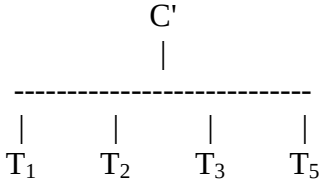
3.6 Calcul d'une mesure d'utilité

On cherche à évaluer une hiérarchie par une mesure d'utilité au sens COBWEB [Fisher, 1987]. Nous allons montrer que le calcul de cette utilité peut se ramener à calculer T_{max} défini en 3.2.

Soit H_{ref} une hiérarchie de concepts de référence $(C_1,C_2,\ldots,C_k,\ldots,C_n)$ modélisée de la façon suivante:



Soit le concept à évaluer C' tel que $C' \in H$ (H étant la hiérarchie à évaluer):



Prédictivité d'avoir les instances d'un concept C' liée à un C_k de Href (i.e. termes):

$$\Pi_k(C') = \frac{1}{card(C_k)} \sum_{i=1}^{card(C')} P(T_i|C_k)^2$$

Probabilité d'avoir le concept C' de H parmi les concepts de H_{ref} :

$$P(C') = P(C_k \mid C')$$

On calcule la prédictivité de C':

$$P(T_i|C_k) = \frac{1}{card(C')} \text{ si } T_i \text{ appartient à } C_k \text{ et à C', 0 sinon}$$

$$\sum_{1}^{card(C')} P(T_i|C_k)^2 = \frac{1}{card(C')} p(C_k)$$

où $p(C_k)$ est la précision $\frac{card(C') \cap card(C_k)}{card(C')}$

$$\Pi_k(C') = \frac{1}{card(C_k) card(C')} p(C')$$

finalement pour un concept C_i appartenant à H:

$$\Pi(C'_i) = \max(\Pi_k(C'_i))$$

Calculons maintenant $P(C'_i)$:

$$P(C'_i) = \frac{card(C'_i) \cap card(C_k)}{card(H)} \quad \text{sachant k pour } \Pi(C'_i)$$

$$PU(C'_i) = P(C'_i).\Pi(C'_i)$$

L'utilité de la hiérarchie H s'écrit :

$$PU(H) = \sum_{i=1}^{card(H)} PU(C'_i)$$

PU(H) se ramène donc à:

$$PU(H) = \sum_{i=1}^{card(H)} \left(\frac{card(C'_i) \cap card(C_k)}{card(H)} \bullet \max\left(\frac{p_k(C'_i)}{card(C_k).card(C'_i)}\right) \right)$$

en utilisant la précision maximale p_{max} et le rappel associé r de C'_i par rapport à C_k:

$$PU(H) = \frac{1}{card(H)} \bullet \sum_{i=1}^{card(H)} \left(\frac{r_k(C'_i) \bullet p_{max,k}(C'_i)}{card(C'_i)} \right)$$

on peut se ramener une somme de $T_{max,k}(C'_i)$ pour chaque C'_i , avec:

$$\tilde{pu}(H) = \frac{1}{card(H)} \bullet \sum_{i=1}^{card(H)} \left(\frac{T_{max,k}(C'_i)}{card(C'_i)} \right)$$

intervalle de réalisation de $\tilde{pu}(H)$:

$$\tilde{pu}(H_{ref}) = \frac{1}{card(H_{ref})} \bullet \sum_{i=1}^{card(H_{ref})} \left(\frac{1}{card(C'_i)} \right)$$

si H n'a aucun terme en commun avec Href alors pmax et r valent 0 donc $\tilde{pu}(H)$ vaut

$$0 \leq \tilde{pu}(H) \leq \tilde{pu}(H_{ref})$$

$$\tilde{pu}(H_{ref}) = \frac{1}{293} \bullet \left(\frac{1}{37} + \frac{1}{37} + \frac{1}{37} + \frac{1}{17} + \frac{1}{20} + \frac{1}{20} + \frac{1}{59} + \frac{1}{59} + \frac{1}{59} + \frac{1}{59} + \frac{1}{59} + \frac{1}{9} + \frac{1}{81} + \frac{1}{81} + \frac{1}{81} + \frac{1}{14} + \frac{1}{60} + \frac{1}{60} + \frac{1}{60} + \frac{1}{60} + \frac{1}{60}\right) = 0.00214$$

Le caractère numérique de $\tilde{pu}(H)$ n'étant pas référencé à un référentiel précis ou absolu on se ramène à des valeurs relatives (i.e. par rapport à $\tilde{pu}(H_{ref})$):

$$0 \leq \frac{\tilde{pu}(H)}{\tilde{pu}(H_{ref})} \leq 1 \quad .$$

$\tilde{pu}$ pour une hiérarchie aléatoire (que l'on appelera de MonteCarlo, H_{MC}), on estime à environ 7 en moyenne le nombre de termes par classe pour 40 classes en tout. Tmax sera de l'ordre de 0.1 (2 termes au plus d'une même catégorie) : $\frac{\tilde{pu}(H_{MC})}{\tilde{pu}(H_{ref})} = 13.7\%$

$\tilde{pu}$ pour une hiérarchie établit par sériation (mots asscociés, 89 termes 11 classes) :

$$\frac{\tilde{pu}(H_S)}{\tilde{pu}(H_{ref})} = 27.8\%$$

$\tilde{pu}$ pour une hiérarchie tablie par réseau de Kohonen (83 termes, 14 classes):

$$\frac{\tilde{pu}(H_K)}{\tilde{pu}(H_{ref})} = 51.3\%$$

si on enlève 2 classes de 3 termes qui ont des termes de médicaments (qui se suivent dans le corpus) : $\frac{\tilde{pu}(H_K)}{\tilde{pu}(H_{ref})} = 39.9\%$

$\tilde{pu}$ pour une hiérarchie établit par CAH (250 termes, 24 classes) :

$$\frac{\tilde{pu}(H_{CAH})}{\tilde{pu}(H_{ref})} = 19.6\%$$

$\tilde{pu}$ pour GALEX (300 termes, 44 classes) :

$$\frac{\tilde{pu}(H_{GALEX})}{\tilde{pu}(H_{ref})} = 36.7\%$$

On voit que les réseaux de Kohonen obtiennent le meilleur score tout en ayant des effectifs nettement plus faibles (termes classés, classes) et des classes correctes de taille très petite. Il faut aussi noter que la mesure peut changer d'une hiérarchie de référence à l'autre pour peu que celle-ci varie en taille. Dans ce cas étudié, d'une manière générale on remarque que les méthodes de classification ont des difficultés à retrouver les classes de la hiérarchie de référence avec un taux de réussite acceptable.

4 Visualisation et retour aux données sources

4.1 L'interface de visualisation

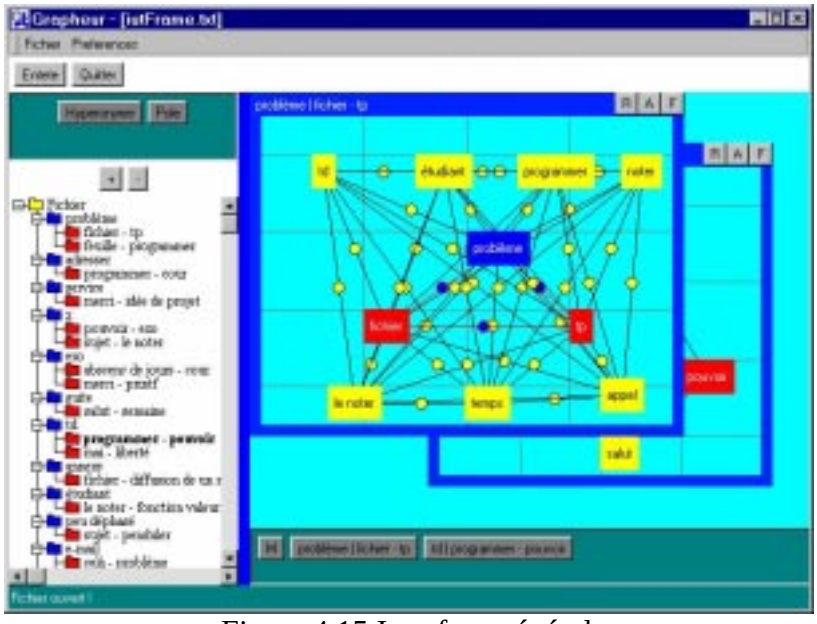


Figure 4.15 Interface générale

Le chargement de données peut être fait en ligne de commande en précisant le nom du fichier, en utilisant « Ouvrir » dans le menu « Fichier » ou encore en faisant appel au constructeur de l'application avec les données sous forme de chaîne depuis une autre application Java. L'interface se présente comme suit (Annexe 12):

Tout d'abord, nous trouvons à gauche l'un des deux arbres permettant la navigation dans les données (figure 4.15).

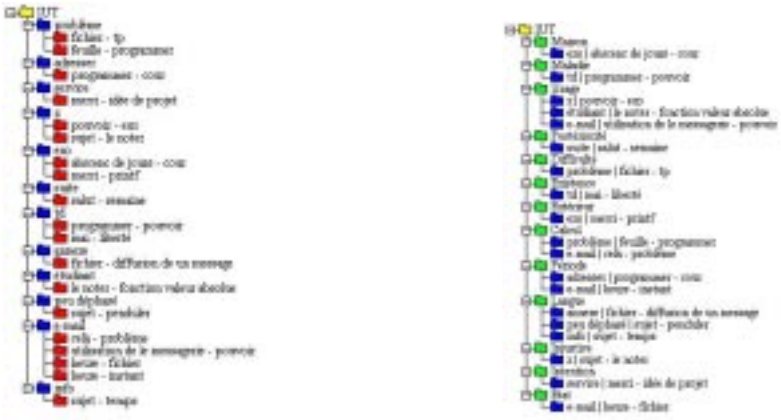


Figure 4.16 Vues de l'arbre

Pour passer à l'autre arbre, il suffit de cliquer sur les boutons au-dessus du cadre de l'arbre qui donnent accès aux deux onglets représentant la classification par pôles et la classification par hyperonymes (figure 4.16).

On peut ainsi faire la recherche à partir du classement par pôles ou du classement par hyperonyme. Lorsqu'on a trouvé la classe que l'on souhaite, il suffit de cliquer sur l'arbre pour l'afficher dans une nouvelle fenêtre fille MDI à droite. En effet, on peut visualiser simultanément plusieurs classes simultanément. Ceci est possible grâce au système de fenêtres MDI (figure 4.17).

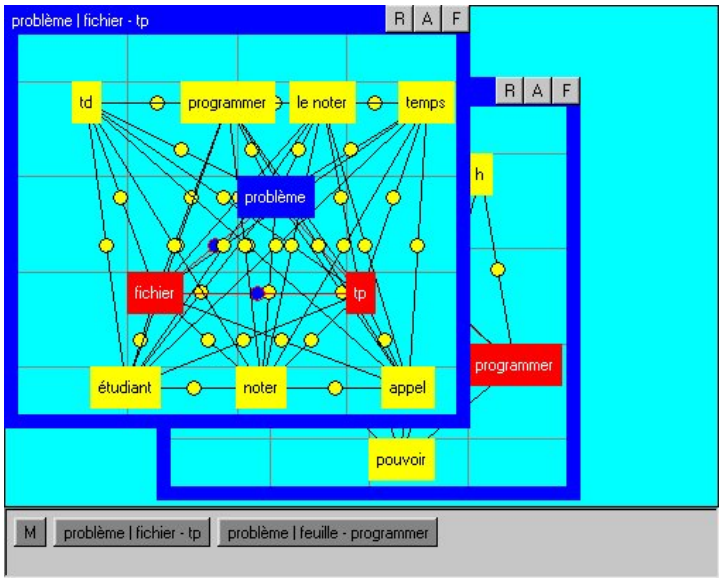


Figure 4.17 Panel d'une classe

On peut ouvrir plusieurs fenêtres en même temps, les faire se chevaucher, se recouvrir ou en placer une qui occupe toute la surface, masquant les autres. Pour chaque fenêtre, un bouton viendra se placer dans la barre de boutons en bas, permettant d'afficher une fenêtre que l'on ne voit pas ou de restaurer une fenêtre réduite. Le bouton M permet un affichage des fenêtres ouvertes en mosaïque (figure 4.18).

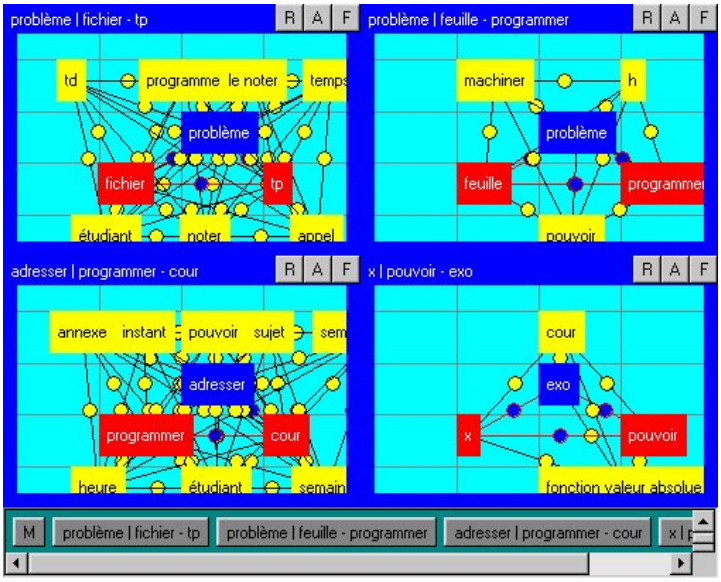


Figure 4.18 Vue en mosaïque

Maintenant, concentrons-nous sur une classe. Toute classe s'affiche dans un panel spécialement conçu à cet effet (figure 4.19).

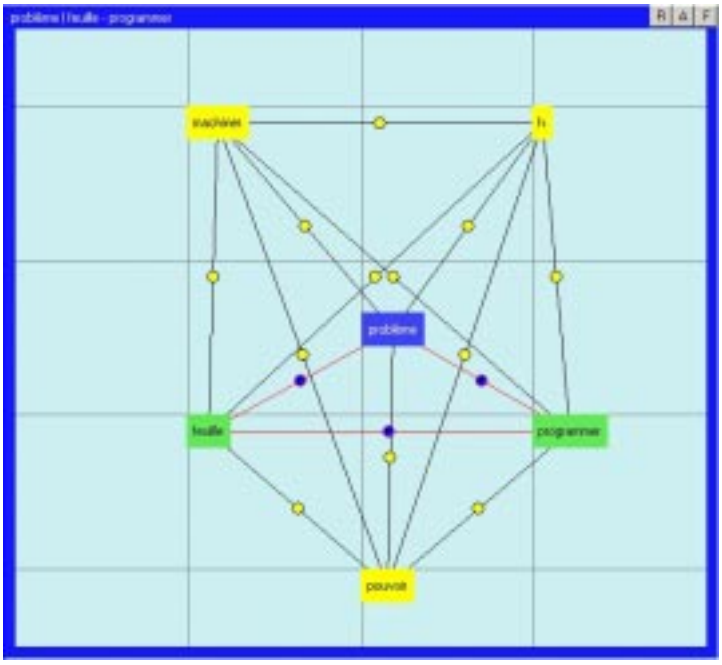


Figure 4.19 Graphe d'une classe

Nous avons ici un graphe de classe avec 6 termes ou nœuds et autant de relations d'un terme à un autre qu'il n'y a de ligne sur le graphe. L'application permet de différencier le pôle (en bleu), les pivots (en vert) et les autres termes (en jaune). On peut aussi remarquer la différence entre les relations importantes liant le pôle aux pivots et les autres relations liant

l'un de ces trois à un autre terme ou deux autres termes entre eux. Maintenant approchons le curseur d'un nœud. On peut constater que le curseur change d'aspect, indiquant la possibilité d'actions sur ce nœud. La première action par *glisser-déposer* de souris (angl., drag and drop) permet de déplacer le nœud sur le graphique. Cela permet de réorganiser celui-ci à volonté, mais aussi d'afficher des relations non visibles qui peuvent être masquées par le nœud dans la présentation par défaut. Par clic sur le nœud, on accède à deux fonctions relatives à celui-ci via un menu flottant (figure 4.20).

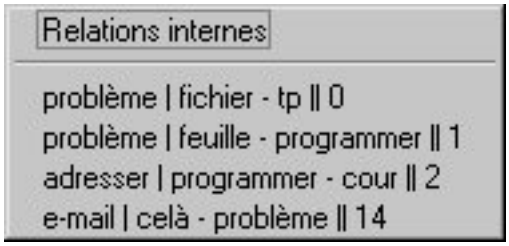


Figure 4.20 Menu déroulant (angl., popup) concernant les propriétés d'un terme

Premièrement, nous pouvons visualiser les relations liant le nœud à lui-même qui n'apparaissent pas sur le graphe pour ne pas le surcharger (ici pour le nœud « fichier ») (figure 4.21).

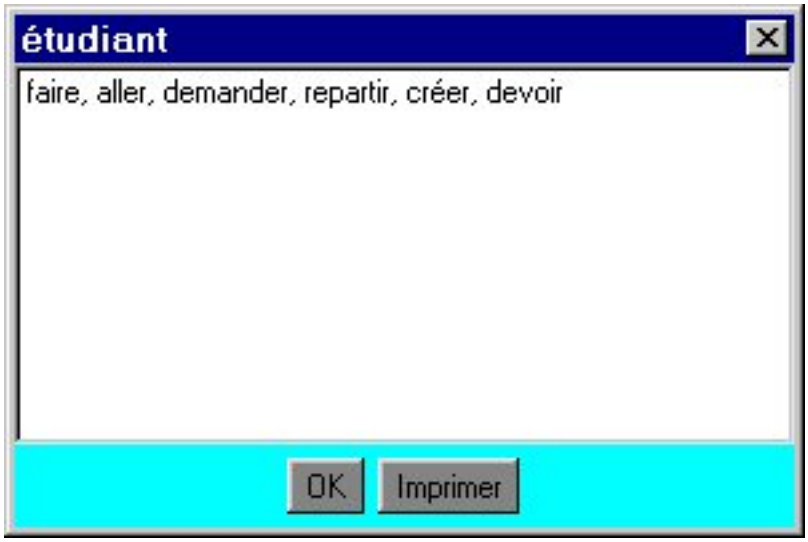


Figure 4.21 Attributs d'un noeud

Deuxièmement, nous pouvons accéder directement, sans passer par l'arbre, aux autres classes contenant le terme. Ceux-ci sont indiqués dans le menu flottant par le terme pôle et les pivots. Maintenant, voyons les relations. Chaque relation est caractérisée par un segment de droite reliant deux nœuds sur le milieu duquel se greffe une petite boule (figure 4.22).

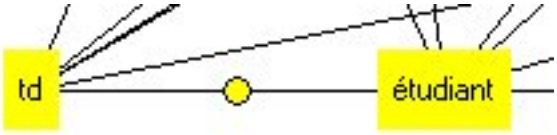


Figure 4.22 Boule identifiant les attributs d'une relation

Lorsque l'on place le curseur sur la boule, celui-ci change ici aussi d'aspect, indiquant qu'on peut agir sur la relation. On ne peut évidemment pas la déplacer, mais on peut visualiser son

contenu en cliquant dessus. Une bulle s'ouvre pour afficher les verbes de la relations s'il y en a (figure 4.23).

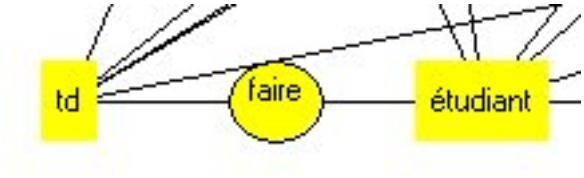


Figure 4.23 Boule ouverte

Le fait de relâcher le bouton de la souris masque à nouveau la bulle.
Cependant, les actions sur la classe ne se limitent pas à cela. Par un clic droit sur une zone vide du panel, on peut accéder à trois fonctions (figure 4.24).

Figure 4.24 Menu déroulant concernant les propriétés d'une classe

La première est une fonction de statistiques. Dans une fenêtre apparaissent les deux hyperonymes et la liste des verbes avec leur répétition par ordre décroissant de répétition (figure 4.25).

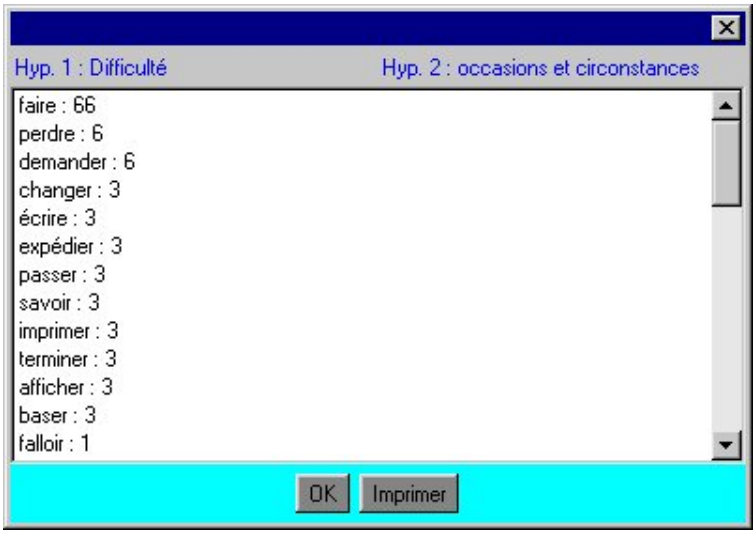


Figure 4.25 Statistiques sur les attributs d'une classe

Les deux autres options permettent la réorganisation des nœuds sur le panel et son impression.

Si l'on veut voir l'entête du fichier, il suffit de cliquer sur « Entête » dans la barre de boutons (figure 4.26).

Figure 4.26 Hyperonymes caractérisant le corpus

Enfin, une barre d'état donne différentes informations à l'utilisateur sur la progression des traitements et sur les erreurs.

4.2 La structure de l'interface graphique

L'application s'articule autour de 3 parties ou trois couches :
affichage, données, entrées / sorties (figure 4.27).

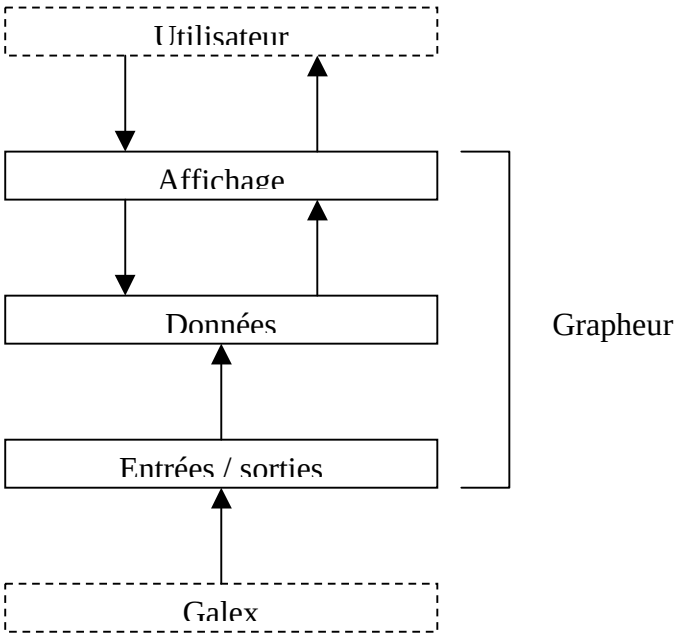


Figure 4.27 Couches de l'application

Premièrement, une couche entrées-sorties opérant l'interprétation du fichier issu du traitement par l'application en amont dans la chaîne, découpage en classes, puis recherche des nœuds et des relations dans chaque classe. Cette opération permet de générer les données qui sont enregistrées dans un fichier qui est ensuite relu par mon application.
Le fichier source contient le découpage en classes, pour chaque classe deux lignes décrivant les deux hyperonymes puis la liste des relations entre les termes du classe. On n'a pas explicitement pour chaque classe la liste des termes. Celle-ci doit être extraite de la liste des relations en comparant les deux membres de la relation à la liste des termes que l'on a déjà.

Hormis le fait que les fichiers sources soient moins volumineux, nous n'avons ici pas de problèmes d'intégrité d'une classe. Il ne peut en effet exister de relation mettant en jeu un terme qui n'existe pas dans cette classe-là puisque ce sont les relations qui déterminent les nœuds dans le fichier source.
Néanmoins, l'application doit être en mesure de lire des fichiers relativement volumineux. Lors des tests, nous avons passé en entrée un fichier d'1,4 Mo. Le plus long dans le traitement reste le chargement du fichier sur le disque. L'étape de découpage du fichier, bien que relativement lente est plus rapide encore.
En sortie de l'étape de lecture et découpage du fichier, nous obtenons les données qui pourront être traitées par l'application.
Les données sont stockées selon le principe de la figure 4.28.

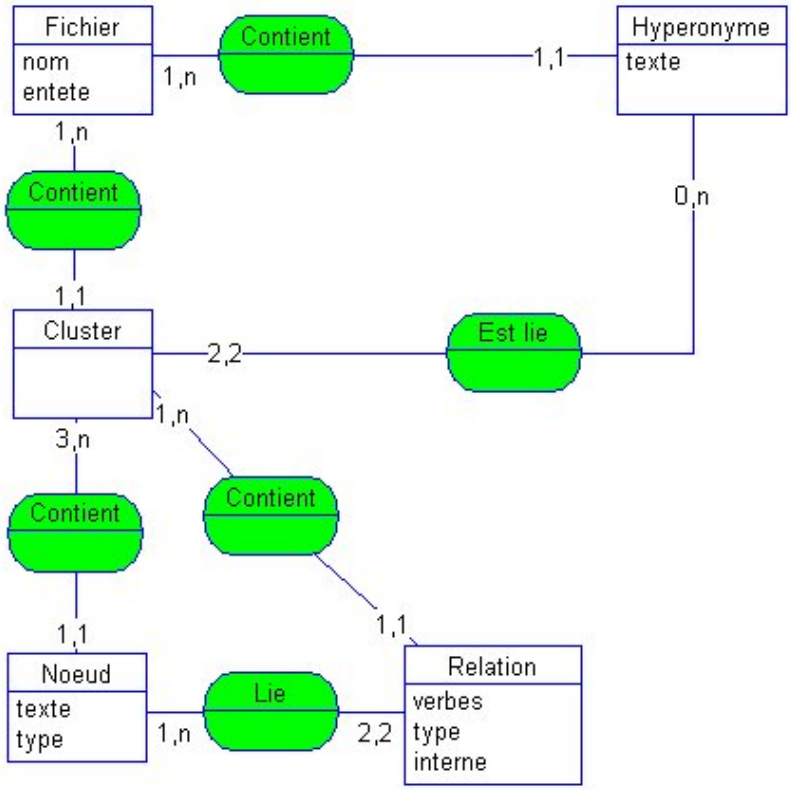


Figure 4.28 Schéma conceptuel du fichier d'entrée stocké en mémoire

Nous avons d'abord l'entité Fichier qui représente en mémoire le fichier que l'on vient de charger du disque, donc le profil choisi dans l'application mère et contient donc une liste de classes ainsi que l'entête du fichier. De même, il contient la liste des hyperonymes.
Chaque classe est liée à deux hyperonymes.

C'est à partir de ces données aussi que peuvent être générés les arbres de navigation.

La couche affichage comprend tous les objets visuels. Les plus importants sont les pendants graphiques des données, permettant tant la visualisation des graphes que leur réorganisation. Mais il faut ajouter les différentes fenêtres qui peuvent apparaître. Chaque composant graphique représentant une donnée se limite à être une représentation avec une référence sur la donnée et ne contient en aucun cas une copie de celle-ci. Mais fournissant plus qu'une simple visualisation interactive des données, la couche affichage offre aussi les possibilités

d'afficher des statistiques sur celles-ci. Les options d'impression sont aussi comprises dans cette partie.

4.3 Le retour aux données sources

On choisit arbitrairement une classe et on extrait les concordances termes/verbes à partir de l'interface graphique. Nous essayons d'expliquer pourquoi les liens termes/verbes entraînent une disparité de composition d'une classe par rapport à la catégorie majoritaire qu'elle représente.

Voici une classe de 6 termes:

nécroser	(terme pôle)
interventriculaire	(terme pivot)
symptomatologie	(terme pivot)
douleurs	
resténose	
spasme	

Ses paramètres d'évaluation donnent Tmax = 0.22 et Pmax = 3/6 =0.5 . Les termes nécrose, resténose et spasme lient majoritairement la classe à la catégorie Pathologie Cardiovasculaire. 104 contextes sont à l'origine de la création de cette classe. Parmi ces 104 contextes 60 sont uniques (annexe 9).

16 contextes font intervenir le terme *symptomatologie* avec les verbes relationnels: survenir, associer, expliquer, proposer, constituer, persister
symptomatologie est sujet de *survenir* et objet *de associer*

18 contextes font intervenir le terme *interventriculaire* avec les verbes relationnels: associer, proposer, constituer, justifier, compliquer, persister, hospitaliser, mettre
interventriculaire est sujet de *associer* et objet de *proposer* et complément d'objet de *constituer* et complément de subordonnée de verbe principal *constituer*,complément d'objet de *persister*

Parmi les autres termes on identifie les relations suivantes:
nécrose est objet de *constituer*
resténose est sujet d'une subordonnée d'une phrase principale du verbe *persister*, sujet de *constituer*, complément d'objet de *hospitaliser* et objet de *mettre* et *compliquer*
douleurs est sujet de *expliquer* et objet de *survenir* et de *justifier*
spasme est complément de *survenir*

On constate que les termes *symptomatologie* et *interventriculaire* ont des relations syntaxiques marquées avec certains verbes relationnels qui induisent l'appartenance des autres termes à la classe. On ne peut donc pas les différencier ni grâce aux verbes relationnels ni grâce à la nature syntaxique que jouent ces relations. Cela est dû à la nature non-exclusive de l'usage des verbes utilisés comme: constituer, survenir, associer. On pourrait supposer que la nature du terme comme terme pivot entraîne une dépendance catégorique marginale. Mais en observant les classes suivantes, on s'aperçoit qu'un terme pivot peut lui-même appartenir à la catégorie majoritaire:

> Classe de catégorie Diagnotic (12 termes, 662 contextes)
> valeur pronostique (terme pôle), branche (terme pivot), échographie ne montre (terme pivot), tronc, test, effort, réseau, postérolateral, épreuve d'effort, paroi, nouveau test, modification.

Classe de catégorie Anatomie Cardiovasculaire (5 termes, 245 contextes)
sévère du tronc commun (terme pôle), porteur d'infarctus du myocarde (terme pivot), tronc (terme pivot), tronc commun, coronaire.

5 Perpectives

5.1 Plan technique

Pour des raisons de maintenance comme la mise à jour et la localisation, il serait judicieux de regrouper les ressources linguistiques dans une base de données comprenant plusieurs tables (table du thésaurus, table du dictionnaire accentué, table du dictionnaire accentué/non accentué, table des mots vides, tables des suffixes). Ces tables seraient d'ailleurs duplicables dans plusieurs langues notamment l'anglais.

5.2 Plan théorique

Dans l'algorithme présenté au §2.2 l'étiquette sélectionnée par consensus concerne une classe et non une super-classe. Il seraient intéressant de tester l'attribution d'une étiquette collective à plusieurs classes ayant le même terme pôle sachant qu'un nombre plus conséquent de termes facilite le consensus.
L'étiquetage n'est pas performant à 100% surtout dans le cas d'un composition de la classe par des terme généraux, de monotermes et l'absence de consensus (codes de fréquence 1 y compris ceux du terme pôle). L'attribution du terme pôle comme étiquette serait peut-être plus avantageuse dans le cas d'une indécidabilité. Il faudrait dans ce cas avoir une mixité de l'étiquetage partagée entre le terme pôle et la catégorie du thésaurus.

6 Travaux antérieurs

La cohésion lexicale est généralement traitée comme la projection d'un thésaurus sur un texte plutôt que comme la projection d'un résultat sur un thésaurus.

[Sussna, 1995] établit une distance entre un mot d'une requête et un document. Il établit des poids à chaque nœud de *Wordnet*. Une distance est développée entre les mots d'un document et la requête. Cette distance se base sur un calcul de profondeur d'un mot dans *Wordnet*. On somme les produits de profondeur d'un mot de la requête et de profondeur d'un mot du document par rapport à chaque mot de *Wordnet*. On associe un centroïde au minimum de la somme des distances entre les mots de la requête et mots d'un document. Tous les documents proches de ce centroïde forment une classe qui est renvoyée comme réponse à la requête.

[Sparck-Jones, 1987[1967]] utilise le *Roget* qui est le thesaurus, de langue anglaise, équivalent au thesaurus que nous utilisons. La problématique de [Sparck-Jones, 1987[1967]] est la reconstitution. En mélangeant des termes de plusieurs catégories, est-il possible de retrouver ces catégories grâce à la morphologie des termes.

[Elman, 2000] établit une statistique de cohésion lexicale centrée autour de relations du *Roget*. Il se sert de la mesure de cohésion lexicale pour différencier des genres littéraires. La cohésion lexicale se définit entre 2 documents contenant la même proportion de liens lexicaux du *Roget*. Ces liens de cohésion sont propres à la structure du thésaurus (1er niveau de généralisation, un 2^{ième} niveau de généralisation, 1 relation d'équivalence intra-classe). Pour une fénêtre de mots donnée ("chaîne lexicale") que l'on balaye du début à la fin d'un texte on recherche les paires de mots qui font partie de la même sous-partie du thésaurus (terme équivalent), de la même catégorie et du même groupe de catégories.

[Loukachevitch, 2000] utilise un thésaurus sur la sociopolitique pour la détection de relations de cohésion lexicale dans les textes. Le thésaurus comprend des relations de : hyperonymie, hyponymie, partie-tout et association. L'inférence de relations de cohésion lexicale est basée sur des relations conceptuelles telles que la transitivité, l'héritage et la symétrie.
(exemple de relation d'héritage conceptuel: *héritage association/hyperonyme→association*, i.e. or/Hyperonyme-métal précieux/Hyperonyme-métaux/Hyperonyme-dépôt/Association). Un texte est projeté (étiqueté) sur le thésaurus pour en dégager un réseau de concepts reliés par les relations du thésaurus. Les relations du réseau obtenu sont qualifiées de relations de cohésion lexicale.

Résumé du chapitre

Dans ce chapitre nous avons présenté une stratégie globale d'évaluation utilisant une hiérarchie de référence développée sur 3 niveaux et permettant d'analyser la cohésion d'une classe de termes. Cette cohésion lexicale se base sur la généralisation de contextes d'usage en commun.

Une première stratégie met en oeuvre un consensus pour décider quelle est l'étiquette sémantique qui se rattache le plus à l'ensemble des termes d'une classe donnée. Cette stratégie exploite un thésaurus général, et s'applique aussi à l'ensemble des classes pour décider quelle étiquette donner au corpus. Cette stratégie permet de savoir si une classe ou le corpus en entier peut être couvert par une étiquette générale identifiant un champ sémantique. Les expériences montrent une difficulté pour étiqueter des classes émanant d'un corpus de courriers électroniques. Les syntagmes ne sont pas référencés dans le thésaurus ou ne recouvrent pas de contextes d'usage en commun. En revanche les corpus de texte encyclopédique sont plus prolifiques avec un taux de 70% d'affectations satisfaisantes. L'attribution d'une étiquette au corpus marche à 100% quel que soit le corpus.

Une autre stratégie exploite une hiérarchie conceptuelle établie d'après des connaissances expertes du domaine. La cohésion lexicale se ramène à rapprocher une classe donnée d'une classe de référence ayant la plus grande proximité. Deux critères principaux sont utilisés: la précision (p) et p couplé avec le rappel (T). On essaie de comparer le comportement de p et T dans un histogramme d'intervalles de valeurs maximales de p ou T de chaque classe. Par exemple on relève le pourcentage u, par rapport au nombre total de classes, des valeurs de p ou T comprises entre 20 et 30 %. Cela signifie que u% des classes ont une valeur maximale de p ou T entre 20 et 30 %. Trois tests sont effectués. Un premier test fait varier les paramètres de Galex pour relever les valeurs les plus intéressantes. Les histogrammes donnent les valeurs optimales: taille de fenêtre = [9-15] termes, nombre de termes en commun = 1 terme, nombre maximal de termes par classe = [10-12] termes. Ce test permet de stabiliser les paramètres du classifieur. Un deuxième test implémente une analyse par MonteCarlo en désordonnant le corpus. La comparaison de l'histogramme des données aléatoires et des données sources montrent un nombre de classes homogènes des données sources nettement supérieur à celui obtenu avec les données aléatoires. Ce test valide l'hypothèse d'analyser les cooccurrences dont l'ordre non aléatoire favorise les associations conceptuelles. Un troisième test compare le comportement des paramètres avec la méthode de notre classifieur et 3 autres méthodes de l'état de l'art: le réseau de Kohonen, la CAH pondérée, la méthode de sériation. Les résultats montrent un taux de classes ayant un $p > 30\%$ supérieur aux autres méthodes et un taux de $p < 20\%$ très inférieur aux autres méthodes. Si 35% des classes ont un taux de précision entre 50 % et 80 %, l'homogénéité reste incomplète. Ce test montre que Galex présente un efficacité par rapport aux autres méthodes, et que les classes présentent une mixité conceptuelle avec une dominante.

CHAPITRE 5

SYSTEME DE FILTRAGE INTELLIGENT DE MESSAGES ELECTRONIQUES

Dans ce 5ème et dernier chapitre nous présentons une application finale exploitant le module de classification. Ce chapitre met en œuvre, en quelque sorte, un type d'évaluation basé sur l'utilisation efficace d'un résultat intermédiaire. En l'occurrence ce résultat intermédiaire est la taxinomie terminologique obtenue aux chapitres précédents. La taxinomie est la pierre d'angle de l'application qui doit démontrer son efficacité par rapport aux méthodes traditionnelles.

L'application met en œuvre l'algorithme présenté au §3 pour déterminer le profil d'un utilisateur à partir d'une base documentaire relatant ses centres d'intérêts thématiques. Nous avons choisi un cadre de traitement d'information dynamique encore mal étudié et prometteur en terme d'analyse de contenu: le filtrage de messages électroniques.

La classification automatique de termes s'interprète comme une partie du modèle de l'utilisateur impliquant aussi le modèle du domaine (application de messagerie). 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 mot-clés (moins d'un dizaine en général) recouvrant faiblement la thématique correspondante. Nous centrons l'analyse du profil sur la réutilisabilité de classes de termes servant de filtre pour retenir des nouveaux messages. Chaque classeur de la messagerie entraîne la génération d'un profil propre. Le nouveau message est confronté à chaque profil. L'ensemble des profils de classeurs constitue le profil de l'utilisateur. Une partie des classes de termes agit comme règle de déduction pour décider le transfert d'un message dans le classeur d'origine de la création des classes de termes.

Une évaluation basée sur les notions classiques de rappel et de précision permet de qualifier la performance des classes de termes dans une tâche de filtrage. Une fonction de gain (angl. "utility function") tient compte des documents filtrés pertinents et non pertinents. L'évaluation opère un apprentissage incrémental par 2 jeux de test. Chaque jeu est composé d'une partie apprentissage (messages connus) et d'un partie de messages tests (messages inconnus). Le 1er jeu contient 60% de messages de la totalité des messages pour réaliser l'apprentissage tandis que le 2nd jeu admet 80% de la totalité des messages. Les 2 jeux de tailles différentes doivent permettre d'étudier l'adaptativité de l'apprentissage. La profil est formé de 3 classeurs, sources de 3 centres d'intérêts, qui seront la cible du transfert d'un message de l'échantillon de test.

1 La messagerie électronique

1.1 Historique

Le courrier électronique possède une double facette:
- échange de messages courts.
- échange de documents courts (<1 Mo).

Quelques chiffres pour mettre en lumière l'importance du nouveau media de communication que représente la messagerie électronique [Rapp, 1998].

En 1980	400 000 utilisateurs
En 1998	80 millions d'utilisateurs de messagerie (20 millions en Europe)
	3 milliards de courriers électroniques échangés chaque mois

Les premières normes techniques d'acheminement de paquets de données remontent à la création d'Internet c'est-à-dire au réseau universitaire américain Arpanet en 1970. En 1992 la société américaine Sprint commercialise le premier système d'accès à Internet. Actuellement les normes définissant les protocoles techniques de codage sont régies par des RFC (Request For Comments).

La messagerie électronique est devenue l'application majeure du réseau aussi bien intranet[7] qu'internet. Ce service de messagerie doit intéresser les entreprises pour 3 raisons :
- le service est asynchrone (le destinataire n'est pas obligé d'être là pour recevoir un message, contrairement au téléphone par exemple).
- le coût est faible puisque le réseau physique est déjà en place.
- les documents ou messages échangés sont réutilisables.

Ces raisons associées avec l'essor du commerce électronique place la messagerie électronique dans une position d'outil ordinaire de communication pour les années à venir.

La messagerie a démontré ces dernières années qu'elle pouvait apporter 2 inconvénients importants. Le premier concerne ce qu'on appelle un "spam" c'est-à-dire un message non désiré et souvent indésirable. Leurs diffuseurs ("spammers") ont repéré des adresses électroniques dans des forums de discussion ou sur des listes de diffusion. Ils inondent plusieurs boîtes aux lettres par des messages en général publicitaires. Le second est la propagation d'un virus électronique. La messagerie ne faisant que transférer de l'information elle ne peut en aucun cas exécuter un virus à l'insu du propriétaire d'un ordinateur ou d'un réseau. Le responsable est l'utilisateur qui lit un document attaché au message sans précaution, et l'application permettant de lire un message. En effet cette dernière propose des fonctionnalités confortables de lecture d'un document grâce à son extension par exemple (*ps*, *doc*, *vbs*, *jpeg*,…etc.).

1.2 Description fonctionnelle

L'acheminement d'un message est réalisé au moyen d'une carte réseau (ou modem) sur le port 21. Le port correspond aux fiches de connexion qui se trouvent sur la carte mère d'un ordinateur. La messagerie exige une machine en connexion permanente comme une station de travail unix par exemple (mode réseau). Dans le cas contraire, si les messages sont écrits et lus sur une machine ponctuellement connectée (par modem, ou un PC en général), un serveur de messagerie doit être mis en place (mode modem). Dans ce cas un client de consultation

[7] Réseau uilisant les protocoles d'internet ,i.e. TCP/IP, mais coupé d'Internet qui est l'interconnexion des réseaux locaux

doit être installé sur la machine de lecture/écriture des messages. L'envoi d'un message vers une machine distante est assuré par un programme standard *SendMail* respectant les protocoles standards (voir http://www.sendmail.org).

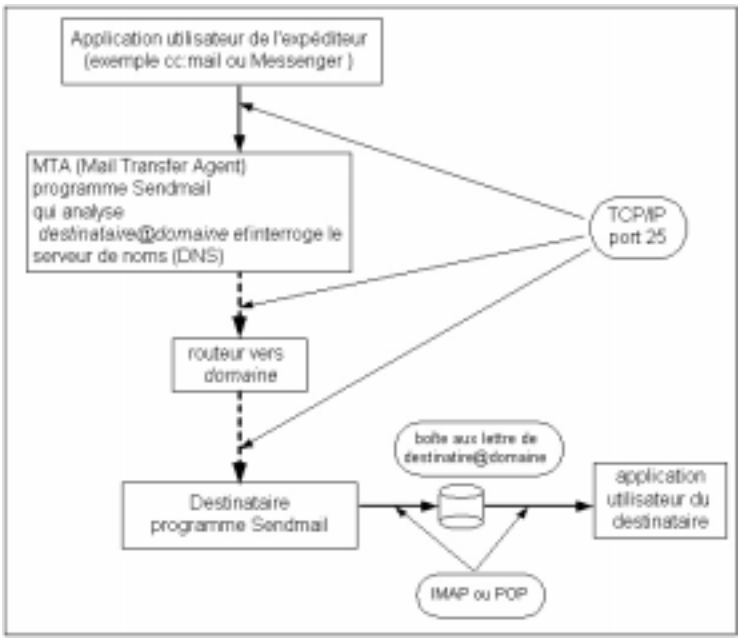


Figure 5.1 Architecture fonctionnelle de la messagerie.

Des clients autonomes (commerciaux) existent et sont assez répandus: Eudora™ (Eudora), Outlook™ (Microsoft), Messenger™ (Netscape), LotusMail™ (IBM). Des opérateurs Internet comme Hotmail ou Bigfoot propose une gestion de la messagerie via une URL et un navigateur courant (Navigator™ de Netscape ou InternetExplorer™ de Microsoft). On trouve des serveurs de messagerie sur le marché comme : LotusDomino™ (IBM), Post-Office™ (NetMessage), SolsticeInternetMail™ (Sun), AltavistaMail™ (Digital), MailServer™ (Netscape), N-Plex™ (Isocor).

1.3 Protocoles et codages

SMTP (Simple Mail Transfer Protocol) fonctionne comme une passerelle et permet une interface entre un réseau local et le réseau Internet. SMTP est défini par la RFC-822.
Le protocole SMTP autorise le transport de messages et peut être appliqué à une diversité de protocoles de réseaux, notamment TCP/IP qui sont les standards des réseaux locaux de transmission. Un réseau doté de SMTP peut être considéré comme standard et communiquer avec d'autres réseaux.

POP3 (Post Office Protocol version 3) est un protocole de lecture à distance des messages stockés sur le serveur de messagerie. POP3 est défini par la RFC-1725.
POP3 permet à une application d'accéder dynamiquement à un service de messagerie où les messages sont stockés sur un serveur hôte. Les messages sont ensuite rapatriés sur la machine cliente.

IMAP4 (Internet Message Access Protocol version 4) est similaire à POP3 mais permet une gestion souple de la messagerie. IMAP4 est défini par la RFC-2060.
Il offre notamment 3 modes de fonctionnement différents:

- mode déconnecté: les correspondances électroniques sont automatiquement renvoyées sur le poste de chaque utilisateur
- mode connecté: le serveur de connexion permanente reçoit et stocke tous les messages qui arrive à destination des utilisateurs du réseau.
- mode autonome: combine les avantages des 2 précédents, chaque utilisateur dispose d'une boîte aux lettres sur le serveur et sélectionne au coup par coup les messages qu'il veut transférer sur son poste. Ce mode est intéressant pour des clients nomades, les messages restent sur le serveur.

MIME (Multipurpose Internet Mail Extension) est une norme technique pour les échanges de données multimédias avec une représentation codée du jeu de caractères et des attachements. MIME est défini par la RFC-2045.
Concerne le codage des caractères et notamment des accents qui sont codés sur 8 bits; sur certaines plate-formes les accents sont codés d'une certaine façon (par exemple uuencode pour Unix) il est alors recommandé d'éviter les accents.

1.4 Architecture physique de notre messagerie

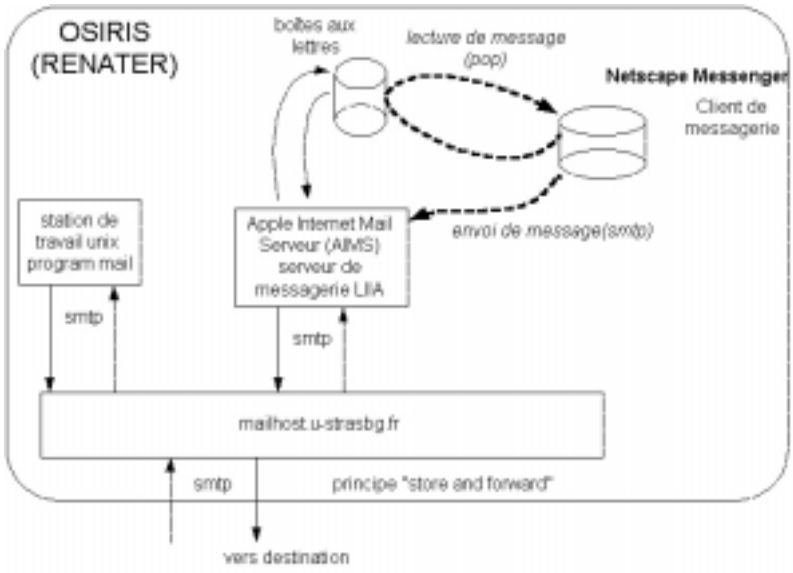


Figure 5.2 Architecture de la messagerie.

Les accès à un serveur de messagerie obéissent aux règles du client/serveur (figure 5.1). Une requête d'un client est envoyée à un serveur avec notamment des informations de connexion et d'autorisation. Pour détecter des messages en mode non connecté il faudrait soit disposer d'un *sniffeur* de trames sur le réseau et détecter des portions de protocole SMTP mais dans ce cas on est limité aux signaux véhiculés sur le réseau local, soit sniffer les signaux de transmission hertzienne sur longue distance permettant de récupérer des portions de protocole à tout vent.
Retour du *daemon* dans le cas d'un destinataire indisponible:
1er cas: serveur de messagerie hors tension, le message reste en attente sur le *mailhost* de l'université (*amon.u-strasbg.fr*) qui réessaie de le transmettre à LIIA une fois par heure pendant 4 jours; si au bout de ce temps LIIA est toujours indisponible alors le message repart à son expéditeur.
2ième cas: LIIA est en marche mais le compte test n'existe pas, alors le message est refusé par LIIA (*unknown user*) et il repart tout de suite vers son expéditeur.

2 ENAÏM: un système intelligent de filtrage d'information

2.1 Modèle de l'utilisateur

Il est très difficile d'anticiper les plans, les buts et les besoins d'un utilisateur avec de surcroît une représentation informatique. L'utilisateur n'est, parfois, pas capable lui-même de définir ses propres caractéristiques. Nous pouvons nous reposer sur 2 types d'objet qui sont à la base du modèle du domaine: la nature du document traité (le message électronique) et les tâches associées aux fonctionnalités exigées d'un client de messagerie. Le modèle du domaine (niveau de tâches et niveau physique) croise le modèle de l'utilisateur. Notamment le niveau des tâches peut engendrer des actions typiques de la part de l'utilisateur: message lu ou ignoré, message supprimé ou conservé, message répondu ou renvoyé, message marqué par une priorité, durée de lecture d'un message, ordre de lecture des nouveaux messages.

Dans notre projet d'application nous nous fixons plus spécifiquement sur l'analyse des documents. L'analyse des documents va donc contribuer essentiellement à construire notre modèle de l'utilisateur. Nous appelons ce modèle un profil d'utilisateur. Ce profil de l'utilisateur pourrait être complété par un modèle cognitif de l'utilisateur que nous n'aborderons pas.

L'adaptivité se réfère à un système qui s'adapte selon une tâche exécutée par le système. Cette fonctionnalité s'oppose à un système statique qui définit une action une fois pour toute. Dans le cas d'un système qui gère un flux de documents dépendant du temps avec un contenu variable, l'adaptivité devient primordiale. La classification automatique répond à cette exigence.

Le traitement d'un document peut comporter différentes considérations qui ne relèvent pas nécessairement du terme-clé. Ce traitement peut considérer la structure électronique du document: comptage des mots, base de données auteur avec données concernant l'expéditeur d'un message, adresses de courrier électronique contenues dans les messages, URLs contenues dans le message, code de programme dans le message, style d'écriture (nombre d'espaces, de sauts de ligne…), message contenant des structures connues (CV, formulaire, signature…), message contenant des attachements, message respectant une chronologie avec d'autres messages…

Nous tentons d'approcher le modèle de l'utilisateur par le traitement du corpus de ses messages consultés et validés. L'analyse du corpus va permettre de dégager des tendances thématiques du contenu de l'ensemble des messages.

L'utilisateur fournit (est caractérisé par) un ensemble de classeurs regroupant chacun des messages associés à un thème donné.
Chacun de ces classeurs est soumis à une classification automatique qui permet de lui associer un ensemble de classes de termes. Cet ensemble de classes, appelé centre d'intéret, sera un des éléments constitutifs du profil de l'utilisateur.
Ce profil réutilisable peut servir à identifier de futurs messages sémantiquement proches et donc propres à être classés par le même classeur dont est issu le profil.

Dans ce modèle de génération de profil l'utilisateur crée lui-même ses classeurs de rangement. L'amorçage du processus de filtrage demande une contribution de l'utilisateur, qui, soit participe manuellement au stockage progressif des messages pour que le profil puisse être généré, soit importe une archive de messages prête à être analysée.

2.2 Architecture du système

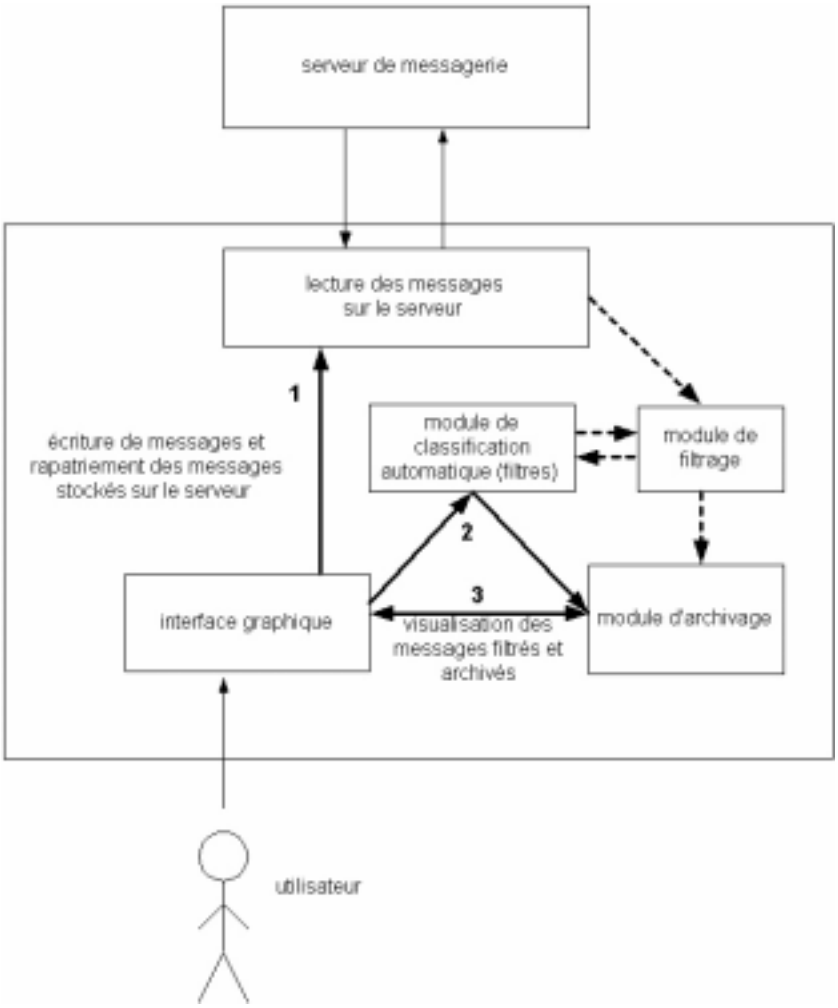


Figure 5.3 Architecture de Enaïm.

Nous sommes partis de l'idée de faire fonctionner nos module sur une plateforme Windows. Nous avons essayé d'intégrer nos modules dans l'architecture ("open source") de Netscape mais la technicité et l'absence de documentation nous a abandonné cette idée et à développer notre propre client de messagerie. La figure 5.3 nous montre les principaux composants du système de filtrage dans lequel l'utilisateur tient une place fondamentale. L'utilisateur interagit avec le système par une interface graphique générale. Cette interface lui propose 3 actions possibles:

- rapatriement des nouveaux messages stockés sur le serveur (1)
- lecture des messages archivés dans des classeurs qu'il a lui-même créés (2)
- calcul de son profil pour chacun des classeurs thématiques (3)

La figure 5.3 représente en pointillé le traitement effectué sur chaque nouveau message reçu pour autoriser le filtrage. Le profil de l'utilisateur est consulté par le module de filtrage qui archive ou pas le message dans le classeur approprié.

Le profil de l'utilisateur est calculé par classification automatique du contenu des messages. Tous les messages d'un classeur sont réunis pour former un corpus plein texte. Le résultat de l'analyse de ce corpus est un ensemble, de classes de termes, sensé représenter les centres d'intérêt thématiques de l'utilisateur. L'aspect thématique est guidé par l'utilisateur lui-même qui amorce la création du classeur en y stockant un certain nombre de messages soit à la main soit par import d'un classeur existant (une fonction d'import permet d'importer un classeur de client classique (Netscape, Microsoft ou Eurora). Les classes sont calculées pour chacun des

classeurs. On a donc des sous-ensembles de classes où chaque sous-ensemble est rattaché à un classeur. La réunion de ces sous-ensembles constitue le profil de l'utilisateur du point de vue global. Un message sera confronté à chaque sous-ensemble pour accepter son transfert dans le classeur relatif.

2.3 Structure des fonctions principales

L'application se présente sous forme d'une interface graphique à peu près équivalente à celle d'un client de messagerie.

Une barre d'outils (fig. 5.4) propose différentes fonctionnalités de traitement: réception de messages, écriture de messages, construction d'un profil, réponse à un message, redirection d'un message, impression d'un message, suppression d'un message. 3 sous-parties permettent: à gauche, de sélectionner un classeur (arbre de classement à gauche), au centre de sélectionner un message, en dessous de visualiser le message sélectionné (fig. 5.6).

Un menu (fig. 5.5) propose plusieurs fonctionnalités comme la création (création d'un classeur dans l'arborescence) ou l'acquisition d'archives (import d'une archive existante de messages).

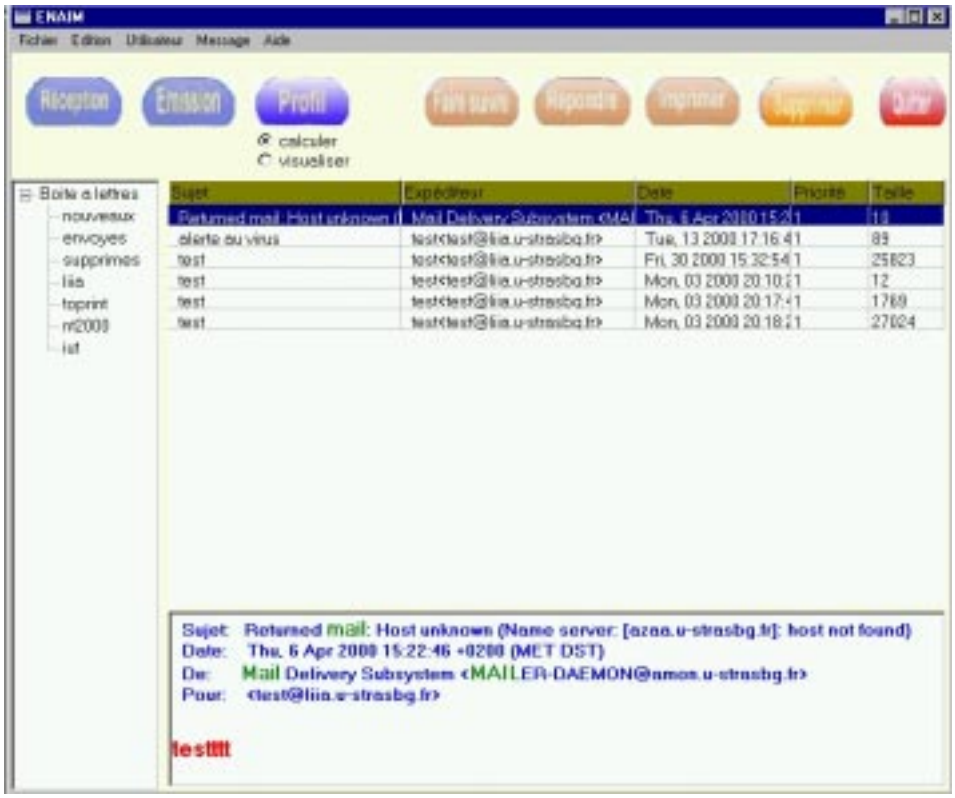


Figure 5.4 Fenêtre principale de l'interface utilisateur.

Figure 5.5 Menu proposant de créer un nouveau classeur et d'importer une archive.

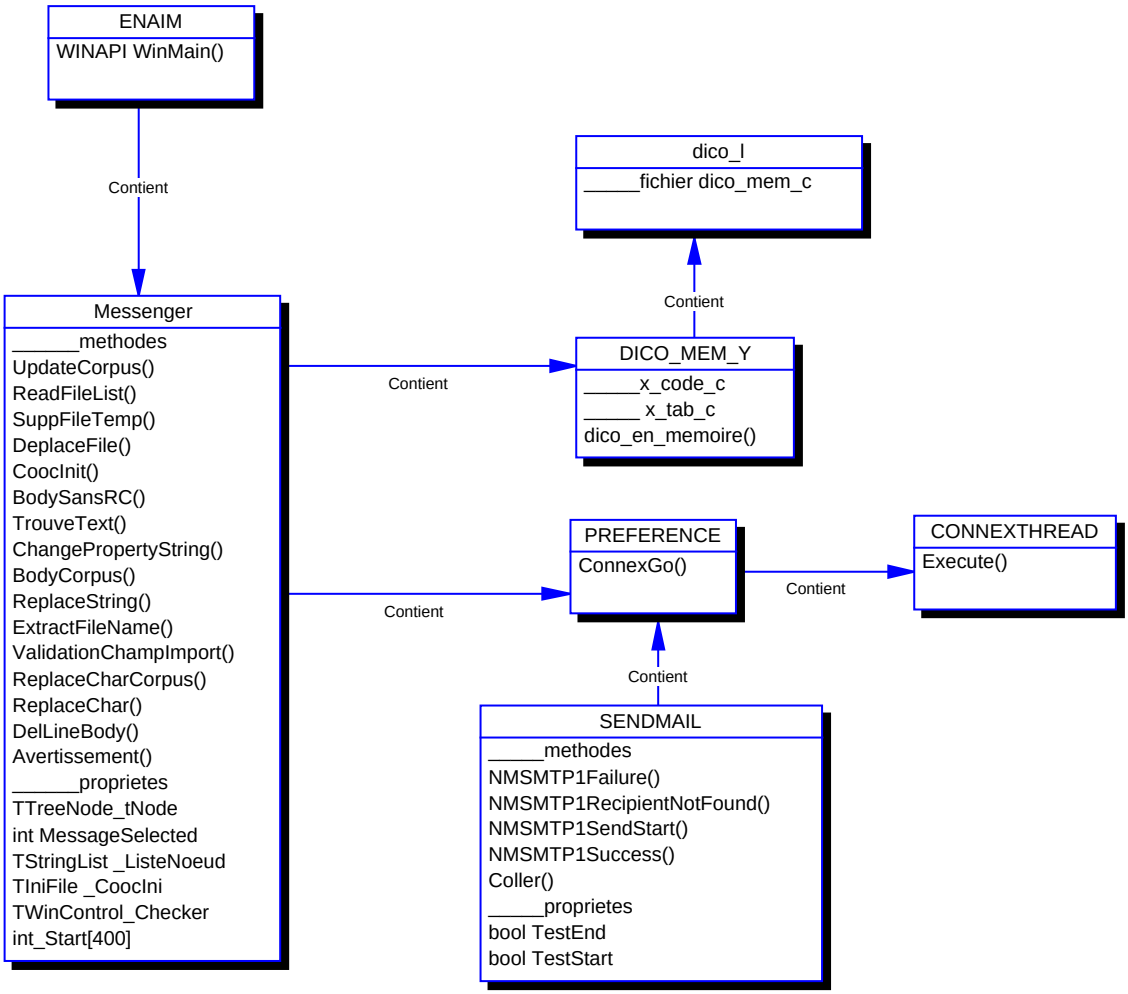


Figure 5.6 Schéma conceptuel des classes générales dans ENAÏM.

2.4 Structure des fenêtres graphiques

Les fenêtres graphiques assurent un confort à l'utilisateur dans la gestion des traitements réalisés (fig. 5.10). Une fenêtre lui permet d'écrire un message de façon traditionnelle en respectant les champs nécessaires (adresse, sujet, corps) (fig. 5.9). Une fenêtre accessible par le menu de la fenêtre principale présente les paramètres généraux, réseau et de la classification nécessaires aux fonctionnalités majeures (accès au serveur de messagerie, création du profil) (fig. 5.7). Finalement une fenêtre avertit l'utilisateur du déroulement de la recherche des messages sur le serveur et de leur nombre (fig. 5.8).

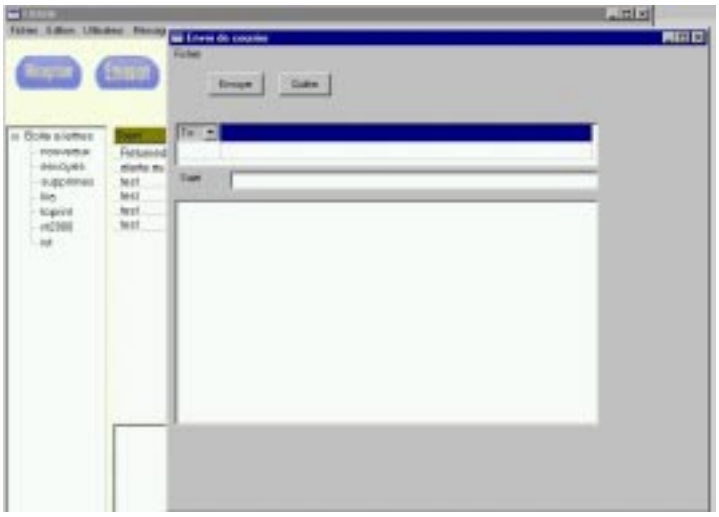


Figure 5.7 Fenêtre d'écriture d'un message à envoyer (action émission).

Figure 5.8 Fenêtre des préférences (réseau, classification, affichage).

Figure 5.9 Fenêtre d'avertissement du rapatriement des messages non lus (action réception).

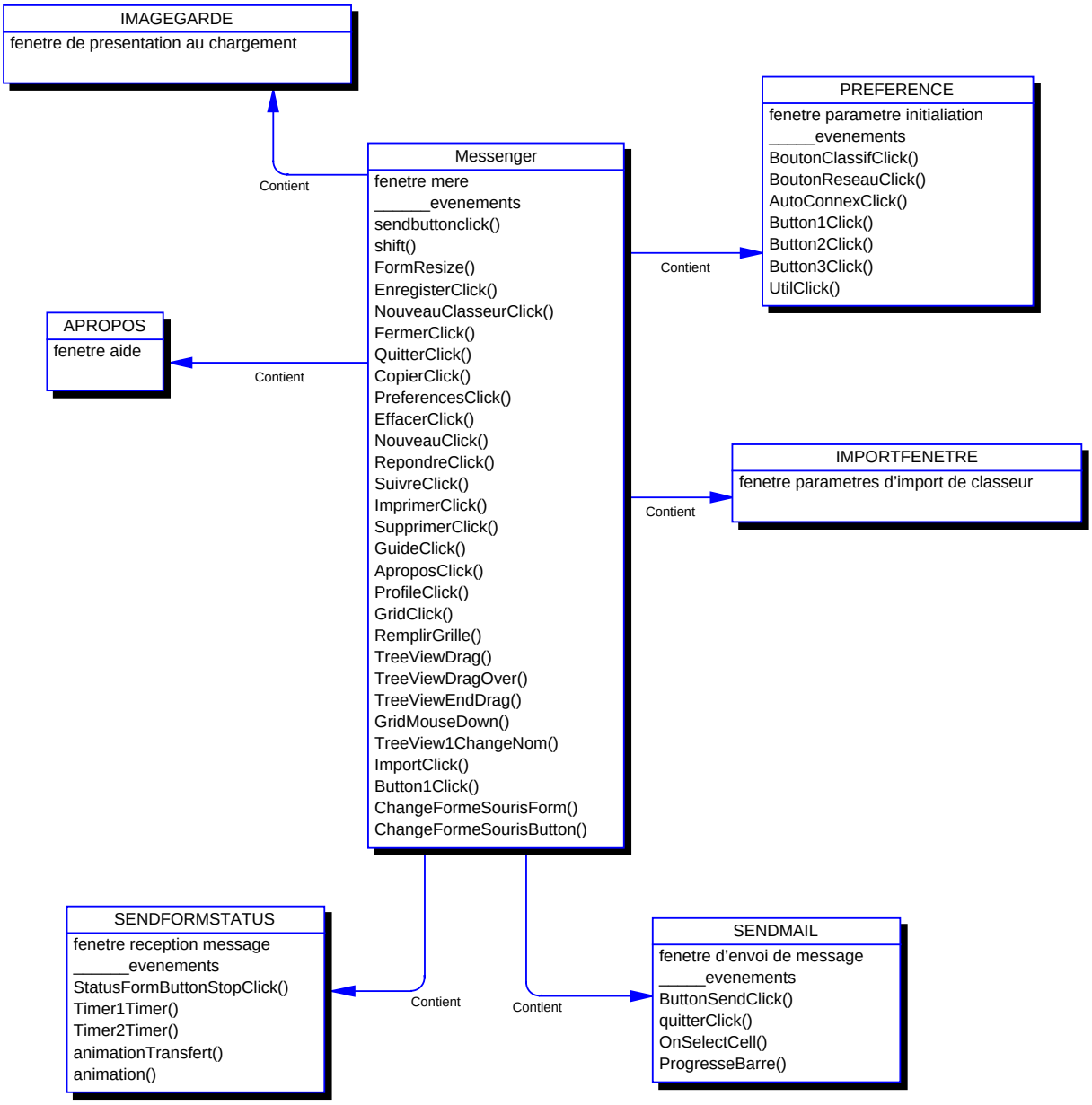


Figure 5.10 Schéma conceptuel des fenêtres graphiques dans Enaïm

2.5 Modèle des connaissances et structure du profil de l'utilisateur

La modélisation des connaissances qui donne le caractère "IA" à notre projet se traduit grâce à l'acquisition des habitudes de consultation d'un utilisateur.

Traditionnellement on considère l'IA à travers la modélisation des connaissances du domaine, traitées par une application, grâce à des règles de ce domaine, déployées dans le cœur des fonctionnalités du système.

Le domaine de ENAÏM est la messagerie électronique. Or, en tant que telle la messagerie ne dispose d'aucune connaissance sur les besoins de l'utilisateur hormis la fonction de consultation. Le domaine est, entre autres, délimité par des *règles d'induction* qui pourraient faire profiter l'utilisateur d'une fonction "améliorée". De telles règles ne peuvent pas émaner de la connaissance technique du domaine que constitue la messagerie. On s'appuie sur les habitudes de consultation de l'utilisateur pour *déduire de telles règles* aboutissant ainsi à un modèle de l'utilisateur (i.e. modèle des connaissances du comportement de l'utilisateur). Les habitudes sont considérées par les unités textuelles archivées (suite de messages). L'extraction de classes d'unités sémantiques, comme les termes, vont permettre d'approcher les concepts usuels intéressant l'utilisateur [Turenne, 1998c].

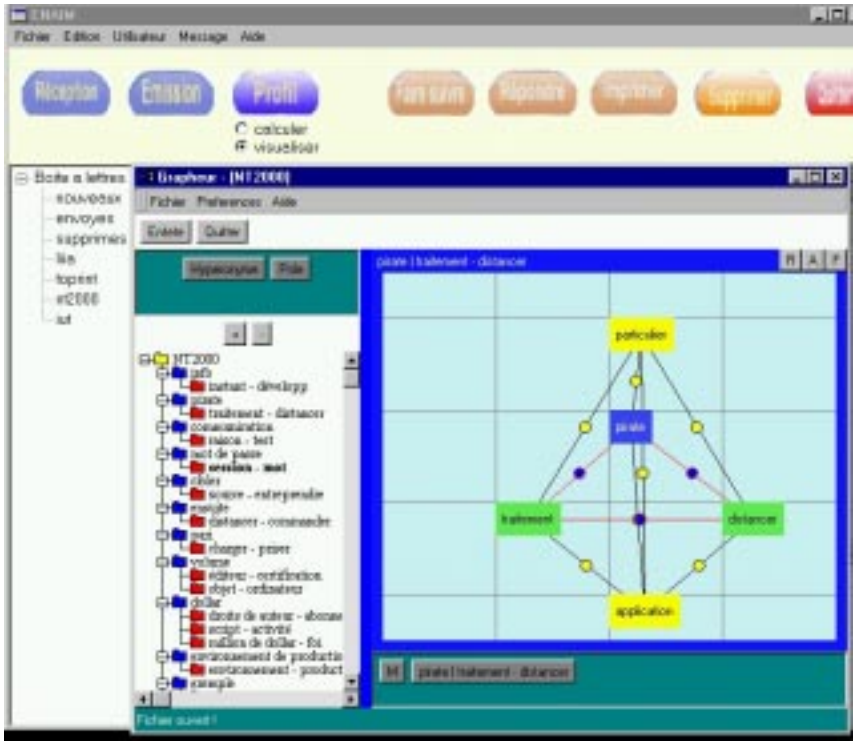


Figure 5.11 Affichage du profil d'un classeur (action *profil-afficher*).

Plus concrètement les archives ou classeurs de l'utilisateur vont servir à constituer le profil de l'utilisateur. On pourrait créer un corpus global avec tous les messages archivés. Pour éviter les recoupements d'associations parasites on préfère juxtaposer des suites de classes. C'est-à-dire que l'on fait correspondre une suite de classes (i.e. centre d'intérêt) à une archive, il y a donc autant de suites de classes que de classeurs de rangement. L'ensemble des suites de classes définit le profil de l'utilisateur. La figure 5.11 montre une classe provenant du centre d'intérêt calculé à partir des messages de l'archive « Windows NT ».

Le calcul du profil utilise les applications vues au chapitre 3 pour extraire les classes. Les programmes vont extraire un centre d'intérêt pour une archive donnée. Ce processus est

commandé par l'utilisateur grâce à l'interface graphique principale et se produit unitairement pour chaque archive (fig. 12).
Finalement un centre d'intérêt aura la même allure que l'annexe 10.

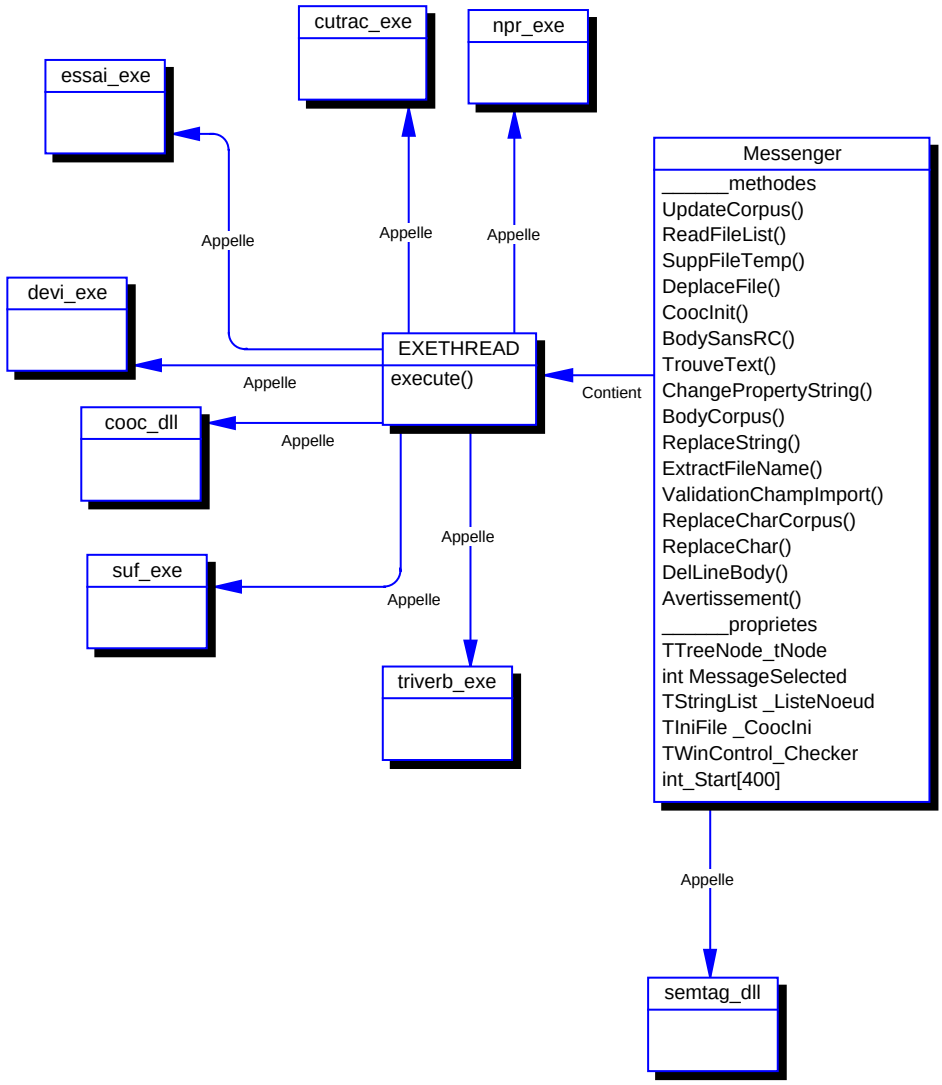


Figure 5.12 Schéma conceptuel des classes de calcul de profil dans Enaïm.

2.6 Règles de filtrage

Le filtrage est un mécanisme d'inférence. L'inférence est une règle logique qui permet d'atteindre une conclusion sachant que les prémisses sont vraies. En l'occurrence dans le cas d'un classement la conclusion est binaire: document à transférer / document à ne pas transférer. Les prémisses sont des conditions de nature diverse. Dans le cas d'un message électronique 3 facteurs sont pris en compte: l'expéditeur, le sujet et les phrases du contenu.

Dans la réalité, les prémisses sont complexes et échappent aux seuls critères écrits dans un message. Des facteurs environnementaux sont pris en compte pour établir l'intérêt ou la priorité. Nous n'avons accès qu'aux formes écrites dans les messages et notre mécanisme portera uniquement sur ces formes. A titre anecdotique on peut remarquer que *filtrage*

d'information, en anglais *information filtering*, se dit IF sous forme d'acronyme anglais et qui veut dire "si" dans la même langue.
Dans ce cadre simplifié voici quelques exemples de règle d'inférence :

Si "le message parle d'astronomie" **alors** *transférer*
Si "l'auteur est F Dupont" **alors** *transférer*
Si "le message contient Windows" **alors** *transférer*

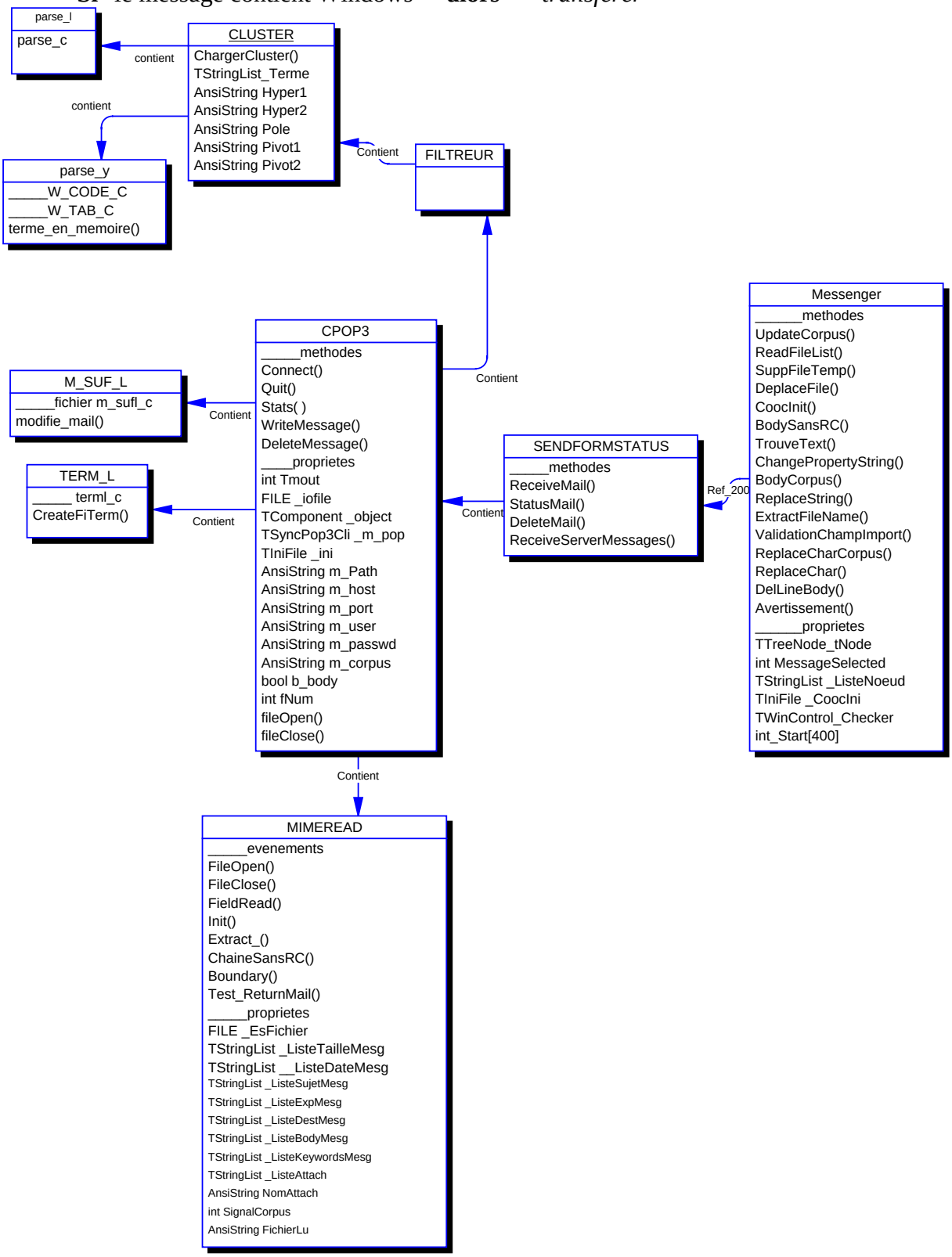


Figure 5.13 Schéma conceptuel des classes de réception/filtrage dans Enaïm

L'apprentissage est tourné vers des exemples positifs. Des règles positives sont traitées. En effet l'utilisateur, dans un modèle de tâche ordinaire, ne traite pas de messages non désirés que l'on appelle "spam" (§1.1). L'utilisateur pourrait créer un classeur "messages non souhaités", y stocker les messages non souhaités pour constituer un corpus de messages non souhaités et construire un "centre de contre-intérêt" lié à ce type de message pour ensuite obtenir des règles de rejet de type:
Si "le message contient Windows" **alors** *supprimer*

Les règles conditionnelles sont directement computationelles. Nous utiliserons les centres d'intérêts pour générer des règles propositionnelles inductives.

En terme de qualification des résultats, nous nous sommes posé la question de savoir comment sélectionner le meilleur candidat classeur pour effectuer le transfert sans pour autant qualifier la pertinence à l'aide d'un seuil. Deux possibilités sont envisageables : soit un raisonnement de maxima de score, soit d'avoir des règles strictes assurant directement le transfert.

Les règles strictes sont fonctionnelles quand on est sûr à 100% de la pertinence des instances présentes dans le document et relatives à un classeur (propriété discriminante). Nous ne nous positionnons pas au niveau d'une discrimination des classeurs vis-à-vis d'un ensemble de termes malgré le fait que cette opération puisse être complémentaire à notre approche d'ensemble de classes. Nos classes qualifiant une archive sont, en principe, typiques de la thématique du classeur mais pas forcément discriminantes c'est-à-dire que ces classes peuvent se partager d'un centre d'intérêt à l'autre suivant leur recouvrement thématique.

Il reste donc la propriété du maximum parmi les scores obtenus pour chaque centre d'intérêt. Nous développons 6 stratégies possibles, c'est-à-dire des caractéristiques de filtrage au niveau : de la classe, des classes d'un même pôle ou de l'ensemble des classes soit par une propriété de seuil minimal soit par une propriété de maximum.

Un message M est décomposé en termes puis les scores suivants peuvent être
calculés pour chacun des centres d'intérêt C :
1. S1: la taille max de l'intersection des termes de M avec une classe de C;
2. S2: la taille max de l'intersection des termes de M avec l'union des classes de C de même pôle;
3. S3: nombre de termes de M présents dans une classe de C + nombre de termes de M égaux à un pôle de classe de C;

Des seuils minimaux ayant été fixés pour chacun des 3 scores, 2 stratégies
d'affectation (filtrage) sont considérées :
1. affectation du message à tous les centres d'intérêt pour lesquels le score sélectionné est supérieur au seuil correspondant fixé;
2. affectation du message à tous les centres d'intérêt pour lesquels le score sélectionné est maximal. Si les scores sont égaux le message est affecté aux classeurs correspondants.

Algorithme

```
1- Découpage du message en unités syntagmatiques
2- Lemmatisation des unités
```

```
3- Stockage des unités dans un tableau T
4- Stockage d'un centre d'intérêt dans un tableau C
                    // stratégie du score seuillé //
5- a- Tester si au moins N termes de T font partie d'une classe de C
                alors autoriser le transfert
   b- Tester si au moins N termes de T font partie des classes d'un pôle
                alors autoriser le transfert
   c- Si un terme de T appartient à une classe de C
            score total= score total +1
      Si un terme de T est pôle d'une classe de C
            score total= score total +2
      Si score total est supérieur à un seuil de C
            alors autoriser le transfert
6- Répéter 4- tant qu'il reste un centre d'intérêt
                    // stratégie du score maximal //
7- a- Calculer le max du score total, du score par classe et par pôle
   b- Transférer le message dans le classeur ayant le score maximal.
```

2.7 Evaluation

L'évaluation se passe en 2 phases. Pour un ensemble de N messages on considère 2 jeux d'apprentissage/test (n_{A1} , n_{T1}) et (n_{A2} , n_{T2}).
Avec $n_{A1} + n_{T1} = N$, $n_{A1} = 60\%N$, $n_{T1} = 40\%N$ et $n_{A2} + n_{T2} = N$, $n_{A2} = 80\%N$, $n_{T2} = 20\%N$;

1ère phase
Un premier apprentissage utilise n_{A1} documents pour construire le profil d'un classeur;
On teste le filtrage avec les n_{T1} messages;
2ième phase
Un second apprentissage plus important avec n_{A2} documents;
On teste le filtrage avec les n_{T2} documents.

L'évaluation se fait par un calcul de performance de la qualité du filtrage [Hull, 1998]. On considère la notion d'efficacité. Cette notion tient compte d'un effet mixte: l'échantillonnage (angl. Sampling) et la sélection des meilleurs classés (angl. Pooling). La sélection des meilleurs classés diminue l'effet de l'absence de borne d'un ensemble de documents extraits. Dans le filtrage cette sélection correspond beaucoup plus à un échantillonnage, car la soumission d'un document se fait à la volée à partir d'un échantillon de test qui demande un classement binaire: vrai (doit être classé) ou faux (ne doit pas être classé). L'ensemble des documents à soumettre au système est équivalent à un ensemble de documents présélectionnés devant être extraits. Cet ensemble correspond aussi à un échantillon pris au hasard d'un ensemble réel beaucoup plus vaste mais de taille inconnue.

Voici la fonction d'efficacité (angl.,utility function) pour le classeur c_i :

$$F_1(c_i) = A \cdot R^+ - B \cdot N^+ \qquad (1)$$

où :
A est le nombre de documents extraits pertinents;
B est le nombre de documents extraits non pertinents;
R^+ est le nombre de documents pertinents filtrés;
N^+ est le nombre de documents non pertinents filtrés.

Cette fonction F_1 tient compte du différentiel entre les documents pertinents et non pertinents filtrés dans le classeur i. Cette mesure peut varier d'un classeur à l'autre et d'un système à l'autre suivant des ensembles de documents différents. Pour ramener cette mesure à une mesure comparable on la normalise par la mesure d'efficacité liée au nombre de documents

non pertinents maximal que l'utilisateur peut tolérer (i.e. on fixe un nombre de documents non pertinents s et on en déduit $F_1(c_i)$ = B. s). On tient aussi compte aussi du nombre maximal de document pertinents possibles $M_u(c_i)$. Ces 2 paramètres vont définir la fonction appelée *fonction de gain* u normalisée entre –1 et 1. Un système sera d'autant meilleur que la fonction u se rapprochera de +1.
Soient $M_u(c_i)$ = max($A.R^+$) pour le classeur c_i ∀le système S,
et $L_u(c_i)$ = min($B.N^+$) pour le classeur c_i ∀le système S. On fixe N+ arbitrairement, par exemple N+ = 2 documents non pertinents tolérés.
La fonction s'écrit:

$$u(S,c_i) = \frac{\max(\ F1(c_i),\ L_u(c_i)) - L_u(c_i)}{M_u(c_i) - L_u(c_i)} \qquad (2)$$

Baseline est le niveau de non filtrage (0 document filtré). Plus le système s'écarte de baseline en direction positive plus ses résultats sont satisfaisants.

$$u^*(S,c_i) = u(S,c_i) - u(\text{Baseline},c_i) \qquad (3)$$

Le résultat global est la moyenne sur tous les p classeurs:

$$u^*_{tot} = \frac{1}{p}\sum_{i=1}^{p} u^*(S,c_i) \qquad (4)$$

	Requête 1					
	R+	N+	F1	F1 classé	u	u* réduit
système	0	5	-10	1	0	-0,4
baseline	0	0	0	2	0,4	0

	Requête 2					
	R+	N+	F1	F1 classé	u	u* réduit
système	4	4	4	4	0,2	0,1
baseline	0	0	0	1,5	0,1	0

	moyenne	
	F1 classé	u* réduit
système	2,5	-0,15
baseline	1,75	0

Figure 5.14 Exemple de calcul de F1 et u*

La figure 5.13 nous donne un exemple de calcul. R+ et N+ permettant le calcul de F1 suivant la formule (1), on calcule u* grâce à (3). On réduit u* en lui soustrayant u*(baseline, R+=0 et N+=0).

Nous avons réalisé 2 bancs de tests pour mettre en oeuvre la fonction de filtrage et l'adaptivité de l'apprentissage grâce à l'acquisition des classes de termes.
L'annexe 11 résume les résultats pour les 2 tests. L'opération s'applique à 3 archives (qui sont de thématiques différentes mais voisines car touchant à l'informatique):
- LIIA (messages reçus de membres de mon *laboratoire*),
- NT (messages d'une *liste de diffusion sur les technologies NT*)
- et IUT (messages de l'*IUT télécom-réseaux* où j'enseignais en 1999-2000).

Les répertoires ne contiennent pas plus d'une cinquantaine de messages.

Le protocole du banc de test est le suivant:

1- diviser chaque archive en 2 : une partie d'apprentissage (A) et une partie test (T),
2- apprendre le profil avec chaque partie *test*,
3- utiliser les *messages test* inconnus pour valider le filtrage (stratégie du score seuillé par classe),
4- recommencer 1,2,3 avec un échantillon d'apprentissage plus grand,
5- calculer le taux de réussite.

Voici quelques exemples de filtrage
(liste de termes du message inconnu participant au succès du filtrage):
(a) termes du score seuillé par classe (message NT):
carrière, système, réseau, domaine, astuce, émail, serveur, numéro;
(b) termes du score seuillé par classe (message LIIA, contenu lié à la soutenance de l'auteur):
certitude, new, incertitude;
(c) termes du score seuillé par classe (message LIIA, contenu lié à la soutenance de l'auteur):
sujet, application, texte;
termes du score total:
titrer, texte, sujet, application, décembre, accord, école, juillet.

Malheureusement l'effet adaptatif recherché n'a pas pu être mis en évidence. L'archive LIIA présente un corpus de messages trop petit pour assurer une bonne extraction de classes. 0 classe pour le test 1, et 2 classes pour le test 2. Ces nombres sont non significatifs et inexploitables pour un filtrage. L'archive NT a saturé dès le premier test car l'apport de nouveaux messages n'a pas permis d'augmenter le nombre de classes extraites du corpus. En effet malgré le petit nombre de messages dans l'archive (29) leur taille (en moyenne 20 Ko par message) a permis de créer un corpus conséquent et homogène. Il reste l'archive LIIA dont le contenu est variable. Le test 2 montre une augmentation du rappel mais le couple rappel-précision ne varie pas significativement. D'autant plus que les messages sont parfois très courts comme celui ci:

Sujet: Re: jury de these (Turenne)

B
je suis pour le moment libre jusqu au 15 decembre - pas apres
YK

Nous constatons que l'apprentissage se passe bien pour une archive d'au moins 50 à 100 Ko.
En deçà la quantité d'information n'est pas suffisante.
En ce qui concerne les messages de la liste de diffusion NT on obtient un taux de filtrage avec 100% de précision concernant les messages de la liste pour le classeur associé. Il se peut que les messages soient transférés dans un autre classeur causant un certain bruit de pertinence mais cela peut être résolu en jouant sur le paramètre du *scoretotal*.
En ce qui concerne les messages de l'archive LIIA. On remarque dans les résultats du test 2 que les messages 17/19 (92%) parviennent au classeur LIIA en prenant en compte du score total pour le classeur de tous les messages LIIA.

La mesure totale u^*_{tot} donne 0.12 pour le premier test et 0.07 pour le second ce qui positionne le système en bonne position par rapport aux autres systèmes de l'étude TREC-7 dont les meilleurs scores sont : +0.01 pour le système de Claritech, +0.005 pour le système

Microsoft/Université de Sheffield, et –0.001 pour les systèmes de l'Université de Iowa et Queens College. Dans les épreuves de Trec-7 aucun système bayésien n'a participé bien que ce type de système fasse partie des approches de l'état de l'art. De tels système existent [Sahami et al, 1998] et il serait intéressant de pouvoir comparer notre système système avec cette approche.

Cette mesure n'est pas simple à interpréter car dans le cas de l'archive NT les tests 1 et 2 donnent une valeur de –0.02 et –0.08 alors que le filtrage transfert tous les messages *test* NT dans le bon classeur. Le filtrage transfert aussi des messages d'autres archives ce qui cause du bruit et donc une baisse de la mesure.
D'autres part les tests se passent dans des conditions réelles mais de taille limitée. Il nous faudrait un échantillon de plusieurs milliers de messages.

Les matrices de confusion offrent une grille de lecture différente d'un calcul de score. Cette matrice permet de croiser, pour un classeur donné, les document s légitimes de ce claseur et les documents étrangers (non légitimes) au classeur en comptabilisant les documlents biens classés et mal classés.

Voici la matrice de confusion pour le classeur windows NT (apprentissage avec 29 messages):

	bien classés	mal classés	total
légitimes	14 (100%)	11	14
non légitimes	31 (50%)	23	54
total	45 (66%)	34	68

Voici la matrice de confusion pour le classeur LIIA (apprentissage avec 115 messages):

	bien classés	mal classés	total
légitimes	11 (58%)	12	19
non légitimes	5 (100%)	4	5
total	16 (71%)	16	24

Les répertoires LIIA et NT reflètent une thématique voisine concernant l'informatique, c'est pourquoi la stratégie de multiclassement choisie conduit à de nombreux documents mal classés. En interdisant le multiclassement le taux de documents mal classés est divisé environ par 10. Pour améliorer le multiclasssment il faudrait tenir compte d'une péréquation entre le score par classe (similarité) et le score sur l'ensemble du centre d'intérêt (dissimilarité).

2.8 Multilinguisme

Le multilinguisme a pris de nos jours des proportions démesurées. Les échanges économiques internationaux alliés à un regain des sentiments nationaux ont mis sur le devant de la scène les sirènes de la tant convoitée traduction automatique. De grands projets européens (Human Language Technology) subventionnent des consortiums européens pour développer des ressources et des applications multilingues.
L'application Enaïm est fondée sur le traitement de la classification automatique (chapitre 3) et de l'étiquetage des classes (chapitre 4). Ce double traitement fait appel à 3 ressources:

(a) un dictionnaire de mots-lemmes;
(b) une base de suffixes;
(c) un thésaurus présentant un ensemble de catégories thématiques.

Les ressources (a) et (b) sont dédiées à la lemmatisation (prétraitement de la classification). La ressource (c) est centrale dans l'étiquetage des classes.
Nous utilisons ces ressources pour le français dans le cadre de nos applications. Pour d'autres langues ces ressources seraient nécessaires dans le cadre d'un portage multilingue. Par

exemple pour l'anglais, le thésaurus Roget est très similaire au thésaurus que nous utilisons. Il est disponible gratuitement sur Internet. La base de suffixes a été établie rapidement. Nous ne disposons pas encore d'un dictionnaire de lemmes.
Pour d'autres langues ces ressources doivent être constituées et demanderaient un certain temps de travail bien que limité.

Les algorithmes de classification et d'étiquetage sont indépendants de toute ressource lexicale et sont complètement portables au multilinguisme.

3. Autres applications de la classification automatique de termes

3.1 Applications de haut niveau

Les applications de haut niveau sont des applications qui proposent une fonctionnalité intégrable dans un système d'information. Dans ce cadre la classification de termes est un chaînon du traitement répondant à la fonctionnalité.
Voici une liste d'applications (la plupart sont classiques). Entre parenthèses on donne le rôle de la classification de termes considéré par l'application:

(a) filtrage/routage de documents.
(génération de règles),
(b) classification de documents
(génération de fonctions de prédiction),
(c) résumé automatique
(sélection de phrases),
(d) création d'un thésaurus
(détection de groupes thématiques),
(e) commerce électronique
(création du profil type d'un client ou d'un produit par des classes de termes).

Le système ENAÏM admet les fonctionnalités de (a). Actuellement nous menons une coopération franco-russe pour développer une architecture caractéristique de (e). Cela assoit notre but constant de valoriser et critiquer les propriétés de la classification automatique de termes.

3.2 Applications intermédiaires

De la même façon que pour les applications présentées ci-dessus il est envisageable de positionner la *classification de termes* comme *sous-tâche* d'un processus lui-même sous-tâche d'une application finale.
Voici une liste de tâches (certaines sont classiques). Entre parenthèses on donne le rôle de la classification de termes considéré par la tâche:

(a) expansion de requête
(classe de termes complétant les termes d'une requête [Grefenstette, 1997]),
(b) reformulation de requête
(l'utilisateur sélectionne des termes d'une classe),
(c) désambiguation de sens d'une paire de termes
(comparaison de termes cooccurrents avec ceux d'une classe),
(d) validation de dépendances entre termes
(génération de règles inductives *si (termeA et termeC) alors termeB*),
(e) sélection de termes discriminants
(détection des formes les plus représentatives grâce aux associations répétées).

Nous avons eu l'occasion d'étudier le comportement de la classification de termes dans le cadre d'une des sous-tâches citées ci-dessus (b): la reformulation de requête (d'un système de recherche d'information sur Internet) [Turenne & Rousselot, 1998b]. Dans cet article nous présentons un nouveau système de reformulation alors baptisé Saros (Système d'Aide à la Reformulation par Opérations Successives). Les fonctionnalités du module existent encore au sein du moteur d'indexation Etat Partenaire de l'ADIT qui en est propriétaire. Le développement de ce module a été réalisé au sein de l'ADIT en partenariat avec le LIIA, les sociétés ECILA et CISI. La solution n'implémente pas d'algorithme nouveau mais intègre plusieurs logiciels existants (Sampler™ qui classe des termes, Ecila™ qui cherchent des pages web, Search97™ qui indexe en mode plein texte, Genet™ qui extrait les syntagmes nominaux).

Cette solution est similaire à celle implémentée dans le module AltaVista/LiveTopics, avec la différence que les classes de termes sont calculées à partir du contenu de la base et non des champs mot-clés des documents. De plus Saros utilise des multitermes et pas uniquement des mots simples.

L'architecture du serveur est composée de 8 éléments: un crawler utilisé pour extraire les pages web jugées intéressantes, un programme de gestion de la base de documents, un module qui permet l'accès à l'information, un extracteur de syntagme et de mots simples, une interface de dialogue initiale avec l'utilisateur, un classifieur, une interface interactive de dialogue utilisateur, un extracteur de documents (i.e. moteur d'indexation plein texte). L'architecture contrôle le flux dynamique d'information provenant du web, et offre un système d'aide à la reformulation d'un besoin en information aussi dynamique que le contenu de la base. Cette aide à la reformulation indique une manière de suggérer des éléments d'un thème relatif à une requête générale.

Le traitement comprend 2 phases. La 1ère phase est un pré-traitement qui indexe la base dans un mode plein texte, extrait les termes (segments répétés et mots discriminants) et classifie les termes. La seconde phase est un traitement en quasi-temps réel de la requête initiale de l'utilisateur qui appelle les suites lexicales à partir desquelles l'utilisateur accède à des classes. A partir de ces classes, l'utilisateur choisit quelques termes pour les ajouter à la requête . La requête est soumise à l'indexeur plein texte qui la traite par un procédé classique (modèle vectoriel) pour renvoyer la liste des documents qui correspondent au résultat classé par ordre de pertinence.

4. Perpectives

4.1 Plan technique

Enaïm réalise un filtrage de messages électroniques. Ces messages proviennent du protocole SMTP par lequel les messages ont été envoyés. Ce sont des messages de destinataire à expéditeur. D'autres messages électroniques sont répandus sur d'autres protocoles comme NNTP. Ce protocole gère les forums de discussion qui sont relativement typés du point de vue thématique puisque les forums sont en général créés pour des utilisateurs partageant un centre d'intérêt commun. Les messages transitent par un serveur NNTP et sont récupérés par un client avec NNTP. L'intérêt de travailler avec des forums consiste à tester l'application par une montée en charge sachant que les forums génèrent des flux considérables de messages; les flux peuvent atteindre plus de 100 messages par jour. C'est également l'occasion de tester la stabilisation de profil avec l'apprentissage adaptatif.

Une adaptation de l'application en vue de test plus poussé sur des données Internet (pages web, forums) est la prise en compte de texte en anglais. L'adaptation nécessite le stockage en base de données du Roget et la récupération d'un dictionnaire de lemmatisation.

4.2 Plan théorique

Les améliorations théoriques concernent la génération des règles d'inférence assurant le filtrage.
D'une part la stratégie de transfert semble jouer des rôles complémentaires; score par classe: similarité avec un centre d'intérêt, score total: dissimilarité avec d'autres centres d'intérêt. Il serait intéressant de coupler les deux propriétés de transfert.
D'autre part 2 considérations sont à éprouver et s'expriment en terme de couplage de techniques.
(1) peut-on coupler une méthode d'étude probabiliste des dépendances fonctionnelles des paires (causalité) avec une méthode d'analyse globale des relations?
(2) comment représenter un arbre de décision s'inspirant du modèle d'utilisateur et de la structure du document ?
Son exploitation suppose une intégration des classes de termes pré-existantes.

5. Travaux antérieurs

En recherche documentaire, la classification de termes est quasi-exclusivement connue pour servir la reformulation ou l'expansion de requête. L'autre usage courant de la classification est la classification de documents servant à la catégorisation c'est-à-dire au rangement des documents dans des rubriques thématiques.
Jusqu'à présent la classification de termes jouissait d'une image de marque dégradée. Certains décriaient son efficacité [Minker et al, 1972][Lewis, 1992]. En effet l'approche de la classification de documents était critiquée par le manque de précision des résultats et par la lenteur des processus (au moins quadratique en temps et donc inexploitable en temps réel pour la navigation dans une base).

Une approche qui a redoré le blason de la classification de document est *Scatter/Gather* [Cutting et al, 1992]. *Scatter/Gather* agit de la façon suivante:
Un vecteur de document est constitué par les fréquences des mots uniques de la base totale, ce vecteur est normalisé; Pour une classe de documents on calcule le vecteur moyen des documents les plus proches; Un sommaire des groupes de documents se fait avec les mots du vecteur moyen qui sont les plus fréquents; la similarité est un calcul du cosinus entre 2 vecteurs. Voici les opérations:
- Application une méthode hiérarchique agglomérative de lien moyen à partir soit d'un ensemble aléatoire limité de documents soit de N/m nœuds (m taille moyenne d'un petit groupe, N la taille de la base). La classification s'arrête dès l'obtention de k classes;
- Affectation de chaque document aux k classes;
- Ajustement soit en joignant une paire de document ayant le même sommaire soit en divisant une paire de document.

L'algorithme est d'ordre O(kn) et interactif par navigation avec l'utilisateur. L'utilisateur choisit les classes de documents qui l'intéressent parmi k classes présentées. Les classes sélectionnées sont réunies, divisées en k classes et proposées à l'utilisateur. Le processus s'arrête dès que l'utilisateur souhaite visualiser les documents.

[Kohrs & Mérialdo, 1999] présentent l'utilisation de la classification hiérarchique traditionnelle dans le cadre du filtrage collaboratif. Le filtrage collaboratif permet, comme un vote majoritaire, de sélectionner l'opinion la plus probable d'un certain nombre d'utilisateurs par rapport à des évaluations numériques d'objets (exemple: opinion entre 1 et 5 de la qualité d'un film). Ce filtrage a été notamment utilisé avec des messages de forum de discussion dont les objets traités étaient des mots pour dégager un profil de vocabulaire de ces messages. [Kohrs & Mérialdo, 1999] montrent que la classification automatique permet d'avoir des résultats comparables à des méthodes classiques d'extraction de fonction discriminante telle

que les moindres carrés ou le coefficient de corrélation de Pearson. Cette approche permet d'avoir une fonction de prédiction ou fonction discriminante représentative d'un ensemble d'opinions de plusieurs utilisateurs sachant que plus il y d'utilisateurs plus la fonction est prédictive.

[Macskassy et al, 1999] mettent au point un agent de routage automatique (*EmailValet*) par apprentissage d'entêtes (expéditeur, destinataire, date, objet). Ils comparent 5 méthodes d'apprentissages bien établies avec une évaluation sur 9000 messages: une méthode d'apprentissage par programmation logique inductive (*Ripper*), une méthode vectorielle simple (*TFIDF*), une méthode de vecteur probabiliste (*Pr-TFIDF*), une méthode bayésienne naïve, une méthode de vote par pondération de caractéristiques (*Winnow*). Les méthodes apprennent leurs règles avec des exemples positifs et négatifs. A la suite de l'apprentissage, ils doivent router un message vers la bonne adresse. L'expérience montre un point de *break-even* (precision=rappel) d'environ 53 % au mieux.

[Roussel, 1998] a mis au point un outil (*GASPER*: Graphe Adaptatif Spécialisé Pour l'Emergence de Relations) qui caractérise au mieux un classeur de messages. Il associe à chaque classeur une base de données. Cette base contient des poids pour chaque donnée et construit un graphe de relations. Le graphe est élagué à un seuil d'émergence de cliques. Une sélection extrait les meilleures cliques. Ensuite les règles sont testées: reconnaissance et élimination des mauvaises règles. Le processus est incrémental et adapté aux données qualitatives. Les règles sont finalement des règles de classement mais qui ne prennent pas en compte le contenu du message.

[Rennie, 1998] développe une méthode bayésienne naïve pour créer des règles de filtrage. L'outil, *IFILE*, opère une sélection des formes avec une notion d'âge d'un mot. Un mot peut être éliminé quand log(âge) > fréquence (âge = nombre de documents filtrés depuis que le mot a été rencontré). Les champs *sujet, expéditeur, destinataire et corps* sont jugés pertinents. Les mots fréquents sont éliminés (524 mots vides) et une troncature est réalisée, basé sur l'algorithme de Porter [Porter, 1980]. Le score maximal (probabilité P(classe i/document à classer)) permet de classer le document dans la classe i. On remarque une variabilité du score suivant le nombre de catégories présentes.

[Boone, 1998] réalise un agent (*Re*:*agent*) qui permet de filtrer des messages en 2 étapes: la première étape extrait les formes ou mots-clés transformés en vecteur (TFIDF). Ces formes proviennent du contenu. La deuxième étape permet d'apprendre des règles en fonction des exemples fournis par des répertoires. Pour apprendre une action 2 types d'approximation: les plus proches voisins et le réseau de neurones par rétro-propagation. Une comparaison est faite par rapport à une méthode classique de la recherche documentaire (extraction des formes, création de vecteurs pondérés et calcul d'un vecteur moyen pour un répertoire). Les résultats sont presque identiques avec un léger avantage pour la méthode des plus proches voisins.

[Cohen, 1995] élabore un système (*RIPPER*: Repeated Incremental Pruning to Produce Error Reduction). Le système construit un modèle initial et l'optimise k fois jusqu'à obtenir une stabilité. La méthode est basée sur une méthode propositionnelle de *FOIL* [Quinlan, 1986]: une phase d'apprentissage, une phase de test. Une règle est une conjonction de caractéristiques. Un ensemble de règle est une conjonction de DNF (formes normales). L'algorithme construit règle après règle. Une fois la règle stockée les exemples positifs et négatifs sont supprimés. La règle est construite à partir de rien et ajoute des prédicats à la suite, A_n=V (attribut nominal), A_c<x ou A_c>x (A_c attribut numérique). Ensuite le système

utilise un ensemble d'exemples d'ajustement, i.e. une règle est ajustée (angl., "pruning") en supprimant des conditions qui maximisent V(règle, prune_pos, prune_neg)= (P+N-n)/(p+N) où P est le nombre d'exemples positifs dans prune_pos, N le nombre d'exemples négatifs dans prune_neg, p le nombre d'exemples dans prune_pos et n le nombre d'exemples dans prune_neg. Le processus est répété jusqu'à ce que V ne varie plus. Le processus s'arrête de rajouter des règles si les règles rajoutées ont une erreur > 50%. Les 100 premiers mots d'un message sont considérés pour l'apprentissage et le routage.

Résumé du chapitre

Ce chapitre essaie de répondre à une question importante concernant la classification automatique: comment mettre en valeur l'efficacité d'une classification conceptuelle dans le cadre d'une tâche spécifique de traitement de l'information. Plus clairement parmi les milliers ou milliards de partitions possibles entre les termes à classer, une fois que nous sélectionnons une partition comment savoir qu'elle est réellement interprétable pour être utilisable.

La recherche documentaire étant le domaine d'application attenant au traitement terminologique d'un corpus et par conséquent d'une classification de termes il nous paraissait intéressant de faire jouer un rôle à des classes de termes dans un système documentaire. Ce système documentaire a été choisi par rapport à des besoins qui font jour grâce au réseau des réseaux : Internet. Ce dernier véhicule des flux importants d'information textuelle mais celle-ci est très peu utilisée dans des processus de traitement avancé et de modélisation.

C'est dans ce cadre de modèle d'un utilisateur de messagerie électronique qu'intervient la classification de termes pour tenter de dégager ou synthétiser les centres d'intérêts thématiques de l'utilisateur à travers les messages qu'il a jugés pertinents. Nous ne modélisons pas le domaine de l'application qui est la messagerie mais le comportement de consultation de l'utilisateur du système. Contrairement à une recherche exacte de type *annuaire*, le modèle va, à la manière d'une fouille de textes, contrôler le *caractère potentiellement intéressant* d'un message par rapport à ce qui s'est déjà lu.

Le processus de sélection d'un nouveau message est réalisé grâce au recouvrement des termes contenu dans le nouveau message vis-à-vis d'une classe (éventuellement des classes ayant un même terme pôle, ou l'ensemble total des classes). Le processus inductif est assuré par un recouvrement minimum dans le cas d'un transfert non exclusif à un classeur, ou par un recouvrement maximum dans le cas d'un transfert exclusif. Nous avons développé un banc de test avec 3 archives et donc 3 classeurs. Après les opérations de filtrage une mesure de gain a été calculée en fonction des documents filtrés avec pertinence, sans pertinence et d'un gain de base (sans documents pertinents et non pertinents). La mesure montre que le système filtre dans des conditions très convenables malgré un bruit non négligeable. Le couplage d'un recouvrement total (dissimilarité entre classeurs) et d'un recouvrement de classes (similarité avec un classeur) pourrait améliorer la qualité du filtrage.

CONCLUSION

Cette thèse présente une étude théorique et applicative de la classification automatique de termes à partir de corpus textuels. Elle présente les performances de la méthode à travers le filtrage d'information au sein d'une maille d'un système d'information: la messagerie électronique.

Intérêt de notre méthodologie de classification

En quoi cette thèse propose d'approfondir les problèmes de la classification tant étudiée ces dernières années?
Nous étudions quels sont les phénomènes linguistiques (morphologie et syntaxe) qui peuvent, potentiellement, améliorer la classification aussi bien en amont (prétraitement) que dans le processus même d'affectation d'un terme à une classe (traitement).
Nous tentons de développer une méthodologie à partir de l'existant qui semble le plus se rapporter à la cohérence d'un rapprochement sémantique de données textuelles compte tenu de leur nature fortement relationnelle. Cette nature relationnelle n'est pas étrangère au caractère sémantique des usages. Nous essayons ensuite de qualifier la pertinence des classes par analyse sémantique en effectuant des calculs de consensus dont le but est de rechercher la catégorie qui couvre le plus le champ sémantique de la classe lexicale. Nos études sur des considérations pré-traitement et post-traitement dépendent, et puisent leur intérêt, de l'apport développé autour du traitement (schéma de relations/motif de graphe).

Prétraitement

Sélection des formes

1- nous étudions quelles formes linguistiques doivent être considérées:
les groupes nominaux, les mots simples, les verbes et proportion de mots simples par rapport aux groupes nominaux

Statistique de dénombrement

2- nous considérons quels phénomènes linguistiques de réduction canonique qui peuvent être considérés pour récupérer toutes les occurrences des formes ayant un rapport conceptuel étroit: **lemmatisation, troncature, dictionnaire des formes standard**, **réduction des formes variantes**

Traitement

Traitement statistico-linguistique basé sur des propriétés intéressant le traitement du langage naturel

3- utilisation de **schémas comme relation de cooccurrence**

4- considération de **motifs de graphe** comme critère d'affectation

Interprétation

Analyse de la cohésion lexicale des classes de termes

5- utilisation d'une hiérarchie conceptuelle de référence **étiquetage des classes par un thésaurus par une notion de consensus** énumérant des liens sémantiques "parle de" (différente de l'hyperonymie)

6- évaluation grâce à des critères de précision et de rappel qualifiant la **superposition des classes obtenue et d'une hiérarchie de connaissances experte de référence**

Exploitation

Performance de l'utilisation de classes lexicales dans un contexte de recherche documentaire

7- exploitation de **classes de termes équivalentes à des règles d'inférence** dans un système de filtrage de documents

8- interprétation de **classes comme partie du modèle d'utilisateur de l'application**

Limites de notre méthodologie

Difficulté liée au postulat du sens unique selon lequel toute occurrence dans un corpus correspond à un sens. On admet que ce postulat se justifie pour un corpus considéré comme homogène et propre à un sujet particulier. En outre si un *crawler* indexe le serveur des ministères du commerce et du ministère de la justice les occurrences du mot avocat des pages parlant de l'importation des avocats seraient mélangées avec celles des pages parlant des avocats du barreau.

Difficulté de la réduction canonique. Par exemple: les formes variantes par conjonction sont très intéressantes mais difficiles à automatiser (exemple: "accélération de l'électron et du proton" $\rightarrow$ "accélération de l'électron" et "accélération du proton", par contre "mise en évidence par l'épreuve d'effort et du caractère très distal" ne peut pas se transformer en "épreuve d'effort" et "épreuve du caractère très distal"). Plus largement ce problème concerne la décomposition des syntagmes à retenir. Autre exemple: on voudrait que *sténose* et *sténosée* soit réduits par une même racine par exemple *sténos*; Si *sténose* appartient au dictionnaire il sera réduit en *sténose* mais *sténosée* sera tronqué en *sténos* à cause du suffixe *-ée* ou lemmatisé en *sténoser* s'il fait partie du dictionnaire. Cet exemple simple met en lumière la difficulté d'atteindre un objectif de réduction "conceptuelle" qui ne marche que dans des cas simples (réduction du genre, de formes conjuguées). Il semble difficile de coupler un dictionnaire et une base de suffixes pour pouvoir faire converger les 2 processus à moins de "lemmatiser" le dictionnaire avec la base de suffixe. De cette manière on peut espérer voir réunir *convergera* et *convergence* du fait que *convergera* sera lemmatisé en *converger* lui même lemmatisé à cause du suffixe *-er* en *converg* et *convergence* sera lemmatisé en *convergence* lui-même *lemmatisé* en *converg* à cause du suffixe *-ence*. Un autre problème se pose quant à la nominalisation des termes. Notre méthodologie suppose une extraction des termes avant lemmatisation, dans ce cas les syntagmes "extraire de l'or" ou "extraction d'or" ne seraient pas considérés comme sémantiquement proches. La seule façon de les apparier est de réduire *extraire* et *extraction* sous une même forme cela suppose de considérer le mot *extrait* en *extraire* et ensuite en *extr* de même *extractions* en *extraction* et en *extr* donc d'avoir *aire* et *action* dans la base de suffixes. Le lemmatiseur lemmatise avec le suffixe le plus long, il ne faut donc pas avoir de suffixe *raction* ou *traction* dans notre base. A ce niveau les effets de bords sont finement à étudier. Une autre solution serait d'avoir un dictionnaire conceptuel avec les familles morphologiques les plus courantes.

Difficulté d'avoir des schémas stables et pertinents. Dans notre étude nous avons considéré que les cooccurrences sont marquées linguistiquement par des relations de type schémas morphosyntaxiques (association *terme/verbe*). Dans des applications spécifiques il est possible d'envisager d'autres types de schémas: comme association *adresse email/ terme* d'un message ou association d'une *date/terme* ou d'une *date/Nom propre*. Les schémas sont variés. Une étude approfondie doit être menée pour connaître les schémas intéressants par leur apport relationnel suffisamment marqué sémantiquement.

Difficulté d'identifier une gamme de motifs pertinents. Dans notre étude nous avons développé une étude empirique qui nous a amené à considérer un motif basé sur un terme central autour duquel gravitent 2 termes "pivots" qui contraignent l'association d'autres termes à ce triplet. Les motifs sont innombrables car développables autour d'une géométrie à symétrie plan/rotation. On peut disposer d'un motif à 2 termes centraux, à 3 termes pivots, de géométries cycliques dont les termes centraux forment une chaîne…etc. Notre modèle de motif ne prédit pas combien de motifs efficaces peuvent être générés. Il est clair que notre motif n'est pas unique. Par contre tout motif ne conduit pas forcément à des classes facilement interprétables. L'identification d'un motif repose donc sur un modèle phénoménologique et intuitif qui ne généralise pas les modèles de motifs exploitables.

Finalement les résultats obtenus par notre filtreur forgent une idée sur les capacités d'une approche basée sur la classification de termes pour classer des messages dans des répertoires thématiques. Mais une évaluation à la fois plus qualitative (grâce à un groupe d'utilisateurs-testeurs) et plus quantitative (grâce à des volumes de documents importants, plusieurs milliers de messages de forum de discussion par exemple) est nécessaire pour juger d'une efficacité substantielle.

Perpectives globales

Nécessité de traiter les anaphores basiques pour diminuer les biais de calcul de la matrice de cooccurrences.

La stratégie de sélection d'une classe par rapport à un autre pourrait être améliorée en utilisant la fonction décrite en au chapitre 3 §6 mais en utilisant les prédicats verbaux par exemple pour savoir quelle est la meilleure classe à conserver si elles ont plus de n termes en commun.

Peut-on isoler certains attributs à usages confinés pour dégager des structures associatives typiques ou groupes conceptuels?

Peut-on imaginer une représentation intermédiaire entre les paquets de mots issus des méthodes de classification et les logiques de prédicat issues de l'analyse en constituants phrastiques (au sens de Chomsky) ?

Une application telle que le filtrage d'information constitue une enceinte de modélisation informatique propice à l'imbrication d'un modèle interactif avec un modèle sémantique des connaissances issu des textes.
Comment mélanger le modèle du domaine et le modèle de l'utilisateur (facette *contenu*) ?
D'autres part la classification constitue une démarche globale d'association ayant un pouvoir discriminant faible pour un nombre restreint de relations. Une archive qui stockerait des messages d'une personne aurait la seule relation "possède ou ne possède pas l'adresse expéditeur"; cette règle ne dépendrait pas vraiment du contenu explicite des messages de l'archive. Dans ce cas la dépendance est plus liée à la structure du document.

Le laboratoire LIIA est impliqué dans une coopération avec un laboratoire russe (AIReC) de Pereslavl-Zalessky. En effet notre laboratoire accueille un chercheur dont la mission est d'intégrer différents modules existants, dont Galex, pour développer une application multi-agent de commerce électronique. Cette application doit permettre de constituer un catalogue automatiquement. Dans ce cadre, l'utilisation d'une méthode de classification automatique de termes aboutissant à une hiérarchie de thèmes permettrait de catégoriser les documents par profil de produit. Cette structure documentaire faciliterait la tâche d'un agent extracteur d'identifiants propres à générer une base de produits.

Remerciements

Les recherches, dans le cadre de ce projet de thèse, ont été effectuées au sein de l'équipe de recherche en ingénierie des connaissances de Strasbourg. Je tiens à remercier B.Keith, directeur de cette thèse, de m'avoir accueilli dans son laboratoire de recherche.

Je tiens à exprimer ma gratitude à F.Rousselot pour avoir toujours su encadrer mes travaux tout en m'accordant confiance, et initiative pour mener mes travaux de recherche au sein du laboratoire LIIA.

Que soient ici remerciés pour leur appui constant et leur participation active, F. de Beuvron et B. Migault, aussi bien sur le plan technique qu'organisationnel tout au long de ce parcours jusqu'à la rédaction de cette thèse.

Je suis reconnaissant aux spécialistes en cardiologie Dr.Frey et Dr.Barthel pour leurs commentaires éclairés du domaine médical, ainsi qu'à P.Guterl (Centre Réseau Communication de l'ULP) pour ses commentaires techniques sur la messagerie. Je suis également reconnaissant aux responsables des sociétés Neurosoft et Adit de m'avoir supporté pendant plusieurs mois dans leur établissement au cours de ma thèse.

Je n'oublie pas les rapporteurs du jury qui m'ont apporté des critiques précieuses sur ce mémoire et une autre perspective de mon travail.

Je tiens finalement à exprimer tous mes remerciements à ma famille ainsi que tous mes collègues et amis du laboratoire LIIA, qui par leurs participations, leur aide, leurs conseils et leur amitié m'ont permis d'accomplir ce travail de façon agréable.

BIBLIOGRAPHIE

ADBS, Le filtrage d'ïnformation, chapitre 5, ADBS, 1998

AGARWAL R., *Semantic Feature Extraction from Technical Texts with Limited Human Intervention*, University of Mississipi, Dissertation, 1995

ALBERT R., JEONG H. and BARABASI A.L. "Diameter of the World Wide Web", Nature, 401, 130-131, 1999

ALBERT R., BARABASI A.L., JEONG H. and BIANCONI G. "Power-law Distribution of the World Wide Web", Scince, 287, 2115a, 2000

ANDERBERG M.R., Cluster Analysis for Application, 1973

ASSADI H., *Construction d'ontologies à partir de textes techniques. Application aux systèmes documentaires*, thèse de l'Université Paris 6, 1998

AUGUSTSON J.G, MINKER J., "Derinving Term Relations for a Corpus by Graph Theoretical Clusters" in Journal of the American Society for Information Science, vol 21 n°2

BAKER L., MCCALLUM A., "Distributional Clustering of Words for Text Classification", dans les actes de *Special Interest Group on Information Retrieval Conference (ACM-SIGIR)*, 1998

BARABASI A.L. and ALBERT R. "Emergence of Scaling in Random Networks", Science, 286, 509-512, 1999

BASILI R., PAZIENZA M-T, VELARDI P. "A Context Driven Conceptual Clustering Method for Verb Classification in Corpus", Processing for Lexical Acquisition ed. B.Boguraev,J.Pustejovsky MIT 1996

BASILI R. ,M DI NANNI M.T PAZIENZA, "Engineering IE Systems: An Object-Oriented Approach" , SCIE99 ed Springer-Verlag , 1999

BEDECARRAX C., HUOT C., "Analyse relationnelle: des outils pour la documentation automatique", *la veille technologique:l'ïnformation scientifique,technique et industrielle, ed. Dunod*, 1991

BENZECRI J-P La taxinomie (T1). L'analyse des correspondances (T2) Dunod Paris 1973

BENZECRI J.P. Histoire et préhistoire de l'Analyse des Données Cahiers de l'Analyse des Données vol1 1976 n°1,2,3,4

BISSON G., "Clustering and Categorization", dans les actes de *International Centre for Pure and Applied Mathematics School (CIMPA)*, Nice (France), 1996

BOONE G., "Concept Features in Re :agent, an Intelligent Email Agent", *Conf on Autonomous Agents*, 1998

BOURIGAULT D., *LEXTER un extracteur terminologique*, Thèse de doctorat univ Paris 8, 1994

BRILL E. A Corpus-Based Approach to Language Learning PhD Thesis, U. Pennsylvania 1993

BROOKS T., "Topical Subject Expertise and the Semantic Model of Relevance Assessment", *Journal of Documentation*,Vol(51) n°4, 1995

BROWN P., PIETRA V.J.D., DESOUZA P.V., LAI J.C., MERCER R.L. Class-based N-gram Models for Natural Language Computational Linguistics vol.18 n°4 1992

BUSH V. As we may think Athlantic Monthly 176, July 1945 p101-216

CARPENTER G.A., "Distributed Learning, Recognition, and Prediction by ART and ARTMAP Neural Networks", *Neural Networks*,Vol(10) n°8, 1473-1494, 1997

CARPINETO C., ROMANO,G, "A Lattice Conceptual Clustering System and it Application to browsing Retrieval", *Machine Learning*,Vol(24), 95, 1996

CHANOD J., "Finite-State Composition of Fench Verb Morphology", *Xérox Technical paper*, 1994

CHANOD J., TAPANAINEN P., "Creating a Tagset, Lexicon and Guesser for a French Tagger", dans les actes de *Association for Computational Linguistics (ACL-SIGDAT)*, 1995

CHANOD J.P., TAPANAINEN P., "Tagging French - comparing a Statistical and a Constraint-based Method", dans les actes de *European Association for Computational Linguistics (EACL)*, Dublin, 1995

CHARNIAK E., *Statistical Language Learning*, MIT Press, 1993

CHEESEMAN P., KELLY J. SELF M. STUTZ J. TAYLOR W. FREEMAN D., "AutoClass: A Bayesian Classification System", dans les actes de *5th International Conference on Machine Learning*, San Francisco (Ca) Morgan Kaufman, 1988

CHEN H., HSU P., ORWIG R., HOOPES L., NUNAMAKER J., "Automatic Concept Classification", *Communications of the ACM*,Vol(37) n°4, 56-73, 1994

COHEN W., "Learning Rules that Classify Email", *Technical Report ATT*, 1995

CUTTING D., KARGER D., PEDERSEN J., TUKEY J. "Scatter/Gather : A Cluster-based Approach to Browsing Large Document Collections", dans les actes de *Special Interest Group on Information Retrieval Conference (ACM-SIGIR)*, 1992

DECAESTECKER, *Apprentissage en classification conceptuelle incrémentale*, Thèse Université Libre de Bruxelles, 1992

DEERWESTER S., DUMAIS S. FURNAS G. LANDAUER T. HARSHMAN R., "Indexing by Latent Semantic Analysis", *Journal of the American Society for Information Science*,Vol(41), pp391-407, 1990

DIDAY E., *La méthode des nuées dyamiques*, thèse de doctorat de l'Univerisité Paris 6, 1971

DIJKSTRA E.W., "Some Theorems on Spanning Subtrees of a Graph", Proc. Amsterdam 22, 1960 196-199

ELMAN J. , "On the Generality of Thesaurally derived Lexical Links" , dans les actes de *Journées internationales d'analyse statistiques de données textuelles (JADT)* ,Lausanne (Switzerland), 2000

ENGUEHARD C., *Apprentissage naturel automatique d'un réseau sémantique*, Thèse de doctorat Université de Compiègne, 1992

ESSONO J.M. Précis de linguistique générale ed. L'Harmattan 1998

ESTOUP J.B. Gammes sténographiques 4th édition Paris 1916

FAURE D., NEDELLEC C., "ASIUM: Learning Subcategorization Frames and Restrictions of Selection", dans les actes de *TextMining Workshop of the European Conference of Machine Learning (ECML)*, Chemnitz (Germany), 1998

FELDMAN R., "The Fact System", *http://www.cs.biu.ac.il :8080/*, 1996

FELDMAN R., DAGAN I., "Knowledge Discovery in Textual Databases (KDT)", dans les actes de *1st international Conference on Knowledge Discovery (KDD)*, Montréal (Canada), 1997

FIRTH J.R., *A Synopsis of linguistic theory 1930-1955*, In Studies in Linguistic-Analysis pp1-32 Oxford Philological Society Reprinted in F.R. Palmer (ed.) Selected Papers of J.R Firth 1952-1959, 1968

FISHER D., "Knowledge Acquisition via Incremental Conceptual Clustering", *Machine Learning*,Vol(2), 1997

FJALLSTROM P.O., "Algorithms for Graph Partitioning: A Survey", *Linkoping Electronic Articles in Computer and Information Science*,Vol(3) n°10, 1998

FONTENELLE T., W BRULS J JANSEN L THOMAS T VANALLEMERSCH S ATKINS U HEID B SCHULZE G GREFENSTETTE, "DECIDE Project", *CE Technical Report MLAP 93/19*, 1994

FORGY E.W. Cluster Analysis of Multivariate Data:Efficiency versus Interpretability of Classifications Biometrics Society Meetings, Riverside, California (Abstract in: Biometrics 21,3,768) 1965

GALE W.A., CHURCH K.W., YAROWSKY D., "A Method for Disambiguating Word Senses in a Large Corpus", *Computers and the Humanities*,Vol(26) n°5-6, 1992

GENNARI J., "A Model of Incremental Concept Formation", *Artificial Intelligence*,Vol(40), 1994

GEY F., "Inferring Probability of Relevance using the Method of Logisitic Regression", dans les actes de 17th annual ACM SIGIR Conf edited by W. Croft C Van Rijsbergen, SpringerVerlag London 1994 p222-241

GODIN R., MINEAU G., MISSAOUI R., MILI H., "Méthodes de classification conceptuelle basées sur les treillis de Galois et applications", *Revue d'Intelligence Artificielle*,Vol(9) n°2, 1995

GREFENSTETTE G. "Evaluation Techniques for Automatic Semantic Extraction: Comparing Syntactic and Window Based Approaches", in Corpus essing for Lexical Acquisition ed. B.Boguraev,J.Pustejovsky MIT 1996

GREFENSTETTE G., "SQLET : Short Query Linguistic Expansion Techniques, Palliating On-Word Queries by providing Intermediate Structure to Text", *Actes de la Conférence Recherche d'Information Assistée par Ordinateur (RIAO)*, Montréal (Canada), 1997

GREFENSTETTE G., TEUFEL S., "Corpus-based Method for Automatic Identification of Support Verbs for Nominalizations", dans les actes de *European Association for Computational Linguistics (EACL)*, Dublin (Ireland), 1995

GREFENSTETTE G. "SEXTANT: Extracting Semantics from Raw Text: Implementation Details, Heuristics" *Integrated Computer-Aided Engineering*, vol.1 n°6 pp. 527-536 1994

GROSS G., "Classes d'objets et description des verbes", *Langage*,Vol(115), 1994

GUTTMAN L. The Quantification of a Class of Attributes: a Theory and Method of a Scale Construction, in: The Prediction of Personal Adjustement; P Horst (Ed) SSCR New York 1941

HABERT B., NAULLEAU E., NAZARENKO A., "Symbolic Word Classification for Medium-Size Corpora", dans les actes de *Computational Linguistics Conference (COLING)*, Copenhague (Danemark), 1996

HAHN U., SCHNATTINGER K., "Knowledge Mining from Textual Sources", dans les actes de *International Conference on Information and Knowledge Management (CIKM)*, 1997

HANSEN P., JAUMARD B., "Cluster Analysis and Mathematical Programming", *Internal Report du GERAD G-97-10*, Montréal (Canada), 1997

HANSON S., BAUER M., "Conceptual Clustering Categorization and Polymorphy", *Machine Learning*,Vol(3), 343-372, 1990

HARRIS Z., *Mathematical Structure of Language*, ed. Wiley, 1968

HEARST M., "Contextualizing Retrieval of Full-Lenght Documents", *Report No UCB/CSD 94/789 University of California*, 1994

HEARST M., "Untangling Text Data Mining", dans les actes de *Association for Computational Linguitics Conference (ACL)*, University of Maryland (USA), 1999

HINDLE D., "Noun Classification from Predicate Argument Structures", dans les actes de *Association for Computational Linguistics (ACL)*, 1990

HULL D., "The TREC-7 Filtering Track : Description and Analysis", dans les actes de *Text Retrieval Conference (TREC)*, University of Maryland, 1998

IBEKWE-SAN JUAN F., *Recherche des tendances thématiques dans les publications scientifiques. Définition d'une méthodologie fondée sur la lingusitique*, thèse de doctorat de l'université de Grenoble3, 1997

JAMBU M. , *Classification automatique pour l'analyse des données (T.1)* , ed Dunod , 1978

KELUS A., LUKASZEWICZ J. Taksonomia wroclawska w zastosowaniu do zagadnien seroantropologii Archiwum Immunol. terap. Doswiadizalnej 1 245-254 , 1953

KARTTUNEN L., J.P CHANOD G GREFENSTETTE A SCHILLER, "Regular Expressions for Language Engineering", *Natural Language Engineering*,1-24, 1997

KETTERLIN A., *Découverte de concepts structurés dans les base de données*, Thèse de doctorat Université L Pasteur Strasbourg, 1995

KODRATOFF Y. , "Faut-il choisir entre science des explications et sciences des nombres?" , *in Induction symbolique et numérique à partir de données , Y.Kodratoof&E.Diday (eds) ed. Cepaduès.* , 1991

KOHONEN T., *Self-Organization and Associative Memory*, ed. Springer Verlag, 1989

KOHONEN T., "Exploration of Very Large Databases by Self-Organizing Maps", dans les actes de *International Conference on Neural Networks (ICNN)*, 1997

KOHRS A., MERIALDO B., "Clustering for Collaborative Filtering Applications ", actes de CIMCA, 1999

KOLODNER, "Reconstructive Memory a Computer Model", *Cognitive Science*,Vol(7), 281-328, 1983

KOWALSKI G., *Information Retrieval Systems, Theory and Implementation*, ed. Kluwer Academic Publishers, Chapter Document and Term Clustering, 1997

KRUSKAL J.B., "On the Shortest Spanning Subtree of a Graph and a Travelling Salesman Problem", dans les actes de *Amer. Math. Soc.*, 1956

LEBART L., SALEM A., BERRY L., *Exploring Textual Data*, ed. Kluwer Academic Publishers, 1998

LEBOWITZ, "Generalization from Natural Language Text", *Cognitive Science*,Vol(7), 1-40, 1983

LECLERC B., "Consensus of Classifications: the Case of Trees", dans les actes de Workshop of Classification Society of North America, Armherst (USA), 1996

LELU A., HALLEB M., DELPRAT B., "Recherche d'information et cartographie dans des corpus textuels à partir des fréquences de n-grammes", *Actes du colloque Journées d'Analyse Statistique des Données Textuelles (JADT)*, 1998

LERMAN I.C., "Sur l'analyse des données préalable à une classification automatique. Proposition d'une nouvelle mesure de similarité", Revue Mathématique et Sciences Humaines, 32, pp.5-15. , 1970

LEWIS D., "An Evaluation of Phrasal and Clustered Representations on a Text Categorization Task", dans les actes de *Special Interest Group on Information Retrieval Conference (ACM-SIGIR)*, Copenhague (Danemark), 1992

LI H., ABE N., "Word Clustering and Disambiguation Based on Co-occurrence Data", *Nec Lab Technical Report*, 1998

LINGRAS P., "Classifying Highways: Hierarchical Grouping vs Kohonen Neural Networks", *Technical Report, Algoma University College SaultMarys (Canada)*, 1994

LOUKACHEVITCH N., DOBROV B. Thesaurus as a Tool for Automatic Detection of Lexical Cohesion in Texts Journées internationales d'analyse statistique de données textuelles (JADT) 2000 Lausanne (Suisse)

LOVINS J.B. Development of a Stemming Algorithm Mechanical Translation and Computational Linguistics 11(6) pp 22-31 1968

MACSKASSY S., DAYANIK A., HIRSH H., "EmailValet: Learning User Preferences for Wireless Email", dans les actes de *Workshop User Modeling of the International Joint Conference in Artificial Intelligence (IJCAI)*, Stockholm (Suède), 1999

MACQUEEN J.B., "Some Methods for Classification and Analysis of Multivariate Observations", dans les actes de 5th Symposium on Mathematics, Statistics and Probability, Berkeley:University of California Press,pp 281-97 1967

MANNING C., SCHUTZE H., *Foundations of Statistical Natural Processing*, ed. MIT Press, 1999

MARCOTORCHINO F., "La classification automatique aujourd'hui", *Technical Report IBM*, 1991

MARI A., SAINT-DIZIER P. Nature et formation de classes sémantiques de verbes pour l'extraction de connaissances dans des textes Journées internationales d'analyse statistique de données textuelles (JADT) 2000 Lausanne (Suisse)

MARKOV A.A. Ob odnom primenenii statistitcheskogo metoda (une application de méthode statistique) Izvestia Imperialisticheskoï Akademii Naouk, 6(4) pp239-42 1916

MAYR E.G, LINSLEY E.G, USINGER R.L. Methods and Principles of Systematic Zoology ed MacGraw-Hill New-York 328pp 1953

MCKUSICK, LANGLEY P., "Constraints on Tree Structure in Concept Formation", *International Journal of Policy and Informatics Systems*, 1980

MEMMI D., GABI K., MEUNIER J.G., "Dynamical Knowledge Extraction from Texts by Art Networks", dans les actes de *4th Internationl Conference on Neural Networks and their Applications (NeuroAp)*, Marseille (France), 1998

MICHALSKI R., "Knowledge Acquisition through Conceptual Clustering. A Theoretical Framework and Algorithm for, Partitioning Data into Conjunctive Concepts Analysis", *International Journal of Policy and Informatics Systems*, 1980

MICHELET B., *L'analyse des associations* , Thèse de doctorat Paris 7, 1988

MINKER J., G WILSON B ZIMMERMAN, "An Evaluation of Query Expansion by Addition of clustered Terms for a Document Retrieval System", *Information, Storage and Retrieval*,Vol(8), 1972

MURTAGH F. , *Mutivariate Data Analysis* , D.Reidel Publishing Company, Kluwer Group , 1987

MURTAGH F., "A survey of recent advances in hierarchical clustering algorithms", The Computer Journal, 26, 354-359, 1983

MURTAGH F. ,GUILLAUME D., "Distributed Information Search and Retrieval for Astronomical Resource" , *Proceedings Library & Information Science Astronomy III, astro. Soc of the Pacific* , 1998

NAZARENKO A. ,ZWEIGENBAUM P., BOUAUD J., HABERT B., "Corpus-Based Identification and Refinement of Semantic Classes" , *Proceedings of the Annual Symposium of Computer Applications in Medical Care* ,Nashville (USA), 1997

NEDELLEC C. , "Apprentissage automatique de connaissances à partir de corpus" , *Séminaire Atala Mars Paris* , 1999

NEEDHAM R.M., "Research on Information Retrieval, Classificaton and Grouping", *PhD thesis C.L.R.U Cambridge University*, 1961

NIWA Y., Y NITTA, "Co-occurrence Vectors from Corpora vs. Distance Vectors from Dictionaries", dans les actes de *Computational Linguistics Conference (COLING)*, 1994

OAKES M., *Statistics for Corpus Linguistics*, ed. Edinburgh Textbooks in Empirical Linguistics, 1998

OSIPOV G.S., *Knowledge Acquisition by Intelligence Systems. Theory and Technology*, ed. Fiziko-Matematichekaya Literatura, 1997

OUESLATI R., "Manuel de Startex", *note Interne du LIIA*, 1996

OUESLATI R., "Une méthode d'exploration de corpus pour l'acquisition automatique de relations syntaxiques", *Technical Report laboratoire LIIA Strasbourg*, 1997

PEARSON K. On the Coefficient of Racial Likeness Biometrika 18 105-117 1926

PEREIRA F., N TISHBY, L LEE, "Distributional Clustering of English Words", dans les actes de *30th conference of the Association for Computational Linguistics (ACL)*, Jerusalem (Israel), 1993

PHILLIPS M. , Lexical Structure of text Discourse Analysis Monograph 12, English Language Research, University of Birmingham 1989

PINCEMIN B., *Diffusion ciblée automatique d'informations: conception et mise en oeuvre d'une linguistique textuelle pour la caractérisation des destinataires et des documents*, Thèse de doctorat Paris 4, 1999

PLOUX S., VICTORRI B., "Construction d'espaces sémantiques à l'aide de dictionnaires de synonymes", *Traitement Automatique de la Langue (TAL)*, 1998

POLANCO X., GRIVEL L., ROYAUTE J., « How to do Things with Terms in Informatics :Terminological variation and Stabilization as Science Watch Indicators », dans les actes de 5th International Conference on Scientometrics & Informetrics, Chicago (USA), 1995

PORTER M.F. An Algorithm for Suffix Stripping Program 14 p130 1980

PRIM R.C. Shortest Connection Matrix Network and Some Generalisations Bell System Techn. J. 36 1957 1389-1401

QUINLAN R., "Induction Decision Trees", *Machine Learning*,Vol(1) n°1, 1986

RAJMAN M. ,BESANCON R., "Text Mining: Natural Languages Techniques and Text Mining Applications" , *Proceedings of the International Federation for Information Processing (IFIP)* , 1997

RAPP L., "La messageire électronique", Que-sais-je, ed. PUF, 1998

RASTIER F., "Le terme: entre ontologie et linguistique", *La Banque des Mots*,Vol(7), 1995

REGNIER S Sur quelques aspects mathématiques des problèmes de classification automatique ICC Bulletin 4 1965 p173-191

REINERT M., "Un logiciel d'analyse lexicale (Alceste)", *Les Cahiers de l'Analyse des Données*,Vol(4), 471-484, 1986

RENNIE J,"IFile: an Application of Machine Learning to Email",

http://www.cs.cmu.edu/~jr6b/papers/ifile98.html, 1998

RESNIK P., "Wordnet and Distributional Analysis. A Class-Based Approach to Lexical Discovery.", *Workshop §Notes, Statistically-Based NLP Techniques, page 54-64 AAAI*, 1992

ROGERS D.J., TANIMOTO T.T. A Computer Program for Clasifying Plants Science 132 99115-1118 1960

ROUSSEL A., *Bali : la boite aux lettres intelligente*, Thèse de doctorat Université de Caen/ CNET, 1998

ROUSSELOT F., FRATH P. , OUESLATI R., "Exploration conceptuelle par repérage de segments répétés, synthèse et utilisation de schémas morphosyntaxiques", dans les actes de *Workshop on Corpus-Oreinted Semantic Analysis of the European Conference on Artificial Intelligence (ECAI), Budapest (Hongrie)* 1996

ROUX M. Algorithmes de classification ed Masson 1985

RUGG D. Experiments in wording questions Public Opinion Quarterly 5 91-92 1941

SAHAMI M., DUMAIS S., Heckerman D., Horvitz E. "A Bayesian Approach to Filtering Junk E-Mail" in proceedings of the AAAI Symposium, 1998

SALTON G, McGILL M.J. Introduction to Modern Information Retrieval New-York: McGraw-Hill 1983

SAPORTA G., *Probabilités, analyse de données et statistique*, ed. Technip, 1988

SILVERSTEIN C., HENZINGER, "Analysis of a very large web search engine query log", SIGIR Forum, vol 33, N°3, 1999

SMADJA F., MCKEOWN K., "Automatically Extracting and Representing Collocations for Language Generation", dans les actes de *Association for Computational Linguistics Conference (ACL)*, Pittsburgh (USA), 1990

SMADJA F., MCKEOWN K., "Automatically Extracting and Representing Collocations for Language Generation", dans les actes de *Association for Computational Linguistics Conference (ACL)*, Pittsburgh (USA), 1990

SOKAL R.R., SNEATH P.H.A. Principles of Numerical Taxonomy ed W.H. Freeman and Company 1963

SPARCK-JONES K., *Synonymy and Semantic Classification*, ed. Edinburgh University Press, 1987

SPARCK-JONES K. What is the role of NLP in Text Retrieval in Natural Language Information Retrieval ed. T Strzaklwski, Kluwer A.P. 2000

SUSSNA M., "Information Retrieval using Semantic Distance in Wordnet", *University of California (San Diego), Technical Report*, 1995

TISHBY N., F PEREIRA W BIALEK, "The Information Bottleneck Method", dans les actes de *37th Annual Allerton Conference on Communication Control and Computing*, 1999

TORRES-MORENO J.M, VELASQUEZ-MORALES P., MEUNIER J.G Classphères: une réseau increémental pour l'apprentissage non supervisé appliqué à la classification de textes Journées internationales d'analyse statistique de données textuelles (JADT) 2000 Lausanne (Suisse)

TRIER J., *Der Deutsche Wortschatz im Sinnbezirke des Verstandes*, Die Geschichte eines Sprachlichen Feldes, Heidelberg, 1931

WARD J.H. Hierarchical Grouping to Optimize an Objective Function Journal of the American Statistical Association, 58(301) pp236-44 1963

WATERMAN S. Distinguished Usage in Corpus Processing for Lexical Acquisition ed. B.Boguraev,J.Pustejovsky MIT 1996

WILHEIM, MAURER, *Les compilateurs : théorie, construction, génération*, Masson, 1990

WILLIAMS W.T., LANCE G.N. Logic of Computer-based Intrinsic Classifications Nature 207 1965 159-161

WONG M.A. A Hybrid Clustering Method for Identifying High Density Clusters Journal of the American Statistical Association 77 841-847 1982

YAROWSKY D., "Word-Sense Disambiguation using Statistical Models of Roget's Categories trained on Large Corpora", dans les actes de *Computational Linguistics Conference (COLING)*, Nantes (France), 1992

YULE G.U. The statistical Study of Literacy Vocabulary Cambridge University Press, 1944 Reprinted in 1968 by Archon Books Hamden Connecticut

ZERNIK U. Train 1 vs Train 2: Tagging Word Sense in a Corpus in Zernik,U (ed.) Lexical Acquisition: Exploiting on-Line Resources to Build a Lexicon, Hillsdale, NJ: Lawrence Erlbaum Associates. 1991

ZIPF G.K The psychology of Language, an Introduction to Dynamic Philology, Houghton-Mifflin Boston 1935

ZYTKOW J., ZEMBOWICZ R., "Contingency Tables as the Foundations for Concepts, Concept Hierarchies, and Rules; the 49er System Approach", *Fundamenta Informaticae*,Vol(30), 383-399, 1997

PUBLICATIONS DE L'AUTEUR

En cours

TURENNE N., "Modèle du consensus pour l'étiquetage de classes de termes", Rapport Technique LIIA, 2000

Avec comité de lecture

TURENNE N., "Term Clusters Evaluation by Montecarlo Sampling", *Actes du colloque Journées d'Analyse Statistique des Données Textuelles (JADT)*, Lausanne (Suisse), 2000

TURENNE N., "Apprentissage d'un ensemble pré-structuré de concepts d'un domaine: l'outil GALEX", *Mathématiques, Informatique et Sciences Humaines*,Vol(148), 41-71, ISSN 0995-2314 1999

TURENNE N., "Learning of User Profile with Clustering" , *Technical Report (accepté aux Journéées de la Société Francophone de Classification SFC'98 Montpellier) , LIIA-ENSAIS* , 1998c

TURENNE N., ROUSSELOT F., "A new Reformulation System: the SAROS Tool", dans les actes de *Knowledge Acquisition Workshop (KAW)*, Banff (Canada), 1998b

TURENNE N., ROUSSELOT F., "Evaluation of 4 Clustering Methods used in Text-Mining", dans les actes de *ECML workshop on textmining*, Chemnitz (Germany), 1998a

Sans comité de lecture

TURENNE N., GUILLERM G. "Neuromail: un outil de filtrage et de routage de messages électroniques", Rapport de projet technologique accepté par l'ANVAR Alsace (Agence Nationale de Valorisation de la Recherche) 1998

TURENNE N., "Dictionnaire des sciences et de l'informatique", ed. La Maison du Dictionnaire CD-ROM LexPro 3.0 ISBN 2-85608-154-1 1998

TURENNE N., THIL J., "La veille sur Internet- des outils avancés pour identifier et extraire l'information", ed La Tribune des Industries de la Langue et de l'information électronique ISSN 1148-7666 n°25-26 1997

GLOSSAIRE

Terme	Traduction anglaise	Définition
Agent	Agent	Module réalisant une tâche automatique de façon autonome avec interaction avec d'autres agents et capable d'auto-organisation
Analyse des Données	Data Analysis	Discipline de la statistique visant à produire une analyse exploratoire des données à base de calcul matriciel, d'analyse de graphe ou calcul probabiliste.
Classe	Cluster	Groupe de termes qui ont des caractéristiques communes
Classement	Classification	Processus d'affectation d'un objet à une classe
Classification	Clustering	Processus de découverte de classes
Clique	Clique	Graphe dont tous les sommets sont liés deux à deux
Concept	Concept	Se définie soit à partir de conditions nécessaires et suffisantes soit à partir d'un prototype soit à partir d'un ensemble d'exemples
Cooccurrence	Co-occurrence	Association syntaxique d'un terme avec un autre dans un texte modulo une fenêtre de n mots (synonyme : collocation)
Corpus	Corpus	Ensemble de textes parlant d'un même sujet
Distribution	Distribution	Suite discrète de valeurs d'une fonction caractérisant une variable
Etiquette	Tag	Nom que l'on attribue à un objet ou un ensemble d'objet
Extraction d'Information	Information Extraction	Domaine de la recherche documentaire visant à extraire dans un texte des valeurs de champs d'un groupe d'attributs pour une entité. (exemple: personne,adresse,téléphone)
Fenêtre	Window	Liste de mots contigus de largeur définie (exemple : la fenêtre de largeur 2 autour de

		‘pot’ dans la phrase ‘j’ai vu le pot sur la table’ est ‘ vu le pot sur la ’)
Filtrage d’Information	Information Filtering	Domaine de la recherche documentaire visant à autoriser ou non le transfert d'un document à une catégorie ou un utilisateur.
Fouille de textes	Text-mining	Domaine applicatif croisé entre la statistique et la traitement automatique de la langue qui vise à identifier des relations dans un texte.
Fréquence	Frequency	Nombre d’occurrences (nombre de positions) dans une suite de messages
Hapax	Hapax	Mot de fréquence 1
Incrémentalité	Incrementality	Caractère d’un objet dont la taille peut être augmentée
Instance	Item ou instance	Elément d’une classe ou d’un centre d’intérêt
Loi de puissance	Power law	loi qui exprime une variable par une puissance ou le logarithme d’une autre (exemple loi de Zipf)
Monoterme	Monoterm	Terme composé d'un seul mot
Multiterme	Multiterm	Groupe de monotermes
Profil utilisateur	User profile	Ensemble de caractéristiques textuelles ou événementielles qui qualifient le comportement et les centres d'intérêts d'un utilisateur
Recherche Documentaire	Information Retrieval	Application consistant à trouver des documents en fonction d'un requête de mots clés.
Terme	Term	Chaîne de caractères significative d'un domaine de connaissances et partagée par les experts du domaine.
Transposer	Transpose	Calculer les coefficients matriciels m’(i,j) = m(j,i) d’une matrice de référence

ANNEXE 1

Outils

Moteurs de recherche Internet commerciaux ("portails" ou "gateways")

Nom de l'outil	Adresse URL
Crawlers	
Altavista	http://www.altavista.com
HotBot	http://www.hotbot.com
Webcrawler	http://www.webcrawer.com
Lycos	http://www.lycos.com
Microsoft	http://search.msn.com/
Opentext	http://index.opentext.net/
AOL	http://www.aol.com/
Voilà	http://www.voila.com
Ecila	http://www.ecila.com
Lokace	http://www.lokace.com
Goto	http://www.goto.com
Google	http://www.google.com
Dejanews	http://www.deja.com
Excite	http://www.excite.com/
Netscape	http://www.netscape.com/
Infoseek	http://www.go.com/
FAST	http://ussc.alltheweb.com/
Classifieurs	
Yahoo	http://www.yahoo.com
Nomade	http://www.nomade.com
Magellan	http://www.mckinley.com/
Looksmart	http://www.looksmart.com
Requêteurs de crawler	
Metacrawler	http://www.metacrawler.com
All in one	http://www.albany.net/allinone/
Internet Sleuth	http://www.infosleuth.com
Search Com	http://www.search.com
The inquirer	http://www.inquirer.com
Search	http://www.search.com/
Internet sleuth	http://www.isleuth.com/
Mamma	http://www.mamma.com/
Beaucoup	http://www.beaucoup.com
Askjeeves	http://www.askjeeves.com
All-4-one	http://all4one.com
Requêteur "intelligents"	
Pacprospector	http://www.pacprospector.com
Northern light	http://www.nlsearch.com
Caloweb	http://www.caloweb.com/

Agents personnalisés	
Copernic	http://www.copernic.com
Entry Point	http://www.pointcast.com
Bbackweb	http://www.backweb.com
Net Attaché Pro	http://www.tympani.com/
Webseeker	http://www.pacprospector.com
Autonomy	http://www.hotbot.com
Tierra Highlight	http://www.tierra.com/
Périclès	Datops
Netseeker	The Coriolis Group

Moteurs d'indexation commerciaux

Nom de l'outil	date de création	Auteurs/Société
Altavista	1994	Digital
Basis	1985	Basis
Darwin/Intuition	1990	Cora
Fulcrum	1985	Fulcrum
PLS	1996	Personal Library Software
Spirit/Sense	1972	TGID
Smart	1971	G Slaton, Buckley
Zy Filter	1997	ZyLab
RetrievalWare		Excalibur Mitre
Topic/Search97	1990	Verity corp
TextWise/DR-LINK		InfoMall
MatchPlus		HNC

Moteurs d'indexation et de diffusion

Nom de l'outil	Auteurs/Société
PointCast	http://www.pointcast.com
FishWrap	http://fishwrap-docs.www.media.mit.edu/docs
SFGate	http://www.sfagte.com
MESSIE	ENST
Etat Partenaire	http://www.adit.fr/

Logiciels commerciaux (classification sur des données textuelles)

Nom de l'outil	Auteurs/Société
Airs/mediator	Université de Paris 6
Alceste	Image
Aleth class	Lexiquest (Erli)
Aperto Libro	Inforama
AutonomyAgentWare	Autonomy
Calliope	Dextria
Cambio	Data Junction
ClearStudio	ClearForest

Cross-Reader	Insight
DataSet	Intercon Systems
Dataview	Université de Marseille
DigOut4U(noémic-taiga)	Arisem
DocsFulcrum	Hummingbird
Evalog	Ingénia
Gingo	Trivium
Hyperbase	Université de Nice
Infoscan	Logiciels Machina Sapiens
InQuery	Dataware
Intelligent Text Miner	IBM
Intex	Université de Paris 6
KeyviewIntranetSpider	Verity,
KNOT	Interlink
KOD	Cisi
Lexiquest	Lexiquest (Erli)
LinguistX	Inxight
Monarch	Datawatch Corporation
Neuronav	Université de Paris 8
NeuroText	Grimmer Logiciels
Periclès	Datops
Sampler	Cisi
SelectResponse	MindWave Software
Sémiomap	Semio
Spad/T	Cisia
Sphinx Lexica	Le Sphinx
SRA	SRA
Taiga	Krummech(Noemic)/Thomson
TechOptimizer	InventionMachine,
Tétralogie	Université de Toulouse
Tewat (technology watch)	IBM
TextAnalyst	Megaputer
TextSmart	SPSS
ThemeScape	Cartia
TKN	Triada
Tks (Text Knowledge Server)	IBM
Tropes	Acetic
Umap	Trivium
V-Strat	Digimind
Websom	Helsinki University
WordStat	Provalis,
Xcize	Brosis

Logiciels commerciaux de filtrage automatique

Fournit une description automatique de sujets sélectionnés apparaissant dans une variété de sources internes et externes Fournit les moyens d'établir des agents pour localiser l'information d'intérêt Utilise des agents basés sur l'extraction de profile pour router l'information. La plupart de ces agents utilise des sujets ou évènements spécifier à l'avance par l'utilisateur.

Nom de l'outil	Auteurs/Société
Agent Server	Verity
Agent Studio	Roving Software Inc
Agentware Knowledge Server	Autonomy
Classifier	Technology in Marketing Limited
Compass Server	Netscape
Convectis	Aptex
DocuMine	COM.sortium
dynaSigth	Arcplan
Echo	InfoMation Publishing Company
Empower Alert	Muscat
FDF 3 Fast Data Finder	Paracel
GroupMaster	Revenet Systems
InfoMagnet	CompassWare Development Inc
InfoScout	Mayflower Software
InfoTap	SAS Institute
InfoWarning	Dataops
InRoute	Soveriegn Hill Software
Intellagents KnowledgeSets	Sageware
IntraExpress	Diffusion
Intraspect Knowledge Server	Intraspect
KnowledgeX	IBM
MyEureka	Information Advantage
NewsEDGE	Desktop Data
Pattern:Alert	Magnify Inc
Perspecta	Perspecta
Plumtree Server	Pumtree Software
PortalWare	Glyphica
Profile Publisher	Extelligent Inc
Relevant Personal Edition	Ensemble Information Systems Inc
SageWave	SageMaker
2Share	2Bridge Software
Verge Insigth	Intraspect
WebMind	Intelligenesis Corporation

ANNEXE 2

Critères de qualification d'un logiciel de Fouille de Texte

1 Utilisation Off-Line
2 Utilisation On-Line
3 Elimination des doublons
4 Constitution d'un thésaurus
5 Conserve un historique des visites
6 Permet des requêtes sur les résultats
7 Mobile sur le réseau
8 Classement par type de site
9 Offre un descriptif de chaque résultat
10 Personnalisable
11 Propose un dictionnaire de synonyme pour affiner les requêtes
12 Mise à jour automatique de l'agent
13 Programmation horaire disponible
14 Alerte par e-mail
15 Permet de spécifier l'adresse spécifique d'un site à explorer
16 Analyse les pages en profondeur en suivant les liens hypertextes
17 Avertit quand un mot clé précis est publié sur un site
18 Analyse la sémantique des textes
19 Permet la séléction des moteurs de recherche à consulter
20 Permet le classement des moteurs de recherche à explorer
21 Ajout de nouveaux mots clés en fonction des mots clés saisis
22 Organisation des signets
23 Trouve les pages qui sont liées à un U.R.L
24 Recherche sur les nouveaux sites Web (du jour ou de la semaine)
25 Recherche par titre de pages web
26 Autonome
28 Possibilité d'entrer des commentaires sur les résultats obtenus
29 Souligne les mots que l'on recherche dans les textes trouvés
30 Recherche en fonction de notre connaissance du domaine
31 Possibilité d'enrichir le dictionnaire par des synonymes
32 Partage avec d'autres experts du domaine
33 Donne la structure d'un site
34 Spécifie le type de fichiers à aspirer dans un site (ou à traiter)
35 Limite le nombre de niveau à aspirer (ou à traiter)
36 Fait état de tous les liens disponibles dans une page
37 Recherche de mots sur titres et contenus des pages aspirées
38 Recherche les adreses e-mail d'un site
39 Recherche les pages qui contiennet un mot clé précis sur un site
40 Aspire tous les types de liens (HTML,ASP,CGI,HTX,SHTML)
41 Renvoie à l'endroit du texte où le mot est trouvé
42 Utilise la proximité des mots
43 Donne un descriptif complet du site (DNS,traffic,rapidité...)
44 Classe par concept
45 gère le cache du navigateur (création d'index)
46 Limite les domaines par pays
47 Recherche les DNS commençant par le mot recherché
48 Recherche par page similaire
49 Génère un index des phrases associés à la recherche
50 Affiche les résultats suivant les différents filtres mis en place

ANNEXE 3

Tableau des distances

p_a,g_a et p_b,g_b sont respectivement les poids et centres de gravité des deux classes a et b
x est une coordonnée
i et j sont deux points

distance euclidienne usuelle	$d(i,j)=\left(\sum_k \left(x_i^k - x_i^k\right)^2\right)^{1/2}$
distance de Mahalanobis	$d^2 = (x-\mu)'S^{-1}(x-\mu)$ x est un vecteur ou une matrice à p colonnes S est la matrice de covariance $(p \times p)$ μ est vecteur moyen de longueur p
distance de Minkowski	$d(i,j)=\left(\sum_k \left(x_i^k - x_i^k\right)^q\right)^{1/q}$
distance L1	$d(i,j)=\left(\sum_k \left\|x_i^k - x_j^k\right\|\right)^{1/2}$
distance de Ward	$\delta(a,b)= \frac{p_a p_b}{p_a + p_b} d^2(g_a, g_b)$
distance du chi-2	$d(i,j)=\left(\sum_k \frac{n}{n_{.j}} \left(\frac{x_i^k - x_j^k}{p} \right) \right)^{1/2}$
distance de Manhattan/City-Block	$d(i,j)=\sum_k \left\|x_i^k - x_j^k\right\|$
Distance de Hamming	$d(i,j)=\sum_k \left(x_i^k - x_i^k\right)$
distance euclidienne moyenne	$d(i,j)=\left(\frac{\sum_k \left(x_i^k - x_i^k\right)}{n} \right)^{1/2}$
Différence moyenne de caractéristiques	$d(i,j)=\frac{\sum_k \left\|x_i^k - x_j^k\right\|}{n}$
distance de Chark	$d(i,j)=\left(\frac{1}{n} \sum_k \left(\frac{x_i^k - x_i^k}{x_i^k + x_i^k} \right)^2 \right)^{1/2}$
distance de Pearson	$d(i,j)=\frac{1}{n} \sum_k \left(\frac{\left(\bar{x}_i^k - \bar{x}_i^k\right)^2}{\frac{s_i^{k\,2}}{n_i} + \frac{s_i^{k\,2}}{n_j}} \right) - \frac{2}{n}$
distance de Rogers-Tanimoto (semi-métrique)	d(i,j)= - ln s(i,j) où s est un coefficient d'association, 0<s<1
semi-métrique d'association (ne vérifie pas l'axiome de transitivité)	d(i,j)=1-s(i,j) où s est un coefficient d'association

distance euclidienne généralisée $q^{kk'}=1$ forme usuelle, $q^{kk'}$ est la matrice unité $q^{kk'}=\frac{1}{\sigma_k^2}$ formule du nuage centré réduit $q^{kk'}=\frac{1}{\max(x_{ik}-x_{ik'})}$ formule pondérée par l'écart maximum	$d(i,j)=\sqrt{q^{kk'}(x_{ik}-x_{jk})(x_{ik'}-x_{jk'})}$
distance de Levenshtein	$D(A_i,B_j)=\begin{cases} D(A_i,B_{j-1})+D_{insert}(b_j) \\ D(A_{i-1},B_j)+D_{insert}(a_i) \\ D(A_{i-1},B_{j-1})+D_{substitute}(a_i,b_j) \end{cases}$ où $D_{insert}(x)$ est le coût pour insérer x et $D_{substitute}(a_i,b_j)$ est le coût pour substituer x à y
distance de l'information mutuelle	$d(i,j)=\sqrt{H(I,J)-H(Q(i,i'),J)}$ avec $H(I,J)=H(f_{IJ},f_If_J)=\sum_{i,j}f_{ij}\ln(\frac{f_{ij}}{f_if_j})$ et $H(Q,J)=H(f_{QJ},f_Qf_J)=\sum_{q,j}f_{qj}\ln(\frac{f_{qj}}{f_qf_j})$
distance de Kullback-Leibler ou de l'entropie relative	$d(p_n\|p_n')=\sum_v p_n(v).\frac{p_n(v)}{p_n'(v)}$
distance entre graphes	$d(i,j)=\text{card}(G(i)\,\Delta\,G(i'))$ G(i) et G(i') sont des graphes associés aux rangs k(i,j) et k(i',j) G(i)={(j,j'); j≠j'; j>j'; k(i,j)<k(i,j')} G(i) Δ G(i')={(i,i'); j≠j'; j>j'; k(i,j)<k(i,j') … k(i',j)>k(i',j')} ∪{(j,j'); j≠j'; j>j';…k(i,j)>k(i,j') et k(i',j)<=k(i',j')}
distance de l'écart maximum	$d(i,j)=\max_{k\in K}(x_i^k-x_j^k)$
distance de Benzécri	$d(i,j)=\sqrt{k(\bar{x}_j)\left(\frac{x_{ij}}{x_i}-\frac{x_{i'j}}{x_{i'}}\right)^2}$
distance de Canberra	$d(i,j)=\sum_k\left(\frac{\lvert x_i^k-x_i^k\rvert}{x_i^k+x_i^k}\right)$
distance du cosinus	$d(i,j)=\frac{\sum_i a_i.b_i}{\sqrt{(a_i)^2.(b_i)^2}}$

Tableau des coefficients

a = nombre de caractéristiques communes
b = nombre de caractéristiques possédées par i et pas par j
c = nombre de caractéristiques possédées par j et pas par i
d = nombre de caractéristiques que ne possèdent ni i ni j

	+	-	
+	a	c	nj+
-	b	d	nj-
	ni+	ni-	n

ni+=a+b (nombre de caractéristiques que possède i)
ni-= c+d (nombre de caractéristiques que ne possède pas i)
nj+=a+c (nombre de caractéristiques que possède j)
nj-= b+d (nombre de caractéristiques que ne possède pas j)
n est le nombre total de caractéristiques

Jaccard	$\frac{a}{a+b+c}$
Dice ou Czkanowski	$\frac{2a}{2a+b+c}$
Ochiaï	$\frac{a}{\sqrt{(a+b)(a+c)}}$
Russel et Rao	$\frac{a}{a+b+c+d}$
Rogers et Tanimoto	$\frac{a+d}{a+d+2(b+c)}$
Tanimoto	$\frac{a}{b+c}$
Zubin ou simple matching	$\frac{a+d}{a+d+2(b+c)}$
Hamann	$\frac{a+d-(b+c)}{a+d+2(b+c)}$
Yule	$\frac{ad-bc}{ad+bc}$
Phi de Pearson	$\frac{ad-bc}{\sqrt{n_{i+}\, n_{i-}\, n_{j+}\, n_{j-}}}$
Kulczinski	$\frac{b+c}{a+d}$
Kulczinski	$\frac{1}{2}\left[\frac{a}{n_{i+}}+\frac{a}{n_{j+}}\right]$
Sokal & Sneath	$\frac{1}{4}\left[\frac{a}{n_{i+}}+\frac{a}{n_{j+}}+\frac{d}{n_{i-}}+\frac{d}{n_{j-}}\right]$

Sokal & Sneath	$\frac{ad}{\sqrt{n_{i+}\,n_{i-}\,n_{j+}\,n_{j-}}}$
Sokal & Sneath	$\frac{a}{a+2(b+c)}$
Sokal & Sneath	$\frac{a+d}{b+c}$
Sokal & Sneath	$\frac{2(a+d)}{2(a+d)+(b+c)}$
Simpson	$\frac{a}{\min(n_{i+},n_{i-})}$
Kocher-Wong	$\frac{a.n}{n_{i+}.n_{i-}}$
Sokal-Michener	$\frac{a+d}{a+d+b+c}$
Sokal-Michener	$\frac{b+c}{a+d+b+c}$
Roux	$\frac{a+d}{\min(b,c)+\min(n-b,n-c)}$
Roux	$\frac{n-ad}{\sqrt{n_{i+}\,n_{i-}\,n_{j+}\,n_{j-}}}$
Euclidien pondéré	$\frac{(b+c)^2}{a+d+b+c}$
Michelet	$\frac{a^2}{bc}$
Fager-McGowan	$\frac{a}{\sqrt{(a+b)(a+c)}}+\frac{1}{2\sqrt{(a+b)}}$
McConnoughy	$\frac{a^2-bc}{\sqrt{(a+b)(a+c)}}$
Φ^2	$\frac{(ad+bc)^2}{(a+b)(a+c)(b+c)(b+d)}$
MI (infomration mutuelle)	$\ln\left[\frac{a.n}{(a+b)(a+c)}\right]$
MI3 (information mutuelle pondérée)	$\ln\left[\frac{a^3.n}{(a+b)(a+c)}\right]$
X^2(khi 2 avec correction de Yates n/2)	$\frac{n\left(\lvert ad-bc\rvert-\frac{n}{2}\right)^2}{(a+b)(c+d)(a+c)(b+d)}$

Collocation normalisée	$\frac{a}{(b+c-a)}$
Geffroy (ε représente la distance entre les cooccurrents i et j)	$\frac{a}{b}\sum_i \frac{1}{\varepsilon_i}$
Dunning	$2(a\log a+b\log b+c\log c+d\log d)$ $-(a+b)\log(a+b)-(a+c)\log(a+c)$ $-(b+d)\log(b+d)-(c+d)\log(c+d)+$ $(a+b+c+d)\log(a+b+c+d)$
Daille	$H(i,k)=a(i,k)\log a(i,k)$ $-\sum_j a(i,j)\log a(i,j)$

Tableau des coefficients de corrélation

Moment d'ordre 3	$\frac{\sum_i (X_{ij}-X_j)(X_{ik}-X_k)}{\sqrt{\sum_i (X_{ij}-\overline{X}_j)^2 (X_{ik}-\overline{X}_k)^2}}$
Coefficient de Spearman ($\sum qQ$ est la sommes des corrélations des membres d'un groupe avec un autre groupe, Δ est la somme des corrélations entre les membres d'un groupe, et q est le nombre d'éléments)	$\frac{\sum qQ}{\sqrt{q+2\Delta q}\,\sqrt{Q+2\Delta Q}}$
Coefficient de Spearman	$1-\frac{6}{(cardJ-1)cardJ}\sum_j (k(i,j)-k(i',j))$
Coefficient de Kendall	$\frac{2(n^+-n^-)}{cardJ(cardJ-1)}$ n^+ est le nombre de fois où k(i,j)-k(i,j') est de même signe que k(i',j)–k(i',j') n^- est le nombre de fois où ces expressions sont de signe contraire

ANNEXE 4

Algorithmes de partitionement de graphe

Méthode locales

Algorithme génétique

---

on réordonne les nœuds; le nouvel ordre est l'ordre dans lequel les nœuds sont visités par une recherche de premièere-largeur commençant à un nœud aléatoire;

ainsi une population initiale consistant en bisections équilibrées est générée;

Pour former une nouvelle génération on choisit une paire de bisections. Chaque bisection est sélectionnée avec une probabilité qui dépend de sa taille de coupure: plus la taille de coupure est petite plus la chance est grande d'être sélectionné;

Une fois q'une paire de parnet est sélectionnée, une bisection équilibrée est créée.

Algorithme du recuit simulé

1- S est la partition initiale, on choisit une tempréature initla T, T>0 ;

2- Un voisin S' à la solution S est sélectionné;

3- soit d=q(S)-q(S'), où q(S) est la qualité d'une solution, si d<0 alors S' devient la solution courante, sinon S' remplacer S avec $e^{-d/T}$;

arrêter suivant un critère sinon répéter 2- en changeant T par r.T ou 0<r<1 (taux de refroidissement).

Algorithme de recherche Tabu

---

on commence par un graph bissectionné équilibré choisi au hasard;

un nœud est sélectionné et déplacé de la partie à laquelle il appartient vers l'autre partie après avoir été déplacé le nœud est conservé dans une liste appelé "liste Tabu" pour un certain nombres d'itérations;

3- si le résultat d'un déplacement d'une bisection équilibré aboutit à une diminution de la taille de coupure que les bisections équilibrées précédentes, cette bisection est conservée comme la meilleure;

4- si aucune meilleure bisection n'est trouvée un facteur de déséquilibre est augmenté, initialement ce facteur est nul. Le facteur limite la différence de cardinalité entre les 2 parties d'une bisection.

Algorithme des k-ensembles utiles

1-on donne une bisedction {n1,n2} et un entier l

2-on cherche un k ensemble utile k>l tel que si un dplamcment de S vers N2(N1) réduit la taille de coupure de k

3-si aucun S n'est trouvé on augemen l'utilité sinon on fixe S et l à 0

4-si S≠0 on cherche un ensemble $S' \subset N_1 \cup S$ ou $S' \subset N_2 \cup S$, si il existe on déplae s et S' entre N_1 et N_2 et on fixe l à 2l sinon on fixe l à l/2

5- répéter 2- jusqu'à l=0

Algorithme de KL (Kernighan & Lin, 1970)

$int(v) = \sum_{(v,u)\in E \& P(v)=P(u)} w(v,u)$ où P(v) est l'indice de la partie à laquelle le noeud v appartient et $w(u,v)$ est le poids de l'arrête (u,v)

$ext(v) = \sum_{(v,u)\in E \& P(v)\neq P(u)} w(v,u)$

$g(v) = ext(v) - int(v)$

$$g(v1,v2) = \begin{cases} g(v1) + g(v2) - 2w(v1,v2) si (v1,v2) \in E \\ g(v1) + g(v2) sinon \end{cases}$$

1 enlever les marques de tous les nœuds d'une bissection équilibrée $\{N_1,N_2\}$;

trouver une paire non marquée $v_1 \in N_1$ et $v_2 \in N_2$ pour laquelle $g(v_1,v_2)$ est maximum (mais pas forcément positif). Marquer v_1 et v_2 et mettre à jour les valeurs de g de tous les nœuds non marqués restants comme si on avait échangé v_1 et v_2. (seuls les valeurs de g des voisins de v_1 et v_2 ont besoin d'être mis à jour) ;

On a maintenant une liste ordonnée de spaires de nœuds (v_1, v_2), i = 1,2,…,n. Ensuite on trouve l'indice j tel que $\Sigma g(v_1, v_2)$ est maximum. Si cette somme est positive, on échange la première paire de nœuds j ;

Répéter 2- min(|N1|, |N2|) fois .

Méthodes globales géométriques

Algorithme RCB (bisection de coordonnée récursive)

1-choisir un axe de coordonnées

2-choisir un plan orthogonal à l'axe sélectionné qui coupe le graphe en 2 sous-ensemble de taille égale. Cela consiste à trouver la médiane des valeurs des coordonnées.

Algorithme de la méthode inertielle

Variante de RCB avec un axe du moment angulaire minimum

Algorithme des d-spheres

projeter stéréographiquement les nœuds sur une sphère unité à (d+1) dimensions. Cela veut dire que le nœud v est projeté sur le point où la ligne de v au pôle nord de la sphère coupe la sphère.

Trouver le point central des nœuds projetés. (un point central d'un ensemble de points S dans un espace à d dimensions est un point c tel quechaque hyperplan à travers c divise S dans un rapport d:1 ou mieux. Chaque ensemble S a un point central, il peut être trouvé par programmation linéaire

Positionner de façon conforme les points sur la sphère. Premièrement on les fait tourner autour de l'origine pour que le point central devienne un point (0,…,0,r) sur l'axe (d+1). Deuxièmement dilater le point par (1) projection des points sur l'espace à d dimensions (2) en multipliant leur coordonnées par $\sqrt{(1-r)/(1+r)}$ et (3) en projetant stéréographiquement les points sur la sphère à d+1 dimensions;

Le point central des points positionnés conformément coincide avec le centre de la sphère à d+1 dimensions;

Choisir un hyperplan aléatoire passant par le centre de la sphère unité à d+1 dimensions;
L'hyperplan de l'étape précédente coupe la sphère unité à d+1 dimensions en un grand cercle, une sphère à d dimensions. Transformer cette sphère en inversant le mapping conforme et la projection stéréographqiue. Utiliser la pshère pour bissecter les nœuds.

Méthodes globales non géométriques

Algorithme RGB (angl. Recursive Graph Bisection)

1- choisir une paire de nœud à la plus grand distance l'un de l'autre du graphe;
2- utilisant la recherchede première-largeur commançant au nœud sélectionné, la distance à chaque nœud depuis ce nœud est déterminée;
3- les nœuds sont triés par rapport à ces distances et l'ensemble trié est divié en 2 enseble de taille égale.

Algorithme RSB (angl. Recursive Spectral Bisection)

calculer la matrice laplaciennne L=D-A où D est la mtrice diagonale exprimant les degrés des nœud et A est la matrice d'adjacence
Les vecteurs de Fiedler (vecteurs propres) sont calculé avec un algorithme de Lanczos modifié
On trie mles nœuds par rappor à leur corrdonnée de Fiedler et oin utilise la seconde valeur propore la plus faible pour diviser le graphe en 2 parties

Algorithm RSB-Multiniveau

1- phase grossière (angl. Coarsening) : une séquence de graphes G^i est construite à partir du graphe. Soit G^i, G^{i+1} est une approximation obtenue en calculant $I_{i,}$ un sous ensemble indépendant maximal de N_i. Ce sous ensemble de N_i est tel qu'aucune paire de nœuds dans le sous ensemble ne partage d'arête.Un sous ensemble indépendant est maximal si aucun nœud ne peut être ajouté à ce sous-ensemble;
2- Avec chaque nœud $v \in I_i$ est associé un domaine D_v qui initialement contient seulement v lui-même. Toutes les arêtes de E_i sont non-marquées. Tant qu'il y a une arête non marquée $(u,v) \in E_i$ si u et v appartiennent au même domaine, marquer (u,v) et l'ajouter au domaine, si seulement 1 nœud, disons u, appartient au domaine, marquer (u,v), et ajouter v et (u,v) à ce domaine. Si u et v sont dans différents domaines, disons D_x et D_y alors marquer (u,v) et ajouter l'arête (x,y) à E_{i+1}. Finalement si u et v n'appartienent pas à un domaine alors traiter l'arête à une étape ultérieure;
3- A un certain niveau on obtient un graphe G_m qui est assez petit pour calculer son vecteur de Fiedler correspondant (f^m) en un temp réduit. Pour obtenir f^0 on amorse la phase non-grossière (angl. Uncoarsening). Sachant f_{i+1} on obtient f_i par interpolation et amélioration. Pour chaque $v \in N_{i,}$ si $v \in N_{i+1}$ alors $f_i(v)=f_{i+1}(v)$ sinon $f_i(v)$ est un ensemble égal à la valeur moyenne des composants de f_{I+1} correspondant aux voisns de v dans N_i ($f_i(v)$ est le vecteur correpsondant au nœud v). Ensuite le vecteur f_i est amélioré par l'itération du coefficient de Rayleigh.

Algorithm KL-multiniveau

1- Phase grossière (angl. Coarsening) : une approximation G^{i+1} d'u graphe $G^i=(N^i,E^i)$ est obtenue en cherchant une correspondance maximale M^i. Une correspondance est un sous-ensemble E^i tel qu'aucune arête dans le sous ensemble ne partage de noeud. Une correspondance est maximale si aucune arête ne peut être ajoutée à la correspondance. Pour une correspondance donnée M^i, G^{i+1} est obtenue en réunissant tous les nœuds de la

correspondance. Si $(u,v) \in M^i$ alors les nœuds u et v sont remplacés par un noeud v' dont le poids est la somme des poids de u et v. Une correspondance maximale peut être trouvée comme suit: les nœuds sont visités dans un ordre aléatoire. Si un nœud visité u est non-marqué, on correspond u avec un voisin non-correspondant v tel qu'aucune arête entre u et un voisin non marqué soit plus lourd que l'arête (u,v).

Phase de partitionement du graphe grossier G^m : on démarre à un nœud quelconque les parties grossissent en ajoutant les nœuds voisins à la partie. Un nœud voisin dont l'addition à la partie provoque une diminution de la taille de coupure, est ajouté.

Phase non-grossière, la partition de G^m est successivement transformée en partition du graphe original G. Pour chaque nœud $v \in N^{i+1}$, soit P^{i+1} (v) l'indice de la partie (de la partition du graphe G^{i+1}) auquel v appartient. Soit P^{i+1}, P^i est obtenu comme suit: premièrement, une partition initiale est construite utilisant la projection: si $v' \in N^{i+1}$ correspondant à la paire correspondante (u,v) de nœuds dans N^i, alors

$P^i(u) = P^i(v) = P^{i+1}(v')$, sinon $P^i(v') = P^{i+1}(v')$. Ensuite cette partition initiale est améliorée utilisant une variante de la méthode de KL: un nœud $v \in N^i$ est déplacé d'une partie avec l'indice k, $k \in A^i(v)$, si une des conditions suivantes rencontrées est satisfaite:

(a) $g^i(v,k)$ est positif et maximum parmi tous les déplacements de v qui satisfont la condition d'équilibre

(b) $g^i(v,k)=0$ et $W^i(P^i(v))-w(v) > W^i(k)$, $W^i(k) = \sum_{v \in N^i \,\&\, P^i(v)=k} w(v)$ poids de la partie

Quand un nœud est déplacé, les valeurs de g et les poids de la partie influencée par le déplacement sont mis à jour.

ANNEXE 5

Chaînes de Markov cachées (HMM)

A Le problème et l'algorithme "Forward"

Nous avons un modèle $\lambda = (\Lambda, B, \pi)$ et une séquence d'observations $O = o_1, o_2, ..., o_T$, et $p\{o|\lambda\}$ doit être trouvé. Nous pouvons calculer cette quantité en utilisant des arguments probabilistes simples. Mais ce calcul implique un nombre d'opérations de l'ordre de N^T . C'est très important même si la longueur de la séquence T est modérée.

Ainsi nous devons chercher une autre méthode pour ce calcul. Heureusement il en existe une qui possède une faible complexité et utilise une variable auxiliaire, $\alpha_t(i)$ appelée variable *forward* ("postérieure").

Cette variable *forward* est définie comme la probabilité de la séquence d'observations partielles $o_1, o_2, ..., o_T$, quand elle se termine à l'état i. Mathématiquement,

$$\alpha_t(i) = p\{o_1, o_2, ..., o_T, q_t = i|\lambda\} \quad (1.1)$$

Alors il est facile de voir que la relation récursive suivante apparaît (a_{ij} est la transition de l'état i vers l'état j, $b_i(o)$ est la probabilité d'avoir le symbole o dans l'état i, π_i est la probabilité initiale de l'état i).

$$\alpha_{t+1}(i) = b_j(o_{t+1}) \sum_{i=1}^{N} \alpha_t(i) a_{ij} \quad ,1 \le j \le N, 1 \le t \le T-1 \quad (1.2)$$

où,

$$\alpha_1(j) = \pi_j b_j(o_1) \ , 1 \le j \le N$$

En utilisant la récurrence on peut calculer

$$\alpha_T(i) \ , 1 \le j \le N$$

et alors la probabilité recherchée est donnée par,

$$p\{O|\lambda\} = \sum_{i=1}^{N} \alpha_T(i). \quad (1.3)$$

La complexité de cette méthode, connue comme algorithme *forward* est proportionnel à N^2T, qui est linéaire par rapport à T alors que le calcul direct mentionné plus haut a une complexité exponentielle.

D'une manière similaire on peut définir la variable *backward* ("antérieure") $\beta_t(i)$ comme la probabilité de la séquence d'observations partielles $o_{t+1}, o_{t+2}, ..., o_T$, sachant que l'état courant est i. Mathématiquement,

$$\beta_t(i) = p\{o_{t+1}, o_{t+2}, ..., o_T | q_t = i, \lambda\} \quad (1.4)$$

Comme dans le cas de $\alpha_t(i)$ il y a une relation de récurrence qui peut être utilisée pour calculer $\beta_t(i)$ efficacement.

$$\beta_t(i) = \sum_{j=1}^{N} \beta_{t+1}(j) a_{ij} b_j(o_{t+1}), 1 \le i \le N, 1 \le t \le T-1 \quad (1.5)$$

où,

$$\beta_T(j) = 1, \ 1 \le i \le N$$

En outre nous pouvons voir que,

$$\alpha_t(i)\beta_t(i) = p\{O, q_t = i|\lambda\}, \ 1 \le i \le N, 1 \le t \le T \quad (1.6)$$

Donc ce la donne une autre façon de calculer p{O|λ}, en utilisant à la fois les variables *forward* et *backward* selon l'équation (1.7):

$$p\{O|\lambda\} = p\{O, q_t = i|\lambda\} = \sum_{i=1}^{N} \alpha_t(i)\beta_t(i) \qquad (1.7)$$

(1.7) est très utile, spécialement dans la dérivation de formules requises pour des apprentissages basés sur des gradients.

B Algorithme de Baum-Welch

Cette méthode peut être dérivée en utilisant de simples arguments de "comptage d'occurrences" ou en utilisant un calcul qui maximise la quantité auxiliaire:

$$Q(\lambda, \bar{\lambda}) = \sum_{q} p\{q|O, \lambda\} \log p\{q|O, \bar{\lambda}\}$$

sur $\bar{\lambda}$. Une caractéristique spéciale de l'algorithme est de garantir la convergence .
Pour décrire l'algorithme de *Baum-Welch* (aussi connu sous le nom d'algorithme *forward-backward*), nous avons besoin de définir 2 autres variables auxiliaires, en addition des variables *forward* et *backward* définies à la section précédente. Ces variables peuvent cependant s'exprimer en terme des variables *forward* et *backward*.

La première de ces variables est définie comme la probabilité d'être dans l'état i au temps t=t et dans l'état j au temps t=t+1. Formellement,

$$\xi_t(i, j) = p\{q_t = i, q_{t+1} = j|O, \lambda\} \qquad (1.10)$$

C'est la même chose que:

$$\xi_t(i, j) = \frac{p\{q_t = i, q_{t+1} = j|O, \lambda\}}{p\{O|\lambda\}} \qquad (1.11)$$

En utilisant les variables *forward* et *backward* cela peut aussi s'exprimer par:

$$\xi_t(i, j) = \frac{\alpha_t(i)a_{ij}\beta_{t+1}(j)b_j(o_{t+1})}{\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_t(i)a_{ij}\beta_{t+1}(j)b_j(o_{t+1})} \qquad (1.12)$$

La seconde variable est la probabilité a posteriori,

$$\gamma_t(i) = p\{q_t = i|O, \lambda\} \qquad (1.13)$$

qui est la probabilité d'être dans l'état i au temps t=t, connaissant la séquence d'obervation et le modèle. En terme de variables forward et backward cela peut s'exprimer par:

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N}\alpha_t(i)\beta_t(i)} \qquad (1.14)$$

On peut voir que la relation entre $\gamma_t(i)$ et $\xi_t(i, j)$ est donnée par:

$$\gamma_t(i) = \sum_{j=1}^{N}\xi_t(i, j), \ 1 \le i \le N, \ 1 \le t \le M \qquad (1.15)$$

Maintenant il est possible de décrire le processus d'apprentissage de Baum-Welch, où les paramètres du HMM sont mis à jour pour maximiser la quantité $p\{O|\lambda\}$. En supposant un modèle initial $\lambda = (\Lambda, B, \pi)$, nous calculons les 'α' et 'β' en utilisant les récurrences 1.5 et 1.2 et alors les 'ξ' et 'γ' en utilisant 1.12 et1.15. L'étape suivante consiste à mettre à jour les paramètres du HMM grâce aux eq. 1.16 à 1.18 connaissant les formule de ré-estimation:

$$\overline{\pi_i} = \gamma_1(i), \ \ 1 \le i \le N \qquad (1.16)$$

$$\overline{a_{ij}} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}, \quad 1 \le i \le N, 1 \le j \le N \tag{1.17}$$

$$\overline{b_j}(k) = \frac{\sum_{\substack{t=1 \\ o_t = 1/k}}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}, \quad 1 \le j \le N, 1 \le k \le M \tag{1.18}$$

Ces formules de ré-estimation peuvent être facilement et également modifiées pour des cas de densité continue.

C Le problème du décodage et l'algorithme de Viterbi

Dans ce cas nous voulons trouver la séquence d'état la plus probable pour une séquence donnée d'observations $O = o_1, o_2, ..., o_T$ et un modèle $\lambda = (\Lambda, B, \pi)$.

La solution à ce problème dépend de la façon dont on définit "la séquence d'état la plus probable". Une approche est de trouver l'état q_t le plus probable au temps t=t et de concaténer tous les 'q_t'. Mais parfois cette méthode ne donne pas de séquence d'état physiquement significative.

Par conséquent nous optons pour une autre méthode qui contourne ce problème. Dans cette méthode, couramment appelée *algorithme de Viterbi*, la séquence d'état entière est trouvée grâce à la vraisemblance maximale. De façon à faciliter la computation nous définissons une variable auxiliaire:

$$\delta_t(i) = \max_{q_1, q_2, ..., q_{t-1}} p\{q_1, q_2, ..., q_{t-1}, q_t = i, o_1, o_2, ..., o_{t-1} | \lambda\},$$

qui donne la plus haute probabilité qu'une séquence d'observation partielle et une séquence d'état jusqu'à temps t=t peut avoir, quand l'état courant est i.
Il est facile d'observer la relation récurrente suivante:

$$\delta_{t+1}(j) = b_j(o_{t+1}) \left[\max_{1 \le i \le N} \delta_t(i) a_{ij} \right], \quad 1 \le i \le N, 1 \le t \le T-1 \tag{1.19}$$

où,

$$\delta_1(j) = \pi_j b_j(o_1), \quad 1 \le j \le N$$

Ainsi la procédure de trouver la séquence d'état la plus probable commence par le calcul utilisant la récurrence 1.8, tandis qu'on garde toujours un pointeur sur l' "état gagnant" dans l'opération de recherche du maximum.
Finalement l'état j* est trouvé par:

$$j^* = \arg\max_{1 \le j \le N} \delta_T(j)$$

et commençant par cet état, la séquence des états est poursuivi comme un pointeur dans chaque état indiqué. Cela donne l'ensemble des états recherchés.
L'algorithme global peut s'interpréter comme une recherche dans un graphe dont les nœuds sont formés par les états du HMM à chaque instant t, $1 \le t \le T$.

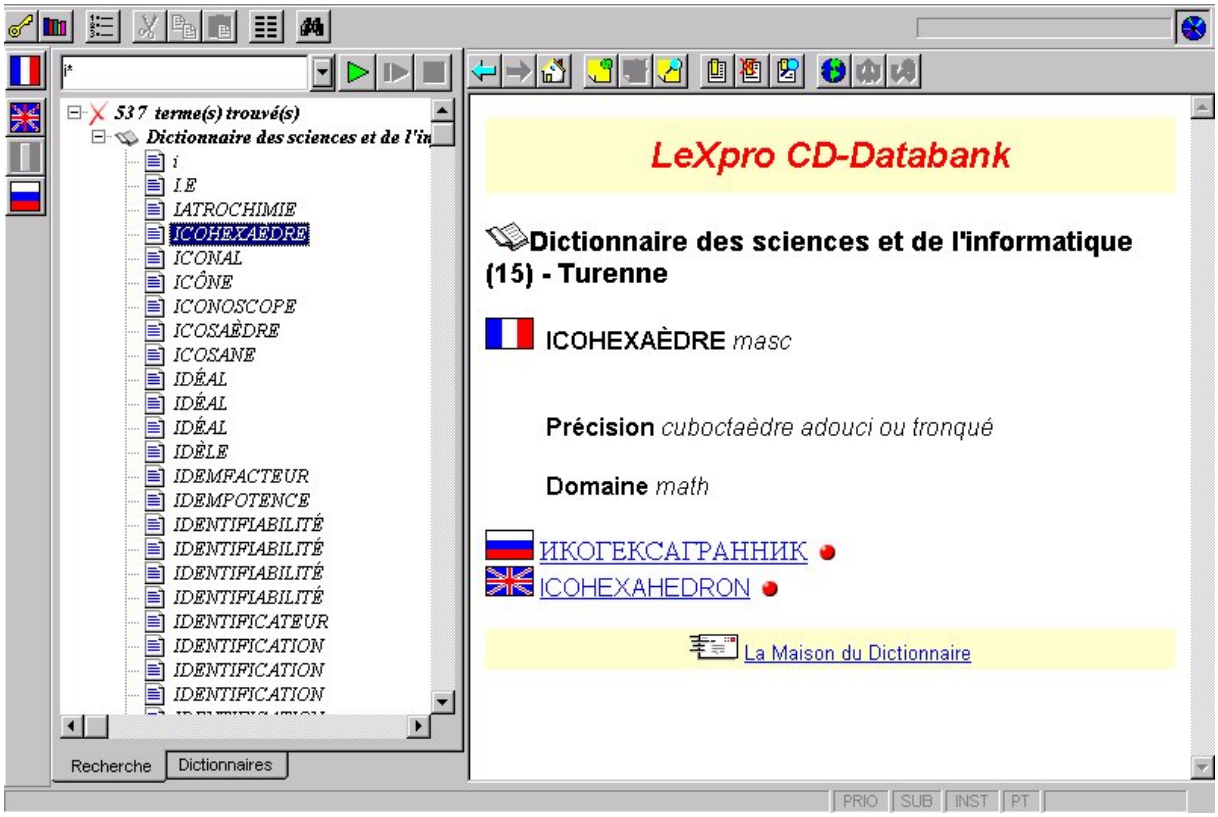
ANNEXE 6

Interface de LexPro avec le dictionnaire des sciences

Le dictionnaire des sciences inclus 8000 termes en français traduits en anglais et en russe soit au total 24000 termes. Chaque terme possède une définition et un champ thématique (physique, astronomie, mathématique, chimie, informatique, aéronautique, technique).

Le dictionnaire a été réalisé par l'auteur de cette thèse entre 1994 et 1996. Il est commercialisé sous CD-ROM avec le logiciel de navigation LexPro™ interfacé avec le logiciel de bureautique MS-Word™.

Lexpro 3.0 est un logiciel réalisé par la société *LCI* (Paris) sous l'égide de l'éditeur *La Maison du Dictionnaire* (Paris).



ANNEXE 7

Hiérarchie de référence des concepts en médecine

caractéristiques générales

26 classes dont 23 qui ont ou plus de 3 éléments regroupant 354 termes, 59/354 d'inclassables soit 16,7%

9 classes conceptuelles
17 sous-classes conceptuelles
6 sous-sous-classses conceptuelles

schéma général de la classification (entre parenthèses nombre de termes de la classe)

c1 Anatomie Cardiovasculaire (53)
- anatomie générale (16)
- artère (24)
- artère ou veine (5)
- coeur (8)

c2 Physiologie Cardiovasculaire (17)

c3 Pathologie Générale (21)
- généralité (17)
- maladie (4)

c4 Pathologie Cardiovasculaire (60)
- généralité (3)
- cardiopathie
  - généralité (6)
  - trouble du rythme (10)
    - valvulopathie (6)
    - coronaropathie (25)
- maladie des vaisseaux (10)

c5 Facteur de Risque (9)

c6 Diagnostic (81)
- généralité (23)
- imagerie (26)
- ecg (30)
- paramètres physiologiques servant au diagnostic (2)

c7 Symptomatologie (14)

c8 Thérapeutique (60)
- généralité (24)
- chirurgie (11)
- cathétérisme (13)
- médicaments/agents traitants
- nom commercial (4)
- famille d'agent (7)
- dci (1)

c9 Information (1)

(mots soulignés = mots appartenant à plusieurs classes)

C1 **CLASSE** **ANATOMIE CARDIOVASCULAIRE**

sous-classe: anatomie générale

antéro
caractère distal des lésions
ensemble du réseau
genou inférieur
membre inférieur droit
membres inférieurs
paroi
partie moyenne
plan

portion
réseau
segment
segment proximal
territoire
tiers proximal
voie

artère

anatomie des artères coronaires
aorte
arbre coronarien distal
artère
artères
carotide
circonflexe
coronaire
coronaires
coronarienne
coronariennes sévères
intraaortique
iva
mammaire interne
marginale gauche
orifices coronariens
part de l'artère
primitive droite
pulsion intraaortique
racine de l'aorte
ras du tronc
tronc
tronc commun
vertébrale gauche

artère ou veine

axes iliaques
branche
branches
vaisseaux du cou
vasculaires périphériques

coeur

cavités droites
chemine dans le sillon
interventriculaire
postérolatérale
sillon auriculoventriculaire
valves aortiques
ventriculaire
ventricule

C2 **CLASSE** **PHYSIOLOGIE CARDIOVASCULAIRE**

basse pression
cardiovasculaire
cinétique ventriculaire gauche
circulation
courant de lésion
débit cardiaque
flux
fonction
fraction d'éjection
fréquence
importante collatéralité

index cardiaque
pression
pressions
revascularisée par collatéralité
rythme sinusal
tension artérielle

C3 **CLASSE PATHOLOGIE GENERALE**

sous-classe: généralité

atteinte
complications reliées
crise
état
évolution
hypertrophie
lésion
lésions
malade
malades
mort subite
patient
patiente présente
plupart de ces malades
récidive
satisfaisante chez ce patient
sympathique patient

maladies

bronchite chronique
diabète non insulino-dépendant
insulino-dépendant
ulcère gastrique

C4 **CLASSE PATHOLOGIE CARDIOVASCULAIRE**

sous-classe: généralité

cardiovasculaire
nécrose
poussée d'insuffisance

cardiopathie

sous-sous-classe: généralité

anévrisme du ventricule gauche
arrêt cardiaque
dyskinésie
fibrillation
hypokinésie
mauvaise fonction ventriculaire gauche
passage en fibrillation
souffrance myocardique

trouble du rythme
hypocinésie antérieure
arythmie ventriculaire
arythmies ventriculaires
tachycardie ventriculaire
akinésie
bradycardie sévère
extrasystoles ventriculaires
trouble de la cinétique segmentaire

valvulopathie

fuite mitrale
insuffisance
persistance d'une sténose
resténose
sténose
sténoses

maladies des vaisseaux coronaires ou coronaropathie

angine
angor
athéromatose coronarienne significative
athérosclérose coronarien
diamètre de l'obstruction
discrètement athéromateuses
étendue de l'infarctus
idm inférieur
infarctus
ischémie
nécrose
obstructions coronariennes
occlusif de la coronaire droite
origine ischémique
plaque
porteurs d'un infarctus du myocarde
réapparition d'un angor
reprise de l'angor
séquelle d'infarctus inférieur
serrée
serrée de l'iva
seuil ischémique
sévère du tronc commun
spasme
tritronculaires sévères

maladies des vaisseaux

accident vasculaire cérébral
accidents thromboemboliqes
hypertension artérielle
ischémie
nécrose
oblitération de
occlusion
origine ischémique
seuil ischémique
subocclusive

C5 **CLASSE** **FACTEUR DE RISQUE**

ancien tabagique
diabète non insulino-dépendant
facteurs de
facteurs de risque
haut risque
hérédité coronaire
hypertension artérielle
risque
tabagisme modéré

C6 **CLASSE** **DIAGNOSTIC**

sous-classe: généralité

aspects techniques
bilan

conclusion
considéré positif
contrôle
degré de sévérité
effectuée sans problème
élévation enzymatique
évaluation de la maladie coronarienne
examen
limites de la normale
milieu hospitalier
minimes irrégularités
modification
normales inférieure
normales ou peu lésées
nouveau test
pratiquement normalise
sévérité de lésions
tableau clinique
taux de
technique
test
valeur pronostique

imagerie

amplificateur de brillance
apparition d'une sténose
cathéters
champ d'aval
dissection
échographie ne montre
estimation du diamètre
existence de lésions athéromateuses
flot d'aval
image résiduelle
incidence oblique
incidences
introduction des cathéters
lit d'aval
lits d'aval
multiples incidences transverses
oblique antérieure
obliques antérieures
observe pas de trait de dissection
profil franc
résiduelle non significative
scintigraphie
seule incidence
substance de contraste
techniques du cathétérisme cardiaque
trait de dissection

ecg

angulation craniocaudale
bicyclette ergométrique
bloc de branche
cathéter électrode
cathétérisme
cathéters
conduction ventriculaire
coronarographie
début de l'examen
décalage
différentes épreuves d'effort
ECG
effort

élévation de l'onde
épreuve d'effort
épreuves d'effort
hémiaxiales
majoration du courant de lésion
maximale négative
modifications électrocardiographiques
négativation des ondes
normalise son ECG
percritique
percutanée fémorale
peu altérée
ponction percutanée
précocement positive
susdécalage
tapis roulant
ventriculographie

paramètres physiologiques servant au diagnostic

débit cardiaque
tension artérielle

C7 **CLASSE** **SYMPTOMATOLOGIE**

altération
caractère distal des lésions
devenait symptomatique
douleur
douleurs
douloureuse
effort
peu altérée
précordialgies
présente des précordialgies
symptomatologie
syndrome angineux
syndrome douloureux
troubles

C8 **CLASSE** **THERAPEUTIQUE**

sous-classe: généralité

actuelle hospitalisation
administrée par voie
antérieur thrombolyse
dose efficace
effectuée sans problème
état
évolution
garder en hospitalisation sous traitement anticoagulant
heure de
hospitalisé en urgence
injection non sélective
injections
matin midi
médecin traitant
milieu hospitalier
poursuite du traitement
pratiquement normalise
préférable de le garder en hospitalisation sous traitement anticoagulant
retrait des bétabloquants
satisfaisante chez ce patient
soins intensifs
stabilisé médicalement
traité par fibrinolyse intraveineuse

traitement

chirurgie

chirurgie
correction chirurgicale
geste de revascularisation
greffons aortocoronariens
intervention de pontage
pont mammaires
pontage
pontages aortocoronariens
prêter à des pontages aortocoronariens
revascularisation chirurgicale
triple pontage

cathétérisme

cathétérisme
angioplastie
ponction percutanée
dilatations
cathéters
techniques du cathétérismes cardiaque
bénéficie d'une angioplastie
laboratoire de cathétérisme cardique
introduction des cathéters
pacing auriculaire
prothèse endocoronaire
percutanée fémorale
possible de dilater

médicaments/agents traitants

sous-sous-classe: nom commercial

aspegic250
bitildiem
glucophage retard
isoptine120

famille d'agents

administration d'héparine intraveineuse
agents antiplaquettaires
beta
betabloquant
continue de nitroglycérine
dérivés nitrés
hbpm

DCI (Dénomination Commune Internationale)

maléate d'ergométrine

C9 CLASSE INFORMATION

étude récente

C10 CLASSE PATIENT

consentement informé
proposer à Monsieur Mojpat11

Mots inclassables

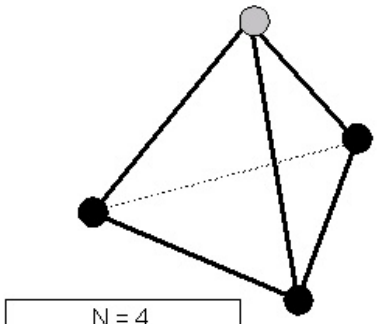
absence

anti

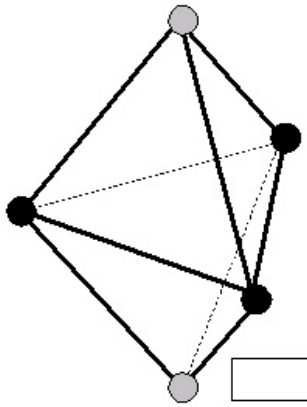
deux
double
en général
façon significative
fait
fut décrite
grosse
invalidant donc
met en évidence
mettre en évidence
mise en
modification
montrait
nécessaire d'envisager
ny
pas
permis de
petit
petite
premier temps
première
prise en charge
relativement rares
strictement
système doit permettre
trois
utile de remplacer
égales ou supérieures
ciné
limite
excellent
donc pas très invalidant
faux
entrée
inconvénients suivants
indication
mettait en évidence
mis en
montre
nouvelle
permet de déceler
hors mis
probablement en rapport
pratiquée de routine
prendre en
pouvant éventuellement
remerciant de ta confiance
reste
mal définie
réalisation d'une
surviennent partout
longue
type
présence
verrons plus loin

ANNEXE 8

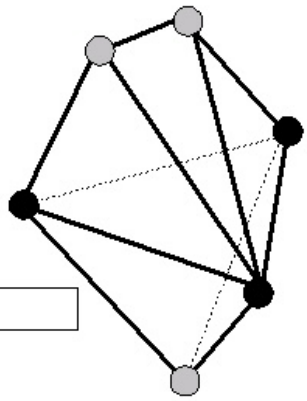
Graphes en 3-D



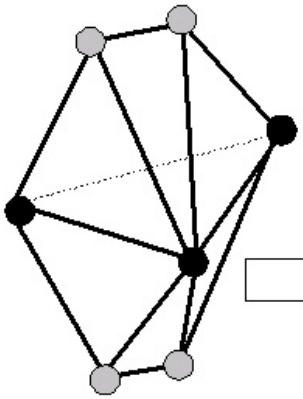
N = 4



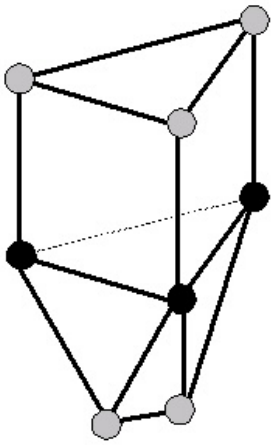
N = 5



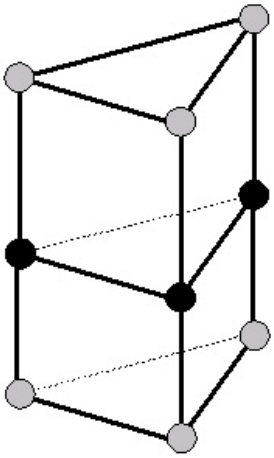
N = 6



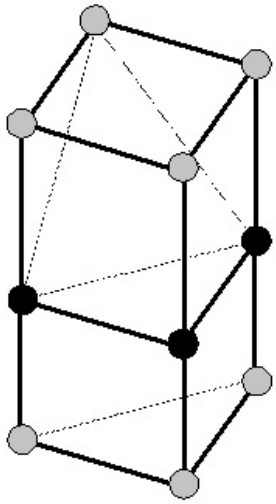
N = 7



N = 8



N = 9



N = 10

ANNEXE 9

Contextes terme/verbe liés à une classe

Ne <**provoquer>** pas de douleur precordiale.La <**symptomatolog**> ne <**sembler>** pas être en rapport avec

âge de an qui présenter un <**symptomatolog**> douloureuse atypique mai <**survenir**> plus volont avoir le effort dans

altération de le électrocardiogramme et sans <**spasmer**> coronaire <**constituer**> un test négatif chez le malade

au niveau de le <**interventriculair**> antérieur <**associer**> au petit trait de dissection nous avoir donc préférer proposer de embler le réalisation de un double pontag <**interventriculair**> anterieure-diagonale MADXMOR malade de an ancien

avoir avoir le originer de le <**interventriculair**> postérieur le fonction ventriculaire gauche être normale.L aortograph savoir sigmoid ne <**montrer**> pas de anomalie au niveau

avoir prendre en fonction de le <**symptomatolog**> clinique et du désir du patient qui ne <**sembler**> pas être commodes.Je te remercier

cardiaque survenir dans du conditions identique <**associer**> avoir un <**symptomatolog**> douloureuse precordial le échographier réaliser avoir

certain et un hypercholesterolem qu il ne connaître pas mai que nous avoir retrouver avoir gr l.Depuis quelques jour il <**présenter**> du <**douleurs**> retrosternal et epigastr qui avoir être

conclusion le patient âgé de an <**hospitaliser**> pour angioplast coronaire de un <**restenos**> de le IVA elle avoir présenter

conclusion patient âge de an avoir <**constituer**> en un probable <**nécroser**> antérieur avoir coronaire normale il être

coronaire droite <**donner**> uniquement le artère <**interventriculair**> postérieur comme brancher terminale aucun branché posterolateral ne provenir de le artère coronaire droite

coronarograph est pratiquer avoir distancer elle ne montrer aucun lésion significatif sûr le réseau coronaire mai un image de thrombu résiduel dans le <**interventriculair**> antérieur qui <**justifier**> le poursuite de

de <**restenos**> il parer logique de <**persister**> dans le indication de un angioplast itterat si un <restenos> se constituer le patient être avertir

de dissection nous avoir donc préférer <**proposer**> de embler le réalisation de un double pontag <**interventriculair**> anterieure-diagonale MADXMOR malade de an ancien

de le année <**repriser>** de un <**symptomatolog**> important ainsi un angioplast de le coronaire droite pouvoir être <**proposer**> si le traitement médical réadapter être

de un angioplast itterat si un <**restenos**> se <**constituer**> le patient être avertir de ce

dernier motiver le actuel hospitalisation.La coronarograph ne <**montrer>** pas de évolution lésionnel et le <**symptomatolog**> est en relation avec un ischem

dilater ne avoir pas présenter de <**restenos**> dans le intervalle le patient avoir développer un zona intercostal thoracique qui avoir <**justifier**> un traitement intra veineux relayer actuellement

distal chez un patient qui avoir <**constituer**> un infarctus ambulatoir en rapport avec un occlusion de le <**interventriculair**> antérieur revascularise par un réseau collatéral

dominante son trois tronc coronar être <**constituer**> de un part de le artère <**interventriculair**> antérieur et de autre part de

du inhibit du enzyme de conversion.Les <**douleurs**> pouvoir si <**expliquer**> être par du hepatalg de effort

et de TILDIEM si le angor <**persister**> un reinterven être alors légitimer MADXLAMMAPBSEB patienter âgé de an qui présenter un <**symptomatolog**> douloureuse precordial de effort ainsi qu

gastro-duodenale avec antrobulbit érosif qui pouvoir <**expliquer**> si le thallium est négatif le <**symptomatolog**> actuel traitement de sortie GLIBENES x

hospitalisation le coronarograph avoir permettre de <**mettre**> en évidence un stenos récent et très serrer de le IVA proximal et le absence de <**restenos**> au niveau de le circonflexe le

image de thrombu résiduel dans le <**interventriculair**> antérieur qui <**justifier**> le poursuite de un traitement anti

le coronaire droite compliquer de un <**restenos**> immédiat le contrôle angiograph <**mettre**> en évidence un excellent dilatation de

le coronaire droite sembler si être <**constituer**> dans le quatre dernier mois de le année repriser de un <**symptomatolog**> important ainsi un angioplast de le

le ergometr en revancher si il <**survenir**> du <**douleurs**> angin durer le examen celles-ci devoir

le exposer avoir un taux de <**restenos**> important.Le contrôle angiograph ne être <**justifier**> que si le angor réapparaître ou

le persistance de un risque de <**restenos**> il parer logique de <**persister**> dans le indication de un angioplast

le persistance de un risque de <**restenos**> il parer logique de persister dans le indication de un angioplast itterat si un <**restenos**> se <**constituer**> le patient être avertir de ce

le persistance de un risque de restenos il parer logique de persister dans le indication de un angioplast itterat si un <**restenos**> se <**constituer**> le patient être avertir de ce

le réalisation de un double pontag <**interventriculair>** anterieure-diagonale MADXMOR malade de an ancien tabagique avoir faire un IDM inférieur en <**compliquer**> de OAP et qui avoir bénéficier

le résultat être insuffisant et il <**persister**> un longue plaquer au niveau de le <**interventriculair**> antérieur associer au petit trait de

le séquelle antérieur de un grand <**nécroser**> avec du pression de remplissage elevees.Les suite immédiat avoir être <**compliquer**> de un petit pousser de insuffisance

lésion MADXMORMAPGPOR le patient avoir être <**hospitaliser**> dans le service pour réalisation de un double angioplast <**interventriculair**> anterieure-diagonale un semaine après celui de

longue plaquer au niveau de le <**interventriculair**> antérieur associer au petit trait de dissection nous avoir donc préférer <**proposer**> de embler le réalisation de un

MADXCHI MAPYFOU malade de an déjà <**hospitaliser**> en pour un angor de prinzmet en rapport avec un stenos serrée de le <**interventriculair**> antérieur ce stenos avoir être dilater

moins de page le présence de <**douleurs**> thoracique sans altération de le électrocardiogramme et sans spasmer coronaire <**constituer**> un test négatif chez le malade

ne provoquer pas de douleur precordiale.La <**symptomatolog**> ne <**sembler>** pas être en rapport avec un pathologie coronaire par contrer le contexte rhumatolog nous conduire avoir <**proposer**> avoir le patient un nouveau consultation

occlusion exposer avoir un taux de <**restenos**> supérieur avoir le valeur habituel MADXMORMAPHIZG patient âge de an avoir présenter un infarctus inférieur en <**hospitaliser**> pour un récidive en territoire antérieur

oppression thoracique non typique de un <**symptomatolog**> angin et sans modification electrocardiograph per-critique il si <**agir>** en faire de un récidive de de compensation cardiaque <**survenir**> dans du conditions identique associer avoir

par un poussée de oedème pulmonaire <**associer**> avoir du oppression thoracique non typique de un <**symptomatolog**> angin et sans modification electrocardiograph per-critique

patient être asymptomat le coronarograph avoir <**mettre**> en évidence un stenos inhomogen de le <**interventriculair**> antérieur proximal responsable de un grand

présenter du precordialg de effort.La coronarograph <**mettre**> en évidence un <**restenos**> de le IVA siéger au niveau

quelques temps du precordialg spontané avec <**douleurs**> atypique dorsale et <**douleurs**> brachial non simultanée le épreuve de effort réaliser dans le service être maximale negative.L échographier ne montrer pas de anomalie le coronarograph

réaliser avoir le eme heure elle <**mettre**> en évidence un double stenos serrer de le <**interventriculair**> antérieur distal modérément calcifie le second

récemment réaliser si avérer positiver avec <**douleurs**> precordial alors que le échographier ne <**montrer**> pas de hypertrophier myocard nette

service pour contrôler avoir mois après <**mettre**> en placer de un prothèse endocoronair au decour de un angioplast de le coronaire droite compliquer de un <**restenos**> immédiat le contrôle angiograph <**mettre**> en

si accompagner de arythm ventriculaire qui <**survenir**> chez environ le moitié du malade avec <**spasmer**> coronar symptomatique on pouvoir noter également

si le angine de poitrine le <**justifier**> en présence de <**douleurs**> angin surtout spontané et de coronaire

significatif on pouvoir donc être rassure.La <**symptomatolog**> être de ailleurs atypique **rester** avoir <**expliquer**> le anomalie electrocardiograph le discret hypertrophier ventriculaire gauche peutl <**expliquer**> en partir mai taire savoir comme

simple le coronarograph de contrôle être <justifier> si du <douleurs> réapparaître mai pouvoir être éviter si

sou former spontané le nouveau coronarograph ne montrer pas de restenos sûr le <interventriculair> antérieur mai le apparition de un

spontané avec <douleurs> atypique dorsale et <douleurs> brachial non simultanée le épreuve de effort réaliser dans le service être maximale negative.L échographier ne montrer pas de anomalie le coronarograph

temporairement le chirurgie le coronarograph avoir <mettre> en évidence un <restenos> de le circonflexe au niveau du

test au metherg est négatif il ne **provoquer** pas de douleur precordiale.La <**symptomatolog**> ne **sembler** pas être en rapport

un angioplast de le coronaire droite <compliquer> de un <restenos> immédiat le contrôle angiograph mettre en

un récidive de de compensation cardiaque <**survenir**> dans du conditions identique **associer** avoir un <**symptomatolog**> douloureuse precordial le échographier **réaliser** avoir

ANNEXE 10

Classes étiquetées

Cluster n°1
Hyperonyme1: Os
Hyperonyme2: corps
(1): accident vasculaire cérébral
(2): pont mammaire
(3): repriser de le angor
(4): angor
(5): coronaire
(6): ventriculaire
(7): coronarograph
(8): tritronculair sévère
(9): traitement
(10): territoire
(11): décalage
(12): dilatation

Cluster n°2
Hyperonyme1: Transports par route
Hyperonyme2: transports
(1): accident vasculaire cérébral
(2): lésion
(3): tapir rouler
(4): douleur
(5): plan
(6): effort
(7): patient
(8): épreuve de effort
(9): évolution
(10): fonction
(11): mammaire interner
(12): ischem

Cluster n°3
Hyperonyme1: Apparition
Hyperonyme2: existence
(1): administration de hepar intraveineuse
(2): segment
(3): ventriculograph
(4): normale ou peu léser
(5): voir
(6): modification
(7): médecin traiter
(8): administrer par voir
(9): léser
(10): angioplast
(11): infarctus
(12): angor

Cluster n°4
Hyperonyme1: Cœur et vaisseaux
Hyperonyme2: corps
(1): administrer par voir
(2): incidence
(3): malade
(4): test
(5): tronc
(6): fréquence
(7): examen
(8): coronar
(9): coronarograph
(10): altération
(11): anatomie du artère coronaire
(12): angine

Cluster n°5
Hyperonyme1: Quantité
Hyperonyme2: quantité
(1): altération
(2): effort
(3): circonflexe
(4): taux de
(5): spasme
(6): sévère du tronc commun
(7): séquelle de infarctus inférieur
(8): segment proxim
(9): segment
(10): scintigraph
(11): risque
(12): réseau

Cluster n°6
Hyperonyme1: Cœur et vaisseaux
Hyperonyme2: corps
(1): anévrisme du ventricule gauche
(2): malade
(3): ventricule
(4): résiduel non significatif
(5): pontag aortocoronar
(6): pontag
(7): obstruction coronar
(8): mauvais fonction ventriculaire gauche
(9): arythm ventriculaire
(10): angioplast
(11): lésion
(12): interventriculair

Cluster n°7
Hyperonyme1: Os
Hyperonyme2: corps
(1): atteindre
(2): pontag
(3): cathéter
(4): tronc
(5): tronc commun
(6): évolution
(7): facteur
(8): valeur pronostiquer
(9): vasculaire périphérique
(10): artère
(11): devenir symptomatique
(12): cinétique ventriculaire gauche

Cluster n°8
Hyperonyme1: Cause
Hyperonyme2: causalité
(1): douloureuse
(2): coronaire
(3): heure
(4): angioplast
(5): facteur de risque
(6): facteur
(7): extrasystol ventriculaire
(8): fibrilla
(9): circonflexe
(10): examen
(11): épreuve de effort
(12): ECG

Cluster n°9
Hyperonyme1: Médecine
Hyperonyme2: médecine
(1): majoration du courant de lésion
(2): angioplast
(3): cathéter
(4): traiter par fibrinolys intraveineuse
(5): flux
(6): precordialg
(7): crise
(8): paroi
(9): présenter du precordialg
(10): territoire
(11): tension artériel
(12): test

Cluster n°10
Hyperonyme1: Cœur et vaisseaux
Hyperonyme2: corps
(1): mammaire interner
(2): interventriculair

(3): vaisseau du cou
(4): revascularisa chirurgical
(5): examen
(6): segment
(7): ventriculaire

Cluster n°11
Hyperonyme1: Dissemblance
Hyperonyme2: identité
(1): minime irrégularité
(2): épreuve de effort
(3): symptomatolog
(4): plaquer
(5): réseau
(6): réapparition de un angor
(7): modification electrocardiograph
(8): paroi
(9): obliquer antérieur
(10): modification
(11): marginal gauche
(12): incidence

Cluster n°12
Hyperonyme1: Cœur et vaisseaux
Hyperonyme2: corps
(1): paroi
(2): circulation
(3): interventriculair
(4): atteindre
(5): brancher

Cluster n°13
Hyperonyme1: Cœur et vaisseaux
Hyperonyme2: corps
(1): peu altérer
(2): lésion
(3): precordialg
(4): ventriculaire
(5): ventriculograph
(6): hypertension artériel
(7): incidence
(8): bilan
(9): cinétique ventriculaire gauche
(10): heure

Cluster n°14
Hyperonyme1: Proximité
Hyperonyme2: situation
(1): plaquer
(2): arythm ventriculaire
(3): ECG
(4): ventriculaire
(5): segment proxim
(6): serrer
(7): oblitération
(8): patient
(9): patienter présenter
(10): tronc
(11): tiers proxim

Cluster n°15
Hyperonyme1: Supériorité
Hyperonyme2: quantité
(1): possible de dilater
(2): brancher
(3): plan
(4): flux

Cluster n°16
Hyperonyme1: Maladie
Hyperonyme2: santé et hygiène
(1): récidiver
(2): cathéter
(3): milieu hospitalier
(4): infarctus
(5): évaluation de le maladie coronar
(6): élévation enzymat
(7): ECG
(8): coronarograph
(9): douleur

Cluster n°17
Hyperonyme1: Répétition
Hyperonyme2: quantité
(1): récidiver
(2): coronaire
(3): réseau
(4): caractère dist du lésion
(5): évolution

Cluster n°18
Hyperonyme1: Multiplication
Hyperonyme2: nombre
(1): séquelle de infarctus inférieur
(2): lit de aval
(3): precordialg
(4): modification electrocardiograph
(5): nécroser
(6): oblitération
(7): interventriculair
(8): fraction de éjection
(9): fonction
(10): coronar
(11): complication relier
(12): facteur de risque

Cluster n°19
Hyperonyme1: Cœur et vaisseaux
Hyperonyme2: corps
(1): seul incidence
(2): malade
(3): risque
(4): amplificateur de brillanc
(5): lit de aval
(6): obliquer antérieur
(7): arbre coronar dist
(8): dissection
(9): coronaire
(10): artère

Cluster n°20
Hyperonyme1: Dimension
Hyperonyme2: dimensions
(1): seul incidence
(2): diamètre de le obstruction
(3): possible de dilater
(4): angioplast

Cluster n°21
Hyperonyme1: Médecine
Hyperonyme2: médecine
(1): substance de contraste
(2): évolution
(3): obliquer antérieur
(4): technique
(5): test
(6): traitement
(7): ventricule
(8): ventriculograph

Cluster n°22
Hyperonyme1: Médecine
Hyperonyme2: médecine
(1): sympathique patient
(2): betabloquant
(3): trait de dissection
(4): risque
(5): pontag
(6): coronar
(7): état
(8): infarctus
(9): heure

Cluster n°23
Hyperonyme1: Cause
Hyperonyme2: causalité
(1): sympathique patient
(2): médecin traiter
(3): patient
(4): facteur de risque
(5): diabète non insulino-dependant

Cluster n°24
Hyperonyme1: Quantité
Hyperonyme2: quantité
(1): syndrome douloureux
(2): décalage
(3): hypertrophier
(4): évolution
(5): ECG
(6): doser efficace

Cluster n°25
Hyperonyme1: Transports par route
Hyperonyme2: transports
(1): tapir rouler
(2): coronarograph
(3): technique
(4): bicyclette ergometr
(5): artère
(6): angioplast

Cluster n°26
Hyperonyme1: Méthode
Hyperonyme2: ordre
(1): technique du catheter cardiaque
(2): symptomatolog
(3): technique
(4): infarctus
(5): bloc de branché
(6): revascularisa chirurgical
(7): circonflexe
(8): malade

Cluster n°27
Hyperonyme1: Cœur et vaisseaux
Hyperonyme2: corps
(1): technique du catheter cardiaque
(2): aorte
(3): artère
(4): segment proxim
(5): repriser de le angor
(6): substance de contraste

Cluster n°28
Hyperonyme1: Méthode
Hyperonyme2: ordre
(1): technique du catheter cardiaque
(2): lésion
(3): examen
(4): angor
(5): cathéter

Cluster n°29
Hyperonyme1: Cœur et vaisseaux
Hyperonyme2: corps
(1): tension artériel
(2): douleur
(3): fraction de éjection
(4): ventriculaire
(5): artère
(6): altération

Cluster n°30
Hyperonyme1: Cœur et vaisseaux
Hyperonyme2: corps
(1): tension artériel
(2): coronaire
(3): infarctus
(4): occlusion
(5): effort
(6): fréquence
(7): bilan

Cluster n°31
Hyperonyme1: Cœur et vaisseaux
Hyperonyme2: corps
(1): trait de dissection
(2): bilan
(3): test
(4): angor
(5): extrasystol ventriculaire
(6): artère
(7): arythm ventriculaire
(8): fonction

Cluster n°32
Hyperonyme1: Médecine
Hyperonyme2: médecine
(1): trait de dissection
(2): dissection
(3): coronarograph
(4): patienter présenter
(5): peu altérer
(6): plupart de ce malade

Cluster n°33
Hyperonyme1: Maladie
Hyperonyme2: santé et hygiène
(1): traiter par fibrinolys intraveineuse
(2): coronarograph
(3): risque
(4): devenir symptomatique
(5): bilan

Cluster n°34
Hyperonyme1: Médecine
Hyperonyme2: médecine
(1): traiter par fibrinolys intraveineuse
(2): chirurgie
(3): lésion
(4): traitement
(5): ECG

Cluster n°35
Hyperonyme1: Droite
Hyperonyme2: situation
(1): tritronculair sévère
(2): anévrisme du ventricule gauche
(3): conclusion
(4): dissection
(5): évolution
(6): fraction de éjection
(7): incidence

Cluster n°36
Hyperonyme1: Grammaire
Hyperonyme2: langue
(1): tritronculair sévère
(2): ECG
(3): fonction
(4): pontag

Cluster n°37
Hyperonyme1: Plaisir
Hyperonyme2: émotions
(1): troubler
(2): douleur
(3): douloureuse
(4): spasme
(5): symptomatolog
(6): angor

Cluster n°38
Hyperonyme1: Absence
Hyperonyme2: existence
(1): troubler de le cinétique segmentair
(2): coronarograph
(3): pression
(4): ventriculograph
(5): complication relier
(6): circonflexe
(7): cheminer dans le sillon

Cluster n°39
Hyperonyme1: Cœur et vaisseaux
Hyperonyme2: corps
(1): vaisseau du cou
(2): fraction de éjection
(3): patient
(4): coronaire
(5): état

Cluster n°40
Hyperonyme1: Quantité
Hyperonyme2: quantité
(1): valeur pronostiquer
(2): fréquence
(3): douleur
(4): modification
(5): pression

Cluster n°41
Hyperonyme1: Présence
Hyperonyme2: existence
(1): valve aort
(2): examen
(3): voir
(4): crise

(5): correction chirurgical
(6): dilatation
(7): effort
(8): diamètre de le obstruction

Cluster n°42
Hyperonyme1: Cœur et vaisseaux
Hyperonyme2: corps
(1): valve aort
(2): coronaire
(3): coronar
(4): aorte
(5): circulation

Cluster n°43
Hyperonyme1: Cœur et vaisseaux
Hyperonyme2: corps
(1): valve aort
(2): chirurgie
(3): hypertension artériel
(4): infarctus
(5): insuffisance
(6): ischem

Cluster n°44
Hyperonyme1: Entrée
Hyperonyme2: mouvements et directions
(1): vasculaire périphérique
(2): injecter
(3): introduction du cathéter
(4): malade
(5): plan
(6): profil franc
(7): sévère du tronc commun

Codes corpus: [1] Cœur et vaisseaux(331), [2] Maladie(383), [3] Médecine(391)

ANNEXE 11

Résultats de filtrage avec 2 tests et 3 classeurs

Test 1

Concepts	taille (ko)	A1	nombre classes	T1	Documents pertinents filtrés R+	Documents non pertinents filtrés N+	F1	P	R
nt	350	21	645	14	14	23	-4	38%	100%
liia	101	81	271	54	26	11	56	70%	48%
iut	21	27	0	18	/	/	/	/	/
total	472	129		86	40	34	26	54%	74%

Concepts	u*(S,T)	u*-base
nt	0,07	-0,02
liia	0,34	0,25
base	0,09	
moyenne		**0,12**

Test 2

Concepts	taille (ko)	A2	nombre classes	T2	R+	N+	F1	P	R
nt	417	29	625	5	5	12	-9	29%	100%
liia	142	115	399	19	11	4	25	73%	58%
iut	28	39	2	6	0	0	0	/	/
total	587	183		30	16	16	8	51%	79%

Concepts	u*(S,T)	u*-base
nt	0,10	-0,08
liia	0,41	0,23
base	0,18	
moyenne		**0,07**

	Test 1	Test 2
borne supérieure	3*68=204	3*31=90
borne inférieure	-2*10=-20	-2*10=-20

* signifie tranféré dans le bon classeur
/ signifie pas transféré dans le bon classeur

test nt A1	résultat
1	*
2	*
3	*
4	*
5	*
6	*
7	*
8	*
9	*
10	*
11	*
12	*
13	*
14	*

test liia A1	résultat	score total			
1	/	liia 13	28	*	
2	/	liia 8	29	*	
3	*		30	/	liiaA1 42 liia 44
4	*		31	*	
5	*		32	/	liiaA1 11 liia 42
6	*		33	/	liiaA1 10 liia 69
7	*		34	/	liiaA1 23 liia 64
8	*		35	/	liiaA1 23 liia 89
9	/		36	/	liia 30
10	*		37	/	liiaA1 16 liia 76
11	*		38	*	
12	/		39	*	
13	/	liiaA1 27 liia 28	40	*	
14	*		41	*	
15	*		42	*	
16	/	liiaA1 21 liia 38	43	/	liia 8
17	*		44	/	liiaA1 15 liia 34
18	/		45	*	
19	/	liiaA1 23 liia 26	46	/	liiaA1 31 liia 8
20	/	liiaA1 31 liia 37	47	/	liiaA1 18 liia 32
21	*		48	*	
22	/		49	*	
23	*		50	*	
24	/	liia 23	51	/	liiaA1 26 liia 37
25	*		52	/	liiaA1 19 liia 88
26	/		53	/	liiaA1 52 liia 22
27	/	liiaA1 48 liia 25	54	/	liiaA1 16 liia 26

test nt A2	résultat	score total
1	*	
2	*	liia 345 nt 659
3	*	
4	*	liia 294 nt 539
5	*	liia 382 nt 659

test liia A2	résultat
1	/
2	/
3	/
4	/
5	/
6	/

test liia A2	résultat	score total
1	*	
2	*	
3	*	
4	*	
5	*	
6	/	tout liia 45 score classe:3 termes
7	*	
8	/	tout liia 53 score classe:3 termes
9	/	tout liia 8 score classe:2 termes
10	/	tout liia 34 score classe:2 termes
11	*	
12	/	
13	/	
14	*	
15	*	
16	*	
17	*	
18	/	tout liia 22 score classe:2 termes
19	/	tout liia 26 score classe:1 termes

ANNEXE 12

Détails d'implémentation

Structure de données du dictionnaire

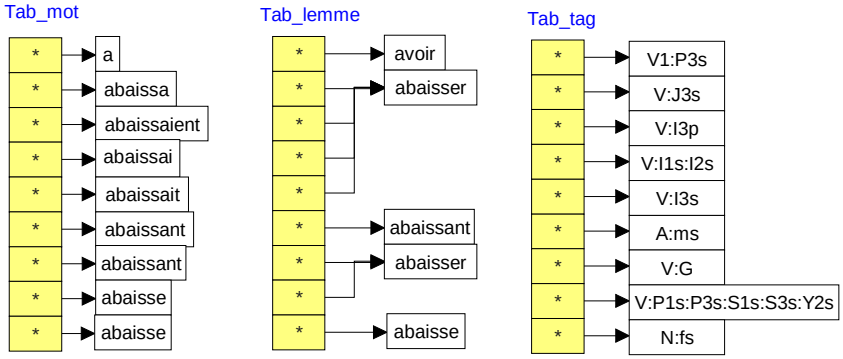


Figure 1 Les 3 tableaux contenant le dictionnaire

Tab_mot, **Tab_lemme** et **Tab_tag** sont des tableaux de pointeurs sur des chaînes de caractères (char **) (figure 1).
Il aurait été intéressant de minimiser la place allouée en faisant, par exemple, pointer les lemmes équivalents des mots « **abaissant** » et « **abaisse** » au lemme du mot « **abaissa** ». Mais ceci demande un algorithme plus lourd. Par conséquent, la mise en mémoire du dictionnaire aurait été plus longue.

Grammaire et unité lexicale pour le dictionnaire

Les langages « **Lex** » et « **Yacc** » sont utilisés pour optimiser le traitement du dictionnaire.
La grammaire de Yacc coïncide avec la strcuture du dictionnaire (figure 2).

```
Fichier_dictionnaire :    '\n' | Liste_de_lignes   |
'\n' Liste_de_lignes
                 | Liste_de_lignes '\n'   | '\n'
Liste_de_lignes '\n'
                     ;

Liste_de_lignes : Ligne   |Liste_de_lignes '\n' Ligne
 ;

Ligne : mot1 ',' lemme '.' tag ;

mot1 : UNMOT ;

lemme : UNMOT ;

tag : UNMOT ;
```

Figure 2 fichier dico_mem.y

Les unités lexicales générées par Lex découpe le dictionnaire (figure 3).

```
MADEFINITION    [a-zA-Z0-9éèêëäâàûüùöôçïî:' ]

%%
[\n][\t \n]*
[ \t\015]+
{MADEFINITION}+
.
```

Figure 3 Fichier dico_mem.l

Les fichiers dico_mem.l génère dico_mem.c et dico_mem.y génère w_tab.h, w_code.c, w_tab.c associant des automates à états finis aux grammaires ainsi définies.

Grammaire et unité lexicale pour les suffixes

```
Char     [a-zA-Zéèàùçêâûôîïüëö]
Sig      (({Char}[\.])+){Char}?
Separ    -
Apostr'
racine {Char}{Char}{Char}+
Mot    ({Char}+)|({Char}[
]?{Apostr})|({Char}+{Separ}{Char}+)
%%


[ \t\r]*  {;}
[\n]+ {fprintf(ptrfile,"\n");}
{racine}alités  { Action(6,4); }
{racine}ilités  { Action(6,5); }
{racine}alité   { Action(5,6); }
{racine}cités   { Action(5,7); }
{racine}ières   { Action(5,8); }
{racine}ilité   { Action(5,9); }
{racine}alement { Action(7,10); }
```

Figure 4 Fichier m_suf.y

Mémorisation des termes

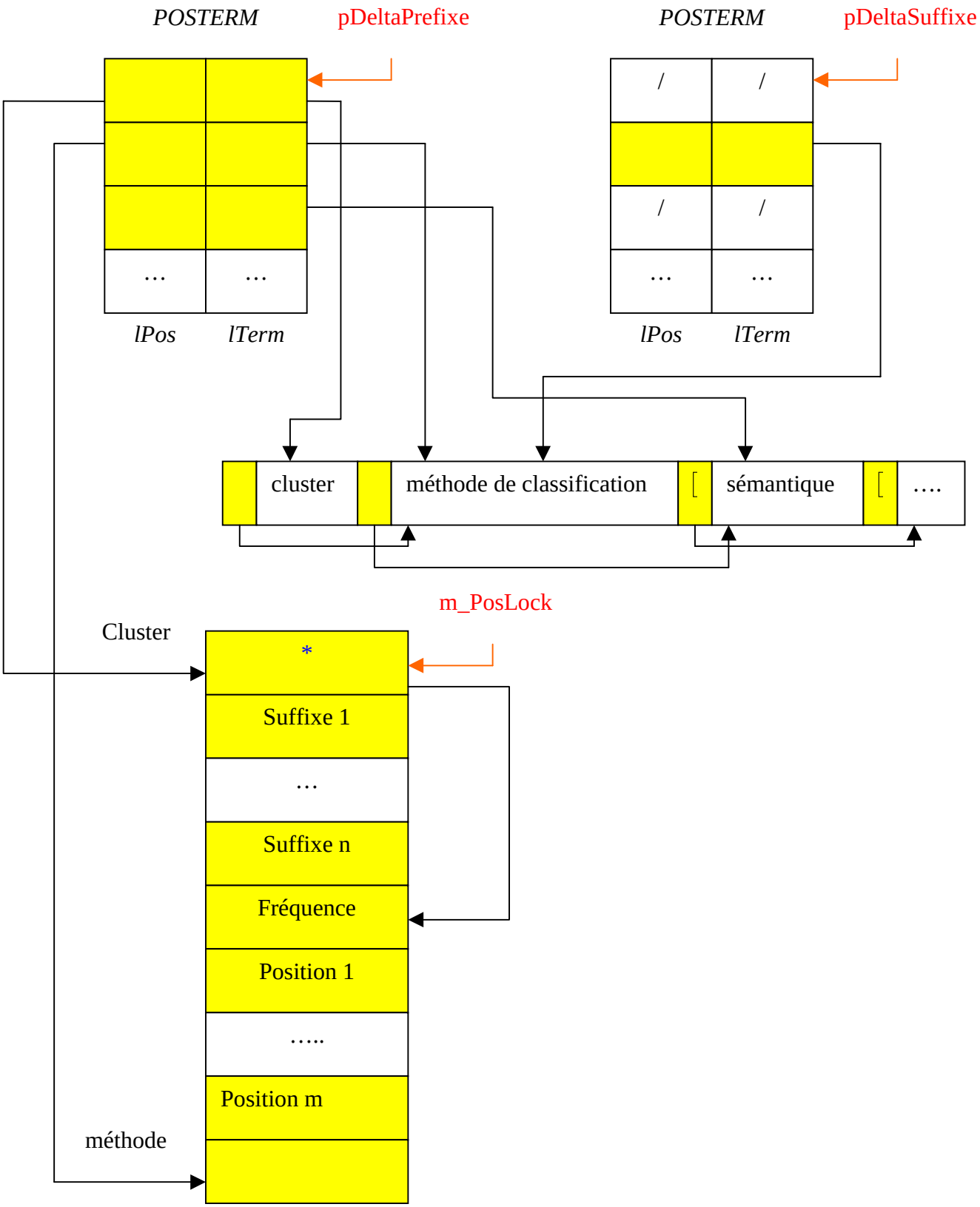


Figure 5 Stockage des termes (monotermes ou multitermes)

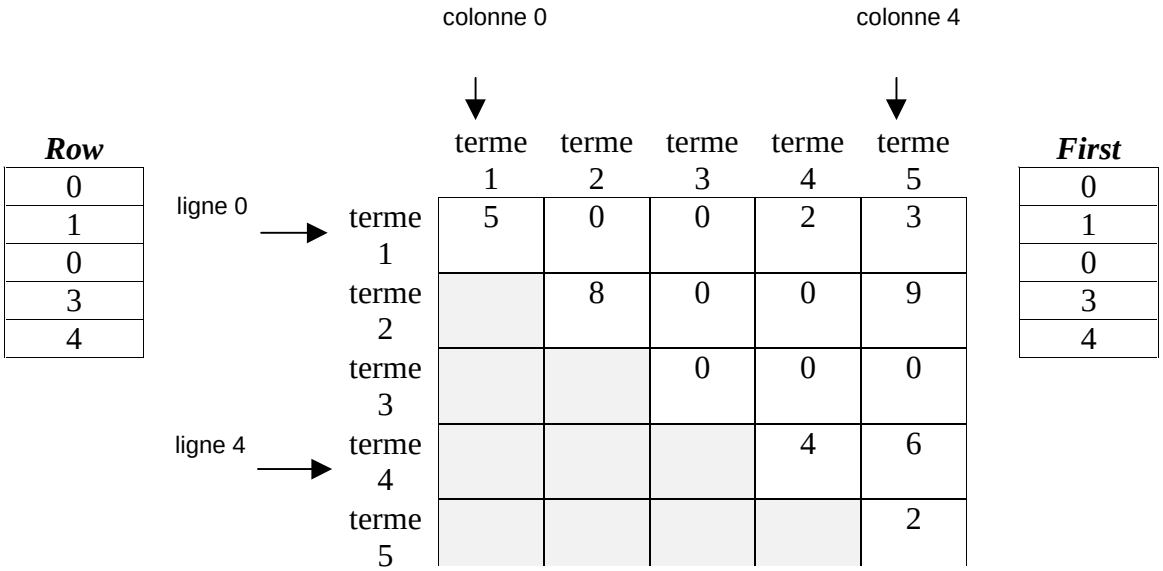
La figure 5 présente la structure de données qui assure le stockage en mémoire des termes lus en entrée du module de classification. La structure est une suite de quatre octets de chaînage suivi du terme. L'octet de chaînage indique la position du terme dans la liste. Le terme est

indexé dans une structure POSTERM qui possède 2 tables : les préfixes (premier mot du terme) et les sufixe (dernier mot du terme). L'indice du terme renvoie à une structure qui donne ses suffixes rencontrés dans le corpus, sa fréquence et ses positions dans le corpus.

Etudions cet algorithme de compression ; la matrice est représentée par un tableau ***Row*** indexé par les numéros de termes, et dont les composantes sont des adresses de lignes vers trois tableaux ***Delta***, ***Valid*** et ***Link***.

Etant donné que les lignes sont creuses, nous disposons les différentes lignes dans un tableau unidimensionnel ***Delta***, nous faisons « glisser » la ligne le long du tableau, jusqu'à ce qu'il n'y ait plus de collision entre les entrés non vides des lignes déjà écrites, et nous inscrivons alors les entrées non vides de la ligne. Dans **Row*[i]*** nous inscrivons l'index dans ***Delta*** à partir duquel la i-ième ligne est allouée.

Mais la matrice ainsi représentée a perdu la capacité de reconnaître les entrées qui restent indéfinies, lorsque par exemple ***Delta(i,j)*** à une valeur nulle et que ***Delta[Row[i]+j]*** contient une entrée non vide pour un terme $k \neq i$. C'est pourquoi il nous faut un deuxième tableau ***Valide*** de même longueur que ***Delta***, et qui indique à quels termes correspondent les entrées qui se trouvent dans ***Delta*** ; ***Valide[Row[i] + j] = i*** lorsque ***Delta(i,j)*** est défini (figure 6).



Delta	5	-	8	2	3	9	4	6	2	-	-
Valid	1	0	2	1	1	2	4	4	5	0	0
Link	3	0	4	4	0	0	4	0	0	0	0

Figure 6 Vecteurs : row, first et delta pour coder la matrice

La fonction d'accès à un élément *(Ligne,Col)* de la matrice est la suivante :

Si Valid[Row[Ligne] + Col] == Ligne + 1
    Alors (Ligne,Col) = Delta[Row [Ligne]+ Col];
    Sinon (Ligne,Col) = 0

Voici le code en C pour exécuter cet algorithme :

```
DWORD Valeur,  Ligne = 1,  Col = 5 ;
if (lpMatrix->m_pValid[lpMatrix->m_pRow[Ligne] + Col]== Ligne + 1)
        Valeur = ((DWORD*)lpMatrix->m_pDelta)[lpMatrix->m_pRow[Ligne]+ Col];
Else
        Valeur = 0 ;
```

Rem : ***LpMatrix*** est la structure globale.
m_pRow et ***m_pDelta*** sont les pointeurs sur les tableaux ***Row*** et ***Delta***.

Le vecteur ***Row*** indique donc la position dans le tableau de la première colonne de la ligne passée en paramètre (indice du vecteur).

Le vecteur ***First*** indique la position de la première Colonne dont l'élément est non nul (le numéro de ligne est l'indice du vecteur).
Exemple : pour accéder à la première valeur non nulle de la ligne 2, il faut écrire :

Delta [Row[1] + First[1]]

Rem : La ligne 2 est l'indice 1, car les indices des vecteurs commencent à 0.
Enfin, le vecteur ***Link*** indique la valeur suivante non nulle de la ligne. La valeur est l'indice à ajouter à la valeur du vecteur ***Row***.
Les vecteurs ***First***, ***Link*** ont été conçus pour un parcours rapide de tous les éléments (recherche de cliques d'ordre 3).
Rem : Les valeurs de la diagonale indiquent le nombre d'occurrence du terme dans le texte.
La figure 7 donnee une représentation plus explicite pour une meilleure compréhension .

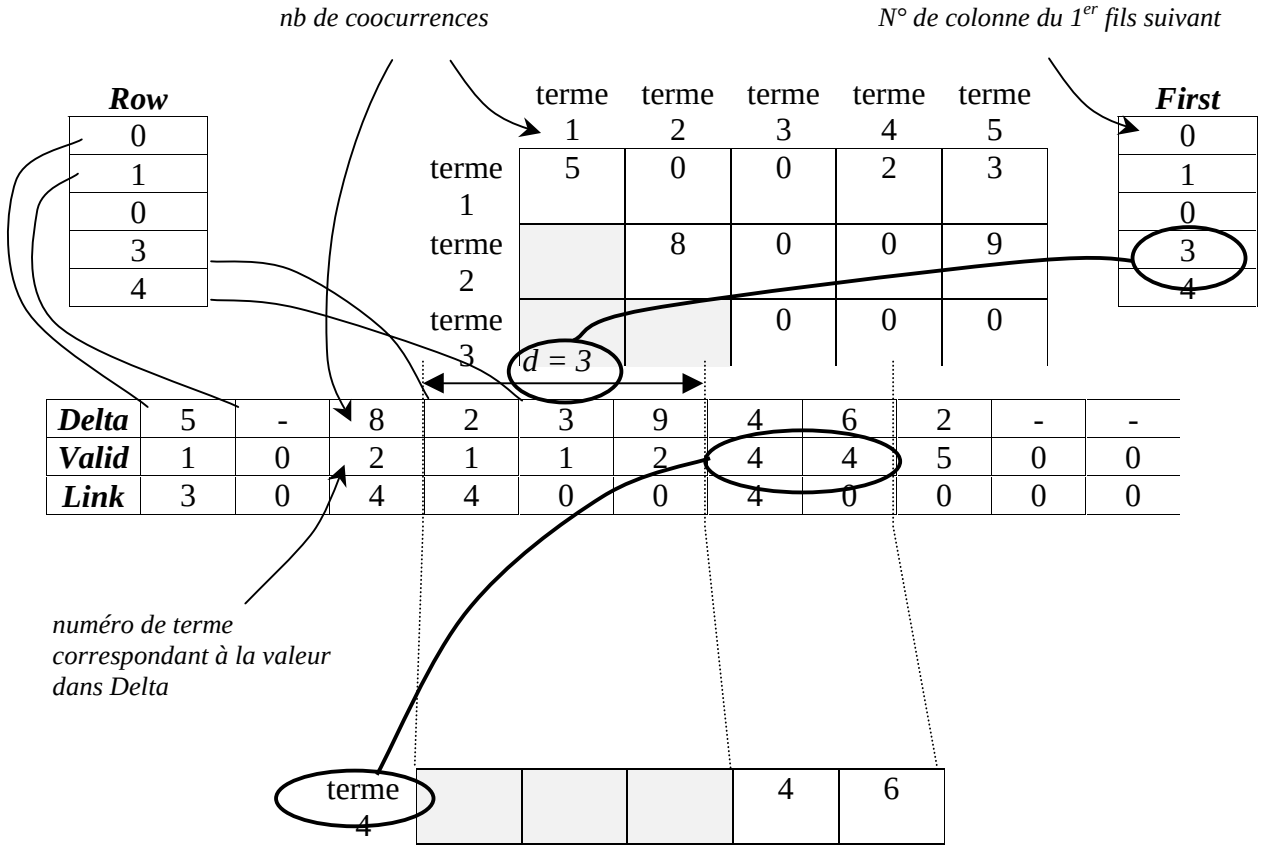


Figure 7 Accès à la matrice

Dépendances des fichiers

Le module classification a été développé en langage C mais avec une organisation objet des fichiers et des fonctions (figure 8). Chaque fichier représente l'équivalent d'une classe et les fonctions associées sont l'équivalent de ses méthodes.
Les fichiers **.H** sont des fichiers de déclarations des constantes, variables, structures de données et prototypes des fonctions du thème concerné, tandis que les fichiers **.CPP** ou **.C** sont les fichiers de code proprement dit.

Les fichiers .L sont des fichiers de spécifications du générateur d'analyseur lexical **FLEX** (langage **LEX**), tandis que les fichiers .Y sont des fichiers du générateur d'analyseur syntaxique **BYACC** (langage **YACC** de Berkeley).

Le tableau suivant donne une correspondance thème/fichiers.

Lecture du fichier des termes, construction du dictionnaire des termes.	Spill.cpp
Analyse et représentation en mémoire du fichier des positions	Position.cpp Cooc.l Cooc.y YyCooc.cpp YyLex.cpp
Détermination et représentation de la matrice de cooccurrence.	Cooc.cpp Cooc1.cpp Cooc2.cpp Cooc3.cpp
Calcul des termes pôles.	Pole.cpp
Calcul des cliques primaires de cooccurrences.	Cycles.cpp
Calcul des cliques de quatre	Clique4.cpp
Calcul et optimisation des classes	Classe.cpp
Gestion des erreurs.	Error.h Error.cpp
Utilitaires divers.	Utils.h Utils.cpp Util_Mat.cpp

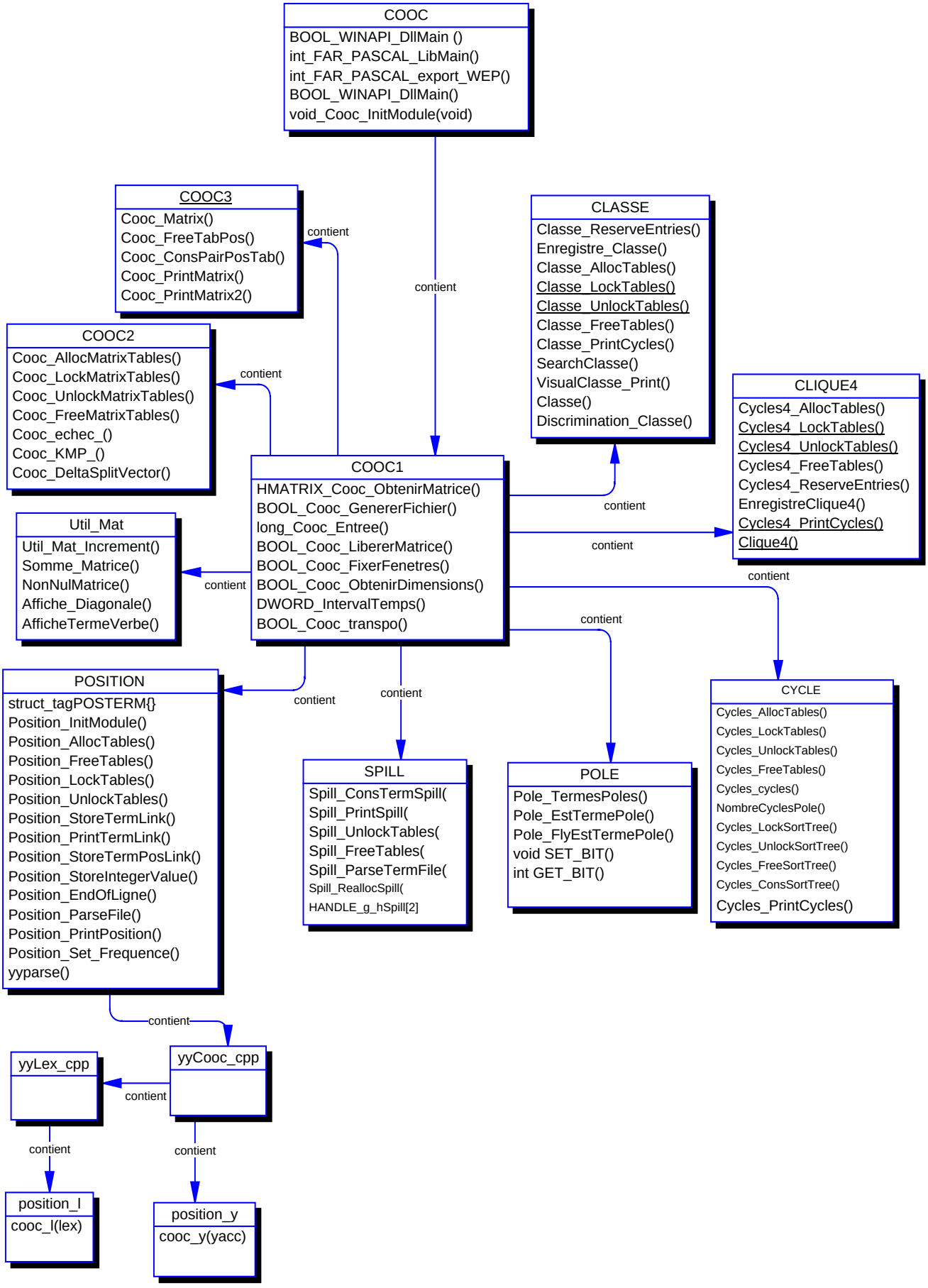


Figure 8 Schéma conceptuel des dépendances de *cooc.dll*

Il existe de plus des fichiers pour :

le debugage : ***Debug.cpp***
la gestion des erreurs : ***Error.cp***p
les fonctions liées au système d'exploitation : ***Sys.cpp***
les fonctions utilitaires : ***Utils.cpp***
la définition d'une chaîne de caractères : ***String.cpp***

Interface graphique de visualisation des classes

L'interface graphique met en scène non moins de 44 classes qui se rangent dans l'un des trois groupes cités plus haut. Cependant, nous pouvons aussi faire d'autres regroupements, notamment pour les composants visuels.

Voici comment les classes sont organisées :

Elles contiennent deux classes. La première contient toutes les fonctions pour charger en mémoire et décortiquer le fichier de données, le découpage en classes, l'extraction des hyperonymes, des nœuds et des relations. La deuxième est simplement une classe intermédiaire pour le découpage des classes. Cependant, pour des raisons de séparation des fonctions d'interprétation du fichier et de l'affichage, certaines fonctions appelant celles-ci et les liant entre elles se situent dans la classe *Grapheur*. La classe d'entrées-sorties contient aussi les fonctions de lecture et d'écriture dans les bases *Access*. Il y a d'abord la fonction qui crée les bases si elles n'existent pas, ensuite la fonction de remplissage qui analyse le corpus et le décompose pour stocker les mots dans la table. Une fonction de simplification de requête dans la base permet de récupérer la donnée que l'on veut à partir de la requête au format SQL standard passée en argument (figure 9). Viennent enfin des fonctions de récupération de listes d'entiers et de chaînes dans le résultat d'une requête.

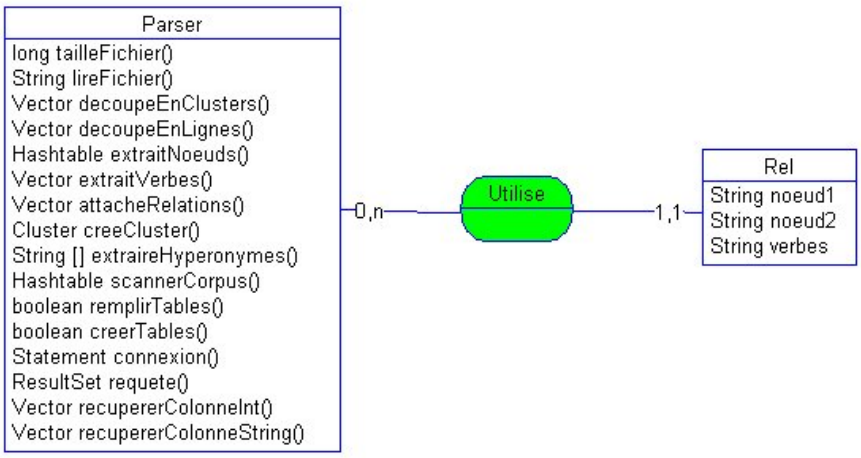


Figure 9 Schéma conceptuel de l'analyseur du fichier d'entrée et de bases de données

Chaque classe est rattachée à deux hyperonymes, ce qui permet de réaliser aussi la classement par hyperonymes. Chaque classe contient aussi la liste des nœuds et relations qu'il contient.

Chaque nœud contient un texte : le terme, la liste des relations qui lui sont associées et un type qui détermine si c'est un pôle, un pivot ou un autre nœud.

Chaque relation contient une référence sur les deux nœuds qu'elle lie, la liste de ses verbes sous forme de liste et sous forme de chaîne et son type suivant les nœuds qu'elle lie, plus un booléen pour accélérer la recherche des relations d'un nœud à lui-même.

Les hyperonymes ne contiennent que leur nom.

C'est elle qui contient tous les composants ainsi qu'un certain nombre de traitements d'événement et les propriétés de préférences. C'est aussi cette classe (*Grapheur*) qui contient la procédure main qui permet de lancer l'application (figure 10). *Grapheur* contient aussi la partie nécessitant un affichage des fonctions d'interprétation de fichier. C'est aussi dans cette classe que se font les traitements liés aux préférences. En effet, les classes *FenPref* (fenêtre de choix des préférences), Nuancier et *TSLPanel* ne sont que des interfaces. Une fonction de la classe *Grapheur* fouille tous les objets présents dans l'application et change leurs propriétés (*changerPreferences*()). C'est aussi la classe *Grapheur* qui contient les fonctions pour générer les arbres visuels contenant les hiérarchies de recherche (*genereArbrePole*(), *genereArbreHyperonyme*()).

La classe a d'abord été conçue pour le chargement de fichiers via un dialogue, cependant pour des raisons d'intégrité de l'application, ces fonctions ont été bridées.

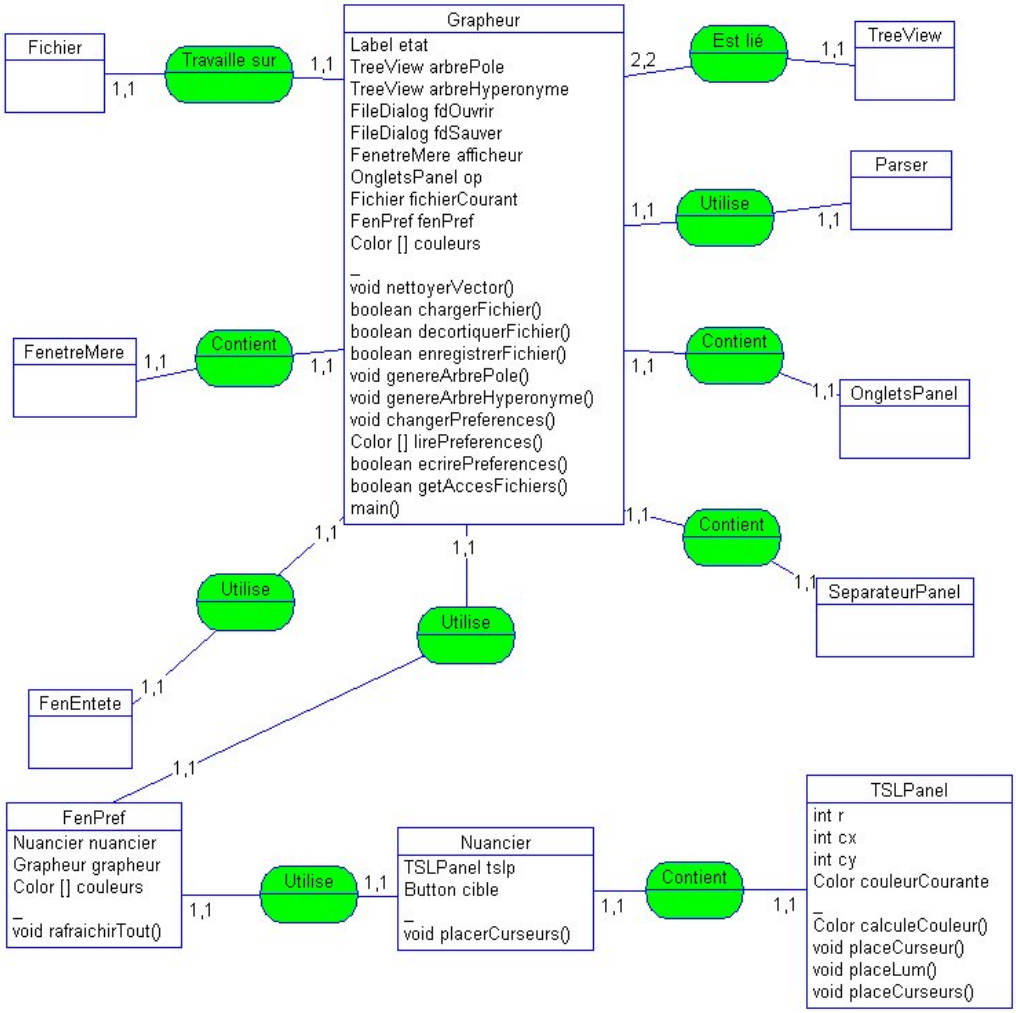


Figure 10 Schéma conceptuel de la classe principale et de ses composants

Sur ce panel (*ClusterPanel*) s'affichent les différents composants visuels qui représentent une classe. Premièrement, nous pouvons lui associer une classe. Cette opération mettra en place un *ClusterVisuel* qui sera le pendant graphique du *Cluster*(figure 11). C'est lui qui contiendra les listes de nœuds visuels et de relations visuelles. Ensuite, nous avons les nœuds (*NoeudVisuel*), répartis en bandes suivant leur type (pôle, pivot, autre terme). Ceux-ci peuvent être déplacés par glisser-déposer de la souris pour les positionner à un autre endroit du panel pour des raisons de présentation ou de lisibilité. Ceci est possible grâce aux écouteurs d'événements souris (*MouseListener* et *MouseMotionListener*) ajoutés à la classe. Les relations (*RelationVisuelle*) ne sont pas à proprement dit des composants visuels. La classe contient en fait les méthodes d'affichage sur un contexte graphique. C'est donc le panel qui doit faire appel à celles-ci. C'est aussi le panel qui détermine, faisant appel à la méthode correspondante de toutes les relations, que le curseur de la souris se trouve dans le point de l'une ou l'autre des relations. Dans ce cas, le curseur correspondant contenu dans le composant visuel doit être assigné au pointeur de souris pour indiquer à l'utilisateur qu'il peut cliquer. S'il y a un clic sur un point, il faut alors que le panel modifie la propriété d'affichage de la bulle pour la relation. Un rafraîchissement prenant en compte ce changement permet alors d'afficher la bulle contenant le texte de la relation. Ce texte est stocké dans la relation visuelle sous forme de liste de chaînes, c'est le texte de départ (les verbes) découpé à la création du panel pour donner le texte formaté pour la bulle.
Le panel de visualisation possède aussi d'autres fonctions. Lorsque l'on clique sur un nœud, on peut choisir d'avoir toutes les relations allant de ce nœud à lui-même. Pour cela, le panel possède une fenêtre contenant une liste qui est remplie lors de l'affichage avec les informations souhaitées (*FenRelPro*). On peut aussi en cliquant sur un endroit libre du panel afficher des statistiques sur l'ensemble de la classe concernant la répétition des verbes des relations.
La gestion de l'impression d'un graphe se trouve aussi dans la classe *ClusterPanel*.

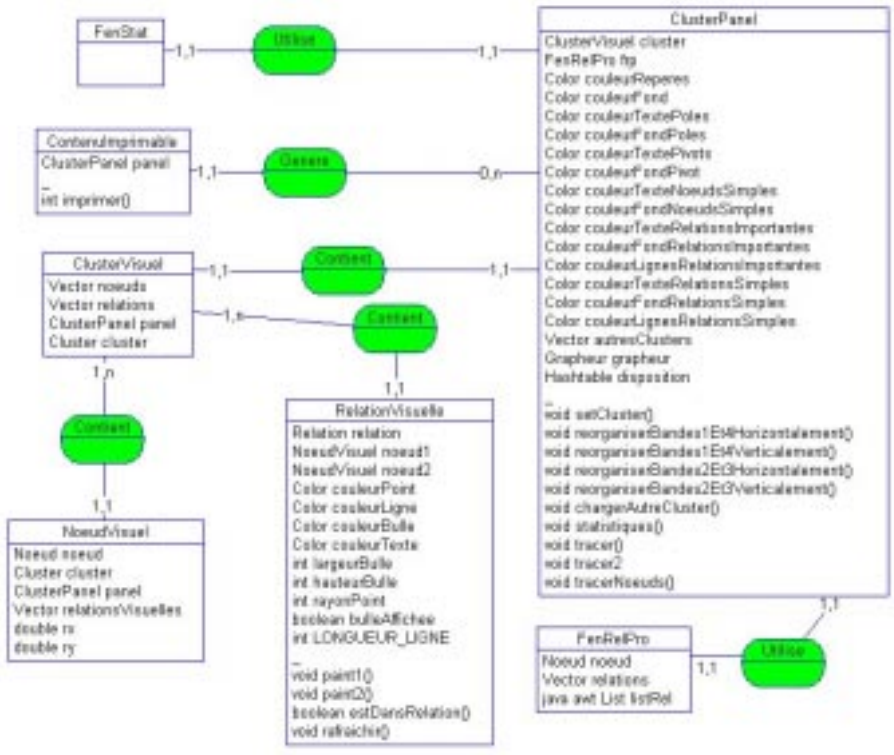


Figure 11 Schéma conceptuel des classes de représentation graphique d'une classe

Le système MDI permet d'afficher dans un espace donné (*TravailPanel*) comme ici une portion de fenêtre plusieurs petites fenêtres (*FenetreFille*) contenant différents documents ouverts simultanément, ici les panels d'affichage de classes (figure 12). Ces fenêtres peuvent à loisir être déplacées, redimensionnées, agrandies, superposées, réduites ou fermées. Une barre de boutons (située dans le bas de la zone contient un bouton pour chaque fenêtre ouverte. Il suffit de cliquer sur le bouton correspondant pour restaurer une fenêtre réduite auparavant ou la passer en premier plan. Il est possible aussi de réorganiser les fenêtres selon une disposition de mosaïque sur deux bandes.

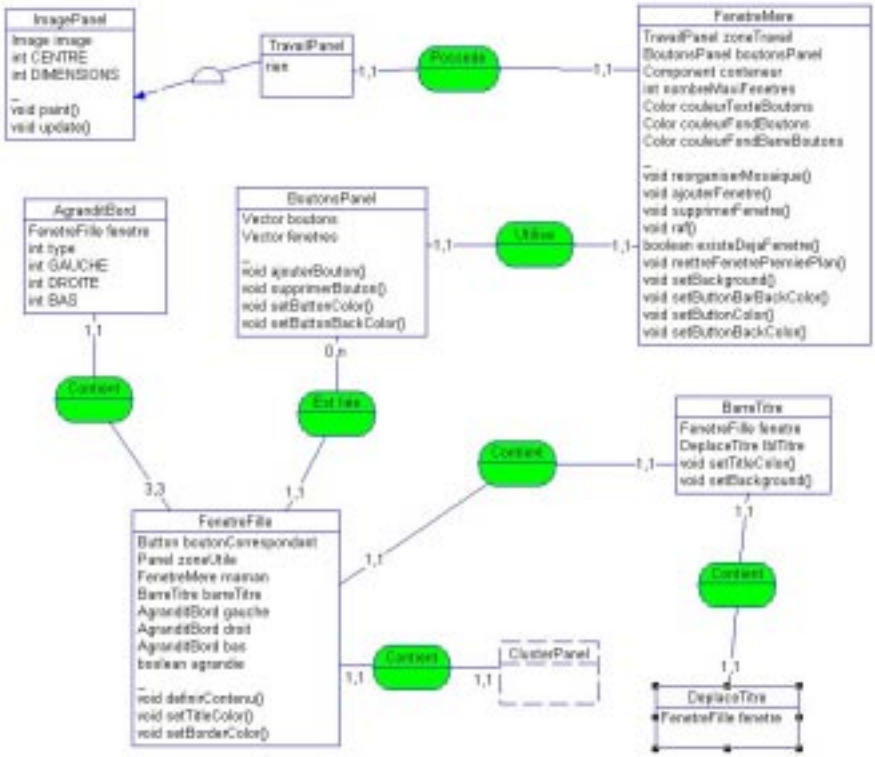


Figure 12 Schéma conceptuel du système MDI

L'arbre de visualisation se compose de deux classes. La première est le nœud (*TreeNode*) (figure 13). Ce nœud contient une référence sur le frère qui le suit verticalement dans l'arbre et une sur son premier fils. De cette manière, il est possible à partir de la racine de balayer récursivement tous les nœuds pour les accès et pour l'affichage. Il possède sa méthode d'affichage sur un contexte graphique donné. Si l'on veut ajouter un frère à cet arbre, une procédure itérative passera dans ses frères jusqu'à atteindre le dernier (propriété next = null). Il suffit de placer alors le nouveau nœud comme frère du dernier trouvé. Si l'on veut ajouter un fils, il y a deux cas de figure : soit le nœud n'a pas de fils, dans ce cas on référence l'arbre fils dans la propriété *child*, soit il y en a déjà un (child ≠ null), dans ce cas il faut balayer les propriétés *next* du fils et de ses frères jusqu'à trouver *null*. On sait que l'on peut alors placer notre nœud à la suite de celui-ci. Grâce aux deux ascenseurs, il est possible de faire défiler l'arbre. Cependant, comme les nœuds ne sont pas des composants graphiques mais possèdent simplement des méthodes graphiques, tout l'arbre sera redessiné au moindre déplacement, d'où une certaine lenteur de l'affichage. Cependant, le principe est simple et peu gourmand en mémoire.

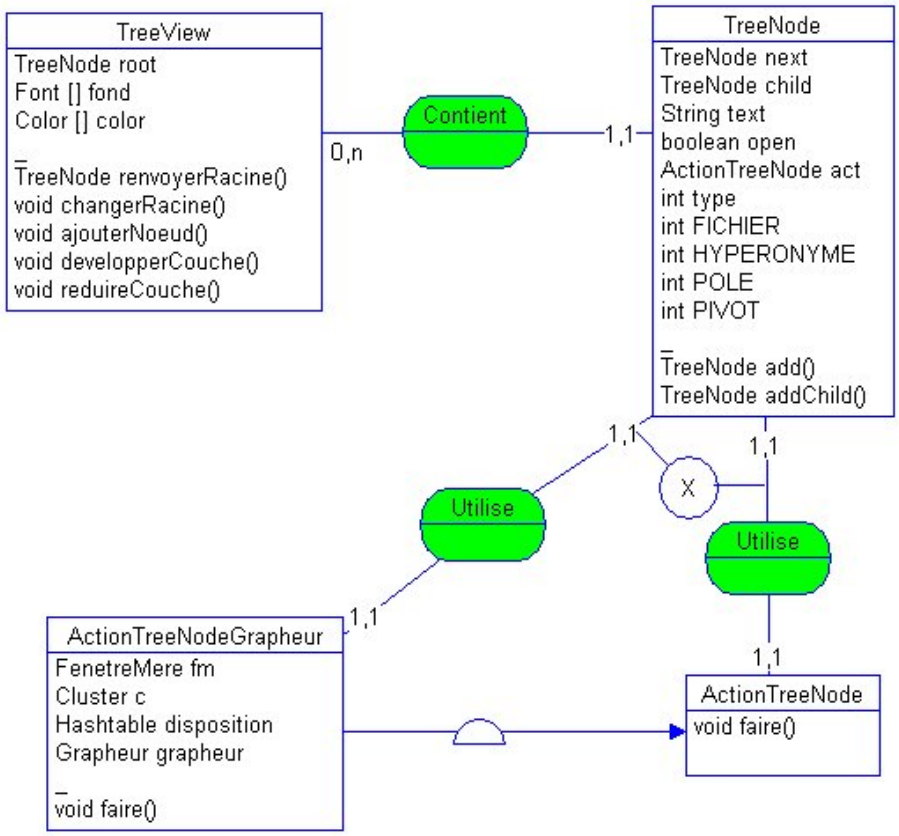


Figure 13 Schéma conceptuel du *treeview*

La présentation du composant panel d'onglets ne ressemble certes pas au traditionnel *TabbedPanel* que l'on trouve dans les produits *Microsoft*, mais ses fonctionnalités sont les mêmes, voire meilleures pour une utilisation de base (figure 14). Chaque onglet est un panel et est lié à un bouton dans la barre de boutons. Le clic sur un bouton de la barre masque le panel affiché s'il y en a un et le remplace par le panel correspondant au dit bouton. Si la taille ou le nombre de boutons dépasse ce qui peut être affiché, un *ScrollPanel* se charge automatiquement de placer les barres de défilement pour la barre de boutons.

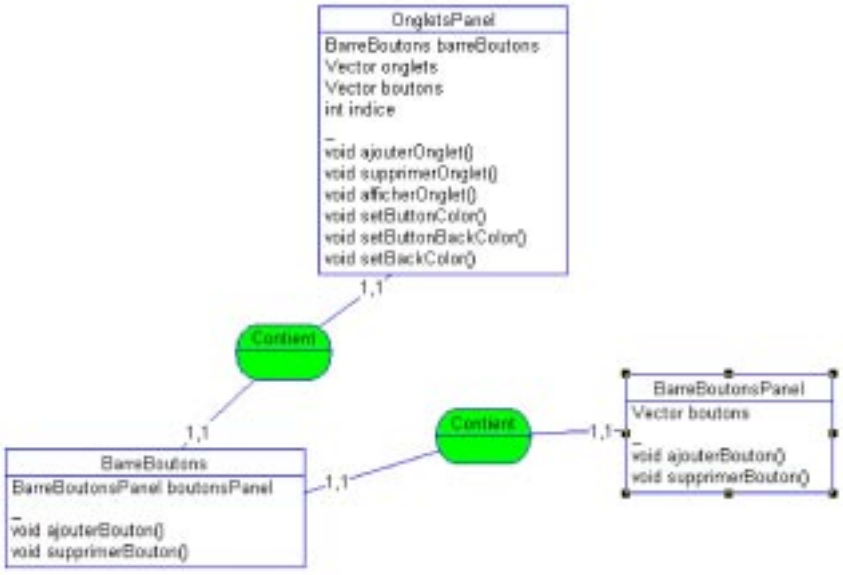


Figure 14 Schéma conceptuel du panel d'onglet

Comme dans *l'Explorateur Windows*, il fallait trouver un moyen de répartir la surface occupée par le *TreeView* à gauche et le système MDI à droite. Le composant séparateur vertical, gère donc un panel auquel on peut ajouter un panel de gauche et un panel de droite qui sont séparés par une barre que l'on peut déplacer horizontalement (figure 15). La première classe est le panel contenant les composants de l'objet (*SeparateurPanel*) (figure 16). C'est sur elle que l'on peut placer les deux panels correspondant aux contenus désirés. La deuxième classe (*Separateur*) sert à déplacer le point de séparation. C'est aussi un panel qui possède lui des écouteurs d'événements sur la souris. Le fait d'utiliser le glisser-déposer en déplaçant horizontalement la souris permet de déplacer le contrôle pour redimensionner les deux zones.

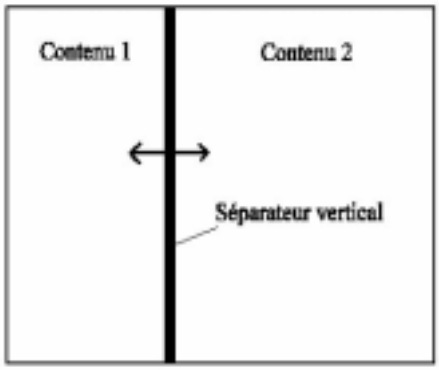


Figure 15 Schéma d'utilisation du séparateur vertical

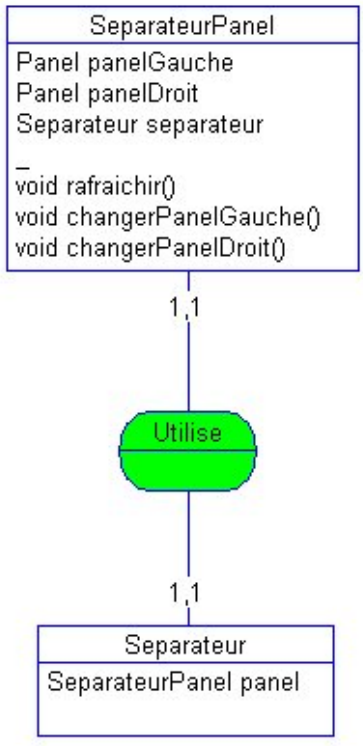


Figure 16 Schéma conceptuel du séparateur vertical

INDEX