Etude et Evaluation d'Approches Multiples d'Expansion de Requêtes pour une Recherche d'Information Intelligente en Santé

Lina Fatima Soualmia

▶ **To cite this version:**

Lina Fatima Soualmia. Etude et Evaluation d'Approches Multiples d'Expansion de Requêtes pour une Recherche d'Information Intelligente en Santé. Informatique [cs]. INSA de Rouen, 2004. Français. NNT : . tel-01371361

HAL Id: tel-01371361

https://theses.hal.science/tel-01371361

Submitted on 25 Sep 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

ECOLE DOCTORALE SPMI

THESE
DE
L'INSA DE ROUEN

Présentée en vue de l'obtention du

DOCTORAT ES SCIENCES

Discipline : INFORMATIQUE

PAR

SOUALMIA LINA FATIMA

INGENIEUR EN INFORMATIQUE

ETUDE ET EVALUATION D'APPROCHES MULTIPLES D'EXPANSION DE REQUETES POUR UNE RECHERCHE D'INFORMATION INTELLIGENTE :

APPLICATION AU DOMAINE DE LA SANTE SUR L'INTERNET

Soutenue publiquement le 14 Décembre 2004 devant le jury composé de :

M. H. ABDULRAB	Professeur INSA de Rouen (Directeur)
M. B.BACHIMONT	Directeur de Recherche INA (Rapporteur)
Mme C.BARRY	Maître de Conférences Université de Picardie (Examinateur)
M. SJ.DARMONI	Professeur Université de Rouen (Directeur)
Mme C.GOLBREICH	Professeur Université Rennes 1 (Examinateur)
Mme MC.JAULENT	Directeur de Recherche INSERM (Rapporteur)
M. P.ZWEIGENBAUM	Professeur associé INaLCO (Président)

Thèse préparée au sein :

du laboratoire PSI, INSA de Rouen, CNRS – FRE 2645
Place Emile Blondel, BP 08, 76131 Mt St Aignan Cedex
&
de la Direction de l'Informatique et des Réseaux du CHU de Rouen
1 rue de Germont, 76031 Rouen Cedex, tel 02 32 88 88 00.

TABLE DES MATIERES

1CHAPITRE 1
RECHERCHE D'INFORMATION :
DE LA REQUETE AUX DOCUMENTS

1.1 INTRODUCTION 27
1.2 INTERNET ET LE WEB 28
1.2.1 HISTORIQUE D'INTERNET 28
1.2.2 CARACTERISTIQUES DU WEB 29
1.2.3 LIMITES DES INFORMATIONS DU WEB 30
1.3 FONDEMENTS DE LA RECHERCHE D'INFORMATION 31
1.3.1 LA RECHERCHE DOCUMENTAIRE 31
1.3.2 LES SYSTEMES DE RECHERCHE D'INFORMATION 32
1.3.3 LA TACHE DE NAVIGATION 33
1.3.4 LA TACHE DE RECHERCHE 34
1.3.5 L'INDEXATION 34
1.3.5.1 L'ESPACE D'INDEXATION 36
1.3.5.2 LES ENTITES D'INDEXATION 36
1.3.5.3 LES LANGAGES D'INDEXATION 37
1.3.5.3.1 Le Langage Libre *37*
1.3.5.3.2 Le Langage Contrôlé *38*
1.3.5.4 LES TYPES D'INDEXATION 39
1.3.5.4.1 Indexation Manuelle *39*
1.3.5.4.2 Indexation Automatique *39*
1.3.5.4.3 Indexation Semi-Automatique *40*
1.3.5.5 CAS DE L'INDEXATION AUTOMATIQUE 40
1.3.5.5.1 Extraction des Termes d'Indexation 40
1.3.5.5.2 Réduction du Langage d'Indexation 43
1.3.5.5.3 Pondération des Termes d'Indexation 44
1.4 MODELES DE RECHERCHE D'INFORMATION ET DE REPRESENTATION 46
1.4.1 LE MODELE BOOLEEN 46
1.4.2 LE MODELE VECTORIEL 47
1.4.3 LE MODELE PROBABILISTE 49
1.4.4 LE MODELE LOGIQUE 50
1.5 ENVIRONNEMENTS DE RECHERCHE 50
1.6 EVALUATION DE LA RECHERCHE D'INFORMATION 51
1.7 LA RECHERCHE D'INFORMATION SUR LE WEB 53
1.7.1 CARACTERISTIQUES DE LA RECHERCHE D'INFORMATION SUR LE WEB 53
1.7.2 LES PROBLEMES DE LA RECHERCHE D'INFORMATION SUR LE WEB 54
1.7.3 PROBLEMES LIES AU PROCESSUS DE RECHERCHE SUR LE WEB 55
1.7.4 APPROCHES EXISTANTES POUR L'AIDE A LA RECHERCHE D'INFORMATION SUR LE WEB 56
1.7.4.1 LES FACTEURS HUMAINS 56
1.7.4.2 LE PROCESSUS DE RECHERCHE 56
1.7.4.2.1 La Tâche de Navigation 57
1.7.4.2.2 La Tâche de Recherche 57

1.7.4.2.3 Les Méta-Moteurs de Recherche d'Information 60
1.7.4.3 LA VISUALISATION DES RESULTATS 61
1.7.4.4 LES AGENTS 62
1.7.4.4.1 Les Agents de Recherche 62
1.7.4.4.2 Les Agents de Recommandation 62
1.7.4.4.3 Approches Multi-Agents 63
1.9 CONCLUSION 64

CHAPITRE 2
RECHERCHE DE DOCUMENTS EN SANTE : CAS DU CATALOGUE CISMEF

2.1 INTRODUCTION 73
2.2 STRUCTURE DES DOCUMENTS 74
2.2.1 LES METADONNEES 74
2.2.2 LE THESAURUS MESH 75
2.2.3 LA TERMINOLOGIE CISMEF 76
2.2.3.1 LES TYPES DE RESSOURCES 76
2.2.3.2 LES METATERMES 77
2.2.4 LE MODELE CISMEF POUR LA RECHERCHE D'INFORMATION 78
2.3 LA METHODOLOGIE DE MISE A JOUR 79
2.3.1 LE RECENSEMENT 79
2.3.2 LA SELECTION 80
2.3.3 LA DESCRIPTION 80
2.4 RECHERCHE D'INFORMATION DANS LE CISMEF 82
2.4.1 ACCES STATIQUE 83
2.4.1.1 LES DEFINITIONS 83
2.4.1.2 LES 'VOIR AUSSI' 84
2.4.1.3 LES ARBORESCENCES 85
2.4.2 ACCES DYNAMIQUE 86
2.4.2.1 LA NAVIGATION DYNAMIQUE 86
2.4.2.2 LA RECHERCHE SIMPLE 87
2.4.2.3 LA RECHERCHE AVANCEE 88
2.4.2.4 LA RECHERCHE BOOLEENNE 89
2.4.2.5 LA RECHERCHE PAS-A-PAS 90
2.4.2.6 OPTIONS DE RECHERCHE 90
2.4.2.6.1 Options par Défaut 90
2.4.2.6.2 Option Arborescence 91
2.4.2.6.3 Option Explosion 91
2.4.2.6.4 Option Majeur/Mineur 92
2.4.3 LES AFFILIATIONS 93
2.4.3.1 AFFILIATION DE QUALIFICATIFS 93
2.4.3.2 AFFILIATION DE TYPES DE RESSOURCES 93
2.4.4 LES REQUETES PREFORMATEES 94
2.4.4.1 LES STRATEGIES DE RECHERCHE 94
2.4.4.2 CISMEF-PATIENTS 95
2.4.4.3 LE PROJET COGNI-CISMEF 99
2.4.5 CATEGORISATION DES DOCUMENTS 100
2.5 QUELQUES PROBLEMES RENCONTRES 102
2.5.1.1 AJOUT D'AUTRES TYPES DE SYNONYMES 104
2.5.1.2 UTILISATION DE CONNAISSANCES 104

CHAPITRE 3
TRAITEMENTS LINGUISTIQUES : DE LA CHAINE DE CARACTERES A LA REQUETE

3.1 INTRODUCTION 109
3.2 PROBLEMES LIES AU TRAITEMENT DE LA LANGUE 111
3.3 LE TRAITEMENT MORPHOLOGIQUE 113
3.3.1 LE MODELE MORPHOLOGIQUE 114
3.3.2 APPLICATION DE LA MORPHOLOGIE A LA RECHERCHE D'INFORMATION 117
3.4 ACQUISITION DE RESSOURCES LINGUISTIQUES 118
3.4.1 LOGICIELS D'ACQUISITION DE TERMINOLOGIE 118
3.4.2 ACQUISITION DE CONNAISSANCES MORPHOLOGIQUES 119
3.4.3 DOMAINE MEDICAL 120
3.4.3.1 METHODE D'ACQUISITION DE RESSOURCES MORPHOLOGIQUES 121
3.4.3.2 LE PROJET UMLF 122
3.5 ACQUISITION DE CONNAISSANCES MORPHOLOGIQUES POUR LE MESH FRANÇAIS 123
3.5.1 DESCRIPTION DE 'LEXIQUE' 123
3.5.2 CONSTITUTION ET EVALUATION DES FAMILLES EXTRAITES 124
3.6 TRAITEMENTS LINGUISTIQUES POUR LA RECHERCHE D'INFORMATION 127
3.6.1 UTILISATION DE CONNAISSANCES MORPHOLOGIQUES 127
3.6.1.1 TRAVAUX INITIATEURS 128
3.6.1.2 ETUDE DU VOCABULAIRE DES UTILISATEURS 128
3.6.1.3 EXPERIENCES AVEC DES RESSOURCES ADAPTEES AU VOCABULAIRE 130
3.6.2 RESULTATS 132
3.6.2.1 DESCRIPTION DES REQUETES 132
3.6.2.2 RESULTATS AVEC LES UNITERMES 132
3.6.2.3 RESULTATS AVEC LES TERMES COMPOSES 133
3.6.3 EXPERIENCES DE PHONEMISATION 133
3.6.3.1 TRAVAUX ANTERIEURS : SOUNDEX / SOUNDEX 2 / PHONEX 134
3.6.3.2 PHONEMISATION DE TERMES MEDICAUX 135
3.6.3.3 APPLICATION A LA RECONNAISSANCE DE TERMES 138
3.6.4 CORRECTION ORTHOGRAPHIQUE 139
3.6.4.1 ALGORITHME 139
3.6.4.2 EVALUATION DES RESULTATS 140
3.7 TRAITEMENTS EN LIGNE 141
3.7.1 APPARIEMENT A BASE DE CONNAISSANCES MORPHOLOGIQUES 141
3.7.2 AUTRES TRAITEMENTS 142

CHAPITRE 4
DECOUVERTE DE CONNAISSANCES DES TEXTES AUX REGLES D'ASSOCIATION

4.1 INTRODUCTION 153
4.2 FOUILLE DE DONNEES 154
4.2.1 FOUILLE DE DONNEES TRANSACTIONNELLES 154
4.2.2 EXTRACTION D'ITEMSETS 158
4.2.3 EXTRACTION D'ITEMSETS FREQUENTS 159
4.2.4 PROCESSUS D'EXTRACTION DE REGLES D'ASSOCIATION 162
4.2.4.1 PREPARATION DES DONNEES 163
4.2.4.2 EXTRACTION DES ENSEMBLES FREQUENTS D'ATTRIBUTS 163
4.2.4.3 GENERATION DES REGLES D'ASSOCIATION 163
4.2.4.4 INTERPRETATION DES RESULTATS 163
4.2.5 AUTRES METHODES 164
4.2.5.1 DECOUVERTE DE PEPITES DE CONNAISSANCES 164
4.2.5.2 EXTRACTION D'ITEMSETS FREQUENTS MAXIMAUX 165
4.2.5.3 CONTRAINTES SUR LES ITEMS 165
4.2.5.4 EXTRACTION D'ITEMSETS FERMES FREQUENTS 165
4.2.6 ALGORITHMES CLOSE ET A-CLOSE 166
4.2.6.1 ALGORITHME CLOSE 167
4.2.6.2 ALGORITHME A-CLOSE 167
4.2.6.3 PROBLEME DE LA PERTINENCE ET DE L'UTILITE DES REGLES D'ASSOCIATION 172
4.2.7 BASES POUR LES REGLES D'ASSOCIATION 173
4.2.7.1 BASE GENERIQUE POUR LES REGLES D'ASSOCIATION EXACTES 174
4.2.7.2 BASE INFORMATIVE POUR LES REGLES D'ASSOCIATION APPROXIMATIVES 176
4.3 AUTRES MESURES STATISTIQUES 179
4.4 FOUILLE DE TEXTES ET FOUILLE DU WEB 182
4.4.1 PRINCIPE DE LA FOUILLE DE TEXTES 182
4.4.2 SIGNIFICATION DES REGLES D'ASSOCIATION 183
4.4.3 PROCESSUS D'EXTRACTION 184
4.4.4 SYSTEMES DE FOUILLE DE TEXTES 185
4.4.5 FOUILLE DU WEB 187
4.5 EVALUATION DES REGLES D'ASSOCIATION 188
4.6 EXPERIENCES D'EXTRACTION DE REGLES D'ASSOCIATION 189
4.6.1 FOUILLE DE DONNEES 189
4.6.1.1 MOTS CLES 191
4.6.1.2 MOTS CLES OU QUALIFICATIFS 191
4.6.1.3 ASSOCIATIONS MOT CLE/QUALIFICATIF 192
4.6.2 FOUILLE DE DONNEES AVEC CATEGORISATION 192
4.6.2.1 MOTS CLES 195
4.6.2.2 ASSOCIATION MOTS CLES/QUALIFICATIFS 195
4.6.3 FOUILLE DE TEXTES 196
4.6.4 FOUILLE DE DONNEES PONDEREES 197
4.6.4.1 MOTS CLES EN MAJEUR 197
4.6.4.2 ASSOCIATIONS MOTS CLES/QUALIFICATIFS EN MAJEUR 198

4.6.5 EVALUATION DES REGLES D'ASSOCIATION 200
4.7 EXPLOITATION POUR LA RECHERCHE D'INFORMATION 202
4.8 CREATION DE REGLES EXPERTES 204
4.9 QUELQUES PERSPECTIVES 206

CHAPITRE 5
REPRESENTATION DES CONNAISSANCES : DU MODELE AU FORMEL

5.1 INTRODUCTION 212
5.2 LES ONTOLOGIES POUR LA RECHERCHE D'INFORMATION 213
5.3 LES MODELES ET LANGAGES DE REPRESENTATION 215
5.3.1 LES RESEAUX SEMANTIQUES 215
5.3.2 LES GRAPHES CONCEPTUELS 216
5.3.3 LES LANGAGES DE FRAMES 218
5.3.4 LA REPRESENTATION DES CONNAISSANCES PAR OBJET 219
5.4 QUELQUES PROJETS DU WEB SEMANTIQUE 220
5.4.1 LE LANGAGE SHOE 221
5.4.2 ONTOSEEK 221
5.4.3 WEBKB 222
5.4.4 CWEB 223
5.4.5 OCML 224
5.5 LE CISMEF DANS L'INFRASTRUCTURE DU WEB SEMANTIQUE 226
5.5.1 REPRESENTATION DES METADONNEES 226
5.5.2 LA TERMINOLOGIE CISMEF : ENTRE TERMINOLOGIE ET ONTOLOGIE 227
5.6 LES LOGIQUES DE DESCRIPTION 229
5.6.1 LES FORMALISMES TERMINOLOGIQUES 230
5.6.2 ASSERTIONS ET REGLES D'INFERENCE 231
5.6.3 LE RAISONNEMENT TERMINOLOGIQUE 232
5.6.3.1 LA RECONNAISSANCE D'INDIVIDUS 232
5.6.3.2 LE RAISONNEMENT SUR LES DESCRIPTIONS 232
5.6.4 EXPRESSIVITE ET COMPLEXITE 233
5.6.4.1 LES PROCEDURES DE CALCUL COMPLETES 233
5.6.4.2 PROCEDURES DE CALCUL INCOMPLETES 233
5.6.4.3 COMPLETUDE ET EFFICACITE 234
5.6.5 LES CONSTRUCTEURS 234
5.6.5.1 LES CONSTRUCTEURS DE CONCEPTS 234
5.6.5.2 LES CONSTRUCTEURS DE ROLES 235
5.6.5.3 LA SYNTAXE DES CONSTRUCTEURS 235
5.6.5.4 LA SEMANTIQUE DES CONSTRUCTEURS 236
5.6.5.5 EXEMPLE 238
5.6.6 RAISONNEMENT TAXINOMIQUE POUR LA RECHERCHE D'INFORMATION 238
5.7 FORMALISATION DE LA TERMINOLOGIE 239
5.7.1 EXPERIENCES AVEC TRIPLE 239
5.7.2 DE OIL A OWL 241
5.7.3 TRAVAUX DANS LE DOMAINE DE LA SANTE 242
5.7.4 PRINCIPES DE MODELISATION 243
5.7.5 DES FICHIERS TEXTES A LA BASE DE DONNEES 244
5.7.6 DE LA BASE DE DONNEES A LA BASE DE CONNAISSANCES 244
5.7.6.1 LES CLASSES OWL 245
5.7.6.2 LES PROPRIETES OWL 246

5.7.6.3 La Propriete Part-Of 246
5.7.6.4 Les Restrictions sur les Domaines des Proprietes 247
5.7.6.5 Representation des Documents 247
5.7.7 Verification de la Consistance et Classification 248
5.7.7.1 Import sous Protege-2000 248
5.7.7.2 Verification de la Consistance 248
5.7.7.3 La Classification 250
5.7.7.4 Ameliorations Possibles 252
5.8 VERS UNE ONTOLOGIE ? LE PROJET ATONANT 254
5.8.1 Terminae 255
5.8.2 Resultats dans le Cadre du Projet ATONANT 255
CONCLUSION GENERALE 265

RESUME

La problématique de nos travaux de recherche se place dans le contexte de la recherche d'information textuelle sur le Web. Nous proposons en ce sens des méthodes de recherche d'information basées sur l'exploitation de connaissances. Nos expérimentations sont réalisées dans le cadre du projet CISMeF (Catalogue et Index de Sites Médicaux Francophones) qui indexe un grand nombre de documents en fonction d'une terminologie structurée du domaine médical. Nous avons développé le prototype KnowQuE (Knowledge-based Query Expansion) pour corriger, préciser et enrichir les requêtes des utilisateurs. Ses modules utilisent les traitements linguistiques, la fouille de données et les mécanismes de raisonnement associés aux logiques de description.

Dans notre application, les connaissances disponibles concernent le vocabulaire du domaine, les utilisateurs et les documents eux-mêmes. A ces connaissances nous avons ajouté, par acquisition et extraction, des connaissances linguistiques mais également « découvert » de nouvelles connaissances contenues dans les documents par un processus de fouille de données.

En effet, partant du constat que les requêtes des utilisateurs correspondent rarement à la formulation exacte effectivement utilisée pour l'indexation, et que de récents travaux ont montré la contribution du traitement morphologique des requêtes en langue française, le premier module du KnowQuE que nous avons développé est composé d'une base de connaissances morphologiques qui ont été acquises en fonction du vocabulaire de la terminologie. Les connaissances linguistiques sont donc de type morphologique.

Le second module exploite des règles d'associations entre termes, extraites à partir du corpus du CISMeF grâce à un algorithme de fouille de données fondé sur une analyse formelle de treillis de concepts. Ces règles d'association sont validées par notre expert.

Toutes ces connaissances ont été modélisées puis formalisées en utilisant un langage formel de représentation. En effet, la gestion et la représentation des connaissances sont à la base des systèmes intelligents en général et du Web Sémantique en particulier et nous souhaitons par cette démarche proposer une méthode qui exploite conjointement toutes les connaissances afin de donner les moyens à la recherche d'information de devenir 'intelligente'.

Le troisième module quant à lui, utilise le raisonnement terminologique. Il est composé d'une base de connaissance terminologique formalisée automatiquement en OWL-DL, nouveau langage standard du Web Sémantique. Les connaissances taxinomiques correspondent à la terminologie et les documents sont considérés comme des instances de concepts de la terminologie formelle. Les mécanismes de raisonnement sont exploités pour la classification automatique, la vérification de la consistance et le processus de recherche d'information.

Une première série d'évaluations concernent des projections automatiques des requêtes et elles sont quantitatives. Les évaluations des projections interactives (évaluation qualitative) avec les utilisateurs sont réalisées grâce à un échantillon d'utilisateurs abonnés au site qui mesurent l'utilité de chacune des trois approches. Nous pouvons conclure que les traitements basiques des requêtes permettent de corriger les requêtes des utilisateurs qui ne connaissent pas les spécificités du vocabulaire médical et évitent ainsi le silence du système. Les règles d'association sont utiles lorsqu'il y a beaucoup trop de réponses et permettent de préciser les requêtes. Enfin le raisonnement terminologique permet d'étendre les requêtes et contribue également à la construction du Web Sémantique (du moins pour le domaine médical).

MOTS CLES :

Recherche d'information ; recherche documentaire ; Internet ; traitement du langage naturel ; fouille de données ; fouille de textes ; représentation des connaissances ; raisonnement terminologique ; ontologies ; métadonnées ; Web sémantique ; domaine médical.

INTRODUCTION GENERALE

Avec le développement des nouvelles technologies de l'information et de la communication, de l'informatique et surtout de l'Internet, le volume d'information stockée électroniquement ainsi que la profusion d'informations accessibles à tous, sont en perpétuelle augmentation et n'ont de cesse de croître. Depuis les années 90, c'est le World Wide Web (également appelé Web ou Toile) qui connaît le plus gros essor au niveau mondial. En effet, ce service de l'Internet met à la disposition de tout Internaute tout type d'informations organisées sous la forme de pages (ou documents) contenant des liens vers d'autres pages et permettant le passage d'une page à une autre très facilement. Cependant, cette prolifération d'informations pose le problème de leur localisation pour leur exploitation par l'utilisateur, chaque document étant noyé dans un énorme fond documentaire (ou corpus) en constante évolution. La quantité d'information accessible est elle-même une richesse mais elle devient très vite un handicap pour l'utilisateur. En effet, il demeure difficile de retrouver de manière pertinente un ensemble d'informations contenu dans un document et notamment de savoir où retrouver l'information recherchée, à moins d'analyser chacun de ces documents. De nombreux outils de recherche ont été développés pour faciliter l'accès à l'information, mais même l'utilisation d'un moteur de recherche ne permet pas toujours de trouver ce dont on a besoin.

Le contexte de nos travaux s'insère dans cette problématique générale qu'est la Recherche d'Information textuelle sur le Web. Nous distinguons le média texte des autres médias que sont l'image, la vidéo et le son. Notre étude porte sur les moyens de récupérer un ensemble de documents pertinents répondant à un besoin d'utilisateur explicité sous la forme d'une requête. Nous considérons que l'utilisateur se charge de récupérer les informations au sein même des documents qui lui sont retournés, tâche qu'il réalise souvent lorsqu'il utilise les outils de recherche disponibles sur l'Internet, ou encore un Système de Recherche d'Information. Pour cela nous étudions et proposons d'appliquer différentes méthodes afin d'aider l'utilisateur dans sa démarche. Ces travaux[1] de thèse ont été effectués en collaboration avec le CHU de Rouen, notamment dans le cadre du projet CISMeF (Catalogue et Index des Sites Médicaux Francophones), et le laboratoire PSI (Perception, Systèmes, Information, CNRS FRE 2645) de l'INSA et de l'Université de Rouen. Ils ont pour origine un besoin du CISMeF de développer et d'améliorer son outil de recherche Doc'CISMeF *(Darmoni et al., 2001)* dont les performances et les résultats n'étaient alors pas satisfaisants.

[1] Travaux financés par une bourse de la Région Haute-Normandie.

Le CISMeF est un catalogue indexant un grand nombre de documents relatifs au domaine de la santé (*n=13 850*) en langue française, mais également un Système de Recherche d'Information dans le sens où son but est de retrouver des documents en réponse à une requête d'utilisateur, de manière à ce que les contenus des documents soient pertinents par rapport au besoin initial d'information de l'utilisateur *(Smeaton, 1989)*. Le CISMeF est un Système de Recherche d'Information puisqu'il traite également de la représentation, du stockage, de l'organisation et de l'accès aux éléments de l'information *(Salton & Mc Gill, 1983)*. Il peut être décomposé en plusieurs modules, d'une part par un module chargé du traitement, de l'indexation et du stockage de l'information : le module indexation qui construit une structure de données organisées de manière à permettre l'accès rapide à l'information. D'autre part, un module permet d'interagir avec les utilisateurs, doté des mécanismes de sélection d'information orientés par les requêtes des utilisateurs : le module interrogation où la recherche se fait par mot clé. Un module d'appariement établit une association entre la requête de l'utilisateur et les documents indexés. Enfin, un autre module permet de visualiser les données et naviguer au sein de celles-ci.

Initialement le sujet traité consistait à étudier et améliorer le module interrogation, dépendant en grande partie du module d'appariement, cela en étudiant des approches linguistiques de traitement des requêtes, notamment par l'application des variations terminologiques, sémantique en utilisant la synonymie, et statistique en utilisant les cooccurrences entre informations contenues dans les documents. Les problèmes auxquels doivent souvent répondre les modules d'interrogation des outils de recherche d'information sont d'ordre syntaxique (orthographe de la requête) et d'ordre sémantique (la polysémie, un terme pouvant avoir plusieurs sens, et la synonymie, plusieurs termes pouvant désigner un même concept). Cependant ces approches ne sont pas seulement relatives à l'interrogation puisque le traitement des documents pour obtenir les cooccurrences concerne le module indexation. D'autre part, il nous est très vite apparu nécessaire de traiter les autres modules du point de vue de leurs performances mais également du point de vue modélisation, gestion et représentation des connaissances disponibles dans le CISMeF. En effet, notre objectif est de rendre l'accès aux informations 'intelligent' dès lors que l'exploitation des connaissances est une technique développée à cet effet.

Nous avons donc pour cela modélisé les différentes connaissances à notre disposition. Elles sont de deux types : les connaissances concernant le domaine et les connaissances concernant l'utilisateur. Les connaissances du domaine sont relatives au domaine d'étude en fonction d'un point de vue partagé par une communauté d'utilisateurs. Dans notre contexte, le domaine considéré est le domaine médical, et lesdites connaissances constituent le langage utilisé pour l'indexation et la recherche d'information. Les connaissances relatives à l'utilisateur permettent de prendre en compte la différence qui peut exister entre les différents types d'utilisateurs, et ce, au-delà des différences du langage et des problèmes de multilinguisme qui en découlent. En effet, les utilisateurs n'ont pas tous les mêmes besoins en information et ils peuvent être classés en différentes catégories, une classification qui est fonction des connaissances que l'utilisateur a plus ou moins du domaine. Les outils de recherche disponibles sur le Web ne tiennent pas compte de ces différences entre utilisateurs. Pour la gestion et la représentation de ces connaissances, nous les avons incluses dans un système à bases de connaissances. De plus, afin d'améliorer le système du point de vue interrogation nous proposons d'inclure d'autres types de connaissances : des connaissances linguistiques acquises à partir de ressources et des connaissances extraites des documents eux-mêmes.

Nous souhaitons par cette démarche proposer une approche générale combinant l'exploitation de connaissances au système de recherche d'information. Dans cette optique, nous avons développé une méthode de recherche d'information fondée sur l'expansion des requêtes des utilisateurs par leur enrichissement à l'aide d'éléments de connaissances. Cette méthode permet à l'utilisateur de corriger, d'affiner et de préciser sa requête pour que le système réponde au mieux à ses besoins informationnels. Aux connaissances déjà disponibles, nous ajoutons des connaissances linguistiques acquises, relatives aux variations terminologiques du vocabulaire utilisé pour l'indexation et l'interrogation. Par ailleurs, de nouvelles connaissances relatives au domaine sont extraites des documents par un processus d'extraction des connaissances, en particulier par un système de fouille de données. Un expert des données du domaine est en charge d'évaluer et de valider ces connaissances acquises et extraites. Notre idée principale est que toutes ces connaissances sont complémentaires lorsqu'elles sont exploitées conjointement dans un système de recherche à base de connaissances.

Du point de vue architecture, le système que nous avons développé, KnowQuE (Knowledge-based Query Expansion), est composé de plusieurs modules de traitement des requêtes, chacun reposant sur une base de connaissances. Le premier module utilise des connaissances sur les variations terminologiques du vocabulaire du domaine. Nous sommes partis du constat que les connaissances morphologiques, et plus particulièrement la flexion (par exemple la forme plurielle d'un terme) et la dérivation (par exemple la forme adjectivale d'un terme), étaient utiles pour la recherche d'information *(Zweigenbaum et al., 2001)*. En effet, même si syntaxiquement elles sont différentes, ces formes sont relatives au même terme, et de ce fait au même concept. De plus, les requêtes des utilisateurs correspondent rarement à la formulation exacte utilisée pour l'indexation. Ce module intervient donc à un niveau basique de traitement de la requête, à savoir son aspect syntaxique. Par exemple, pour la requête *"enfants asthmatiques"* le module retourne des documents traitant d'un *"enfant ayant de l'asthme"*. Cependant, ces connaissances morphologiques, plus généralement les ressources lexicales, ne sont pas disponibles pour le vocabulaire français médical. Il a donc fallu les acquérir.

Le second module exploite des connaissances extraites à partir du contenu des ressources par un processus de fouille de données *(Agrawal & Srikant, 1994)*. Ces connaissances sont exprimées à l'aide de règles d'association entre concepts et sont extraites à partir du corpus indexé à l'aide d'un algorithme fondé sur une analyse formelle de treillis de concepts. Les règles d'association extraites et validées par notre expert sont exploitées en recherche d'information. L'idée principale est que ces règles d'association permettent de guider la recherche d'information à partir des connaissances issues des données. Par exemple, la règle d'association *"prévention du cancer du sein → mammographie"* est extraite car *"prévention du cancer du sein"* et *"mammographie"* sont utilisés fréquemment ensemble dans l'indexation des documents. En appliquant cette règle d'association pour la recherche d'information, une requête sur le terme *"mammographie"* permet de proposer à l'utilisateur des documents traitant de la *"prévention du cancer du sein"*.

Le troisième module utilise le raisonnement sur les connaissances formelles. Parmi les avantages d'une telle approche figurent un langage pour exprimer le contenu des documents, une sémantique associée à ce langage et des moteurs d'inférences associés qui s'appuient sur cette sémantique pour raisonner. Le raisonnement peut être utilisé pour la recherche d'information. Ce troisième module est composé d'une base de connaissances terminologiques représentées dans la logique OWL-DL (Ontology Web Language- Description Logics) *(Horrocks*

et al., 2003) nouveau standard du Web Sémantique *(Berners-Lee et al., 2001)* depuis Février 2004. Le Web Sémantique a pour but de proposer une nouvelle forme de contenu documentaire manipulable par une machine qui spécifie des connaissances associées au document dans un langage formel de représentation des connaissances, condition nécessaire pour permettre à la machine de les reconnaître et ainsi de faciliter la tâche de recherche d'information intelligente. L'indexation des documents s'avère de ce fait conceptuelle. Dans notre cas, les connaissances taxinomiques correspondent aux connaissances du domaine (la terminologie). Elles sont organisées de manière structurée selon une hiérarchie de concepts sur lesquels il est possible de faire des calculs de spécialisation ou de généralisation notamment avec le moteur d'inférence Racer *(Haarslev & Möller, 2001)*. Nous considérons les documents comme étant des instances de concepts de la terminologie formelle. La particularité réside ici dans la grande taille de notre base de connaissances terminologiques, du fait de son nombre de concepts mais également de son nombre d'instances. Par ailleurs elle doit régulièrement être mise à jour en fonction de l'évolution du vocabulaire et des documents. Pour la recherche d'information, la requête *"hépatite"*, par exemple, permet de récupérer des documents plus spécifiques indexés à *"hépatite virale A"* grâce au lien de subsomption qui existe entre *"hépatite"* et *"hépatite virale A"*.

Concernant l'évaluation du système KnowQuE *(Soualmia et al., 2003)*, nous l'avons choisie de deux types : quantitative et qualitative. L'évaluation quantitative se fait essentiellement par une expansion automatique des requêtes sans l'intervention de l'utilisateur afin de mesurer le silence du système. Néanmoins la précision des réponses est évaluée par notre expert. Mais au final, comme c'est l'utilisateur qui doit être satisfait des résultats, l'évaluation de notre prototype n'a pu se faire sans sa participation. Une expansion interactive des requêtes a donc été développée, ainsi qu'un serveur d'évaluation des projections interactives.

Nos travaux de recherche concernent ainsi plusieurs aspects de la gestion des connaissances, à savoir leur modélisation, leur acquisition, leur représentation, cela en vue de leur exploitation dans le cadre de la recherche d'information. La gestion et la représentation des connaissances sont à la base des recherches actuelles sur les systèmes intelligents en général et sur le Web Sémantique en particulier.

BIBLIOGRAPHIE

(Agrawal & Srikant, 1994) AGRAWAL R. & SRIKANT R.(1994) Fast Algorithms for Mining Association Rules in Large Databases. In *Proceedings of the Very Large DataBases Conference 1994*, pp.478-499.

(Berners-Lee, 2001) BERNERS-LEE T. (2001). The Semantic Web. *Scientific American*, 284(5):34-43.

(Darmoni et al., 2001) DARMONI SJ., THIRION B., LEROY JP. et al. (2001). A Search Tool Based on 'Encapsulated' MeSH Thesaurus to Retrieve Quality Health Resources on the Internet. *Medical Informatics & the Internet in Medicine*, 26(3):165-178.

(Haarslev & Möller, 2001) HAARSLEV V. & MÖLLER R. (2001) Description of the RACER System and its Applications. In *International Workshop on Description Logics 2001*.

(Horrocks et al., 2003) HORROCKS I., PATEL-SCHNEIDER PF., VAN HARMELEN F. (2003) From SHIQ and RDF to OWL: the Making of a Web Ontology Language. *Journal of Web Semantics*, 1(1): 7–26.

(Salton & Mc Gill, 1983) SALTON G. & MC GILL MJ. (1983) Introduction to Modern Information Retrieval. *McGraw Hill International Book Company*.

(Smeaton, 1989) SMEATON AF. (1989) Information Retrieval and Natural Language Processing. In P*roceedings of a Conference Jointly sponsored by ASLIB,* p.2.

(Soualmia et al., 2003) SOUALMIA LF., BARRY C., DARMONI SJ. (2003). Knowledge-Based Query Expansion over a Medical Terminology Oriented Ontology. Dojat, Keravnou, Barahona (Eds.), *Lecture Notes in Artificial Intelligence # 2780*, Springer-Verlag, pp.209–213.

(Zweigenbaum et al., 2001) ZWEIGENBAUM P., DARMONI SJ., GRABAR N. (2001) The Contribution of Morphological Knowledge to French MeSH Mapping for Information Retrieval. *Journal of the American Medical Informatics Association*, 8:796–800.

ORGANISATION DU MEMOIRE

Le Chapitre 1 introduit la problématique de la recherche d'information textuelle sur le Web. Les fondements ainsi que les caractéristiques du Web sont décrits afin de présenter le contexte de nos travaux. Les notions de rappel, de précision et de pertinence, ainsi que les difficultés liées au traitement des requêtes y sont évoquées.

Le Chapitre 2 est consacré à la présentation du catalogue CISMeF dans lequel nous réalisons nos expérimentations. Il a été développé pour remédier au problème 'appliqué' de la recherche d'information en santé. Nous modélisons les données que nous avons à notre disposition dans le CISMeF, principalement une terminologie structurée, un ensemble de métadonnées et un ensemble de documents. Cette modélisation prend en compte les spécificités du vocabulaire d'indexation, le type d'utilisateur amené à interroger le système, mais également les caractéristiques des documents indexés qui sont spécifiques aux documents en santé.

Dans le Chapitre 3, nous développons une analyse des apports du traitement linguistique des requêtes, en décrivant plusieurs méthodes d'amélioration de la recherche d'information existantes. Nos recherches se sont axées sur les variations terminologiques qui sont assimilables des connaissances linguistiques. Le problème de l'indisponibilité des lexiques est abordé et les différents moyens pour leur construction automatique sont présentés. Nous proposons également des algorithmes qui interviennent au niveau syntaxique de la requête. Des possibilités de phonémisation y sont évoquées pour la correction orthographique. Enfin plusieurs séries d'expériences permettent d'évaluer les algorithmes présentés. Une correction automatique des requêtes est mise en avant.

Le Chapitre 4 traite de la découverte de nouvelles connaissances par la fouille de données (ou data mining) et la fouille de textes (ou text mining). Nous présentons l'algorithme d'extraction de règles d'associations que nous avons implémenté. Nous montrons également comment nous exploitons les résultats de cette fouille de données, à savoir l'ensemble de règles d'associations obtenu, pour notre problématique de recherche d'information. Dans ce type de recherche par l'exploitation de connaissances, nous privilégions une recherche d'information interactive avec l'utilisateur.

Dans le Chapitre 5 nous abordons les notions d'ontologie et de Web Sémantique. Quelques systèmes de recherche d'information utilisant une approche ontologique sont détaillés. Pour représenter les connaissances que nous avons modélisées dans le Chapitre 2, notre choix s'est orienté vers la logique OWL-DL devenue standard du W3C. Nous proposons également un algorithme de traduction automatique de nos connaissances dans la syntaxe de ce langage. Le but n'est pas d'obtenir une ontologie, travail de longue haleine, mais plutôt une base de connaissances formalisée permettant de réaliser des inférences pendant le processus de recherche d'information.

Nous concluons en dressant le bilan de l'utilisation des trois approches de recherche d'information décrites dans les chapitres 3, 4 et 5. L'idée principale est que l'utilisation conjointe des traitements linguistiques, de la fouille de données et du raisonnement terminologique, malgré le fait que ce sont des approches issues de communautés différentes (Traitement du Langage, Bases de Données et Ingénierie des Connaissances), permet de répondre aux problèmes de la recherche d'information. Nous proposons enfin quelques perspectives de recherche qui nous semblent intéressantes à explorer.

CHAPITRE 1

RECHERCHE D'INFORMATION :

DE LA REQUETE AUX DOCUMENTS

Sommaire

1.1 INTRODUCTION 27
1.2 INTERNET ET LE WEB 28
1.2.1 HISTORIQUE D'INTERNET 28
1.2.2 CARACTERISTIQUES DU WEB 29
1.2.3 LIMITES DES INFORMATIONS DU WEB 30
1.3 FONDEMENTS DE LA RECHERCHE D'INFORMATION 31
1.3.1 LA RECHERCHE DOCUMENTAIRE 31
1.3.2 LES SYSTEMES DE RECHERCHE D'INFORMATION 32
1.3.3 LA TACHE DE NAVIGATION 33
1.3.4 LA TACHE DE RECHERCHE 34
1.3.5 L'INDEXATION 34
1.3.5.1 L'ESPACE D'INDEXATION 36
1.3.5.2 LES ENTITES D'INDEXATION 36
1.3.5.3 LES LANGAGES D'INDEXATION 37
1.3.5.3.1 Le Langage Libre *37*
1.3.5.3.2 Le Langage Contrôlé *38*
1.3.5.4 LES TYPES D'INDEXATION 39
1.3.5.4.1 Indexation Manuelle *39*
1.3.5.4.2 Indexation Automatique *39*
1.3.5.4.3 Indexation Semi-Automatique *40*
1.3.5.5 CAS DE L'INDEXATION AUTOMATIQUE 40
1.3.5.5.1 Extraction des Termes d'Indexation 40
1.3.5.5.2 Réduction du Langage d'Indexation 43
1.3.5.5.3 Pondération des Termes d'Indexation 44
1.4 MODELES DE RECHERCHE D'INFORMATION ET DE REPRESENTATION 46
1.4.1 LE MODELE BOOLEEN 46
1.4.2 LE MODELE VECTORIEL 47
1.4.3 LE MODELE PROBABILISTE 49
1.4.4 LE MODELE LOGIQUE 50
1.5 ENVIRONNEMENTS DE RECHERCHE 50
1.6 EVALUATION DE LA RECHERCHE D'INFORMATION 51
1.7 LA RECHERCHE D'INFORMATION SUR LE WEB 53

1.7.1 CARACTERISTIQUES DE LA RECHERCHE D'INFORMATION SUR LE WEB 53
1.7.2 LES PROBLEMES DE LA RECHERCHE D'INFORMATION SUR LE WEB 54
1.7.3 PROBLEMES LIES AU PROCESSUS DE RECHERCHE SUR LE WEB 55
1.7.4 APPROCHES EXISTANTES POUR L'AIDE A LA RECHERCHE D'INFORMATION SUR LE WEB 56
1.7.4.1 LES FACTEURS HUMAINS 56
1.7.4.2 LE PROCESSUS DE RECHERCHE 56
1.7.4.2.1 La Tâche de Navigation 57
1.7.4.2.2 La Tâche de Recherche 57
1.7.4.2.3 Les Méta-Moteurs de Recherche d'Information 60
1.7.4.3 LA VISUALISATION DES RESULTATS 61
1.7.4.4 LES AGENTS 62
1.7.4.4.1 Les Agents de Recherche 62
1.7.4.4.2 Les Agents de Recommandation 62
1.7.4.4.3 Approches Multi-Agents 63
1.9 CONCLUSION 64

Le but de ce Chapitre 1 est de faire une présentation d'Internet et plus particulièrement du Web. Nous présentons les concepts de la Recherche d'Information pour ensuite introduire les particularités et problèmes de la Recherche d'Information textuelle sur le Web. Nous détaillons enfin les différentes solutions qui ont été proposées dans la littérature afin de remédier à ces problèmes, solutions principalement fondées sur le traitement de la langue.

1.1 Introduction

Avec le développement des nouvelles technologies de l'information et de la communication, de l'informatique et surtout de l'Internet, le volume d'information stockée électroniquement ainsi que la profusion d'informations accessibles à tous sont en perpétuelle augmentation et n'ont de cesse de croître. Depuis les années 90, c'est le World Wide Web (également appelé Web ou Toile) qui connaît le plus gros essor au niveau mondial. Le Web est devenu la source d'informations privilégiée pour quiconque recherche des informations en relation avec ses besoins. En effet, ce service de l'Internet met à la disposition de tout Internaute tout type d'informations organisées sous la forme de pages (que nous appellerons 'documents' par abus de langage) contenant des liens vers d'autres pages et permettant le passage d'une page à une autre très facilement. Cependant, cette prolifération d'informations pose le problème de leur localisation pour leur exploitation par l'utilisateur, chaque document étant noyé dans un énorme fond documentaire (également appelé corpus) en constante évolution. La quantité d'information accessible est elle-même une richesse mais elle devient très vite un handicap pour l'utilisateur. En effet, il demeure difficile de retrouver de manière pertinente un ensemble d'informations contenu dans un document et notamment de savoir où retrouver l'information recherchée, à moins d'analyser chacun de ces documents.

Les Systèmes de Recherche d'Information (SRI) sont conçus à l'origine pour répondre aux besoins d'automatiser la gestion de la documentation. Du fait de leur grand nombre, la localisation des informations pertinentes est un problème. Avec l'avènement d'Internet, le volume des documents et le nombre de personnes à gérer se sont accrus de manière importante: le nombre de pages Web accessibles a augmenté de 320 millions en 1997 à plus de 4,3 milliards en 2004[2]. Le nombre d'utilisateurs est aujourd'hui évalué à des centaines de millions. De nombreux outils de recherche ont été développés pour faciliter l'accès à l'information sur ce média, mais même l'utilisation d'un moteur de recherche ne permet pas toujours de trouver ce dont on a besoin. Les réponses sont souvent non pertinentes par rapport aux attentes et, en suivant des liens à partir d'un document pertinent, l'utilisateur se rend compte qu'il en existe d'autres qui n'étaient pas signalés ou qui étaient signalés après plusieurs centaines d'autres documents non pertinents. L'utilisateur a du mal à se repérer, à identifier les documents intéressants au sein de cette masse informationnelle qui évolue sans cesse. L'utilisateur navigue d'un lien à l'autre et la conséquence directe est que la recherche est abandonnée (loi de Mooers *(Mooers, 1960)*).

[2] Google, 2004

Le problème s'agit donc de disposer de systèmes de recherche les plus performants possibles afin de satisfaire au mieux les attentes de l'utilisateur. Les SRI sont aujourd'hui confrontés à un nouveau défi dû à la disparité et à la quantité des types de documents à gérer autant qu'à la multiplicité des demandes des utilisateurs et les problèmes de RI et d'indexation restent d'actualité. Outre le problème d'identifier l'information contenue dans un document, la Recherche d'Information (RI) *(Mooers, 1950)* doit également permettre à l'utilisateur de formuler sa demande, son besoin d'information, le plus exactement possible, sous la forme d'une requête. Plusieurs recherches sont en cours de développement dans ce domaine, mais les problèmes d'indexation automatique et de recherche d'information sont encore très actuels. Nous présentons ci-après l'Internet et le Web ainsi que les fondements des SRI.

1.2 Internet et le Web

Cette section présente un historique de l'Internet et du Web. Elle met également en évidence les limites et les problèmes liés à ce média concernant la RI.

1.2.1 Historique d'Internet

L'histoire d'Internet a débuté en pleine guerre froide. En 1957, les Soviétiques ont lancé leur satellite Spoutnik et les Américains ont redouté une guerre nucléaire. Le ministère de la défense américain créa alors une agence pour la recherche nommée ARPA. Son but visait à développer un réseau de communication militaire pouvant fonctionner même avec une partie hors service. Une première application du réseau ARPANET est le courrier électronique permettant aux militaires de communiquer. Sa première expérimentation a eu lieu en 1969 aux Etats-Unis. Ce réseau fut rebaptisé Internet (« Inter Networking ») en 1980.

Dès lors, l'Internet a connu une perpétuelle évolution en particulier au travers de l'augmentation du nombre de machines connectées. En revanche, Internet n'était, à ses débuts, destiné qu'à peu d'universitaires qui connaissaient son langage. C'est en Europe qu'a été simplifié le langage d'Internet, avec la notion d'hypertexte, et développé le premier navigateur permettant de visualiser les différents documents disponibles. Basé sur cette approche, le premier navigateur « grand public » nommé Mosaic fut développé en 1993. Grâce à ce navigateur, chaque utilisateur connecté pouvait accéder et parcourir simplement les documents disponibles sur Internet. Cette fonctionnalité est toujours disponible aujourd'hui grâce aux navigateurs actuels (Microsoft Internet Explorer, Netscape…). Cet attrait d'Internet peut être expliqué par le fait qu'il permet de partager instantanément des informations entre toutes les machines connectées.

Au début des années 1990, Internet a connu un véritable essor du fait de l'avènement du service Web. Ce service a permis de simplifier la mise en oeuvre de services multimédia incitant les particuliers ainsi que les entreprises à diffuser leurs informations. Cet essor peut s'expliquer par le fait que le Web repose sur des notions peu complexes et qu'il permet d'interagir avec les autres protocoles disponibles sur Internet. Le Web peut être vu comme une « interface » entre les internautes et les différents services d'Internet tout en proposant un outil facile et puissant d'utilisation pour le parcours des différents documents. Dans le cas du Web, la notion de

documents revêt un caractère particulier et on parle plus couramment de pages que de documents. Nous donnons ci-après la définition de la notion de document :

Définition 1.1 (DOCUMENT) (Définition ISO).

> *Un document est l'ensemble d'un support d'information et des données enregistrées sur celui-ci sous la forme en général permanente et lisible par l'homme et la machine.*

La technologie sur laquelle repose le Web a été développée au CERN (Centre Européen pour la Recherche Nucléaire) en 1989 par Tim Berners-Lee *(Berners-Lee et al., 1994)*. L'objectif était la diffusion d'informations scientifiques entre les chercheurs. L'idée sur laquelle repose le Web était d'organiser les informations sous forme de documents avec possibilités d'insérer des liens vers d'autres documents autorisant le passage d'un document à un autre sans peine.

La norme HTML, développée par le World Wide Web Consortium (W3C[3]), permet à l'utilisateur de décrire les documents qu'il souhaite mettre en ligne sous forme textuelle. Ce langage hypertexte permet également à l'utilisateur d'insérer des liens (ancres) vers tout autre document associé à une URL. Un document sous cette forme en HTML est communément appelé une page web. Par extension, un site Web correspond à une arborescence de pages Web ayant pour racine une page, dite d'accueil, et se trouvant sur un même serveur. Ce standard est aujourd'hui remis en cause par des langages où la distinction entre contenu et présentation est beaucoup plus nette comme XML (eXtensible Markup Langage) *(Bray & Sperberg-Mc Queen, 1996)*. Ce langage de balisage permet de créer des documents en distinguant la structure logique (pour le contenu sémantique) de la structure physique (pour la présentation des données).

1.2.2 Caractéristiques du Web

Les informations disponibles sur le Web peuvent être scindées en deux catégories par rapport aux modes d'accès possibles : le Web caché et le Web visible. Le Web caché *(Bergman, 2000)* correspond à l'ensemble des documents accessibles par l'intermédiaire d'un serveur « dédié » comme un serveur de base de données. Le seul moyen d'y accéder est d'interroger le serveur grâce à une requête adéquate ou à un formulaire. Le Web visible correspond à l'ensemble des documents directement accessibles sans avoir besoin de formuler une quelconque requête ou de remplir un quelconque formulaire.

La grande différence entre ces deux modes d'accès réside dans le fait que les informations accessibles par le Web caché sont plus nombreuses. De plus, les informations que contient le Web caché sont plus « contrôlées » que celles du Web visible. Le Web visible est constitué de plus de 4 milliards de documents Web et est en plein essor avec une évolution approximative de plus de 7 millions de documents par jour *(Murray & Moore, 2000)*.

[3] http://www.w3c.org

1.2.3 Limites des Informations du Web

L'utilisateur a donc facilement accès à un nombre important de documents contenant des informations aussi diverses qu'abondantes. Cependant, outre le volume important d'information disponible, le Web a des limites qui lui sont inhérentes. Ces limites sont *(Baeza-Yates & Ribeiro-Neto, 1999)* :

– la non-persistance de l'information : le Web possède une dynamique très importante et l'information naît, évolue et disparaît rapidement. Un document visité à un moment t ne sera pas forcément le même que celui consulté au moment $(t+\Delta)$. Il a d'ailleurs été estimé que 40% des informations disponibles sur le Web changent tous les mois *(Kahle, 1996)*.

– l'instabilité de l'information : le Web repose sur une architecture informatique qui peut connaître diverses pannes ou dysfonctionnements. De ce fait, l'information n'est pas accessible de façon permanente et il se peut qu'à tout moment celle-ci ne soit plus accessible.

– le manque de qualité de l'information : le Web est un média ouvert, dans le sens où il n'y a pas d'organisme contrôlant les contenus disponibles. De ce fait, les informations disponibles sont souvent sujettes à des problèmes de véracité, de fautes de langages ou erreurs typographiques. De plus, tout un chacun peut créer sa page Web et y insérer les informations qu'il souhaite.

– la redondance d'information : une expérimentation *(Shivakumar & Garcia-Molina, 1998)* réalisée à partir d'une collection de 24 millions de pages Web montre que plus de 30% de l'information est redondante. Cette proportion peut être encore plus importante si l'on considère une redondance sémantique ou partielle des informations.

– l'hétérogénéité de l'information : sur le Web cohabitent des informations dans des médias différents (image, son…), des formats différents (jpeg, mp3…) et des langues différentes (français, chinois…).

– le volume d'information disponible.

Ce dernier point qu'est le volume d'information implique que la couverture du Web par les outils de recherche reste assez faible. La plupart de ces problèmes sont difficilement gérables de façon automatique (stabilité, hétérogénéité de l'information). Certains d'entre eux sont relatifs à la nature humaine (contenu inexact ou mal formé des documents par exemple). Du point de vue de l'internaute, le problème principal du Web vient de son architecture. En effet, il n'existe aucune organisation spécifique des informations, aucun index général référençant les informations existantes. Les informations peuvent être situées n'importe où, voire dupliquées, d'où le problème de la localisation de l'information. Ce problème est d'autant plus important que le nombre de documents disponibles est grand. Cependant, ce problème n'est pas récent, il était déjà d'actualité dès les débuts d'Internet avec les premiers outils de recherche tels que 'Gopher', mais il ne fait que s'accentuer avec le temps.

Nous détaillons dans la section suivante les fondements de la recherche d'information.

1.3 Fondements de la Recherche d'Information

La RI traite de la représentation, du stockage, de l'organisation ainsi que de l'accès à l'information. Un SRI est un ensemble de modèles et de processus permettant la sélection d'informations pertinentes dans une ou plusieurs collections en réponse aux besoins d'un utilisateur. Depuis toujours, la recherche documentaire est subordonnée à la RI. Dans la majorité des cas, un utilisateur recherche une information plutôt qu'un document, mais il accepte qu'un système lui renvoie une liste de documents dans lesquels il est supposé trouver l'information dont il a besoin.

1.3.1 La Recherche Documentaire

(de Loupy, 2000) et *(Fayet-Scribe, 1997)* dressent un historique des méthodes de classement et de recherche documentaire de l'Antiquité à aujourd'hui. La recherche documentaire vise à retrouver des documents textuels répondant à un besoin informationnel spécifié par une requête. La RI cherche des documents répondant à un besoin informationnel ou sujet formulé par une requête. Les documents sont au préalable indexés : chaque mot de chaque document est répertorié dans une table inverse avec ou sans conservation des mots dans le texte d'origine. L'appariement entre la requête et l'index va déterminer les documents qui sont considérés comme répondant le mieux au besoin informationnel initial.

La recherche documentaire se compose de deux processus de base :

– L'indexation qui est un processus de représentation du contenu des textes (les textes étant à la fois les documents et les requêtes),

– La comparaison entre les représentations des textes issues de l'indexation.

Le but de l'indexation est de représenter les documents et les requêtes dans le même espace de représentation à l'aide d'une structure de données. Cependant, les documents et les requêtes peuvent avoir des caractéristiques bien différentes. Par exemple, une requête peut être constituée de deux mots reliés par un opérateur booléen tandis qu'un document peut être un article de vingt pages, paru dans une revue scientifique. Lorsque la différence structurelle entre les documents et les requêtes est trop importante, le processus de représentation des textes est décomposé en deux processus distincts : une fonction d'indexation traite des requêtes formulées dans un langage d'interrogation et une fonction d'indexation traite les documents.

En recherche documentaire, la tâche consiste à retrouver les documents qui correspondent à une requête, ce qui revient à classer tout le corpus en deux classes : les textes correspondant à la requête d'une part, les autres d'autre part. Le principe de toute recherche documentaire repose sur l'appariement d'une question (requête) avec des documents ou des informations contenues dans une base *(Lefèvre, 2000)*. *(Lewis, 1992a)* résume les étapes de la recherche documentaire comme suit :

1. L'indexation de textes : l'opération qui permet de représenter le texte afin qu'il soit exploitable par le système de recherche documentaire.

2. La formulation d'une question, sous diverses formes de requêtes :

– un thème ou un descripteur.

– une requête, construite avec des mots du langage courant, et utilisant des opérateurs booléens, de proximité, de troncature.

– une expression en langage naturel.

– un document entier, utilisé comme exemple du sujet sur lequel on veut obtenir d'autres informations.

– un graphe de concepts. Les concepts, représentés par des termes, peuvent être liés par différentes relations sémantiques.

3. La comparaison entre les requêtes et les documents. La comparaison se fait généralement en utilisant une fonction de similarité.

4. Le *Feedback* : les résultats fournis par le système correspondent rarement aux besoins exacts de l'utilisateur. L'utilisateur doit donc revoir la requête et la reformuler. Si le système de recherche modifie ou reformule la requête on parle alors du *relevance feedback* (bouclage de pertinence).

Le résultat est souvent imparfait à cause de l'ambiguïté et de la redondance de la langue naturelle. L'ambiguïté se produit car un mot peut posséder plusieurs sens selon le contexte, et la redondance car un même concept peut être exprimé par différents mots.

1.3.2 Les Systèmes de Recherche d'Information

Il y a plusieurs définitions d'un SRI, qui sont plus ou moins proches.

Définition 1.2 (SYSTEME DE RECHERCHE D'INFORMATION)*(Strzalkowski, 1993)*

> *La tâche typique de la recherche d'information est de sélectionner des documents dans une base de données, en réponse à une requête de l'usager, et leur rangement par ordre de pertinence.*

Définition 1.3 (SYSTEME DE RECHERCHE D'INFORMATION)*(Smeaton, 1989)*

> *L'objectif d'un système de recherche d'information est de trouver des documents en réponse à une requête d'utilisateur tels que le contenu des documents soit pertinent par rapport au besoin initial de l'utilisateur.*

La RI intègre deux tâches bien spécifiques : la navigation et la recherche. La navigation correspond à l'action de trouver des informations pertinentes au travers d'une base de documents sans connaître, à priori, le contenu et le format des documents contenus dans la base. La recherche correspond à l'action de rechercher des informations au travers d'une base de documents à partir des besoins exprimés sous la forme d'une requête. Dans cette section, nous détaillons ces deux tâches qui mettent en jeu des processus différents.

1.3.3 La Tâche de Navigation

La tâche de navigation permet à l'utilisateur de parcourir l'espace des documents de la collection sans devoir formuler ses besoins. Le principal intérêt de cette tâche est qu'elle permet à l'utilisateur d'acquérir des informations sans nécessairement avoir à connaître, a priori, le contenu, la structure des informations qu'il va rencontrer. Trois modèles ont été définis pour caractériser une navigation :

– Le modèle plat : les documents sont présentés dans un plan ou une liste simple.

– Le modèle structuré : par analogie à un système de fichiers, les documents sont organisés sous la forme d'une arborescence. Ce modèle permet de proposer à l'utilisateur les documents en fonction des thèmes qu'ils abordent.

– Le modèle hypertexte : ce modèle est basé sur la notion d'hypertexte qui étend la notion de fichier texte linéaire (ou séquentiel) en permettant une structuration en graphe *(Julien, 1988)*.

Le modèle hypertexte a été développé pour permettre une consultation non linéaire des documents. Les noeuds peuvent contenir du texte mais également des images (fixes ou animées) et du son. Un lien hypertexte est un lien référentiel établissant des relations non hiérarchiques de sémantique très diverses entre les noeuds. Ils sont généralement orientés et caractérisés par un noeud d'origine et un noeud de destination. L'utilisateur peut ainsi, au travers des liens hypertextes, atteindre une portion du document voire des portions d'autres documents. Les liens sont soit insérés par l'auteur des documents afin de rapprocher les documents traitant du même thème par exemple, ou automatiquement par un processus de classification *(Agosti & Melucci, 2000)*.

Dans le contexte du Web, ce sont les deux premiers modèles sont fréquemment utilisés tout en servant de base à une navigation hypertexte, la navigation la plus courante sur le Web. Un hypertexte local est un sous-ensemble de l'hypertexte centré sur le document visité. L'hypertexte est composé de l'ensemble des documents Web et des liens entre eux. La notion de distance correspond au nombre de liens qu'il existe, par transitivité, entre deux noeuds. La taille de l'hypertexte local dépend de la distance maximale prise en compte par rapport au document considéré. L'hypertexte local évolue donc au fur et à mesure de la navigation de l'utilisateur. Malgré la facilité d'utilisation, il existe deux limites de la navigation hypertexte *(Agosti, 1996)*, *(Baeza-Yates & Ribeiro-Neto, 1999)* :

– La désorientation : l'utilisateur ne sait plus trop quel chemin suivre et il est «perdu dans l'hypertexte». Pour pallier cet inconvénient, il existe des mécanismes de retour en arrière.

– La surcharge cognitive : l'utilisateur réalise un important effort cognitif pour construire une carte mentale de l'hypertexte reflétant l'organisation de l'hypertexte local. Il se produit une surcharge cognitive lorsque l'utilisateur n'arrive plus à mémoriser la structure de l'hypertexte dans lequel il se trouve. Ainsi, de la conception de l'hypertexte (simplicité, organisation…) dépend le bon déroulement de la navigation.

1.3.4 La Tâche de Recherche

La recherche vise à proposer à l'utilisateur des documents en adéquation avec ses besoins appelés requêtes. Pour mesurer cet appariement, le SRI s'appuie sur une représentation commune des besoins de l'utilisateur et du contenu des documents textuels. Ces représentations reposent sur la caractérisation du contenu sémantique des documents et des besoins de l'utilisateur. Ces représentations sont ensuite utilisées au travers d'un modèle de RI permettant de mesurer leur appariement. Afin de construire des représentations comparables entre le contenu sémantique des documents et des requêtes, le SRI applique une phase d'indexation.

1.3.5 L'Indexation

La phase d'indexation analyse le contenu textuel des unités documentaires en vue de construire un ensemble de termes d'indexation (termes significatifs). Ces termes d'indexation représentent le contenu sémantique de l'unité documentaire. L'ensemble de ces termes est appelé langage d'indexation.

Définition 1.4 (INDEXATION)

> *L'indexation est l'identification de l'information contenue dans tout texte et sa représentation au moyen d'un ensemble d'entités appelé index pour faciliter la comparaison entre la représentation d'un document et d'une requête.*

Le processus d'indexation consiste à transférer l'information contenue dans le texte vers un autre espace de représentation exploitable par un système informatique. De manière générale, l'indexation peut être considérée comme un processus de représentation des textes. En effet, certains SRI acceptent comme requête un document entier. Dans certains cas, le meilleur document retrouvé par une première requête est envoyé comme requête au SRI. Cette méthode de modification de la requête par des documents préalablement jugés pertinents, fait partie de l'approche de bouclage de pertinence *(relevance feedback) (Salton & Buckley, 1990).* On peut considérer l'indexation comme le processus de traitement des textes qu'ils soient documents ou requêtes.

L'espace d'indexation, ou espace de représentation de l'information, doit être défini en choisissant les entités d'indexation. Elles correspondent à l'unité de base de l'espace d'indexation. La « structure » assemblant les entités d'indexation permet de construire un index.

Définition 1.4 (CONSTITUTION D'UN INDEX)*(Fluhr, 1992)*

> *Les documents sont lus par un documentaliste qui en déduit les thèmes principaux et les traduit en une liste de mots, dit descripteurs du document. Cet ensemble de mots constitue l'index du document et représente la description du contenu sémantique de celui-ci.*

La définition des techniques d'indexation permettent, à partir du texte, de détecter les entités et de construire les structures d'indexation. Le processus de comparaison permet de choisir les documents répondant au besoin d'information de l'utilisateur, en comparant la base des index du corpus à la représentation de la requête dans le même espace grâce à une fonction de comparaison. Cette phase vise à extraire des caractéristiques sur le contenu sémantique des informations textuelles brutes. Pour cela, le SRI traite les informations textuelles sous la forme d'unités documentaires. Une unité documentaire est le plus petit granule d'information correspondant à un niveau de finesse d'étude du contenu informationnel. Ce granule peut correspondre au contenu intégral d'un document ou encore à un paragraphe voire à une phrase du document. A partir de ces unités documentaires, une phase indexation est réalisée afin d'extraire les caractéristiques de leur contenu informationnel. L'indexation est une phase très importante dans le processus de recherche car de sa qualité dépendra la qualité des réponses lors de l'utilisation du SRI et donc de ses performances. En effet, en terme de performances, l'utilisateur souhaite obtenir toutes les informations répondant à ses besoins (unités documentaires pertinentes) et aucune n'y répondant pas (unités documentaires non pertinentes).

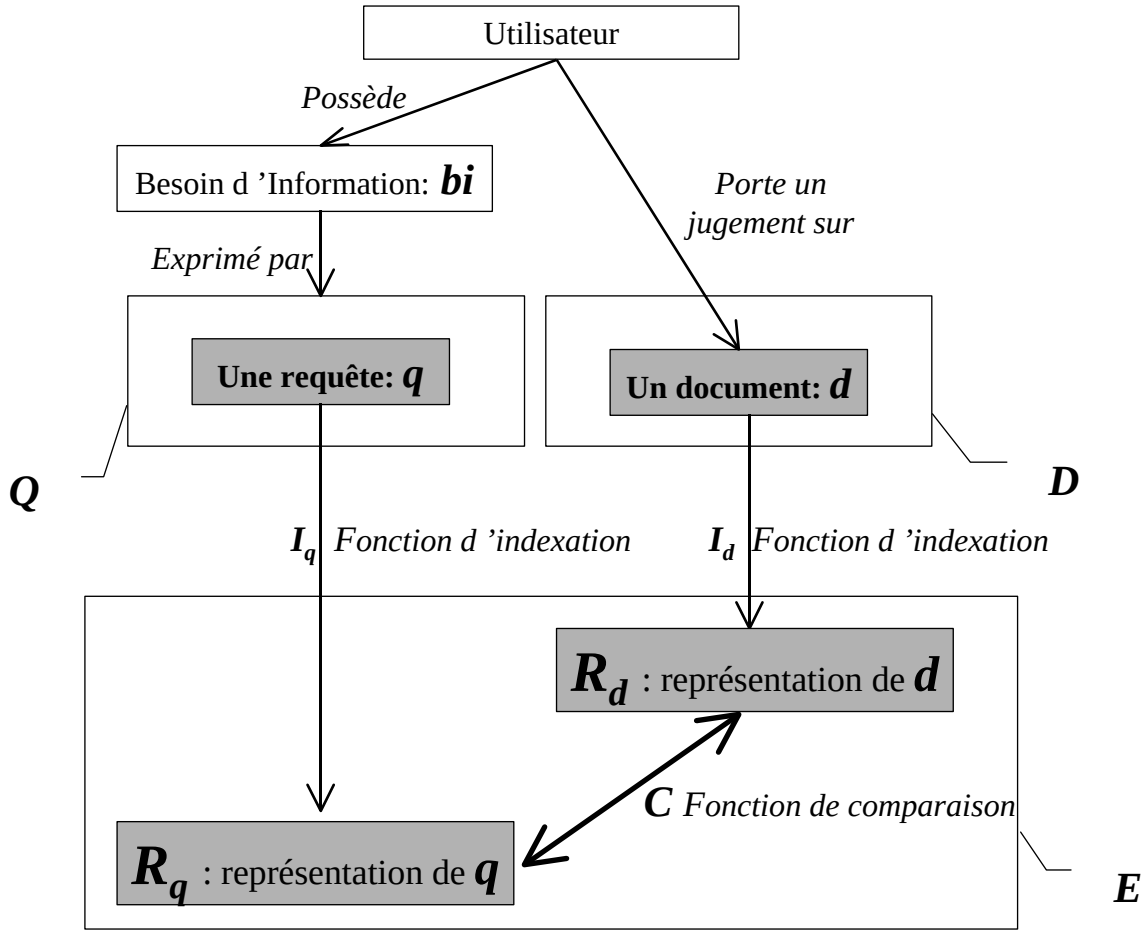


FIG.1.1–Schéma de la recherche documentaire. *(Roussey, 2001).*

1.3.5.1 L'Espace d'Indexation

L'indexation définit l'espace de représentation de l'information *(E)*, et influence ainsi la fonction de comparaison *(C)*. En effet, pour pouvoir être comparés, il faut que la représentation d'un document *d* (R_d) et la représentation d'une requête *q* (R_q) soient exprimées dans le même espace d'indexation *E*. C'est pourquoi différents modèles de SRI ont été créés, définissant à partir d'un nouvel espace d'indexation, toutes ses caractéristiques.

Si un utilisateur possède un besoin d'information *bi,* cet utilisateur doit exprimer *bi* dans le langage d'interrogation du SRI pour former la requête *q*.

Soit :

– Q : l'espace des requêtes et q une requête telle que $q \in Q$.

– D : l'espace des documents et d un document tel que $d \in D$.

– E : l'espace d'indexation du SRI.

La phase d'indexation se décompose en deux fonctions d'indexations I_q et I_d telles que :

– I_q est une application de Q dans E, qui à tout élément q de Q associe une image dans E unique $I_q(q) = R_q$.

$$I_q : Q \rightarrow E$$
$$q \mapsto I_q(q)$$

– I_d est une application de D dans E, qui à tout élément d de D associe une image dans E unique $I_d(d) = R_d$.

$$I_d : D \rightarrow E$$
$$d \mapsto I_d(d)$$

La fonction de comparaison C est une application qui, à tout couple de E, associe une valeur numérique comprise entre 0 et 1, telle que :

$$C : E \times E \rightarrow [0,1]$$
$$(R_q, R_d) \mapsto C(R_q, R_d)$$

avec $R_q = I_q(q)$ et $R_d = I_d(d)$.

L'indexation est le processus de représentation des textes, c'est-à-dire le passage d'un document (ou d'une requête) à sa représentation manipulée par un système documentaire *(Le Loarer, 1994)*. Le type de représentation définit l'espace d'indexation et implique ainsi la fonction de comparaison. Les composantes de l'espace d'indexation sont les entités d'indexation et les index.

1.3.5.2 Les Entités d'Indexation

Il existe au moins quatre entités d'indexation possibles : le groupe de caractères (les n-grammes *(Halleb & Lelu, 1997)*, le mot, le terme et le concept. L'unité lexicale est un élément du

vocabulaire de la langue, auquel sont associées des règles syntaxiques de construction de phrase. Un terme est une unité lexicale correspondant à une unité sémantique. Le terme dénote une notion précise, il est la manifestation linguistique d'un concept dans un texte *(Bourigault & Condamines, 1998)*. Un terme est le label d'un concept dans un contexte précis. Un terme est composé d'au moins un mot, mais il n'y a pas de limite sur son nombre de mots. Par exemple, le terme « rupture d'anévrysme » se compose de trois mots, mais représente un seul concept. La relation entre terme et concept est une relation ambiguë. En effet, un concept peut être représenté par plusieurs termes et un même terme peut représenter, suivant le contexte, des concepts différents.

TAB.1.1–Quelques relations liant les mots, les termes et les concepts.

Relation	Mot	Terme	Concept
Homographie	Car	Car : conjonction de coordination	introduit une explication
		Car : nom masculin singulier	grand véhicule automobile de transport collectif, routier ou touristique
Synonymie	Car	Car : nom masculin singulier	grand véhicule automobile de transport collectif, routier ou touristique
	Autocar	Autocar : nom masculin singulier	
Polysémie	Double	Double : nom masculin	quantité égale à deux fois une autre
			reproduction, copie, duplicata, obtenu au moyen d'un carbone.

Une entité d'indexation peut être également un ensemble de symboles (un nombre, une icône) caractérisant un groupe de mots (ou un groupe de termes ou un groupe de concepts, etc.), jugé valide pour représenter le contenu du document. Cet ensemble de symboles est appelé terme d'indexation (ou descripteur lorsqu'il est utilisé pour indexer un document).

1.3.5.3 Les Langages d'Indexation

L'ensemble des termes d'indexation constitue le vocabulaire du langage d'indexation. Ils peuvent être de deux types *(Abel, 1993)* : libre ou contrôlé. *(Jacquemin, 1998)* définit les langages documentaires comme une liste contrôlée de termes d'indexation ayant fait l'objet d'une validation humaine. Le langage d'indexation peut être une combinaison d'un langage libre et d'un langage contrôlé. Le choix d'un langage influence ainsi la manière dont sont choisis les descripteurs.

1.3.5.3.1 Le Langage Libre

Les termes d'indexation d'un langage d'indexation libre sont extraits des textes et peuvent donc inclure toute la variété du langage naturel. Ce langage d'indexation est construit lors de la phase d'indexation. C'est un langage évolutif dont le vocabulaire est choisi à posteriori et n'est pas limité par un contrôle. Il est composé de tous les descripteurs déterminés librement afin

d'indexer les documents. L'indexation en texte intégral en est un exemple : tous les mots du document sont extraits automatiquement pour constituer l'index du document. Par conséquent, le vocabulaire évolue rapidement et peut contenir des termes synonymes, polysémiques,...etc., entraînant des incohérences et diminuant les performances du SRI. Des documents portant sur le même sujet peuvent être indexés par des descripteurs différents. Inversement, des documents de sujets différents peuvent être indexés par le même descripteur. De plus, ce qui est vrai pour l'indexation des documents et aussi vrai pour l'indexation des requêtes. Ainsi un document sera retrouvé pour une requête parce que son index contient les descripteurs de la requête alors qu'il ne traite pas du sujet de la requête. Inversement, un document pertinent pour une requête ne sera pas retrouvé car ses descripteurs sont différents de ceux de la requête.

1.3.5.3.2 Le Langage Contrôlé

Le langage contrôlé, ou langage documentaire, est un langage normalisé. Les termes d'indexation utilisés sont prédéfinis et limités dans une liste pour éviter les problèmes de polysémie et de synonymie du langage libre. Ce type de langage d'indexation est construit à priori avant de commencer la phase d'indexation.

Cette liste, pour être efficace, ne doit pas contenir de termes polysémiques ou synonymiques. Ainsi, un terme d'indexation ne possède qu'un seul sens. Inversement, un sens n'est associé qu'à un seul terme d'indexation. Par conséquent, l'utilisation d'un langage contrôlé doit permettre de limiter le nombre de représentations possibles du contenu du document, si l'on ne tient pas compte de la subjectivité de l'interprétation des documents et des termes. Construit à priori, ce langage doit être connu avant d'indexer un document et avant de construire une requête. Pour faciliter le choix des descripteurs et appréhender rapidement le vocabulaire du langage contrôlé, l'ensemble des termes d'indexation est organisé dans un thésaurus.

Le thésaurus contient le lexique de tous les termes normalisés (masculin singulier) du langage documentaire. Les termes du thésaurus sont en relations sémantiques structurant le domaine de connaissance. Ces relations, au nombre de trois, sont illustrées par des exemples tirés du thésaurus anglais Global Legal Information Network :

– La relation d'équivalence regroupe les termes jugés équivalents. Le langage documentaire ne différencie pas ces termes les uns des autres. Ces termes peuvent être synonymes ou très proches sémantiquement. Un seul terme, appelé terme préféré, est choisi comme terme d'indexation. Ce terme d'indexation représente le concept identifié par l'ensemble de termes en relation d'équivalence. Par exemple, dans le thésaurus Global Legal Information Network, *Catastrophes*, *Natural Disasters* et *Disasters* sont considérés comme synonymes et *Disasters* est choisi comme terme d'indexation.

– La relation hiérarchique construit une hiérarchie entre les termes d'indexation, du général au spécifique ou d'un tout à ses parties. Par exemple, *Government organisation* a quatre termes spécifiques *Diplomatic & consular service, Administrative agencies, Airports, Colonies*.

– La relation d'association lie des termes d'indexation ayant des connotations entre eux. Dans GLIN, *Government organisation* a pour termes relatifs *Administratives laws, Corporations, Government ; Criminal courts, Employees, Government*.

L'organisation du thésaurus permet de trouver le terme d'indexation le plus approprié pour représenter un concept. Ainsi, l'utilisateur d'un SRI peut utiliser un terme de son vocabulaire

comme entrée dans le thésaurus et, en suivant différentes relations, trouver le terme d'indexation reconnu par le système pour composer sa requête.

Les termes du langage d'indexation peuvent être sélectionnés par une indexation manuelle, semi-automatique ou automatique.

1.3.5.4 Les Types d'Indexation

L'indexation automatique et l'indexation humaine, aussi appelée indexation manuelle, se différencient par l'agent mettant en œuvre le processus d'indexation des documents :

– Dans le cas d'une indexation humaine, c'est le documentaliste qui effectue l'analyse du document, pour identifier son contenu et en construire une représentation.

– Dans le cas d'une indexation automatique, c'est le SRI qui génère les index des documents. L'indexation assistée revient, le plus souvent, à faire valider, ou corriger, par un humain une représentation du document proposée par le système.

1.3.5.4.1 Indexation Manuelle

L'indexation manuelle est réalisée par des documentalistes. Ces experts ont pour tâche de caractériser au mieux les idées contenues dans les unités documentaires. Cette indexation requiert un important effort intellectuel et cognitif pour identifier et décrire l'essence des unités documentaires. Ce type d'indexation permet d'obtenir une caractérisation performante mais subjective car cette approche dépend fortement des connaissances du domaine des documentalistes. L'indexation manuelle trouve ses limites pour de grandes bases de documents qui nécessitent énormément de temps pour être traitées.

L'indexation manuelle est très souvent critiquée pour son coût. En effet, la personne chargée de l'analyse des documents doit posséder les connaissances minimales à la compréhension des centres d'intérêt du document, sous peine d'obtenir une indexation incorrecte. Une autre caractéristique fréquemment soulignée de l'indexation humaine est sa variabilité. En effet, même si l'indexation s'appuie sur un langage documentaire, des descripteurs différents peuvent être proposés pour représenter un même document suivant l'interprétation faite du contenu du document. Par conséquent, la variabilité entraîne une incohérence dans la base des index qui diminue les performances du SRI. Cette variabilité a été repérée aussi bien dans des situations où plusieurs personnes indexaient que dans des situations où une même personne indexait un même document à deux moments différents *(Le Loarer, 1994)*.

1.3.5.4.2 Indexation Automatique

Face à cette situation, l'indexation automatique présente l'avantage d'une régularité du processus, car l'indexation automatique fournit toujours le même index pour le même document. Ce qui constitue une qualité du système, mais qui est différente de la justesse de l'indexation. En effet, l'indexation automatique ne peut pas interpréter un texte et s'adapter à de nouveaux vocabulaires. Par exemple, si le système n'a aucune connaissance lui permettant de

lever les ambiguïtés des termes, il génèrera des erreurs d'interprétation entraînant ici aussi des incohérences dans la base des index.

1.3.5.4.3 Indexation Semi-Automatique

Ce type d'indexation combine les méthodes d'indexation manuelle et automatique en privilégiant toutefois l'intervention humaine. Ainsi, les experts caractérisent les idées contenues dans une unité documentaire sous la forme de méta-informations. Une indexation automatique est ensuite réalisée pour l'unité documentaire en tenant compte de ces méta-informations.

1.3.5.5 Cas de l'Indexation Automatique

Pour de grandes bases, la tendance générale s'oriente vers un processus d'indexation automatique permettant d'extraire rapidement les termes représentatifs des unités documentaires. L'intérêt d'une telle indexation réside principalement dans sa rapidité d'exécution qui est tout à fait adaptée à des volumes très importants mais également dans le fait qu'elle permet de limiter la représentation des documents aux « entrées » utiles permettant de retrouver les informations accessibles. Cette approche repose sur différentes phases correspondant à :

- l'extraction des termes d'indexation,
- la réduction du langage d'indexation,
- la pondération des termes d'indexation.

1.3.5.5.1 Extraction des Termes d'Indexation

Afin de construire le langage d'indexation, le système parcourt le contenu du document pour en extraire les termes d'indexation. Diverses approches pour cette extraction sont envisageables:

- une approche linguistique : elle fait appel à des techniques de traitement du langage naturel pour analyser et comprendre le contenu de l'unité documentaire ou d'une requête. Elle repose sur une recherche du sens même du contenu des unités documentaires.

- une approche morpho-syntaxique : elle repose sur une étude morpho-syntaxique du contenu des diverses unités documentaires.

- une approche mixte : elle combine les deux approches précédentes.

Dans une unité documentaire, l'ensemble des termes d'indexation extraits peut être important. Plus il y a de termes, plus les temps de calculs (notamment lors de l'appariement entre la requête et le document) sont importants. Il est donc nécessaire de limiter le langage d'indexation aux termes les plus représentatifs du contenu d'une unité documentaire.

a) Choix de termes

Dans la recherche documentaire, le document d_j est transformé en un vecteur $d_j=(w_{1j},w_{2j},...,w_{|T|j})$, où T est l'ensemble de termes (ou descripteurs) qui apparaissent au moins une fois dans le corpus (ou la collection). Le poids w_{kj} correspond à la contribution du termes t_k à la sémantique du document d_j. Cependant, la représentation par un vecteur entraîne une perte d'information notamment celle relative à la position de mots dans la phrase.

b) Représentation en « sac de mots »

La représentation de textes la plus simple a été introduite dans le cadre du modèle vectoriel et porte le nom de « sac de mots ». Le principe consiste à transformer les textes en vecteurs dont chaque composante représente un mot. Les mots ont l'avantage de posséder un sens explicite. Cependant, plusieurs problèmes se posent. Il faut tout d'abord définir ce qu'est « un mot » pour pouvoir le traiter automatiquement. On peut le considérer comme étant une suite de caractères appartenant à un dictionnaire, ou bien, de façon plus pratique, comme étant une séquence de caractères non-délimiteurs encadrés par des caractères délimiteurs (caractères de ponctuation) *(Gilli, 1988)* ; il faut gérer les sigles, ainsi que les mots composés ce qui nécessite un pré-traitement linguistique. On peut choisir de conserver les majuscules pour aider, par exemple, à la reconnaissance de noms propres, mais se pose le problème des débuts de phrases. Les composantes du vecteur sont fonction de l'occurrence des mots dans le texte. Cette représentation des textes exclut toute analyse grammaticale et toute notion de distance entre les mots : c'est pourquoi elle est appelée « sac de mots »; d'autres auteurs parlent d'« ensemble de mots » lorsque les poids associés sont binaires. Un grand nombre d'auteurs comme *(Lewis, 1992)*, *(Apté et al., 1994)*, *(Dumais et al., 1998)*, *(Aas & Eikvil, 1999)* utilisent les mots comme termes (c'est à dire comme composantes du vecteur) pour représenter les textes.

c) Représentation des textes par des phrases

Malgré la simplicité de l'utilisation de mots comme unité de représentation, certains auteurs proposent plutôt d'utiliser les phrases comme unité *(Fuhr & Buckley, 1991) (Schütze et al., 1995) (Tzeras & Hartmann, 1993)*. Les phrases sont plus informatives que les mots seuls : elles ont l'avantage de conserver l'information relative à la position du mot dans la phrase. Logiquement, une telle représentation doit obtenir de meilleurs résultats que ceux obtenus via les mots. Cependant les expériences présentées ne sont pas concluantes : si les qualités sémantiques sont conservées, les qualités statistiques sont largement dégradées : le grand nombre de combinaisons possibles entraîne des fréquences faibles et trop aléatoires *(Lewis, 1992)*. Néanmoins, *(Caropreso et al., 2001)* proposent d'utiliser les phrases « statistiques » comme unités de représentation en opposition aux phrases « grammaticales » et obtiennent de bons résultats. Une phrase statistique est un ensemble de mots contigus (mais pas nécessairement ordonnés) qui apparaissent ensembles mais qui ne respectent pas forcément les règles grammaticales. Afin de déterminer les phrases statistiques, ils utilisent des prétraitements tels que l'élimination des mots vides (*stop words*) et le *stemming*.

d) Représentation des textes avec des racines lexicales et des lemmes

Dans le modèle précédent (représentation en « sac de mots »), chaque flexion de mot est considérée comme un descripteur différent et donc une dimension de plus. Ainsi, les différentes formes d'un verbe constituent autant de mots. Par exemple : les mots « déménageur, déménageurs, déménagement, déménagements, déménager, déménage, déménagera, etc. » sont considérés comme des descripteurs différents alors qu'il s'agit de la même racine « déménage ». Les techniques de *désuffixation (*ou *stemming*), qui consistent à rechercher les racines lexicales, et de *lemmatisation* cherchent à résoudre cette difficulté. Pour la recherche des racines lexicales, plusieurs algorithmes ont été proposés. L'un des plus connus pour la langue anglaise est l'algorithme de Porter *(Porter, 1980)*. Une comparaison entre différents algorithmes de recherche de racines lexicales a été menée dans *(Hull, 1996)*. La lemmatisation consiste à remplacer les verbes par leur forme infinitive, et les noms par leur forme au singulier. Un algorithme efficace, nommé TreeTagger *(Schmid, 1994)*, a été développé pour les langues anglaise, française, allemande et italienne. L'extraction des *stemmes* repose, quant à elle, sur des contraintes linguistiques bien moins fortes. Elle se base sur la morphologie flexionnelle mais aussi dérivationnelle *(de Loupy, 2001)*. De ce fait, les algorithmes sont beaucoup plus simplistes et mécaniques que ceux permettant l'extraction des lemmes. Ils sont plus rapides, mais leur précision et leur qualité sont naturellement inférieures.

e) Méthodes basées sur les n-grammes

Un n-gramme est une séquence de *n* caractères. Dans la littérature, ce terme désigne quelquefois des séquences qui ne sont ni ordonnées ni consécutives. Par exemple un 2-gramme peut être composé de la première lettre et de la troisième lettre d'un mot *(Cavnar & Trenkle, 1994)*. *(Caropreso et al., 2001)* considèrent un *n*-grammes comme un ensemble non ordonné de *n* mots après avoir effectué la désuffixation (ou *stemming*) et la suppression de mots vides.

Pour un document quelconque, l'ensemble des n-grammes est le résultat obtenu en déplaçant une fenêtre de *n* cases sur le texte ; ce déplacement se fait par étapes de un caractère pour constituer l'ensemble de tous les n-grammes du document. Le profil n-grammes d'un document est la liste des n caractères les plus fréquents, par ordre décroissant de leur fréquence d'apparition dans le document, ainsi que leurs fréquences elles-mêmes. Un document est caractérisé par son profil n-grammes. Les profils s'obtiennent en temps linéaire grâce à des tables de hachage. La notion de n-grammes a été introduite par *(Shannon, 1948)* qui s'intéressait à la prédiction d'apparition de certains caractères en fonction des autres caractères. Il y a plusieurs avantages à l'utilisation des n-grammes : ils capturent les connaissances des mots les plus fréquents et opèrent indépendamment des langues, alors que les systèmes basés sur les mots sont dépendants des langues. Par exemple, les traitements d'élimination des "mots vides", de "recherche de racine" et de lemmatisation sont spécifiques à chaque langue. De même, la plupart des techniques de n-grammes n'exigent pas une segmentation préalable du texte en mots. Ils sont également tolérants aux déformations liées à l'utilisation de systèmes de reconnaissance de caractères et tolérants aux fautes d'orthographe. Enfin, ces techniques n'ont pas besoin de procéder à l'élimination ni des "mots vides", ni au "Stemming", ni à la lemmatisation, qui améliorent la performance des systèmes basés sur les mots. Si un document

contient plusieurs mots de même racine, les fréquences des n-grammes correspondants augmenteront sans avoir besoin d'aucun traitement linguistique préalable.

1.3.5.5.2 Réduction du Langage d'Indexation

Pour limiter le nombre de termes d'indexation, il est nécessaire de ne conserver que les termes qui contribuent au mieux à la caractérisation de l'unité documentaire. Ainsi, dans un premier temps, les mots qualifiés de «vides» sont supprimés. Ce processus correspond à la suppression des termes d'indexation qui n'apportent pas de sens réel à l'unité documentaire. Ces mots vides se trouvent généralement dans un anti-dictionnaire et sont des mots à contribution purement syntaxique comme les pronoms «à» ou «de» par exemple pour le français. De plus, les lois de Zipf *(Zipf, 1949)* et *(Luhn, 1958)* soulignent que :

– un terme d'indexation apparaissant trop fréquemment dans un texte ne joue qu'un rôle syntaxique (mot vide) et ne doit pas être utilisé dans le langage d'indexation,

– un terme d'indexation présent dans l'ensemble des documents de la base n'apporte aucun pouvoir discriminant à la recherche (le terme est considéré comme un mot vide),

– un terme d'indexation de fréquence intermédiaire est considéré comme significatif. Il représente le contenu sémantique de l'unité documentaire et il appartient au langage d'indexation.

Afin de mesurer un appariement graduel entre une unité documentaire et une requête, les termes d'indexation sont pondérés afin de refléter leur degré d'importance ou de représentativité dans l'unité documentaire.

a) Réduction de la dimension

Avec la représentation en sac de mots, chacun des mots d'un corpus est un descripteur potentiel. Pour un corpus de taille moyenne, ce nombre peut être de plusieurs dizaines de milliers. Les mots les plus fréquents peuvent être supprimés car ils n'apportent pas d'information sur la catégorie d'un texte puisqu'ils sont présents partout. De même, les mots très rares, qui n'apparaissent qu'une ou deux fois sur un corpus, sont supprimés (termes hapax). Après la suppression de ces deux catégories de mots, le nombre de candidats reste encore élevé, et il faut utiliser une méthode statistique pour choisir les descripteurs utiles. Les techniques de réduction de dimension sont issues de la théorie de l'information et de l'algèbre linéaire *(Sebastiani, 2002)*.

b) Réduction locale de dimension

Elle consiste à proposer, pour chaque catégorie c_i un nouvel ensemble de terme T'_i avec $|T'_i|<<|Ti|$ *(Apté et al., 1994)*, *(Lewis & Ringuette, 1994)*, *(Schütze et al., 1995)*, (*Wiener et al., 1995)*, (Ng et al., 1997), *(Li & Jain, 1998)*, *(Sable & Hatzivassiloglou, 2000)*. Ainsi, chaque catégorie c_i possède son propre ensemble de termes et chaque document d_j sera représenté par un ensemble de vecteurs d_j différents selon la catégorie. Généralement, $10 \leq T'_i \leq 50$.

c) Réduction globale de dimension

Dans ce cas, le nouvel ensemble de termes T' est choisi en fonction de toutes les catégories. Ainsi, chaque document d_j sera représenté par un seul vecteur d_j quelque soit la catégorie *(Yang & Pedersen, 1997), (Mladeni´c & Grobelnik, 1998), (Caropreso et al., 2001), (Yang & Liu, 1999)*. Les techniques de réduction de termes peuvent être appliquées soit localement soit globalement.

d) Sélection de termes

Les techniques de réduction de dimensions par la sélection de termes visent à proposer un nouvel ensemble T' avec $|T'| << |T|$. Parmi ces techniques figurent le calcul de l'information mutuelle *(Lewis, 1992b) (Moulinier, 1997) (Dumais et al., 1998)*, ou des méthodes plus simples utilisant uniquement les fréquences d'apparitions *(Wiener et al., 1995)*, *(Yang & Pedersen, 1997)*. D'autres méthodes ont également été testées *(Moulinier, 1996)*, *(Sahami, 1999)*.

e) Extraction de termes

L'objectif des techniques d'extraction de termes est de proposer un sous-ensemble T' avec $|T'|<<|T|$ mais, à la différence des techniques de sélection, le sous-ensemble T' est une synthèse (combinaison linéaire des descripteurs) qui devrait maximiser la performance. On recherche des variables synthétiques pour éliminer les problèmes liés aux synonymies, polysémie et homonymies en proposant des variables artificielles, jouant le rôle de nouveaux «termes». L'une des approches est appelée le «Latent Semantic Indexing (LSI)», proposée par *(Deerwester et al., 1990)*. Le LSI est fondé sur l'hypothèse d'une structure latente des termes, identifiable par les techniques factorielles. Il consiste en une décomposition en valeurs singulières de la matrice dans laquelle chaque document est représenté par la colonne des occurrences des termes qui le composent. D'autres approches sont également proposées et utilisées pour la réduction de dimensions comme « *term clustering* ».

1.3.5.5.3 Pondération des Termes d'Indexation

Le calcul de la représentativité d'un terme d'indexation repose sur sa fréquence d'apparition dans le texte en langage naturel *(Salton, 1983)*, *(Zipf, 1949)*. Afin de caractériser le pouvoir de représentativité (poids) des termes d'indexation, différentes mesures ont été proposées. Elles reposent sur des mesures statistiques dont les plus utilisées sont la fréquence relative d'un

terme d'indexation (TF). Il s'agit de la fréquence d'apparition du terme d'indexation dans l'unité documentaire et la fréquence absolue d'un terme d'indexation dans la collection globale d'unités documentaires (IDF). Il s'agit de la fréquence inverse d'apparition du terme d'indexation dans l'ensemble des unités documentaires de la collection *(Sparck Jones, 1972)*.

a) Codage des termes

Une fois choisies les composantes du vecteur représentant un texte *j*, il faut décider comment coder chaque coordonnée de son vecteur d_j. Il existe différentes méthodes pour calculer le poids w_{kj} . Ces méthodes sont basées sur les deux observations suivantes :

1. plus le terme t_k est fréquent dans un document d_j , plus il est en rapport avec le sujet de ce document.
2. plus le terme t_k est fréquent dans une collection, moins il sera utilisé comme discriminant entre documents.

Soient $\#(t_k, d_j)$ le nombre d'occurrences du terme t_k dans le texte d_j, $|Tr|$ le nombre de documents du corpus et $\#Tr(t_k)$ le nombre de documents de cet ensemble dans lesquels apparaît au moins une fois le terme t_k. Selon les deux observations précédentes, un terme t_k se voit donc attribuer un poids d'autant plus fort qu'il apparaît souvent dans le document et rarement dans le corpus complet. La composante du vecteur est codée $f(\#(t_k, d_j))$, où la fonction f reste à déterminer. Deux approches triviales peuvent être utilisées. La première consiste à attribuer un poids égal à la fréquence du terme dans le document :

$w_{kj} = \#(t_k, d_j)$

et la deuxième approche consiste à associer une valeur booléenne :

$$w_{kj} = \begin{cases} 1 & Si\ \#(t_k , d_j) > 1 \\ 0 & Sinon \end{cases}$$

b) Codage TF × IDF

Les deux fonctions précédentes sont rarement utilisées car ces codages appauvrissent l'information : la deuxième fonction ne prend pas en compte la fréquence d'apparition du terme dans le texte, fréquence qui peut constituer un élément de décision important. La première fonction ne prend pas en compte la fréquence du terme dans les autres textes.

Le codage TF × IDF a été introduit dans le cadre du modèle vectoriel (section 1.4.2) et utilise une fonction de l'occurrence multipliée par une fonction de l'inverse du nombre de documents différents dans lequel un terme apparaît. Ce sigle provient de l'anglais et signifie « '*Term Frequency*'× '*Inverse Document Frequency*' ». Les termes caractérisant une classe apparaissent plusieurs fois dans les document de cette classe, et moins, ou pas du tout, dans les autres. C'est pourquoi le codage TF × IDF *(Sebastiani, 2002)* est défini comme suit :

$TF \times IDF(t_k, d_j) = \#(t_k, d_j) \times log\ (|Tr|/\#Tr(t_k))$

c) Codage TFC

Le codage TF × IDF ne corrige pas la longueur des documents. Pour ce faire, le codage TFC est similaire à celui de TF×IDF mais il corrige les longueurs des textes par la normalisation en cosinus, afin de ne pas favoriser les documents les plus longs. D'autres codages sont également utilisés, comme par exemple le codage LTC *(Buckley et al., 1994)* qui tente de réduire les effets des différences de fréquences, ou encore le codage à base d'entropie. *(Dumais, 1991)* affirme obtenir de meilleurs résultats avec un codage basé sur l'entropie *(Aas & Eikvil, 1999)*.

1.4 Modèles de Recherche d'Information et de Représentation

L'indexation permet au SRI d'obtenir l'essence des unités documentaires par le biais d'un langage d'indexation. Cependant, il est nécessaire d'utiliser un modèle unique de représentation (modèle de recherche) pour la requête et pour les unités documentaires afin d'en apprécier l'appariement. Ces modèles peuvent être divisés en deux catégories : les modèles dits « exacts » qui ne retournent que des documents répondant exactement à la requête (modèle booléen) ou les modèles dits « partiels » (probabiliste, vectoriel…) qui retournent des documents répondant à tout ou partie de la requête. Ces derniers utilisent une valeur réelle (degré de pertinence système) pour rendre compte du degré de l'appariement entre la requête et une unité documentaire. Il est à noter que cette pertinence système est une valeur calculée et qu'elle peut être différente de la pertinence réelle qui découle du jugement de pertinence réalisé par l'utilisateur. L'indexation pondérée permet de donner à chaque descripteur un niveau d'importance. Ainsi on peut connaître le sujet principal d'un texte et les thèmes secondaires abordés. Un poids, affecté aux descripteurs, indique son niveau d'importance par rapport au texte indexé. Ce poids est généralement calculé à l'aide d'une fonction statistique provenant d'un SRI de type vectoriel ou de type probabiliste. Il correspond à la probabilité que le descripteur soit pertinent pour le document. Une collection de textes peut être ainsi représentée par une matrice dont les lignes sont les termes qui apparaissent au moins une fois et les colonnes sont les documents de cette collection. L'entrée w_{kj} est le poids du terme t_k dans le document d_j. On distingue différents types de représentation *(Le Loarer, 1994)*.

1.4.1 Le Modèle Booléen

L'indexation dite à plat considère que les descripteurs ont tous le même statut vis-à-vis du texte à indexer. La représentation du texte est une succession d'entités d'indexation : une liste de descripteurs non ordonnée. Seuls les systèmes documentaires de type booléen manipulent ce genre d'indexation.

Le modèle booléen tire son nom des opérateurs booléens utilisés pour formuler une requête. En effet, une requête est une formule logique, combinant des descripteurs et les opérateurs ET, OU, NON. Les documents sont représentés par une liste de descripteurs. Ces descripteurs peuvent appartenir à un langage libre ou contrôlé. Ils peuvent être extraits automatiquement des documents ou manuellement choisis par des documentalistes. Les index sont stockés dans un

fichier inverse où, à chaque descripteur correspond la liste des documents contenant ce descripteur dans leur index. La fonction de comparaison retrouve les documents dont les index valident la formule logique de la requête. Donc la base de documents est séparée en deux, les documents qui correspondent à la requête et ceux qui n'y correspondent pas. L'inconvénient majeur de ce modèle *(Salton, 1988)* est l'absence d'ordonnancement des documents résultats par la fonction de comparaison.

1.4.2 Le Modèle Vectoriel

Ce modèle représente un document ou une requête par un vecteur dans un espace d'indexation construit à partir des entités d'indexation. Les coordonnées des vecteurs sont les poids indiquant l'importance du descripteur par rapport au document. L'ensemble des coordonnées des vecteurs est contenu dans une matrice. La fonction de comparaison évalue la correspondance entre deux vecteurs (document et requête) ce qui permet de classer les résultats. Le schéma suivant illustre cette méthode.

T_k est un des vecteurs de base de l'espace de représentation.

Il représente *l'entité d'indexation k.*

D_i est le vecteur désignant le *document i.*

$W_{i,k}$ est le poids de *l'entité k* dans le *document i.*

$D_i = (W_{i,1}, W_{i,2}, W_{i,3})$.

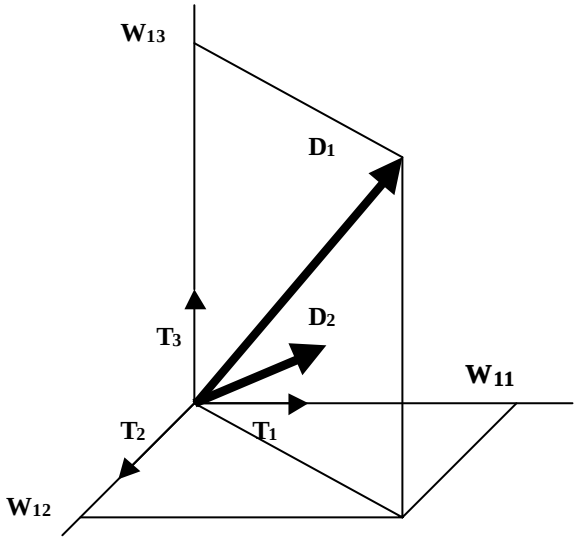


FIG.1.2.–Représentation des documents dans un espace vectoriel intitulé espace des termes.

La matrice représentant ce corpus de deux documents s'appelle « *matrice terme-document* » et s'écrit de la manière suivante :

TAB.1.2.–«Matrice terme-document» de l'espace des termes précédent.

D_1	D_2	
$W_{1,1}$	$W_{2,1}$	T_1
$W_{1,2}$	$W_{2,2}$	T_2
$W_{1,3}$	$W_{2,3}$	T_3

Le modèle vectoriel permet à l'utilisateur de formuler sa requête en langage (pseudo) naturel. Il permet de retrouver des unités documentaires plus ou moins pertinentes par rapport à la requête et ainsi de restituer une liste ordonnée par pertinence système ou bien une liste limitée aux *k* documents les plus pertinents. En revanche, l'inconvénient majeur de ce modèle vectoriel est qu'il ne permet pas de modéliser les associations entre les termes d'indexation : chacun des termes d'indexation est considéré comme indépendant des autres (variables indépendantes dans l'espace du langage d'indexation).

Le système SMART développé par l'équipe de Salton *(Salton et al., 1975)* est l'un des premiers SRI utilisant ce modèle. Les coordonnées des vecteurs sont calculées à partir de la fréquence des mots dans les documents par la formule du TF×IDF. L'indexation est donc totalement automatique.

La pondération des descripteurs tient compte de deux facteurs :

1. TF : la fréquence du terme dans le document.
2. IDF : l'importance du terme comme descripteur dans le corpus de documents, qui est une fonction inverse du nombre de documents indexés par ce terme.

Le poids $W_{i,m}$ d'un descripteur MOT_m pour un document D_i est calculé de la manière suivante :

$TF = fréquence\ du\ MOT_m\ dans\ D_i$

$$IDF_m = \frac{1}{nb\ de\ documents\quad contenant\quad MOT_m}.$$

Ainsi, un descripteur avec une forte pondération est un terme fréquent dans un document et absent des autres documents. Par conséquent, les termes rares qui risquent d'être peu utilisés pour la recherche sont privilégiés. Cette pondération amplifie considérablement l'importance des termes étrangers, des noms propres et des mots mal orthographiés dans l'index.

Dans leur approche, *(Salton et al., 1975)* font l'hypothèse que les mots sont indépendants les uns des autres. La fonction de comparaison, appelée fonction de similarité, se base sur le cosinus de l'angle formé par les deux vecteurs à comparer. Plus l'angle est petit, plus les vecteurs sont similaires. Pour un vecteur document $d = \left(W_1^d, W_2^d, W_3^d\right)$ et un vecteur requête la fonction de comparaison évaluera la similitude entre les deux vecteurs par la formule :

$$Sim(d,q) = \frac{\sum_{j=1}^{3}\left(W_j^d \times W_j^q\right)}{\sqrt{\left[\sum_{j=1}^{3}\left(W_j^d \times W_j^d\right) \times \sum_{j=1}^{3}\left(W_j^q \times W_j^q\right)\right]}}$$

$$q = \left(W_1^q, W_2^q, W_3^q\right)$$

1.4.3 Le Modèle Probabiliste

Ce type de modèle, basé sur la théorie des probabilités, considère la RI comme un espace d'évènements possibles. Un évènement peut être le jugement de pertinence porté par l'utilisateur sur un document par rapport à une requête ou l'association d'un descripteur à document. La représentation des documents est généralement un index pondéré, les poids des descripteurs correspondent à la probabilité que le descripteur soit pertinent pour le document, aussi appelée degré de croyance. Dans le cas du modèle binaire indépendant, seule la représentation de la requête est un index pondéré. Les documents y sont représentés par un index à plat *(Fuhr & Buckley, 1991)*.

La fonction de comparaison calcule la probabilité qu'un document *D* réponde à un besoin d'information d'un utilisateur, formulée par une requête *Q* qui peut être interprétée par la probabilité que l'événement pertinent *R* arrive sachant *D* et *Q : P(R/D, Q)*. Les documents résultats sont ordonnés. Pour calculer la probabilité de l'événement *(R/D, Q)*, il doit être décomposé en événements plus simples par la formule de Bayes, en tenant compte des dépendances entre les différents évènements. Plusieurs paramètres sont pris en compte : les documents, les requêtes, les représentations des documents, les termes d'indexation, etc. Afin d'obtenait une formule de comparaison calculable, les modèles probabilistes font des hypothèses restrictives, comme l'indépendance entre les mots. En revanche, le système INQUERY *(Turtle & Croft, 1991*) tient compte des dépendances entre certains évènements pour calculer *P(R/D, Q)* en parcourant un réseau d'inférence bayesienne.

L'avantage des modèles probabilistes est de tenir compte des imprécisions, en particulier entre le document et sa représentation. Pour évaluer les différentes probabilités du système il est nécessaire de disposer d'un jeu de données initiales. Ces systèmes fonctionnent en deux étapes. Une première étape d'apprentissage calcule les probabilités des évènements à partir d'un jeu de données et une seconde étape de test répond à une nouvelle requête. Les données nécessaires au calcul des probabilités peuvent être :

– La fréquence du mot dans le document,

– Un ensemble de jugement de pertinence de documents par rapport à des requêtes, généralement obtenues par retour de pertinence (relevance feedback) permettant de faire évoluer le système au cours de son utilisation,

– Un corpus de documents préalablement indexés manuellement, un jeu de test contenant des requêtes et leurs documents résultats, etc.

Ces systèmes sont utilisables autant pour l'indexation automatique que pour l'indexation manuelle. En indexation automatique, la probabilité qu'un descripteur soit représentatif du document est évaluée à partir d'un jeu de données. En indexation manuelle, l'événement d'attribution d'un descripteur à un document est connu donc sa probabilité d'apparition n'a pas besoin d'être évaluée. Dans ce cas, la représentation des documents est une indexation à plat. Les systèmes de type probabiliste peuvent autant utiliser une indexation en langage contrôlé qu'en langage libre, tout dépend du jeu de données utilisé au départ pour évaluer les probabilités.

1.4.4 Le modèle Logique

(van Rijsbergen, 1986) modélise la pertinence d'un document répondant à une requête par une implication logique. Soit $\chi(d)$ l'information contenue dans le document d et $\chi(q)$ le besoin d'information formulée par la requête q, tous deux sont des formules logiques. Ce genre de système cherche à évaluer l'ajout minimal d'information nécessaire pour obtenir l'implication $\chi(d) \rightarrow \chi(q)$, permettant de classer les documents résultats. Cette approche améliore l'utilisation des connaissances dans le SRI car les éléments d'information (et non plus les termes) sont les descripteurs du document. Le problème majeur est d'extraire les éléments d'information automatiquement. La proposition a été appliquée à plusieurs théories logiques pour déterminer $\chi(d) \rightarrow \chi(q)$ *(Bruza & Lalmas, 1995)*.

Une des théories, souvent utilisée est la théorie des situations. Les modèles logiques développés à partir de la théorie des situations considèrent qu'un document est identifié à une situation et que les éléments d'information sont des types. Un type possède la valeur vraie dans une certaine situation, et fausse dans une autre situation. Ces valeurs sont déterminées dans la phase d'indexation. Des contraintes sont définies entre ces types provenant, par exemple de relations sémantiques trouvées dans un thésaurus. Ces contraintes définissent la nature du flot d'information existant entre deux situations. La formule de comparaison évalue l'incertitude du flot d'information circulant entre la situation du document et celle de la requête *(Lalmas, 1995)*.

Les modèles logiques développés actuellement ont permis de mieux comprendre fondamentalement la RI en donnant un cadre théorique pour la comparaison entre les modèles existants. En revanche, l'implémentation de ces modèles semble difficile du fait de leur complexité. Les systèmes développés à partir de ce modèle n'ont pas donné de résultats très satisfaisants comparés aux autres systèmes *(Lalmas, 1998)*.

Une étude comparative de ces modèles ainsi que des extensions possibles peuvent être trouvées dans *(Soulé-Dupuy, 2001)*.

1.5 Environnements de Recherche

La tâche de recherche peut être réalisée dans un environnement adhoc ou dans un environnement dynamique qui conditionne le processus de recherche. Dans un environnement ad hoc, l'outil de recherche repose sur une ou plusieurs collections de documents stables et des besoins en informations momentanés et dynamiques. En revanche, dans un environnement dynamique, l'outil de recherche repose sur un flux dynamique de documents. Les besoins de l'utilisateur, contrairement à l'environnement ad hoc, sont relativement stables (profil de filtrage).

La recherche dans un contexte ad hoc peut être résumée en deux étapes :

– d'une part, les unités documentaires devant servir de support à la RI sont traitées pour être insérées dans la base d'indexation. L'ensemble des unités documentaires traitées par l'outil de recherche constitue la base d'indexation,

– d'autre part, l'utilisateur formule ses besoins sous la forme d'une requête (généralement sous la forme d'une liste de mots-clés).

L'outil de recherche identifie dans sa base d'indexation les unités documentaires pertinentes pour les besoins de l'utilisateur. Les unités sont proposées à l'utilisateur pour qu'il puisse à son tour les exploiter.

Le but d'un outil de filtrage est, à partir d'un flux d'unités documentaires, d'identifier celles qui sont susceptibles d'être pertinentes pour un utilisateur par rapport à son profil et une fonction de décision. Au début du processus, l'utilisateur définit ses besoins en information qui sont traduits par le système en un *profil utilisateur*. Pour *(Korfhage, 1997)*, un profil utilisateur est un indicateur des besoins en information, durables ou récurrents, qui sont représentés communément sous la forme d'une liste de mots-clés pondérés. A partir du flux de documents entrants, le système apprécie l'appariement entre le profil utilisateur et les documents. Grâce à une fonction de décision, le système est en mesure de décider, par rapport à l'appariement mesuré, si un document est pertinent ou non pertinent.

Cette dualité entre les outils de recherche ad hoc et les outils de filtrage repose sur le fait que *(Belkin, 1992)* :

– un outil de recherche ad hoc suppose l'existence d'une collection de documents alors qu'un outil de filtrage repose sur un ensemble de profils utilisateurs,

– un outil de recherche ad hoc est utilisé de façon ponctuelle, le besoin en information est de même unique et temporaire tandis qu'un outil de filtrage utilise des besoins à long terme,

– un outil de recherche ad hoc utilise et organise des informations tandis qu'un outil de filtrage vise à les diffuser,

– un outil de recherche ad hoc repose sur une base d'indexation statique alors qu'un outil de filtrage utilise des informations provenant d'un flux dynamique,

– un outil de recherche ad hoc permet de décider si un document est intéressant ou non plutôt que d'aller chercher les documents intéressants,

– un outil de recherche ad hoc repose, comparativement, sur une interaction importante avec l'utilisateur qui consulte les résultats de recherche, juge ces résultats… En revanche, un outil de filtrage est peu interactif puisque l'utilisateur consulte les documents proposés par le système de façon périodique,

– un outil de recherche ad hoc peut proposer à l'utilisateur la liste de tous les documents ordonnés par pertinence système alors que l'outil de filtrage doit décider si un document est pertinent ou non.

1.6 Evaluation de la Recherche d'Information

Afin d'évaluer les systèmes les uns par rapport aux autres, diverses plates-formes proposent un cadre d'évaluation entre les différents systèmes. Parmi les plus importantes, la plate-forme TREC (Text Retrieval Conference) *(Voorhees & Harman, 2001)* propose un cadre expérimental afin d'évaluer différentes applications de la Recherche d'Information (filtrage, recherche ad hoc…) et la plate-forme CLEF (Cross-Language Information Retrieval and Evaluation) *(Peters, 2000)* propose un cadre d'évaluation spécialisé dans la RI multilingue.

Pour comparer les SRI entre eux, des mesures d'efficacité ont été introduites. Ces mesures considèrent que la pertinence d'un document par rapport à une requête est binaire : un document est pertinent ou non. L'évaluation des SRI se fait en fonction de leur capacité à identifier l'ensemble des documents pertinents. Suite à une recherche, le corpus se sépare en quatre ensembles de documents.

TAB.1.3.– Les quatre ensembles de documents résultats en RI.

	documents pertinents	**documents non pertinents**
Documents sélectionnés	documents trouvés	documents trouvés documents hors contexte : **bruit**
Documents non-sélectionnés	documents oubliés : **silence**	documents non trouvés non pertinents

La cardinalité de ces différents ensembles de documents est utilisée pour évaluer la précision, le rappel et la précision moyenne *(van Rijsbergen, 1979)*.

Certains documents retenus par le système peuvent ne pas correspondre à la demande de l'utilisateur. La précision correspond au pourcentage de documents pertinents renvoyés par le système qui répondent effectivement à la requête. On cherche à maximiser ce pourcentage.

$$précision = \frac{documents\ trouvés}{documents\ sélectionnés}$$

On utilise aussi la notion de bruit qui présente le problème selon le point de vue opposé. Le bruit est le pourcentage de textes non pertinents renvoyés par le système :

bruit= 1- précision

Le système peut considérer certains textes comme non pertinents alors qu'ils correspondent à la requête de l'utilisateur. Le rappel désigne le pourcentage de documents pertinents rapportés par le système par rapport au nombre total de documents pertinents qui se trouvent dans la base documentaire. On cherche également à maximiser ce pourcentage.

$$rappel = \frac{documents\ trouvés}{documents\ \ pertinents}$$

On utilise aussi la notion de silence qui est le point de vue opposé. Le silence est le pourcentage de textes pertinents non renvoyés par le système :

silence = 1- rappel

L'évaluation de l'ordre d'un ensemble de documents est un peu plus complexe. On utilise la précision moyenne. Tout d'abord, en partant du premier document de la liste, des ensembles de documents sont construits ayant des niveaux de rappel prédéfinis par exemple (0,1 ; 0,3 ; 0,5 ; 0,7 ; 0,9). Ensuite, la précision est calculée pour chaque groupe de documents. Ce processus est répété pour plusieurs requêtes et une précision moyenne est calculée pour chaque niveau de rappel. La précision moyenne est en fait la moyenne de ces moyennes.

De nombreux facteurs entrent en jeu et cette évaluation ne peut qu'être relative, dépendante de l'application visée, du type de recherche effectué, de la base textuelle interrogée, du juge humain, etc. L'idéal est d'avoir un système dont la précision et le rappel sont toujours égaux à 1

(donc ni de bruit ni de silence), c'est à dire que tous les documents pertinents sont rapportés et seulement ces documents. Mais il est impossible d'obtenir, sur tous les domaines, quel que soit le corpus et quelle que soit la requête, un rappel et une précision de 100 %.

Après avoir décrit les concepts généraux de la RI, nous présentons dans cette section les particularités de celle-ci appliquée au Web.

1.7 La Recherche d'Information sur le Web

Deux types d'utilisateurs peuvent se distinguer sur le Web. Le premier est celui qui ne connaît pas exactement ce qu'il cherche et tente d'explorer la masse de documents à sa disposition. Dans ce cas, la navigation constitue l'outil le plus approprié. Le second type d'utilisateur qui représente le grand nombre, est celui qui définit une requête correspondant à son désir d'information et qui attend une liste, précise et pertinemment ordonnée, des documents. Les modèles de recherche booléen, vectoriels ou probabilistes présentent différentes solutions performantes, robustes et relativement simples à mettre en oeuvre. Sur le Web, les outils de recherche ad hoc correspondent aux moteurs de recherche. Ils reposent sur une méthode d'accès à l'information de type PULL. L'utilisateur suit une démarche active pour retrouver des documents répondant à ses besoins. Il existe toute une panoplie de moteurs de recherche sur le Web qui se différencient notamment par la taille de leur base d'indexation, leur langage d'interrogation, le type d'indexation utilisée... Les outils de filtrage, quant à eux, sont généralement nommés des outils PUSH. A l'instar des moteurs de recherche, les outils PUSH proposent automatiquement des documents pertinents à un utilisateur passif ayant initialement formulé ses besoins.

1.7.1 Caractéristiques de la Recherche d'Information sur le Web

L'utilisateur est au centre du processus de RI. Il intervient à différents niveaux (formulation de la requête, étude des résultats...) et de lui dépend en partie le résultat de la recherche. Cependant, chaque utilisateur est différent et certaines aptitudes sont nécessaires pour le bon achèvement de sa tâche de recherche. D'un point de vue général, *(Shneiderman, 1998)* souligne l'impact de la diversité humaine sur l'utilisation d'une application informatique à-travers différents aspects (physiques et lieu de travail ; cognitifs et sensoriels). Deux éléments conditionnent une RI sur le Web : la connaissance pratique et la connaissance du domaine. Ces deux connaissances jouent un rôle important dans l'apprentissage et les performances de l'utilisateur. *(Höschler & Strube, 2000)* soulignent que les connaissances sont les caractéristiques humaines essentielles de la RI sur le Web. La connaissance pratique représente la connaissance du Web avec tout ce que cela comporte. Nous assimilons à cette catégorie la maîtrise du navigateur, l'utilisation des liens hypertextes, des fonctionnalités offertes par les outils de recherche... (GVU, 1998) souligne également le fait qu'il y a une grande différence entre les utilisateurs novices et experts, qui est due à leur connaissance pratique. Cette étude souligne que, plus un utilisateur est expert, plus il utilise la quantité d'outils mis à sa disposition sur le Web. En revanche un utilisateur novice se limite souvent à un moteur de recherche.

La connaissance du domaine correspond à la connaissance que possède un utilisateur sur les thèmes relatifs à ses besoins. Elle permet une bonne formulation des requêtes (sélection des termes les plus appropriés pour trouver des documents pertinents) ainsi qu'une meilleure évaluation des documents visités *(Pejtersen & Fidel, 1998)*. Par exemple, le passage des besoins « mentaux » aux besoins « explicites » (sous forme de mots-clés) est un réel problème puisque le vocabulaire utilisé influence directement les résultats de la recherche. Des termes trop généraux risquent de générer un nombre de résultats trop important (probabilité d'obtenir un fort bruit). A l'inverse, des termes trop spécifiques risquent de générer un nombre de résultats faible voire nul (probabilité d'obtenir un fort silence). Par ailleurs, un vocabulaire en inadéquation avec les besoins réels de l'utilisateur risque donc tout simplement de produire des documents inadaptés aux besoins réels de l'utilisateur. *(Pejtersen & Fidel, 1998)* soulignent également le fait que les utilisateurs n'ayant pas, ou peu, de connaissances dans un domaine, ont du mal à définir les termes le caractérisant et à concevoir une stratégie de recherche. De plus, cette étude montre que les usagers ont du mal à évaluer si un document correspond ou non au thème recherché.

Concernant la recherche en elle-même, le nombre de documents est si important sur le Web qu'il n'est pas envisageable de stocker le contenu de tous les documents. C'est pour cela que les outils de recherche ad hoc sur le Web utilisent généralement une collection virtuelle de documents (ils ne conservent qu'un minimum d'informations concernant les documents comme son URL ou ses termes d'indexation). De plus, la structure hypertexte sur laquelle repose le Web peut être intégrée au niveau du processus d'indexation *(Li & Rafski, 1997) (Gery, 1999)*. Cette indexation repose généralement sur une indexation automatique grâce à des robots d'indexation nommés « crawlers » ou « spiders » qui parcourent le Web à la recherche de nouveaux documents à indexer. Le modèle de recherche communément utilisé est le modèle vectoriel.

De ces caractéristiques découlent des problèmes de la RI sur le Web. Par exemple, le fait d'utiliser une collection virtuelle implique que le moteur de recherche peut proposer des documents qui n'existent plus ou qui ont été modifiés. D'autres problèmes sont détaillés ci-après.

1.7.2 Les Problèmes de la Recherche d'Information sur le Web

Dans (GVU, 1998) les problèmes auxquels les utilisateurs peuvent être confrontés lorsqu'ils utilisent le Web sont soulignés. Ils concernent :

- l'impossibilité de trouver des informations recherchées (1),
- l'impossibilité d'organiser efficacement les informations retrouvées (2),
- l'impossibilité de trouver une page dont on connaît l'existence (3),
- l'impossibilité de revenir à un document déjà visité (4),
- l'impossibilité de déterminer où l'utilisateur se situe (perdu dans l'hyper-espace) (5),
- l'impossibilité de visualiser où l'utilisateur est allé, où il peut aller (visualisation de portions du site Web visité par exemple) (6),
- la rencontre de liens ne fonctionnant pas (liens morts) (7).

Cette étude montre qu'environ un quart des utilisateurs ne sont pas satisfaits de leur recherche dans le sens où ils ne retrouvent pas les informations recherchées (situations 1 et 3). De plus, l'utilisateur a du mal à organiser et à réutiliser les informations retrouvées (2). Quant aux problèmes 5 et 6, ils sont principalement dus à la surcharge cognitive qu'implique une navigation hypertexte. Le problème 7 provient directement de la structure du Web et surtout de l'évolution rapide à laquelle sont soumis les documents sur ce média (les auteurs modifient le contenu, déplacent les documents…). Cependant, cette étude ne permet pas d'identifier les réels problèmes de la recherche d'information sur le Web. La connaissance du domaine de recherche, le processus de recherche mais également la gestion des informations retrouvées met l'utilisateur face à un grand nombre d'écueils qu'il est important d'éviter. La principale limite de la RI correspond au facteur humain (connaissance pratique et connaissance du domaine) qui a un fort impact sur les résultats d'une RI.

1.7.3 Problèmes liés au Processus de Recherche sur le Web

Pour effectuer une recherche, l'utilisateur s'appuie sur une navigation hypertexte et sur une recherche ad hoc utilisées de façon alternée. Les problèmes liés à la tâche de recherche sont les suivants :

– la couverture du web. Les outils de recherche n'indexent qu'une partie limitée du Web *(Lawrence & Giles, 1999) (Sullivan, 2001) (Notess, 2002)*. Une première raison à cette restriction provient du nombre très important de documents disponibles. Une seconde raison est l'incapacité qu'ont les robots d'indexation à indexer les documents du Web caché. En effet, ces documents ne sont accessibles généralement que par le biais de formulaires que le robot ne peut pas automatiquement remplir *(Seltzer, 1997)*. Les robots ne se concentrent donc que sur le Web visible (portion limitée du Web global),

– le chevauchement des bases de documents des différents outils de recherche. Le chevauchement entre les bases d'indexation des différents outils de recherche est relativement faible *(Notess, 2000)*, ce qui signifie que chacun des outils a préalablement indexé des documents différents et que pour une même requête, il va retourner des documents différents des autres outils.

– la mise à jour des bases d'indexation des outils de recherche. Un facteur important dont pâtissent les outils de recherche est l'évolution rapide des documents. De ce fait, les outils de recherche n'ont pas une base d'indexation à jour et ils proposent à l'utilisateur un grand nombre d'URLs de documents déplacés, supprimés, voire obsolètes.

– la présentation des résultats à l'utilisateur. Même s'ils ne couvrent qu'une partie du Web, les outils de recherche indexent plusieurs millions de documents. De ce fait, suite à une recherche, l'utilisateur se retrouve souvent avec des milliers de documents pertinents (du point de vue système) pour ses besoins. Or, les outils actuels utilisent communément des listes de résultats pour présenter ces derniers. Ce mécanisme ne s'avère pas adapté à la compréhension du résultat dans sa globalité car une liste de résultats ne présente que quelques dizaines de résultats par page.

1.7.4 Approches Existantes pour l'Aide à la Recherche d'Information sur le Web

Outre ces difficultés de recherche liées à la technologie, d'autres difficultés d'ordre général liées à la gestion et à l'organisation des résultats retrouvés ont une incidence sur la RI.

1.7.4.1 Les Facteurs Humains

L'utilisateur doit posséder une bonne connaissance pratique et une bonne connaissance du domaine afin de réaliser des recherches efficaces. La connaissance pratique peut être améliorée au cours de formations, grâce à la lecture d'ouvrages spécifiques ou de sites Web. Contrairement à la connaissance pratique, la connaissance du domaine est sujette à une forte évolution. En effet, les centres d'intérêt de l'utilisateur évoluent au même titre que les informations qui s'y réfèrent. La solution idéale pour augmenter et affiner la connaissance d'un centre d'intérêt est de faire régulièrement des recherches sur le Web pour être au courant des dernières évolutions dans le domaine. Cependant, cette démarche demande de la part de l'utilisateur un lourd investissement.

Une alternative pour réduire en partie cet investissement, consiste à utiliser des outils basés sur le principe de PUSH ou autres agents intelligents. En effet, ces outils permettent de présenter automatiquement et de manière permanente des documents répondant aux centres d'intérêt de l'utilisateur. Par exemple, le méta-moteur de recherche ProFusion[4] propose à l'utilisateur un système d'alertes qui le prévient dès qu'une nouvelle information relative à ses centres d'intérêt apparaît au sein des bases d'indexation des moteurs utilisés. Ces avertissements sont effectués par courrier électronique.

1.7.4.2 Le Processus de Recherche

Plusieurs approches visent à améliorer la navigation au travers de :

- la réduction de l'effort cognitif nécessaire,
- l'aide à l'orientation.

D'autres approches sont liées à la recherche ad hoc au travers :

- des aides à la formulation des besoins,
- des aides à la sélection des outils de recherche,
- des méta-moteurs de recherche,
- des interfaces de visualisation des résultats de recherche,
- des agents de recherche et de recommandation.

[4]http://www.profusion.com/

1.7.4.2.1 La Tâche de Navigation

La navigation souffre essentiellement de problèmes provenant de l'architecture d'hypertexte sur laquelle elle repose. Ces problèmes sont la surcharge cognitive et la désorientation. Afin de limiter l'effort cognitif induit par l'hypertexte, divers outils ont été mis en œuvre pour garder une trace et un cheminement des documents visités. Les principaux navigateurs proposent un historique des différents documents visités lors de la navigation de l'utilisateur présenté sous forme d'une liste organisée par site, par jour ou encore par ordre de visite. Ainsi, l'utilisateur peut aisément revenir à un document précédemment visité dans l'hypertexte *(Hascoët, 2000)*. Une limite de la liste historique est qu'elle ne permet pas à l'utilisateur de s'affranchir totalement de l'effort cognitif lié à la navigation. En effet, les différents documents sont présentés de façon indépendante.

La visualisation de la navigation est une évolution des historiques. Elle permet de représenter graphiquement les documents visités ainsi que les liens éventuels entre ces documents *(Dömel, 1994)*. Grâce à de tels outils, l'utilisateur peut visualiser non seulement les documents qu'il a précédemment visités mais également l'organisation de ces documents. Pour éviter que l'utilisateur perde le cheminement de sa navigation, une cartographie de l'hypertexte local est proposée. Au lieu de ne présenter que les documents visités, ces cartographies visent à présenter les documents visités au sein de leur hypertexte local. Ainsi, les documents visités apparaissent avec les documents liés par des liens hypertextes afin que l'utilisateur puisse avoir une vision globale de l'hypertexte « local » dans lequel il se trouve *(Wood et al., 1995)*.

Grâce aux différentes approches proposées dans la littérature, la tâche de navigation peut être réalisée dans de meilleures conditions. Mais la RI sur le Web fait également appel à une tâche de recherche. La section suivante présente les approches visant à améliorer cette tâche.

1.7.4.2.2 La Tâche de Recherche

Même si la requête de l'utilisateur n'est pas optimale, diverses approches tentent de l'améliorer. En effet, les performances d'un SRI dépendent des requêtes formulées par l'utilisateur. Le plus souvent, il les formule en des termes qui lui sont propres, mais qui ne correspondent pas forcément à ceux utilisés pour indexer les documents pertinents de la base d'indexation du SRI. Pour sélectionner le maximum de documents pertinents en limitant le bruit, il faudrait que l'utilisateur formule ses besoins à partir de termes pertinents directement issus du langage d'indexation du système. Cette tâche s'avère difficile dans la mesure où, en règle générale, sur de gros corpus, il est impossible de connaître le langage d'indexation utilisé. Compte tenu des volumes croissants des bases d'informations, retrouver les informations pertinentes en utilisant seulement la requête initiale est une opération quasi-impossible. En conséquence, de nombreuses recherches passées et actuelles visent à concevoir des systèmes capables de s'adapter aux besoins de l'utilisateur (via le concept de profil par exemple). Ces systèmes sont capables de déterminer le but de la recherche afin de l'aider à cibler son besoin.

Le premier point à prendre donc en compte lors d'une RI, est la formulation des besoins sous la forme d'une requête. En effet, quel que soit l'outil de recherche utilisé, quel que soit le domaine, il est nécessaire pour l'utilisateur d'expliciter au mieux ses besoins pour obtenir des

résultats pertinents. Une inadéquation entre les besoins réels et la requête peut être la cause d'un grand nombre d'échecs par interrogation. De façon générale, la formulation de la requête repose sur l'utilisation d'un langage pseudo-naturel. Ainsi, l'utilisateur utilise des mots simples (mots-clés) visant à représenter ses besoins.

Les limites de cette formulation proviennent essentiellement du fait que :

– l'utilisateur n'a pas une bonne connaissance du domaine de recherche,

– l'utilisateur ne connaît pas le contenu a priori de la base d'indexation des outils de recherche,

– l'utilisateur peut avoir du mal à traduire la représentation mentale de ses besoins en une représentation sous la dorme d'une liste de mots-clés,

– l'utilisateur n'utilise que peu de termes pour représenter ses besoins. Différentes études *(Silverstein et al., 1998)*, *(Jansen et al., 2000)*, *(Spink et al., 2002)* montrent qu'en moyenne moins de trois mots-clés sont utilisés pour formuler la requête. Ce petit nombre de termes ne suffit généralement pas pour représenter un besoin en information de l'utilisateur et les méthodes utilisées pour l'indexation ne sont pas adaptées.

Une première approche est l'interrogation des outils de recherche par médiation. L'idée réside dans le fait que plutôt que de demander à l'utilisateur de formuler ses besoins sous la forme de mots-clés, celui-ci choisit dans un ensemble de classes de documents celles qui correspondent à ses besoins. A partir des classes sélectionnées, le système génère automatiquement la requête correspondante. Comme application de ce principe de médiation, le système WebCluster *(Mechkour et al., 1998)* permet de générer une requête qu'il propose soit directement à un outil de recherche, soit à l'utilisateur pour lui permettre de la modifier. Par ce biais, l'utilisateur peut s'affranchir de la formulation d'une requête.

Tout outil de recherche propose généralement à la fois un service de recherche adhoc basé sur une requête et une classification thématique des documents (par exemple Yahoo). L'utilisateur peut ainsi trouver dans cette approche une alternative à la recherche via une requête en parcourant les thèmes l'intéressant. Le système Cat-a-cone *(Hearst & Karadi, 1997)* est un exemple d'application qui offre à l'utilisateur la possibilité, soit de formuler une requête à l'aide d'un ensemble de mots-clés, soit de naviguer visuellement, à-travers d'une interface 3D, dans une hiérarchie de thèmes décrivant les documents de la base d'indexation.

Une autre approche de l'aide à la formulation de requêtes est également proposée au travers des outils de requêtes dynamiques par interaction. L'utilisateur peut interroger l'outil de recherche par tâtonnement. A partir d'une requête, il peut modifier ses composantes et visualiser immédiatement le résultat des modifications apportées.

On peut également citer la reformulation de requête. C'est un processus ayant pour objectif de générer une nouvelle requête plus adéquate que celle initialement formulée par l'utilisateur. Cette reformulation permet de coordonner le langage de recherche (utilisé par l'utilisateur dans sa requête) et le langage d'indexation du système. Par conséquent, elle limite le bruit et le silence dus à un mauvais choix des termes d'indexation dans l'expression de la requête d'une part, et les lacunes du processus d'indexation d'autre part. Il existe deux approches de reformulation de requêtes, (1) selon qu'elles utilisent les associations entre les termes, ou (2) la pertinence et non-pertinence des documents restitués en réponse à une requête initiale. Les deux principales techniques utilisées sont respectivement l'expansion de requête et la réinjection de la pertinence.

L'expansion de requête se base sur le fait que la simple comparaison du contenu de la requête et des documents de la base d'indexation ne permet pas d'avoir tous les documents correspondant à la requête. Il reste toujours des documents pertinents non restitués par le système. Des travaux de recherche ont proposé de reformuler la requête initiale par l'ajout des termes sémantiquement proches. Ces derniers sont issus :

– soit d'études sur le langage naturel (variantes morphologiques…). Il est ainsi possible d'ajouter à la requête des variantes morphologiques des différents termes employés par l'utilisateur. Le but de ce mécanisme est d'assurer la restitution des documents indexés par des variantes des termes composant la requête. Dans ce cadre, des algorithmes de racinisation et de troncature sont utilisés,

– soit d'études statistiques et d'analyses sur les contenus des documents de la base. On peut ainsi choisir d'ajouter un certain nombre de termes les plus pertinents des documents sélectionnés, ou de n'en conserver qu'un nombre limité parmi les termes initiaux et rajoutés. On peut également proposer d'ajouter des termes voisins ou des termes associés à ceux de la requête. Il s'agit de chercher des associations inter-termes (corrélation entre les termes, classification des termes…).

La reformulation de requêtes par la réinjection de la pertinence (ou « relevance feedback » *(Salton & McGill, 1983)*) peut être utilisée lorsque les documents sont restitués en réponse à une requête initiale formulée par un utilisateur. C'est un processus évolutif et interactif. Son principe fondamental est d'utiliser la requête initiale pour amorcer la recherche, puis modifier celle-ci à partir des jugements de pertinence et/ou non pertinence de l'utilisateur sur les documents restitués, soit pour repondérer les termes de la requête initiale, soit pour y ajouter (resp. supprimer) d'autres termes contenus dans les documents pertinents (resp. non pertinents). La nouvelle requête, obtenue à chaque itération du feedback, permet de corriger la direction de la recherche dans le sens des documents pertinents. Le processus de réinjection de pertinence peut être adapté aux différents modèles de recherche comme le modèle vectoriel avec l'approche de *(Rocchio, 1971)*, le modèle connexionniste *(Boughanem et al., 2000)* mais peut également être réalisé par l'approche des algorithmes génétiques *(Tamine, 2000)*.

Après avoir pris conscience de la façon dont ses besoins peuvent être formulés, l'utilisateur a la lourde tâche de sélectionner l'outil de recherche qu'il souhaite interroger. Les différents outils de recherche (moteurs de recherche, annuaires) disposent, au sein de leur base d'indexation, des mêmes documents. Ceci se traduit par un faible recouvrement des bases d'indexation. La sélection de l'outil de recherche conditionne les résultats de la recherche : un outil de recherche généraliste pour une requête dans un domaine spécifique donnera vraisemblablement des résultats moins « bons » que la même requête posée sur un outil de recherche spécialisé dans le domaine. L'utilisateur doit donc choisir au mieux l'outil qu'il va interroger. Cependant, quel que soit l'outil de recherche interrogé, et du fait de cette faible couverture des bases d'indexation, l'utilisateur ne peut se contenter d'interroger un seul outil de recherche.

Ainsi, un problème de la recherche ad hoc réside dans le fait de bien savoir choisir les outils de recherche pour obtenir les meilleurs résultats puis de réaliser une synthèse des résultats obtenus. Si elle est réalisée manuellement, cette tâche s'avère longue et fastidieuse surtout si chacun des outils de recherche retourne des milliers de documents, ce qui est assez courant à l'heure actuelle. Cette interrogation multiple est d'autant plus difficile à réaliser que chacun des outils de recherche propose son propre langage de requête et qu'il est nécessaire d'intercaler

manuellement les résultats fournis par chaque moteur. Pour l'aider dans cette tâche, l'utilisateur peut faire appel à un méta-moteur de recherche d'information.

1.7.4.2.3 Les Méta-Moteurs de Recherche d'Information

Un méta-moteur de recherche d'information se présente à l'utilisateur comme un outil de recherche « classique ». Cependant, à partir d'une requête, le système crée une multitude de requêtes qu'il soumet en parallèle à un ensemble d'outils de recherche prédéfinis. Chacune d'elles correspond à la traduction de la requête initiale dans le langage spécifique de l'outil interrogé. Du point de vue du résultat obtenu au travers de ces outils, les méta-moteurs peuvent être classés en diverses catégories *(Andrieu, 1998)* :

– les aides à la saisie. Ces outils proposent uniquement une traduction de la requête initiale pour un ensemble d'outils de recherche prédéfinis. Cependant les différents outils de recherche restent indépendants. L'utilisateur doit manuellement procéder à l'interrogation des outils de recherche (par exemple *MetaSearch*[5]),

– les listes de résultats. Ces outils soumettent la requête originale parallèlement à un ensemble d'outils de recherche. Les résultats sont présentés pour chaque outil de recherche indépendamment (par exemple *Internet Sleuth*[6]),

– les listes synthétisées. Ces outils sont des listes de résultats qui fusionnent les résultats issus des différents outils interrogés en une seule liste de résultats. Les résultats en double sont supprimés et les résultats réorganisés. On peut citer l'outil *Copernic*[7] ou encore le service en ligne *MetaCrawler*[8].

Les deux premières catégories de méta-moteurs tendent aujourd'hui à disparaître au profit des listes synthétisées qui fournissent à l'utilisateur un résultat synthétique. Cependant, la plupart des outils disponibles interrogent toutes les sources sélectionnées sans, à priori, vérifier si ces sources sont pertinentes pour la requête. Certains problèmes liés à la RI persistent donc. En effet, comment l'internaute réagit-il face à un nombre de résultats dépassant le millier voire le million de documents? Ce cas est plus que fréquent sur Internet. Différentes études *(Silverstein et al., 1998)*, *(Spink et al., 2002)* montrent que l'utilisateur ne parcourt que les premières pages de résultats (environ 30 documents) alors que des documents intéressants peuvent se situer au-delà. Il est donc important de prendre en compte la présentation à l'utilisateur des résultats de recherche (des millions de documents) dans le processus de la Recherche d'Information pour optimiser et faciliter la tâche de l'internaute. En réponse à cela, nous pouvons souligner l'intérêt des approches qui visent à étudier les résultats pour personnaliser la réponse proposée à l'utilisateur.

Ainsi, le projet *Profildoc (Lainé-Cruzel, 1999)* permet de filtrer les documents retrouvés par rapport au profil de l'utilisateur. Ce système repose sur un profil utilisateur contenant des informations telles que le niveau éducationnel, le champ disciplinaire, le type de recherche. Ce profil est utilisé afin d'identifier au sein des documents retrouvés ceux qui correspondent au profil de l'utilisateur et ainsi réduire le nombre de documents en éliminant ceux qui ne seraient pas pertinents pour l'utilisateur.

[5] http://www.metasearch.com/
[6] http://www.isleuth.com
[7] http://www.copernic.com
[8] http://www.metacrawler.com/index.html

Malgré tout, le nombre de résultat reste très important et il peut s'avérer intéressant de présenter les résultats de façon globale au travers d'une interface de visualisation.

1.7.4.3 La Visualisation des Résultats

Les résultats issus d'un moteur ou encore d'un méta-moteur de recherche d'information sont communément présentés sous la forme d'une liste. Cette liste présente les différents résultats au travers:

– d'un numéro de classement ou une appréciation de la pertinence système,

– d'un nom ou d'une URL associé à un lien hypertexte pour permettre à l'utilisateur d'accéder au document,

– d'un court descriptif présentant les premières lignes du document dans lesquelles apparaissent les termes de la requête.

Cet affichage est facile à mettre en oeuvre et à utiliser mais n'est efficace que pour un nombre réduit de résultats (< 20) *(Cugini et al., 2000)*. Pour un nombre important, les listes de résultats souffrent essentiellement des limites suivantes *(Dubin, 1995)*, *(Zamir, 1998)* :

– la position d'un document dans la liste ne permet pas explicitement de déduire sa similarité avec les autres documents de la liste,

– l'utilisateur peut avoir du mal à comprendre pourquoi le document a été inséré à cet endroit dans la liste et quelle est la relation avec la requête sans avoir visualisé son contenu.

A cela s'ajoute le fait que les résultats soient affichés page par page, ce qui ne facilite pas la vision globale du résultat. Pour obtenir une vision globale des résultats, l'utilisateur doit visiter chacun des documents, un à un, pour en apprécier la réelle pertinence et identifier les liens potentiels entre ceux-ci. De ce fait, il est compréhensible que l'utilisateur se contente en moyenne des 30 premiers résultats en occultant le reste des résultats (éventuellement pertinents) si le nombre de résultats est très important. Cependant, il occulte également un ensemble de documents potentiellement pertinents pour ses besoins.

Pour remédier à cet état de fait, des travaux ont été réalisés dans le domaine des interfaces de visualisation. *(Zamir, 1998)* dresse une classification des techniques de visualisation. Cette classification met en évidence deux types de techniques selon leur but :

– les visualisations des attributs des documents (informations issues des documents). Il y a trois sous-catégories de techniques.
  - la distribution des termes de la requête. Elle permet de savoir comment chaque mot-clé utilisé dans la requête est réparti dans les documents,
  - les attributs prédéfinis. Elle permet de montrer la relation qu'a le document avec des attributs tels que la taille, l'auteur, etc.,
  - les attributs formulés par l'utilisateur. Elle permet de montrer la relation qu'a le document avec des critères choisis par l'utilisateur (requête par exemple…),

– les visualisations de similarité inter-documents. Il y a quatre sous-catégories.
  - les réseaux de documents. Les documents sont reliés entre eux selon leur similarité,

- « les affichages pondérés ». Les documents sont répartis visuellement selon des forces qui les repoussent ou les rapprochent des autres par rapport à leur similarité,
- les « classifications ». Ces visualisations représentent les documents sous forme de groupes de documents (par similarité de contenu, selon les liens hypertextes…),
- les cartes auto-organisatrices. Ces techniques permettent d'afficher sur une « carte » 2D les documents par rapport à leur similarité de contenu.

1.7.4.4 Les Agents

Les approches précédentes reposent généralement sur la méthode PULL. Or, cette méthode d'accès à l'information nécessite une implication continue et importante de l'utilisateur.

Un agent est communément défini comme « *une personne chargée des affaires et des intérêts d'un individu* » (*Petit Robert*). L'application informatique du concept d'agents respecte cette définition et est caractérisée par plusieurs aspects *(Caglayan & Harrison, 1998)*. Les agents de recherche et les agents de recommandation interviennent dans le contexte de l'aide à la RI sur le Web. Les systèmes multi-agents peuvent apporter de la valeur ajoutée dans l'aide à la recherche.

1.7.4.4.1 Les Agents de Recherche

La catégorie des agents de recherche comprend les méta-moteurs de recherche d'information de dernière génération (listes synthétisées) jusqu'aux outils de recherche off-line (*Copernic* par exemple). Ces derniers permettent à l'utilisateur d'initier des recherches d'information qui s'exécuteront même lorsque l'utilisateur ne sera plus connecté à Internet.

1.7.4.4.2 Les Agents de Recommandation

Les agents de recommandation visent à optimiser la RI de l'internaute en lui proposant automatiquement de nouveaux documents au regard de ses besoins ou de ses actions. Ils reposent essentiellement sur une approche PUSH proposant des informations à l'utilisateur et une caractérisation des besoins au moyen d'un profil utilisateur. Les systèmes *Letizia (Lieberman, 1995)*, *WebWatcher (Armstrong et al., 1995)* ou encore *BroadWay (Jaczynski & Trousse, 1997)* sont de bons représentants de cette catégorie. Certains agents peuvent également prendre en considération l'environnement de recherche de l'utilisateur (ensemble des applications) pour déduire ses besoins et lui recommander des documents qui ne sont pas forcément reliés (par des liens hypertextes) aux documents visités.

1.7.4.4.3 Approches Multi-Agents

Les systèmes multi-agents visent à faire coopérer une série d'agents afin d'obtenir un résultat. Ainsi, dans le cadre de la RI, il peut être intéressant de faire coopérer les agents pour répondre à des besoins précis en information d'un utilisateur. Par exemple, l'approche multi-agents proposée au travers du projet *Marvin*[9] permet de construire des collections thématiques de documents. Chaque agent parcourt le Web à la recherche de documents pour un thème donné. L'intérêt des approches multi-agents peut également être vu au travers du projet *Abrose (Carré et al., 1999)* qui permet de construire et de maintenir un profil utilisateur pour le commerce électronique mais qui pourrait être adaptée à la RI. Ce profil utilisateur contient l'ensemble des préférences d'utilisation d'Internet d'un utilisateur qui peut être exploité pour répondre à des requêtes de façon plus fine ou proposer à l'utilisateur de nouveaux documents ou offres plus intéressantes.

1.8 Applications

Nous présentons, ici, les types d'utilisation principaux d'un système de gestion des documents électroniques.

– ***La veille technologique*** : cherche à étudier l'information accessible afin de :
  - déterminer les secteurs en développement, ceux où la demande est la plus forte,
  - étudier l'évolution de la demande (marchés moins porteurs, problèmes émergeants, etc.),
  - évolutions dans un domaine particulier (problèmes liés à un programme, apparition de nouveaux virus, etc.).

Un outil de veille technologique automatique (autre qu'une simple vérification de la modification de telle ou telle page) doit utiliser un moteur de recherche documentaire.

– ***Le résumé automatique*** : même si on se limite à un domaine précis, la quantité d'information disponible peut être trop importante pour être gérée manuellement. Il est alors nécessaire de réduire cette information tout en en gardant la substance. Un module de résumé automatique peut être utilisé dans tous les types de traitement de l'information afin d'aider l'utilisateur dans sa tâche.

– ***Le filtrage d'information*** : le filtrage permet, dans un flux d'information (comme les communiqués de presse ou messages sur des forums), de ne garder que les documents traitant d'un sujet très précis ou d'orienter chaque document vers un forum particulier. Le filtrage de l'information consiste à évaluer la similarité entre un document et un certain besoin.

– ***Réponse à une question*** : dans la majorité des cas, un utilisateur recherche une information plutôt qu'un document, mais il accepte qu'un système lui renvoie une liste de documents dans lesquels il est supposé trouver l'information dont il a besoin. Mais, étant donné la taille des bases textuelles disponibles actuellement comme Internet, les utilisateurs peuvent

[9] http://www.hon.ch/Project/Marvin_project.html

préférer une réponse à leur question plutôt qu'une liste de documents. Par conséquent, les systèmes de réponse à une question sont actuellement en pleine expansion et constituent une évolution logique des services offerts par un système de gestion automatique de l'information documentaire *(Strzalkowski et al., 2000)*. Ils sont subordonnés à un système de recherche documentaire qui trouve les documents pertinents par rapport à une question.

1.9 Conclusion

Le développement d'Internet au niveau mondial a profondément transformé la gestion des documents. Cette révolution technologique a engendré de nouvelles problématiques documentaires pour la RI. Ce chapitre a permis de présenter un court historique de l'Internet et du Web mais également des difficultés rencontrées par les utilisateurs lors de leur recherche d'information sur ce média. La recherche documentaire et ses fondements ont été détaillés. Nous avons vu ce que peut être un SRI. Nous avons rappelé ce qu'est l'indexation dans ces systèmes, c'est-à-dire un processus de projection des documents ou requêtes dans un espace de représentation. Après avoir donné une définition de l'espace de représentation et de ses caractéristiques, nous avons présenté les différents modèles de SRI et leur type d'indexation utilisé. La notion d'agent a également été abordée. En résumé, les moteurs de recherche permettent d'accéder à un grand nombre de documents par une recherche basée principalement sur les mots du texte, les annuaires eux proposent un nombre restreint de documents avec une recherche axée sur la précision des indexations, le but n'étant pas de tout dire sur un document, mais d'orienter la lecture du document.

Dans le Chapitre 2, nous présentons le catalogue de santé CISMeF qui est accessible sur le Web. Nous y présentons les améliorations apportées au cours de ces dernières années en matière d'intégration des éléments présentés dans ce Chapitre 1, le but étant de guider au mieux l'utilisateur dans sa quête d'information dans le domaine de la santé.

BIBLIOGRAPHIE

(*Aas & Eikvil, 1999*) AAS K. & EIKVIL L. (1999). Text Categorization : a Survey. Technical report, *Norwegian Computing Center*.

(*Abel, 199 3*) ABEL, Y. (1993). Indexation Automatique et Traitement du Langage Naturel. *Rapport de DEA Contrôle des Systèmes*, Université de Compiègne.

(*Agosti & Melucci, 2000*) AGOSTI M. & MELUCCI M. (2000). Information Retrieval on the Web. *Lecture Notes in Information Retrieval*. pp.243-285.

(*Agosti & Smeaton, 1996*) AGOSTI M. & SMEATON A. (1996). Information Retrieval and Hypertext, *Kluwer Academic Publisher*.

(*Andrieu, 1998*) ANDRIEU O. (1998) Trouver l'Info sur Internet. *Eyrolles*.

(*Apté et al., 1994*) APTE C., DAMERAU F. J., WEISS S. M. (1994). Automated Learning of Decision Rules for Text Categorization. *ACM Transactions on Information Systems*, 12(3):233–251.

(*Armstrong et al., 1995*) ARMSTRONG R., FREITAG D., JOACHIMS T. (1995). Webwatcher : Machine Learning and Hypertext. *Proceedings of the 1995 AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments*.

(*Baeza-Yates & Ribeiro-Neto, 1999*) BAEZA-YATES R. & RIBEIRO-NETO B. (1999) Modern Information retrieval. *CM Press Books, Addison-Wesley*.

(*Belkin & Croft, 1992*) BELKIN N. & CROFT B. (1992). Information Filtering and Information Retrieval: Two sides of a Same Coin ? *Communications of the ACM,* 35(12) :29-38.

(*Berckley et Al., 1994*) BUCKLEY, C., SALTON, G., ALLAN, J., AND SINGHAL, A. (1994). Automatic Query Expansion using SMART : TREC 3. In *Text REtrieval Conference*.

(*Bergman, 2000*) BERGMAN MK. (2000). The Deep Web: Surfacing the Hidden Value, *BrightPlanet ed.*

(*Berners-lee et al., 1994*) BERNERS-LEE T., CAILLIAU R., NIELSEN H.F., SECRET A. (1994). The World Wide Web. *Communications of the ACM*, 37(8).

(*Boughanem et al., 2000*) BOUGHANEM M., CHRISMENT C., SOULÉ-DUPUY C., TAMINE L. (2000). Connectionnist and Genetic Approaches to Achieve IR. *Soft Computing in Information Retrieval Techniques and Applications*, F. Crestani & G. Pasi (ed.), pp 173-198.

(*Bourigault & Condamines, 1998*) BOURIGAULT D. & CONDAMINES A. (1998). Terminologies et Ingénierie des Connaissances. In *Bulletin de l'AFIA N°32*. pp. 19-24.

(*Bray & Sperberg-Mc Queen, 1996*) BRAY T. & SPERBERG-MC QUEEN CM. (1996) Extensible Markup Language (XML), *W3C Working Draft*, WD-xml-961114.

(*Bruza & Lalmas, 1995*) BRUZA, P.D. & LALMAS M. (1995). Logic-Based Information Retrieval: Is it really worth it? *Working Notes of Workshop on the treatment of Uncertainty in Logic-based Models of Information Retrieval Systems*.

(*Caglayan & Harrison, 1998*) CAGLAYAN A. & HARRISON C. (1998). Les Agents, *InterEditions*.

(*Caropreso et al., 2001*) CAROPRESO, M. F., MATWIN, S., AND SEBASTIANI, F. (2001). A Learner-Independent Evaluation of the Usefulness of Statistical Phrases for Automated Text Categorization. *Text Databases and Document Management : Theory and Practice*, pp. 78–102.

(*Carré et al., 1999*) CARRE J., MACHONIN A., GLIZE P. (1999). Un Système Multi-agents Auto-organisateur pour l'Apprentissage d'un Profil Utilisateur. *7èmes journées francophones d'Intelligence Artificielle et Systèmes Multi-Agents* (*JFIADSMA'99*), pp 207-221.

(*Cavnar & Trenkle, 1994*) CAVNAR,W.B., TRENKLE, J. M. (1994) N-gram-based Text Categorization. In Proceedings of *SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, pp.161–175.

(*Chi, 2000*) CHI ED. H. (2000) A Taxonomy of Visualization Techniques Using the Data State Reference Model. *INFOVIS'2000*, pp 69-75.

(*Cugini et al., 2000*) CUGINI J.V., LASKOWSKI S., SEBRECHTS M. (2000) Design of 3-D Visualisation of Search Results: Evolution and Evaluation », *12th International Symposium on Electronic Imaging: Visual Data Exploration & Analysis (SPIE 2000),* pp 198-210.

(*De Loupy*, 2000) DE LOUPY. (2000) Evaluation de l'Apport de Connaissances Linguistiques en Désambiguïsation Sémantique et Recherche Documentaire. *Thèse de l'Université d'Avignon et des Pays du Vaucluse.*

(*De Loupy, 2001*) DE LOUPY C. (2001) L'apport de Connaissances Linguistiques en Recherche Documentaire. Actes *conférence Traitement Automatique du Langage Naturel (TALN'01).*

(*Deerwester et al., 1990*) DEERWESTER S., DUMAIS S., LANDAUER T., FURNAS G., HARSHMAN R. (1990). Indexing by Latent Semantic Anlysis. *Journal of the American Society of Information Science*, 416(6) :391–407.

(*Dömel, 1994*) DÖMEL P. (1994) Webmap - a Graphical Hypertext Navigation Tool, *2nd International World Wide Web Conference (WWW'1994).*

(*Dubin, 1995*) DUBIN D. (1995) Document Analysis for Visualization. *18th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp 199-204.

(*Dumais et al., 1998*) DUMAIS S., PLATT J., HECKERMAN D., SAHAMI M. (1998) Inductive Learning Algorithms and Representations for Text Categorization. *Proceedings of CIKM-98, 7th ACM International Conference on Information and Knowledge Management,* pp.148–155.

(*Dumais, 1991*) DUMAIS, S. (1991). Improving the Retrieval of Information from External Sources. *Behavior Research Methods, Instruments & Computers*, 23(2) :229–236

(*Fayet-Scribe, 1997*) FAYET-SCRIBE S. (1997) Chronologie des Supports, des Dispositifs et des Outils de Repérage de l'Information; Le savoir et ses outils d'accès : repères historiques. *Dossier Solaris, No. 4.*

(*Fluhr, 1992*) FLUHR C. (1992) Le Traitement du Langage Naturel dans la Recherche d'Information. In *Interface intelligente dans l'information scientifique et technique (INRIA)*. pp.103-130.

(*Fuhr & Buckley, 1991*) FUHR, N., BUCKLEY, C. (1991) A probabilistic learning Approach for Document indexing. In *ACM Transactions on Information System*. Vol. 9, N°3. pp.223-248

(Gery, 1999) GERY M. (1999) « Smartweb : recherche de zones de pertinence sur le world wide web », Actes du *17ème Congrès INFORSID*.pp 133-147.

(*Gilli, 1988)* GILLI Y. (1988) Texte et Fréquence. *Université de Besançon, Paris*.

(*Grefenstette et al., 2000*) GREFENSTETTE G., HULL D., ROUX C. (2000) Recherche d'Information en Français et Traitement Automatique des Langues. *TAL,* 41(2):473-493.

(*GVU, 1998*) 10th WWW User Survey, Graphic, visualisation & usability center (GVU).

(*Halleb & Lelu, 1997*) HALLEB M. & LELU A. (1997) Hypertextualisation automatique multilingue à partir des fréquences des n-grammes. *Hypertextes et hypermédias*. Vol. 1, N°2-3-4. pp.275-287.

(*Hascoët & Beaudouin-Lafon, 2001*) HASCOËT M. & BEAUDOUIN-LAFON M. (2001) Visualisation interactive d'information, *Revue Information-Interaction-Intelligence (I3).*

(*Hascoët, 2000*) HASCOËT M. (2000) A user interface combining navigation aids. *11th International ACM Hypertext Conference*. pp.224-225.

(*Hearst & Karadi, 1997*) HEARST M.A. & KARADI C. (1997). Cat-a-cone: an interactive interface for specifying searches and viewing retrieval results using a large category hierarchy. *20th International ACM SIGIR Conference on Research and Development in Information Retrieval*. pp.246-255.

(*Höschler & Strube, 2000*) HÖLSCHER C. & STRUBE G. (2000). Web search behavior of internet experts and newbies. *9th International Conference on the World Wide Web*.

(*Hull, 1996*) HULL D. A. (1996) Stemming algorithms : A case study for detailed evaluation. *Journal of the American Society of Information Science,* 47(1) :70–84.

(*Jacquemin C., 1998*) Analyse et inférence de terminologie. In *Revue d'Intelligence Artificielle*, 1998, Vol. 12, N°2. pp. 163-205.

(*Jaczynski & Trousse, 1997*) JACZYNSKI M. & TROUSSE B. (1997) Broadway: a world wide web browsing advisor reusing past navigations from a group of users. *3rd UK Case-Based Reasoning Workshop (UKCBR'97).*

(*Jansen et al., 2000*) JANSEN B.J., SPINK A., SARACEVIC T. (2000) Real life, real users, and real needs: a study and analysis of user queries on the web. *Information Processing and Management*. Vol 36, pp.207-227.

(*Julien, 1988*) JULIEN C. (1988) Bases d'informations généralisées : contribution à l'étude des mécanismes de consultation d'objets multimédia. *Thèse de Doctorat Toulouse III*.

(*Kahle , 1996*) KAHLE B. (1996) Archiving the internet. *Scientific American.*

(*Korfhage, 1997*) KORFHAGE R.R. (1997) Information storage and retrieval. *Wiley Computer Publishing*.

(*Lainé-Cruzel, 1999*) LAINE-CRUZEL S. (1999) Profildoc – Filtrer une information exploitable. *Bulletin des Bibliothèques de France* (BBF), 44(5), pp. 60-64.

(*Lalmas, 1995*) LALMAS, M. (1995) From a Qualitative towards a Quantitative Representation of Uncertainty in a Situation Theory based model of an Information Retrieval System. *Working Notes of the Workshop on the treatment of Uncertainty in Logic-based Models of Information Retrieval Systems*.

(*Lalmas, 1998*) LALMAS, M. (1998) Logical Models in Information Retrieval: Introduction and Overview. In *Information Processing and Management*. Vol. 34, N°1. pp. 19-33 .

(*Lawrence & Giles, 1999*) LAWRENCE S., GILES L. (1999) Accessibility of information on the web », *Revue Nature*. Vol. 400, pp. 107-109.

(*Le Loarer, 1994*) LE LOARER P. (1994) Indexation automatique, recherche d'information et évaluation. In *Cours INRIA Le traitement électronique du document*. pp 149-201.

(*Lefèvre, 2000*) LEFEVRE, P. (2000). La recherche d'information - du texte intégral au thésaurus. *Hermès Science*.

(*Lewis & Ringuette, 1994*) LEWIS DD. & RINGUETTE M. (1994). A comparison of two learning algorithms for text categorization. *SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, pp.81–93.

(*Lewis, 1992a*) LEWIS, DD.(1992). An evaluation of phrasal and clustered representations on a text categorization task. *SIGIR-92, 15th ACM International Conference on Research and Development in Information Retrieval*, pp.37–50.

(*Lewis, 1992b*) LEWIS, DD. (1992). Representation and learning in information retrieval. *PhD thesis, Department of Computer Science, University of Massachusetts*.

(*Li & Jain, 1998*) LI Y.H. & JAIN A. K. (1998). Classification of text documents. *The Computer Journal*, 41(8) :537–546

(*Li & Rafski, 1997*) LI Y. & RAFSKI L. (1997). Beyond relevance ranking: hyperlink vector voting", *5th International Conference on Computer Assisted Information Retrieval, RIAO'97*, pp.648-651.

(*Lieberman, 1995*) LIEBERMAN H. (1995). Letizia: an agent that assists web browsing. *International Joint Conference on Artificial Intelligence (IJCAI'95)*.

(*Luhn, 1958*) LUHN HP. (1958). The automatic creation of literature abstracts », *IBM Journal of Research and Development,* 2(2), pp.159-165.

(Mechkour et al., 1998) MECHKOUR M., HARPER D.J., MURESAN G. (1998). The webcluster project: using clustering for mediating access to the world wide web. *21st International ACM SIGIR Conference on Research and development in Information Retrieval*, pp. 357-358.

(*Miller et al., 1999*) MILLER, E., SHEN, D., LIU, J., NICHOLAS C. (1999). Performance and Scalability of a Large-Scale N-gram Based Information Retrieval System. *Journal of Digital Information,* 1(5).

(*Mladenic & Grobelnik, 1998*) MLADENIC D. & GROBELNIK M. (1998). Word sequences as features in text-learning. *In Proceedings of ERK-98, the Seventh Electrotechnical and Computer Science Conference*, pp.145–148.

(Moeers, 1950) MOEERS C. (1950) The theory of digital handling of non-numerical information and its applications to machine economics. *Technical bulletin No. 48*. Cambridge University.

(*Mooers, 1960*) MOOERS C. (1960) Mooers' Law or, Why Some Retrieval Systems Are Used and Others Are Not; *American Documentation*, 11(3).

(*Moulinier, 1996*) MOULINIER I (1996). Une approche de la catégorisation de textes par l'apprentissage symbolique. *Thèse de Doctorat, Université Paris 6*.

(*Moulinier, 1997*) MOULINIER I (1996) Feature selection : a useful preprocessing step. *BCSIRSG-97, the 19th Annual Colloquium of the British Computer Society Information Retrieval Specialist Group*.

(*Murray & Moore, 2000*) MURRAY BH. & MOORE A. (2000) Sizing the Internet: A white paper, *Cyveillance ed*.

(*Ng et al., 1997*) NG H.T., GOH W.B., LOW K.L. (1997). Feature selection, perception learning, and a usability case study for text categorization. *SIGIR-97, 20th ACM International Conference on Research and Development in Information Retrieval*, pp.67–73.

(*Notess, 2000*) NOTESS G.R. (2000) Search engine statistics: database overlap. *SearchEngineShowdown*.

(*Notess, 2002*) NOTESS G.R. (2002) Search engine statistics: database total size estimates. *Search Engine Showdown*.

(*O'Neill et al., 1998*) O'NEILL E., LAVOIE B., MCCLAIN P. (1998) Characterizing the web and web-accessible information. *Report of the Web characterization workshop, W3C web characterization group conference*.

(*Pejtersen & Fidel, 1998*) PEJTERSEN A.M. & FIDEL R. (1998) A framework for centered evaluation and design: a case study of information retrieval on the web. Working Paper for *MIRA Workshop*.

(*Peters, 2000*) PETERS C. (2000) Introduction. *Workshop of the Cross-Language Evaluation Forum (CLEF), Lecture Notes in Computer Science #2069*. pp.1-6.

(*Porter, 1980*) PORTER M.F. (1980) An algorithm for suffix stripping. *Program*, Vol. 1(3), pp.130-137.

(*Rijsbergen, 1979*) VAN RIJSBERGEN, CJ. (1979). Information Retrieval, 2nd edition. *Dept. of Computer Science, University of Glasgow.*

(*Rijsbergen, 1986*) RIJSBERGEN, C.J. (1986) A new theorical framework for information retrieval . In *Proceedings of the 9th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* pp.194-200

(*Rocchio, 1971*) ROCCHIO, J. (1971) Relevance feedback in information retrieval. *The SMART Retrieval System Experiments in Automatic Document Processing*, pp.313–323.

(*Roussey, 2001*) ROUSSEY C. (2001) Une méthode d'indexation sémantique adaptée aux corpus multilingues ; *Thèse d'Informatique de l'INSA de Lyon.*

(*Rücker & Polanco, 1997*) RÜCKER J. & POLANCO M.J. (1997) Siteseer: personalized navigation for the web. *Communications of the ACM*, 40(3), pp.73-75.

(*Sable & Hatzvassiloglou, 2000*) SABLE C. L. & HATZIVASSILOGLOU V. (2000) Text-based approaches for non-topical image categorization. *International Journal of Digital Libraries*, 3(3) :261–275.

(*Sahami, 1999*) SAHAMI M. (1999) Using Machine Learning to Improve Information Access. *PhD thesis, Computer Science Department, Stanford University.*

(*Salton & Buckley, 1990*) SALTON G. & BUCKLEY C. (1990) Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science,* Vol.41, n°4, pp.288-297.

(*Salton & McGill, 1983*) SALTON G. & MCGILL M. (1983). Introduction to Modern Information Retrieval. *McGraw-Hill, New York.*

(*Salton et al., 1975*) SALTON G., WONG A., YANG C.S. (1975) A Vector Space Model for Automatic Indexing. *Communication of the ACM,* Vol. 18, N°11. pp.613-620.

(*Salton, 1988*) SALTON, G. (1988) A Simple Blue Print for Automatic Boolean Query Processing. *Information Process Management,* Vol. 24, N°3. pp.269-280.

(*Schmid, 1994*) SCHMID H. (1994) Probabilistic part-of-speech tagging using decision trees. In *International Conference on New Methods in Language Processing*.

(*Schütze et al., 1995*) SCHÜTZE H., HULL D.A., PEDERSEN J.O. (1995) A comparison of classifiers and document representations for the routing problem. *Proceedings of SIGIR-95, 18th ACM International Conference on Research and Development in Information Retrieval*, pp.229–237.

(*Sebastiani, 2002*) SEBASTIANI, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys,* 34(1) :1–47.

(*Seltzer, 1997*) SELTZER R. (1997) Altavista, Understanding the limits of accuracy. *Internet Search Advantage, Cobb group publishing, ZD Journal.*

(*Shannon, 1948*) SHANNON C. (1948) The Mathematical Theory of communication. *Bell System Technical Journal,* 27 :379–423 et 623–656.

(*Shivakumar & Garcia-Molina, 1998*) SHIVAKUMAR N. & GARCIA-MOLINA H. (1998) Finding near-replicas of documents on the web, *International workshop on the web and databases* (WebDB).

(*Shneiderman, 1998*) SHNEIDERMAN B. (1998) Designing the user interface. *Addison-Wesley.*

(*Silverstein et al., 1998*) SILVERSTEIN C., HENZINGER M., MARAIS H., MORICZ M. (1998) Analysis of a very large web search engine query log. *SRC technical note #1998- 014.*

(*Smeaton, 1989*) SMEATON AF. (1989) Information retrieval and natural language processing. In *proceedings of a conference jointly sponsored by ASLIB, University of York,* pp.2.

(*Soulé-Dupuy, 2001*) SOULE-DUPUY C. (2001) Bases d'informations textuelles : des modèles aux applications. *Habilitation à Diriger des Recherches, Université Toulouse III.*

(*Spark Jones, 1972*) SPARK-JONES K. (1972) A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation.* Vol. 28(1), pp 11-20.

(*Spink et al., 2002*) SPINK A., JANSEN B.J., WOLFRAM D., SARACEVIC T. (2002) From e-sex to e-commerce: web search changes. *IEEE Computer.* Vol. 35 (3), pp.107-109.

(*Strzalkowski et al., 2000*) STRZALKOWSKI T., STEIN G. C., WISE GB., BAGGA A. (2000) Towards the next generation information retrieval. In *6eme Conférence de Recherche d'Information Assistée par Ordinateur (RIAO'2000)*. pp.1196–1207.

(*Strzalowski, 1993*) STRZALKOWSKI T. (1993) Natural language processing in large-scale text retrieval tasks. In *Text REtrieval Conference (TREC-1)*, pp.173.

(*Sullivan, 2001*) SULLIVAN D. (2001) Search engines sizes. *The search engine report*.

(*Tamine, 2000*) TAMINE L. (2000) Optimisation de requêtes dans un système de recherche d'information. *Thèse de l'Université Paul Sabatier*.

(*Turtle & Croft, 1991*) TURTLE H. & CROFT BW. (1991) Evaluation of an inference Network-Based Retrieval Model. In *ACM Transactions on Information System*, Vol. 9, N°3. pp.187-222.

(*Tzeras & Hartman, 1993*) TZERAS K. & HARTMANN S. (1993) Automatic indexing based on Bayesian inference networks. *Proceedings of SIGIR-93, 16th ACM International Conference on Research and Development in Information Retrieval*, pp.22–34.

(*Voohrees & Harman, 2001*) VOOHREES EM. & HARMAN D. (2001) Overview of TREC 2001. *10th Text REtrieval Conference (TREC-2001)*.

(*Wiener et al., 1995*) WIENER E.D., PEDERSEN J.O., WEIGEND A.S. (1995). A neural network approach to topic spotting. In Proceedings of *SDAIR-95, 4th Annual Symposium on Document Analysis and Information Retrieval,* pp. 317–332.

(*Wood et al., 1995*) WOOD A., DREW N., BEALE R., HENDLEY B. (1995) Hyperspace: web browsing with visualization. *3rd International World Wide Web Conference (WWW'95)*.

(*Yang & Liu, 1999*) YANG Y. & LIU X. (1999). A re-examination of text categorization methods. *SIGIR-99, 22nd ACM International Conference on Research and Development in Information Retrieval*, pp.42–49.

(*Yang & Pedersen, 1997*) YANG Y. & PEDERSEN JO. (1997). A comparative study on feature selection in text categorization. *ICML-97, 14th International Conference on Machine Learning,* pp.412–420.

(*Zamir, 1998*) ZAMIR, O. (1998) Visualisation of search results in document retrieval systems, General Examination, *University of Washington*.

(*Zipf, 1949*) ZIPF GK. (1949) Human behavior and principles of feast effort, *Addison Wesley ed*.

CHAPITRE 2

RECHERCHE DE DOCUMENTS EN SANTE :

CAS DU CATALOGUE CISMeF

Sommaire

2.1 INTRODUCTION 73
2.2 STRUCTURE DES DOCUMENTS 74
2.2.1 LES METADONNEES 74
2.2.2 LE THESAURUS MESH 75
2.2.3 LA TERMINOLOGIE CISMEF 76
2.2.3.1 LES TYPES DE RESSOURCES 76
2.2.3.2 LES METATERMES 77
2.2.4 LE MODELE CISMEF POUR LA RECHERCHE D'INFORMATION 78
2.3 LA METHODOLOGIE DE MISE A JOUR 79
2.3.1 LE RECENSEMENT 79
2.3.2 LA SELECTION 80
2.3.3 LA DESCRIPTION 80
2.4 RECHERCHE D'INFORMATION DANS LE CISMEF 82
2.4.1 ACCES STATIQUE 83
2.4.1.1 LES DEFINITIONS 83
2.4.1.2 LES 'VOIR AUSSI' 84
2.4.1.3 LES ARBORESCENCES 85
2.4.2 ACCES DYNAMIQUE 86
2.4.2.1 LA NAVIGATION DYNAMIQUE 86
2.4.2.2 LA RECHERCHE SIMPLE 87
2.4.2.3 LA RECHERCHE AVANCEE 88
2.4.2.4 LA RECHERCHE BOOLEENNE 89
2.4.2.5 LA RECHERCHE PAS-A-PAS 90
2.4.2.6 OPTIONS DE RECHERCHE 90
2.4.2.6.1 Options par Défaut 90
2.4.2.6.2 Option Arborescence 91
2.4.2.6.3 Option Explosion 91
2.4.2.6.4 Option Majeur/Mineur 92
2.4.3 LES AFFILIATIONS 93
2.4.3.1 AFFILIATION DE QUALIFICATIFS 93
2.4.3.2 AFFILIATION DE TYPES DE RESSOURCES 93

2.4.4 LES REQUETES PREFORMATEES 94
2.4.4.1 LES STRATEGIES DE RECHERCHE 94
2.4.4.2 CISMEF-PATIENTS 95
2.4.4.3 LE PROJET COGNI-CISMEF 99
2.4.5 CATEGORISATION DES DOCUMENTS 100
2.5 QUELQUES PROBLEMES RENCONTRES 102
2.5.1.1 AJOUT D'AUTRES TYPES DE SYNONYMES 104
2.5.1.2 UTILISATION DE CONNAISSANCES 104S

Dans ce Chapitre 2, nous présentons le contexte dans lequel nous réalisons nos expériences, à savoir le catalogue de santé CISMeF. La structure du catalogue pour la description des documents se fonde sur un ensemble de métadonnées et une terminologie du domaine médical qui a été adaptée à la problématique de la recherche d'information. Nous détaillons les améliorations entreprises au cours de ces 3 dernières années en terme de recherche d'information et de navigation dans le catalogue.

2.1 Introduction

Avec l'explosion du Web et la prolifération des connaissances biomédicales, les utilisateurs ont potentiellement accès à des informations de plus en plus nombreuses mais en réalité, ils sont obligés de naviguer dans un vrai labyrinthe de pages. Les outils généralistes (tels que Yahoo, Google, Altavista…) ne permettent pas d'obtenir une présentation claire et organisée de l'information disponible en médecine. Leur utilisation est ainsi très vite limitée. A ce manque de spécificité des outils de recherche généralistes, s'ajoute le problème de la prise en compte de la qualité des informations disponibles. C'est dans ce contexte que le catalogue CISMeF (Catalogue et Index de Sites Médicaux Francophones) *(Darmoni et al., 2001) (Soualmia et al., 2002) (Soualmia & Darmoni, 2003a)* a été développé pour assister la recherche d'information en santé sur le Web.

Un grand nombre de ressources (ou documents) francophones (n=13 850 en septembre 2004) sont sélectionnées en fonction de critères stricts par une équipe de documentalistes et sont répertoriées selon une méthodologie de mise à jour. La description de ces ressources se fait à l'aide de *notices* en se basant sur un ensemble de méta-données et une terminologie structurée du domaine médical. Les notices et la terminologie sont stockées dans une base de données relationnelle.

Le but initial était d'assister les professionnels de santé dans leur quête d'informations sur l'Internet. D'autres catégories d'utilisateurs peuvent également être intéressées par un accès facilité aux informations de santé, tels que les étudiants en médecine ou encore les patients ou le grand public. Trois axes prioritaires ont ainsi été définis : le recensement des ressources (une ressource correspond à un site ou un document) concernant l'enseignement, la médecine factuelle (recommandations pour la bonne pratique clinique et conférences de consensus) et les ressources spécialement destinées aux patients. Composé au départ d'une simple liste de signets utiles aux professionnels de l'établissement (modèle à plat), ce service s'est progressivement amélioré et développé, pour devenir un véritable catalogue, reposant sur une structure complexe.

Nous détaillons dans les sections suivantes les différentes améliorations apportées au cours de ces 3 dernières années. Les améliorations (aussi bien conceptuelles que techniques) visent l'optimisation de la recherche d'information. La structure hiérarchisée du thésaurus MeSH et de la terminologie CISMeF permet d'optimiser la recherche d'information en précisant ou en étendant une requête par l'opération d'explosion des concepts en allant des concepts les plus génériques aux plus spécifiques. La structure du thésaurus et sa composition basée sur la possibilité d'affiliation de qualificatifs à des mots clés, permettent de mettre en place des options

de recherche permettant de préciser les requêtes. Enfin, le choix d'un thésaurus standardisé et maintenu par un organisme professionnel reconnu permet une interopérabilité avec d'autres catalogues en ligne utilisant le même outil. En fonction des besoins des utilisateurs, des améliorations portent sur l'enrichissement du vocabulaire employé (traduction des définitions des mots clés, création de synonymes ...). Les besoins spécifiques de la recherche d'information sur l'Internet, et notamment l'hétérogénéité du type des ressources accessibles, entraînent la création de niveaux de concepts supplémentaires. Ainsi la terminologie CISMeF intègre deux niveaux de concepts inexistants originellement au sein du thésaurus MeSH : les types de ressources et les métatermes. Nous détaillons tous ces éléments dans les sections suivantes.

2.2 Structure des Documents

Les documents répertoriés dans le CISMeF sont décrits à l'aide d'un ensemble de métadonnées et d'un ensemble de termes issus de la terminologie CISMeF, construite à partir du thésaurus MeSH.

2.2.1 Les Métadonnées

Les métadonnées (metadata en anglais) sont, au sens littéral du terme, des "données sur des données", une "information sur une information". Elles constituent une information secondaire au sujet d'une ressource primaire. Le but des métadonnées est de faciliter le travail des robots d'exploration qui indexent automatiquement les pages Web, en leur proposant des données descriptives normalisées. L'objectif final est de permettre aux moteurs de recherche de proposer des résultats plus pertinents lors des requêtes des utilisateurs.

Pour déterminer les champs nécessaires à la description des ressources incluses dans CISMeF, l'équipe s'est d'abord basée sur les travaux du projet Dublin Core[10] *(Baker, 2000)* qui propose un format pour l'écriture de métadonnées *(Thirion et al., 2004)*. Il existe aujourd'hui un groupe de travail DCMI (Dublin Core Metadata Initiative) pour la proposition d'une norme internationale de description des ressources de l'Internet. Une sélection a été faite parmi les 15 champs proposés par le Dublin Core, et les champs suivants ont été retenus : "titre" et "sous-titre", "auteur", "description", "site éditeur", "date", "identifiant" (URL), "format", "langue", "mots clés", "types de ressources". Les champs sont répétitifs et non obligatoires.

Le format IEEE 1484 Learning Object Metadata (LOM)[11] est un standard de description spécifique aux données pédagogiques. Un des trois axes prioritaires de CISMeF est le recensement des ressources pédagogiques. LOM permet de déterminer quelques champs pour répondre aux besoins générés par le recensement des ressources de ce type. Il s'agit des champs "cycle", "année", "numéro de question d'internat" et "intitulé de diplôme".

D'autres champs ont été créés pour répondre aux besoins spécifiques de description de ressources électroniques propres à CISMeF : "accès" (réservé, partiellement réservé, libre), "coût", "date de création de la notice descriptive", "date de consultation de la ressource",

[10] http://www.dublincore.org
[11] http://ltsc.ieee.org/

"département", "institution", "ISBN", "ISSN", "parrainage", "pays", "public concerné", "source", "ville".

Il a été également nécessaire de créer d'autres champs caractéristiques de certains types de ressources dont le recensement est prioritaire selon la politique éditoriale de CISMeF. Ainsi, les champs "indication du niveau de preuve" et "méthodologie suivie" ont été inclus pour répondre aux besoins des professionnels recherchant des recommandations de bonne pratique clinique (*Darmoni et al., 2003a).*

2.2.2 Le Thésaurus MeSH

Le thesaurus MeSH[12] (Medical Subject Headings) est produit depuis 1960 par la bibliothèque nationale américaine de médecine, la NLM (National Library of Medicine). L'organisation de l'information dans CISMeF repose sur ce thésaurus qui est utilisé notamment pour la base de données bibliographiques Medline, base de données la plus utilisée au monde en médecine. Il est précis, rigoureux et mis à jour annuellement par des experts du domaine. Il comprend, dans sa version 2004, 22 568 mots clés organisés selon 15 grandes catégories. Son adaptation annuelle comprend en moyenne entre 800 et 1000 nouveaux termes. Son utilisation courante par les professionnels de la santé a été un élément décisif dans le choix de cette terminologie pour organiser l'information dans CISMeF. CISMeF utilise la version française[13] du thesaurus MeSH, réalisée depuis 1986 par le DISC (Département de l'Information Scientifique et de la Communication) de l'INSERM (Institut National de la Santé et de la Recherche Médicale).

Les mots clés du thésaurus correspondent à des concepts médicaux. Ils sont organisés en 11 niveaux hiérarchiques allant du terme le plus général en haut de la hiérarchie, aux termes les plus spécifiques en bas de la hiérarchie. Une arborescence décrit d'abord des notions générales puis des notions très spécifiques en fonction du niveau. Par exemple, le mot clé *aberration chromosomique* est plus général que le mot clé *trisomie*. Chaque terme MeSH est identifié par un numéro : le "MeSH Tree Number" qui dépend de sa position dans l'arborescence. Un même mot clé peut appartenir à plusieurs arborescences. A ce jour (septembre 2004), 10 416 mots clés sont utilisés par CISMeF, soit près de 45% du thesaurus MeSH.

```
LT = POMPES IONIQUES
UF = POMPES A IONS
NT = ANTIPORTEURS
NT = PROTEINES DE TRANSPORT ANIONS
NT = PROTEINES DE TRANSPORT CATIONS
NT = SYMPORTEURS
RT = CANAL MEMBRANAIRE
RT = TRANSPORT BIOLOGIQUE ACTIF
RT = TRANSPORT IONIQUE
```

```
LT= Terme principal
UF= Synonyme
NT= terme spécifique
RT= Voir Aussi
```

FIG.2.1.–Fichier texte fourni par l'INSERM

Les fichiers MeSH sont traités automatiquement pour renseigner la terminologie CISMeF afin qu'elle soit exploitable au niveau du site. La médecine étant un domaine qui évolue

[12] http://www.nlm.nih.gov/mesh/MBrowser.html
[13] http://disc.vjf.inserm.fr:2010/basismesh/mesh.html

constamment (nouvelles maladies, nouveaux traitements…etc.), ces fichiers sont mis à jour chaque année (nouveaux mots clés, qualificatifs, nouvelles organisations dans les hiérarchies).

D'après l'exemple de définition de la Fig.2.2, le terme associé au concept ayant l'identifiant unique D006521 est *Hépatite Chronique*. Le code cat.MeSH indique à quel niveau ce concept est situé dans la hiérarchie. On peut en déduire que *Hépatite Chronique (*C06.552.380.350) est subsumé par *Hépatite* (C06.552.380).

Des qualificatifs peuvent être affiliés aux mots clés pour en préciser le sens en limitant leur étendue à certains aspects. Ils sont à utiliser contextuellement. Le thesaurus MeSH comprend 84 qualificatifs dans sa version 2004. Par exemple, l'association du qualificatif *diagnostic* au mot clé *lombalgie* (notée *lombalgie/diagnostic*), permet de qualifier un aspect particulier de la lombalgie, en l'occurrence le diagnostic. La ressource qui sera indexée avec ce couple (mot clé/qualificatif) traite donc du diagnostic de la lombalgie. Les qualificatifs sont également organisés de manière hiérarchique, tout comme les mots clés.

2.2.3 La Terminologie CISMeF

Deux niveaux conceptuels ont été ajoutés à la terminologie CISMeF dans le but d'améliorer la recherche d'information dans le catalogue : *les types de ressources* et les *métatermes*.

2.2.3.1 Les Types de Ressources

Les types de ressources sont une généralisation des types de publication de Medline et correspondent à la nature de l'information véhiculée par la ressource. Cette liste a été créée selon les besoins spécifiques d'un catalogue de ressources Internet en santé et est alimentée régulièrement. A ce jour, cette liste comprend 145 types de ressources. Il existe en effet différents types de documents présents sur Internet dans le domaine médical recensés dans le CISMeF. De ce fait, un simple texte est différencié d'une ligne directrice de pratique médicale, d'un cours, d'un QCM (Question au Choix Multiple) ou d'un APP (Apprentissage par Problèmes).

Les documents recensés dans le CISMeF peuvent correspondre soit à un site dans sa globalité, soit à un ensemble de documents publiés sur un site, soit à un document particulier. Le "type de ressource" (TR) se définit comme la nature de l'information véhiculée par le site ou le document. Le type de ressource est à différencier des mots clés. Ainsi, un *compte-rendu de conférence de consensus* sur l'*hépatite C* sera indexé avec le mot clé *hépatite C* et le type de ressource *conférence de consensus*, tandis qu'un *article* traitant de la *méthodologie des conférences de consensus* sera indexé avec le mot clé *conférence de consensus* et le type de ressource *article de périodique*.

Certains types de ressources peuvent être un mot clé décrivant le sujet d'une ressource (par exemple *service cardiologie hôpital*), ou encore un type de ressource donnant lieu à l'élaboration d'un répertoire (ici le *répertoire des services hospitaliers en cardiologie*). Dans le thésaurus MeSH, certains mots clés correspondent à des types de ressources mais cette liste n'est pas suffisante et ne permet pas d'apposer un type de ressource à chaque ressource. La base de données bibliographiques Medline possède sa propre liste de "types de publication" spécifique aux articles de périodiques scientifiques. Pour répondre aux spécificités des ressources recensées par CISMeF, une liste de types de ressources a été créée, incluant certains

types de publication de Medline ainsi que les mots clés MeSH utilisables. Elle est adaptée en permanence selon les nouveaux besoins qui émergent. Ainsi, par exemple, le type de ressource *association* a été créé pour permettre d'indexer les sites web d'associations médicales.

Les types de ressources sont organisés en arborescences, contrairement aux types de publication de Medline. Par exemple, le type de ressource *association* généralise des types de ressources plus précis selon la hiérarchie suivante :

association

association patients

association professionnels santé

syndicat

L'introduction du concept de types de ressource permet d'inclure un niveau de hiérarchie supplémentaire dans le modèle du CISMeF.

2.2.3.2 Les Métatermes

Les mots clés ont été regroupés dans CISMeF en fonction de spécialités médicales (n=66) intitulés *métatermes* (Par exemple : *Cardiologie*). Ce sont des super-concepts qui permettent une vision plus globale concernant une spécialité en offrant un niveau supplémentaire d'abstraction. Les métatermes permettent en effet de connaître l'ensemble des termes MeSH qui sont répartis dans plusieurs arborescences mais qui concernent une même spécialité. Le thésaurus MeSH, dans sa structure d'origine, ne permet pas d'obtenir de vision globale d'une spécialité médicale.

En effet, il existe 15 catégories concernant des notions générales. Les têtes d'arborescences ("top terms") sont organisées à plat de la façon suivante :

A - ANATOMIE

B - ORGANISMES

C - MALADIES

D - PRODUITS CHIMIQUES, BIOLOGIQUES ET PHARMACEUTIQUES

E - ÉQUIPEMENTS ET TECHNIQUES ANALYTIQUES, DIAGNOSTIQUES ET THERAPEUTIQUES

F - PSYCHIATRIE ET PSYCHOLOGIE

G - SCIENCES BIOLOGIQUES

H - SCIENCES PHYSIQUES

I - ANTHROPOLOGIE

J - TECHNOLOGIE, ALIMENTS ET BOISSONS

K - ARTS ET SCIENCES HUMAINES

L - SCIENCES INFORMATION

M - INDIVIDUS

N - SANTE (ADMINISTRATION DES SOINS)

Z - EMPLACEMENTS GEOGRAPHIQUES

Cette organisation ne permet pas d'accéder facilement à tous les mots clés concernant une spécialité médicale ou biologique particulière. Ainsi, pour retrouver tous les mots clés concernant la *neurologie* par exemple, il faut d'abord consulter l'arborescence ANATOMIE pour obtenir les mots clés concernant l'*anatomie du système nerveux*, puis l'arborescence MALADIES pour les mots clés concernant les *maladies du système nerveux*, l'arborescence PRODUITS CHIMIQUES ET PHARMACEUTIQUES pour les substances utilisées en *neurologie*, etc… C'est un travail laborieux, qui nécessite une connaissance approfondie du thésaurus MeSH.

Pour simplifier cet accès, le concept de "métaterme" a été créé. Les métatermes permettent de connaître l'ensemble des termes MeSH répartis dans plusieurs arborescences mais concernant une même spécialité biologique ou médicale. Ces liens sémantiques concernent tous les niveaux hiérarchiques du modèle de structuration de l'information de CISMeF, c'est-à-dire non seulement les mots clés mais aussi les qualificatifs et les types de ressources.

Ainsi, le métaterme correspondant à la spécialité médicale *cancérologie* est lié, notamment, au mot clé *cancérogènes* de l'arborescence PRODUITS CHIMIQUES, BIOLOGIQUES ET PHARMACEUTIQUES, au mot clé *tumeurs* de l'arborescences MALADIES, mais aussi au qualificatif *radiothérapie*, ainsi qu'au type de ressource *service oncologie hôpital*.

2.2.4 Le Modèle CISMeF pour la Recherche d'Information

L'introduction de deux niveaux conceptuels supplémentaires par rapport au MeSH, les métatermes et les types de ressources, permet d'exprimer des requêtes complexes dans CISMeF comme des '*recommandations en cardiologie'* ou encore des *'cours en virologie'* ce qui n'est pas possible avec la structure du MeSH et sa seule utilisation. Chaque métaterme est en association avec une ou plusieurs arborescences de mots clés, qualificatifs et types de ressources. Par exemple le métaterme *Chirurgie*, est en association avec les mots clés : *complications post-opératoires*, le qualificatif : *transplantation*, et le type de ressource : *chirurgie hôpital*. Chaque terme peut avoir un ensemble de synonymes. Il peut appartenir à plusieurs arborescences. Par exemple : le terme *tumeur peau* sera associé aux métatermes *dermatologie* et *cancérologie*. Les « fils » d'un terme vont dépendre de l'arborescence dans laquelle il se trouve. Un même terme peut figurer dans les hiérarchies des mots clés, des qualificatifs ou des types de ressources. Par exemple le terme *virologie* est un mot clé et un qualificatif (et c'est également un métaterme).

Un terme appartenant à un des niveaux de concept de la terminologie CISMeF, c'est-à-dire qui est soit un métaterme (ou un synonyme de métaterme), un mot clé américain ou français (ou un synonyme américain ou français), un qualificatif (ou un synonyme de qualificatif), ou un type de ressource (ou un synonyme de type de ressource), est appelé "terme réservé".

Ce modèle de structuration de l'information est utilisé dans le processus mis en place pour la recherche d'information, mais également pour la catégorisation des ressources.

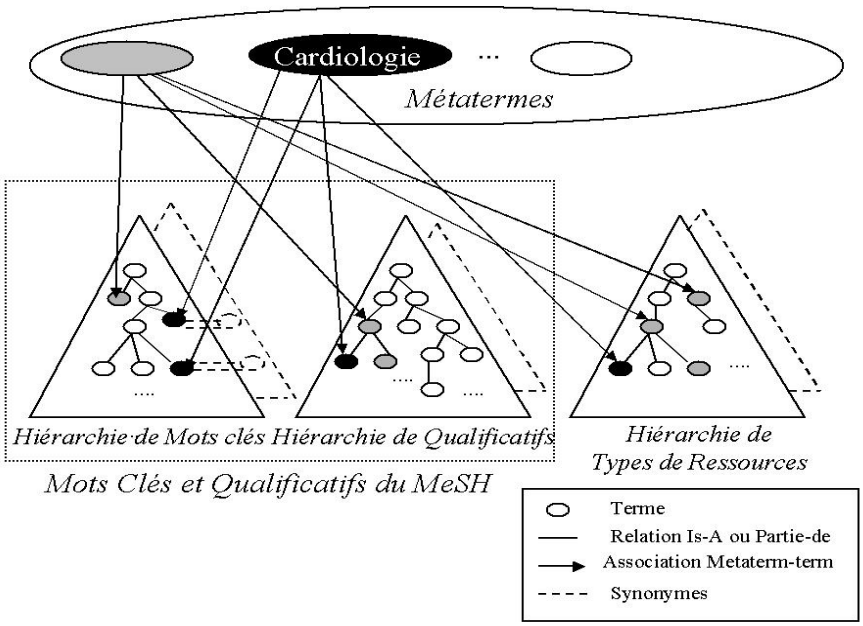


FIG.2.3.– Structure de la terminologie CISMeF *(Soualmia et al., 2002)*

2.3 La Methodologie de Mise à Jour

Pour permettre le catalogage de nouvelles ressources, une méthodologie de mise à jour et de maintenance a été mise en place. Le catalogage des ressources s'effectue selon quatre étapes successives : le recensement des ressources, la sélection, la description et enfin l'indexation.

2.3.1 Le Recensement

Le recensement des nouvelles ressources est en partie automatisé, grâce à un outil de veille qui permet de repérer les changements apparus sur une sélection de pages Web. Une veille quotidienne est effectuée sur des annuaires pluridisciplinaires francophones tels que Yahoo, Nomade, Toile du Québec, etc... Une veille complémentaire hebdomadaire ou mensuelle est réalisée sur des sites producteurs d'informations de santé (ministères, universités, agences gouvernementales, sociétés savantes...). La découverte de nouveaux documents par la consultation des rubriques "Liens" ou "Favoris" ou par la lecture de périodiques spécialisés vient compléter cette démarche de veille. Cette première étape de recensement des ressources inclus un aspect plus "passif", mettant en œuvre la méthode PUSH, par le biais d'abonnements à des services d'envoi de nouveautés gérés par les sites producteurs d'informations, ainsi qu'à des listes de diffusion spécialisées. Ce recensement peut également s'effectuer grâce au signalement de nouvelles ressources par les webmasters eux-mêmes.

2.3.2 La Sélection

L'étape de sélection permet de filtrer les sites et documents recensés lors de l'étape précédente. Une politique de sélection a été mise en place de façon à ce que CISMeF recense en priorité les ressources émanant de sources institutionnelles ou officielles. Cette exigence par rapport à la source des informations est primordiale et s'avère être le principal critère de sélection des ressources, garant de la qualité des informations véhiculées. En effet, la validité de l'information présente sur l'Internet doit être systématiquement remise en cause. Contrairement à l'édition papier validée par des comités de lecture de revues scientifiques, l'information disponible sur le réseau n'a pas été évaluée dans la grande majorité des cas. Pour ces raisons, une politique rigoureuse de sélection des ressources a été mise en place.

Plusieurs travaux ont été réalisés pour déterminer un ensemble de critères de qualité favorisant l'évaluation de sites et documents médicaux. Le CHU de Rouen a participé au groupe de travail français Centrale Santé, fédéré par l'École Centrale de Paris et qui a réuni médecins, bibliothécaires médicaux, ingénieurs et juristes. Ce groupe de travail a donné lieu à l'élaboration de la grille du NetScoring[14], référentiel regroupant les critères de qualité de l'information en santé *(Darmoni et al., 1999)*. Le NetScoring est une grille comprenant 49 critères de qualité regroupés en 8 classes principales : la crédibilité, le contenu, la qualité des hyperliens (notamment avec la mesure du Web Impact Factor *(Soualmia et al., 2002b) (Soualmia et al., 2002c)*), le design, l'interactivité, les aspects quantitatifs, la déontologie et l'accessibilité.

Pour sélectionner les ressources à inclure, CISMeF utilise les principaux critères de cette grille du NetScoring. Une attention particulière est portée à l'identité de l'auteur, à sa fiabilité et à la date de mise à jour des informations. Certaines ressources sont rejetées parce qu'elles ne respectent pas des critères de qualité essentiels, en particulier sur le plan déontologique. Cette étape de sélection nécessite parfois la consultation d'un expert de la discipline.

2.3.3 La Description

Chaque ressource sélectionnée est ensuite décrite à l'aide d'un certain nombre de champs inspirés de normes de description de métadonnées. Les ressources sont également indexées en fonction de la terminologie CISMeF. Celle-ci a été construite à partir des concepts du thésaurus MeSH. A chaque ressource d'information de santé est associée une notice ou fiche descriptive. Ces notices sont similaires à des «annotations». Elles permettent d'indexer les ressources sélectionnées. Ces notices sont créées par les membres de l'équipe qui se chargent de déterminer le contenu des champs des métadonnées (*auteur, titre, contenu*...etc), d'indexer les ressources avec les termes de la terminologie CISMeF, mais aussi de décrire le contenu informationnel de la ressource à l'aide d'un résumé (le champ *contenu*). Aucune modification et annotation ne sont effectuées sur la ressource elle-même.

La politique d'indexation mise en place par l'équipe consiste à indexer les ressources avec les mots clés les plus fins concernant le sujet traité. Ainsi, une ressource concernant la grippe sera indexée au mot clé *grippe* et non pas au mot clé *maladies virales* qui est un terme générique du terme *grippe*. Il existe un certain nombre de descripteurs obligatoires[15] (appelés aussi "check

[14] http://www.chu-rouen.fr/netscoring
[15] http://disc.vjf.inserm.fr:2010/basismesh/m2004_lst_chtags.html

tags") correspondant à des notions que les indexeurs doivent prendre systématiquement en compte dès qu'elles apparaissent dans le texte d'un document. Le descripteur obligatoire doit apparaître dans la liste des mots clés. Par ailleurs, CISMeF n'utilise que les termes qui concernent l'être humain.

Dans de rares cas, il arrive que des ressources ne puissent pas être parfaitement indexées avec des termes du MeSH. Un « mapping » manuel est alors effectué pour retrouver le(s) terme(s) MeSH qui décrivent au mieux la ressource considérée. Par exemple les ressources traitant de *dysmélie* (n'existant pas dans la terminologie) ont été indexées avec le terme MeSH *ectromélie*.

Le niveau de finesse de l'indexation dépend également du degré d'intérêt de la ressource en question. Ainsi, une recommandation de pratique médicale, unique sur l'Internet, particulièrement difficile d'accès, et essentielle pour les professionnels sera indexée de façon beaucoup plus fine qu'une association de patients contre le cancer par exemple. En moyenne, 6,7 mots clés ou couples (mots clé/qualificatif) sont utilisés. Le nombre de mots clés peut varier d'une ressource à une autre en fonction du niveau de finesse de l'indexation décidé au départ. Le nombre maximal de mots clés utilisés pour l'indexation d'une seule et même ressource est de 301 à ce jour. Outre les mots clés et les qualificatifs, chaque ressource est indexée avec un ou plusieurs "types de ressources".

La notion de « Major Topics » de Medline a été adaptée dans le CISMeF. Cela a permis de redéfinir la politique d'indexation, dans le but d'améliorer les possibilités de recherche d'information. En effet, l'indexation ne tenait compte que des thèmes traités de manière importante dans les documents. Chaque mot clé choisi était de facto considéré comme étant "majeur". L'intégration de la notion de *pondération* des mots clés permet, au moment de l'indexation des ressources, de déterminer si les mots clés sont majeurs ou mineurs, c'est-à-dire si les thèmes correspondants sont traités de façon importante tout au long du document, ou si au contraire, ils sont abordés uniquement dans une petite partie du document. Cette notion de pondération est représentée par une étoile (*) dans les notices des ressources.

Cette notion de pondération qui existe dans Medline pour les mots clés et les qualificatifs a été étendue, dans le CISMeF, aux types de ressources. Ainsi, par exemple, un cours comprenant une image sera indexé aux types de ressources **cours* (en majeur) et *illustration médicale* (en mineur).

Nous avons *(Dahamna & Soualmia, 2002)* réalisé une retro-ingénierie de la structure de la base de données relationnelle (qui ne satisfaisant aucune contrainte d'intégrité) gérée à l'époque avec Access, vers une base de données relationnelle sous Oracle permettant ainsi des traitements de saisie, de génération des notices et de réponse aux requêtes des utilisateurs plus performants. La saisie de tous les champs s'effectue sous Oracle, avec un écran de saisie de présentation Access[16]. A partir de cette saisie sont générées automatiquement plusieurs ensembles descriptifs. Des notices descriptives sont créées sous différents formats, plus ou moins longs, et le format RDF (into HTML) est utilisé pour présenter les métadonnées. L'outil de recherche Doc'CISMeF associé au catalogue, a également été amélioré au cours de ces dernières années pour prendre en compte les différents besoins des utilisateurs et leur permettre une vision globale de la terminologie du domaine.

[16] L'architecture du système utilise le serveur Tomcat (sous NT) qui regroupe un serveur applicatif (entre autres la saisie des informations des notices) et un serveur Web. Tomcat dispose d'un moteur de servlets. Tout le principe de l'interrogation du catalogue (par Doc'CISMeF) est fondé sur l'appel de servlets Java en passant en paramètre la méthode Get.

Titre : Comparaisons internationales sur la prévention sanitaire

Auteur(s) : Mme Jourdain-Menninger D ; Mme Lignot-Leloup M
Site éditeur : Documentation Française (La)
Contenu : de la prévention à la promotion de la santé, une démarche partagée par la Finlande, le Québec et le Royaume-Uni (contexte commun aux pays développés, implication croissante des organisations internationales dans les politiques nationales de promotion de santé), méthode : stratégie élaborée au plan national et mise en œuvre locale (...) ; 171 pages
Langue : français
Pays : France
Publié le : 01/01/2003
Mots-clés : accident travail
administration santé publique
*alcoolisme/prévention et contrôle
Allemagne
*Angleterre
*appareil cardiovasculaire, maladies/prévention et contrôle
*causes décès
communication
comportement alimentaire
*contrôle social formel
*délivrance soins
délivrance soins/organisation et administration
(...)
troubles mentaux
tumeur sein/diagnostic
tumeur sein/prévention et contrôle
*tumeurs/diagnostic
*tumeurs/prévention et contrôle
Spécialités : *médecine préventive
*environnement et santé publique
*thérapeutique
*psychiatrie
*virologie
*oncologie médicale
*toxicologie
(...)
pédiatrie
Type(s) : étude comparative
rapport technique
Tarif : gratuit
Accès : libre
Format(s) : html ; pdf
Création de la notice : 07/08/2003
Consultation de la ressource : 07/08/2003
URL(s) : http://www.ladocumentationfrancaise.fr/brp/notices/034000473.shtml

FIG.2.4.–Exemple de notice. http://doccismef.chu-rouen.fr/html/nl/12/012081.html

2.4 Recherche d'Information dans le CISMeF

La terminologie CISMeF est utilisée dans le processus de recherche d'information. Celle-ci peut être statique, par l'utilisation d'un index alphabétique ou thématique (en fonction des spécialités), ou dynamique par l'intermédiaire de l'outil Doc'CISMeF.

2.4.1 Accès Statique

CISMeF contient un classement alphabétique qui utilise les termes MeSH de la traduction française du thesaurus MeSH et indique également les termes d'origine américains. Ce classement donne accès aux mots clés, ainsi qu'aux qualificatifs et aux types de ressources utilisés dans CISMeF. A chaque terme correspond une page HTML présentant les caractéristiques du terme en question.

Les pages des mots clés sont particulièrement développées et mentionnent notamment les éléments suivants :

– la définition du mot clé,

– ses synonymes,

– des renvois d'orientation,

– les arborescences auxquelles le mot clé appartient,

– des requêtes pré formatées utilisant l'outil de recherche et proposant notamment un accès aux ressources destinées aux professionnels, aux patients ou aux étudiants,

– les notices descriptives abrégées des ressources indexées avec ce mot clé. Ces notices sont classées par qualificatifs puis par types de ressources.

Un classement thématique propose également un accès par grandes spécialités médicales et biologiques.

2.4.1.1 Les Définitions

La traduction française du thésaurus MeSH effectuée par l'INSERM porte uniquement sur les descripteurs, tandis que les définitions (appelées "Scope Note" par la NLM) ne sont disponibles qu'en anglais, ce qui entraîne une difficulté supplémentaire dans la manipulation du thésaurus. Les définitions sont extrêmement utiles aussi bien pour les indexeurs que pour les utilisateurs. Pour les indexeurs, les définitions permettent de s'assurer que les termes d'indexation sont utilisés de façon normalisée, et donc en cohérence d'un indexeur à un autre. Pour les utilisateurs recherchant de l'information, les définitions aident à cerner le sens d'un mot clé, et donc à éviter des incompréhensions pour les mots clés dont le libellé peut parfois être imprécis ou ambigu.

Entreprendre la traduction de ces définitions est une entreprise lourde mais cependant nécessaire à l'amélioration de l'accès au thésaurus. A ce jour, l'équipe CISMeF a traduit 2 892 définitions qui apparaissent sur les pages statiques des mots clés dans CISMeF.

FIG.2.5.–Page du mot clé *hépatite*.

Cela permet à l'utilisateur, lors de la consultation de la page du mot clé *abêtalipoprotéinémie* par exemple, et avant de consulter les ressources concernant ce mot clé, d'accéder à la définition indiquent que c'est une "perturbation héréditaire du métabolisme". (*abêtalipoprotéinémie : perturbation héréditaire du métabolisme lipidique caractérisée par la quasi-absence d'apolipoprotéines B et de bêtalipoprotéines dans le plasma. La protéine microsomale de transfert des triglycérides est déficitaire ou absente dans les entérocytes. Les résultats cliniques et de laboratoire incluent acanthocytose, hypocholestérolémie, neuropathie périphérique, dégénération de colonne postérieure, ataxie et stéatorrhée. Les capacités intellectuelles peuvent aussi être diminuées).*

CISMeF est non seulement un catalogue de ressources mais aussi fournisseur d'information primaire grâce à ce travail entrepris sur les définitions du thésaurus MeSH. En priorité, le travail s'est d'abord axé sur les définitions des mots clés ayant un libellé imprécis ou ambigu et il se poursuit par la traduction systématique des nouveaux mots clés introduits dans CISMeF. Les définitions des mots clés concernant la *virologie* sont toutes traduites.

2.4.1.2 Les 'Voir Aussi'

Toujours dans le but de faciliter la recherche d'information pour l'utilisateur, il a aussi été nécessaire d'enrichir les relations de type voir aussi entre les termes du thésaurus MeSH. Ces "voir aussi" facilitent la navigation des pages statiques. Ils renvoient l'utilisateur vers des mots clés appartenant à une arborescence différente, ou vers des qualificatifs, des types de ressources

ou encore des métatermes. Par exemple, la page du mot clé *diarrhée nourrisson* mentionne qu'il est aussi possible de consulter les pages des mots clés *antidiarrhéiques* et *solutions réhydratation*. La même démarche a été entreprise concernant la mise en place de relations d'exclusion. Ainsi, de nombreux "ne pas confondre avec..." ont été introduits.

2.4.1.3 Les Arborescences

La structure hiérarchisée du thésaurus MeSH permet, de mener une recherche par arborescences. Le principal avantage de cette hiérarchisation des termes est effectivement la possibilité de "naviguer" à l'intérieur des ensembles d'arborescences. La possibilité de visualiser les arborescences permet d'affiner ou au contraire d'élargir une recherche, voire même de la rediriger vers des termes voisins en prenant connaissance des termes génériques et/ou spécifiques. La structure arborescente du thésaurus permet de prendre connaissance des termes représentant les concepts d'un domaine particulier. De plus, cela permet d'avoir une approche visuelle de la structure du thésaurus, ce qui facilite la recherche d'information. Rechercher par arborescence permet une navigation au sein du thésaurus sans avoir une connaissance approfondie du vocabulaire employé. On peut ainsi savoir que le mot clé *virémie* fait partie de l'arborescence *maladies virales* et en déduire que la virémie est une maladie virale. De même on peut déduire que l'*hépatite* est une maladie de l'appareil digestif.

Chaque page concernant un mot clé du thésaurus MeSH propose un accès statique aux arborescences auxquelles celui-ci appartient. Cet accès indique les termes génériques des arborescences et permet de déployer la totalité de l'arborescence. Par exemple, pour le mot clé *migraine*, la page du mot clé indique qu'il appartient aux arborescences *appareil cardiovasculaire, maladies* et *système nerveux, maladies*. Un clic sur *appareil cardiovasculaire* permet de visualiser la totalité des termes de l'arborescence. Il s'agit néanmoins des seuls termes MeSH utilisés dans CISMeF et non de la totalité du thésaurus.

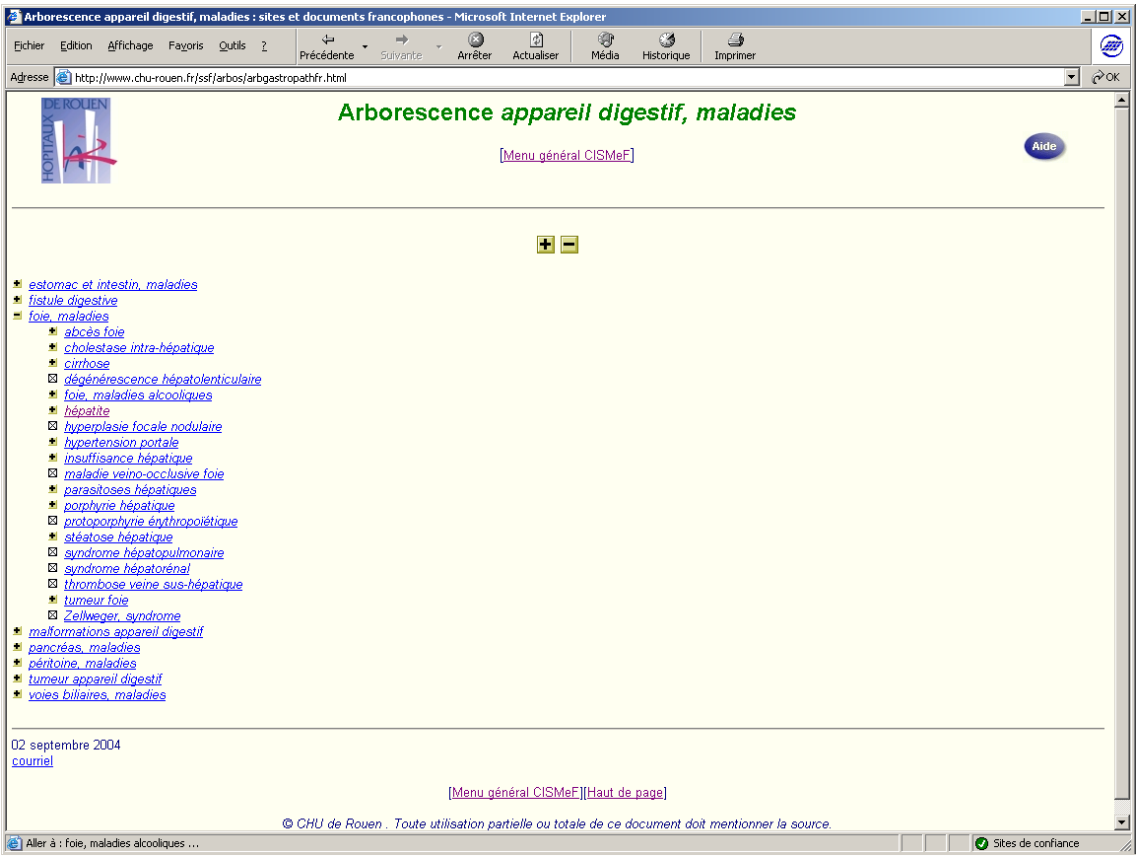


FIG.2.6.–Arborescence *Appareil Digestif, Maladies* à laquelle appartient le mot clé *hépatite*.

2.4.2 Accès Dynamique

2.4.2.1 La Navigation Dynamique

Le second type d'accès possible est une navigation dynamique dans les différentes arborescences. Ce type de présentation permet de visualiser en une seule fois les différentes positions du mot clé au sein de toutes les arborescences auxquelles il appartient. Il permet également de naviguer dynamiquement dans les arborescences. Ainsi, cliquer sur un autre mot clé permet de visualiser immédiatement ces différentes positions dans les arborescences. Pour chaque mot clé, deux liens sont proposés : un lien vers le site de PubMed (base de données Medline) et un lien vers la page statique du mot clé dans CISMeF. Cet accès par navigation dynamique dans les arborescences présente la liste de tous les mots clés du thésaurus MeSH, et non plus seulement les mots clés utilisés par CISMeF.

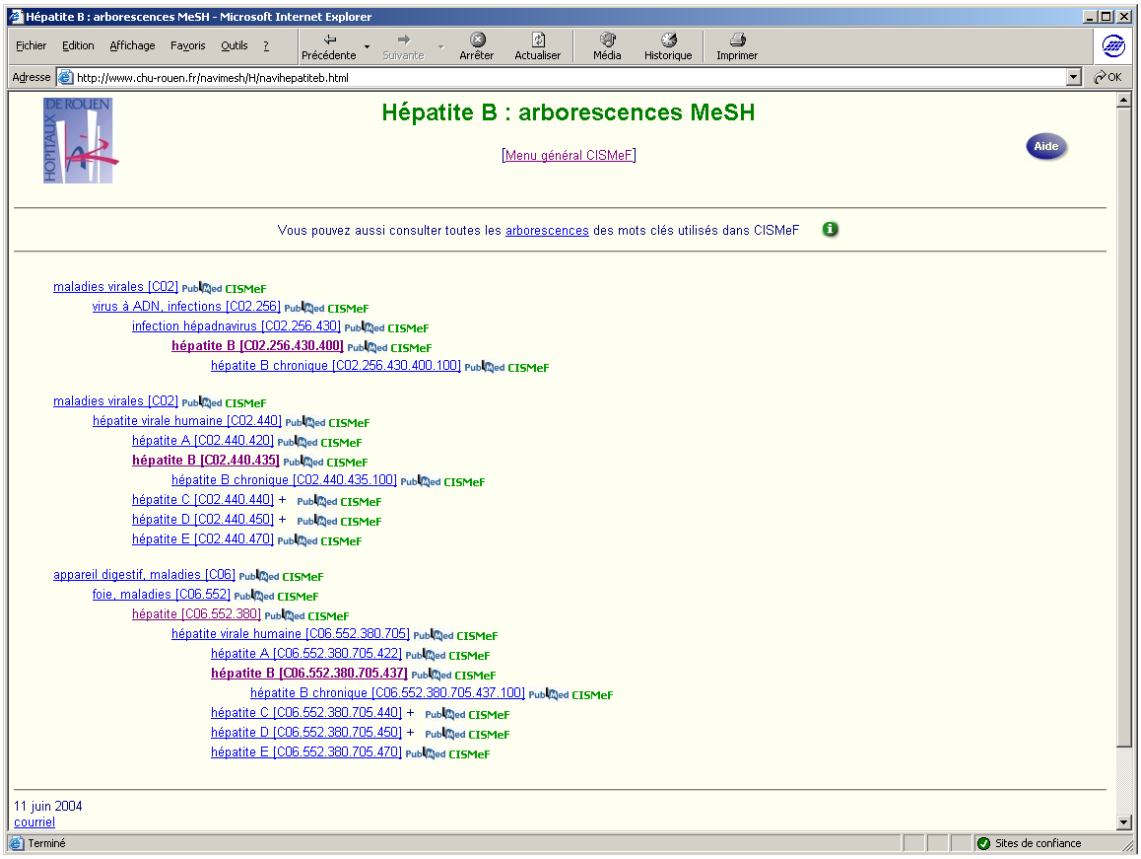


FIG.2.7.–Navigation au sein des arborescences du mot clé *hépatite B*.

Une page[17] récapitule tous les termes des têtes d'arborescences (Top Terms) afin d'offrir un accès par thèmes du MeSH. Elle propose des liens vers le déploiement des arborescences ainsi que vers les arborescences "dynamiques" permettant de visualiser les différentes positions du terme dans les arborescences du MeSH.

Face au nombre croissant de nouvelles ressources intégrées au catalogue (50 environ par semaine), de simples classements alphabétique et thématique ne permettent plus de répondre aux besoins de formulation de requêtes de plus en plus précises. Depuis juin 2000, CISMeF dispose d'un outil de recherche associé : Doc'CISMeF qui propose des possibilités de recherche plus étendues et plus performantes. Différents types de recherche sont disponibles : une recherche simple, une recherche avancée, une recherche booléenne et une recherche pas-à-pas.

2.4.2.2 *La Recherche Simple*

Ce type de recherche permet de saisir un terme unique (par exemple, le terme *migraine*) ou une expression (par exemple, *hépatite virale humaine*). De façon générale, les termes saisis peuvent l'être en minuscules, en majuscules, accentués ou non, en français ou en anglais.

[17] http://www.chu-rouen.fr/ssf/arborescences.html

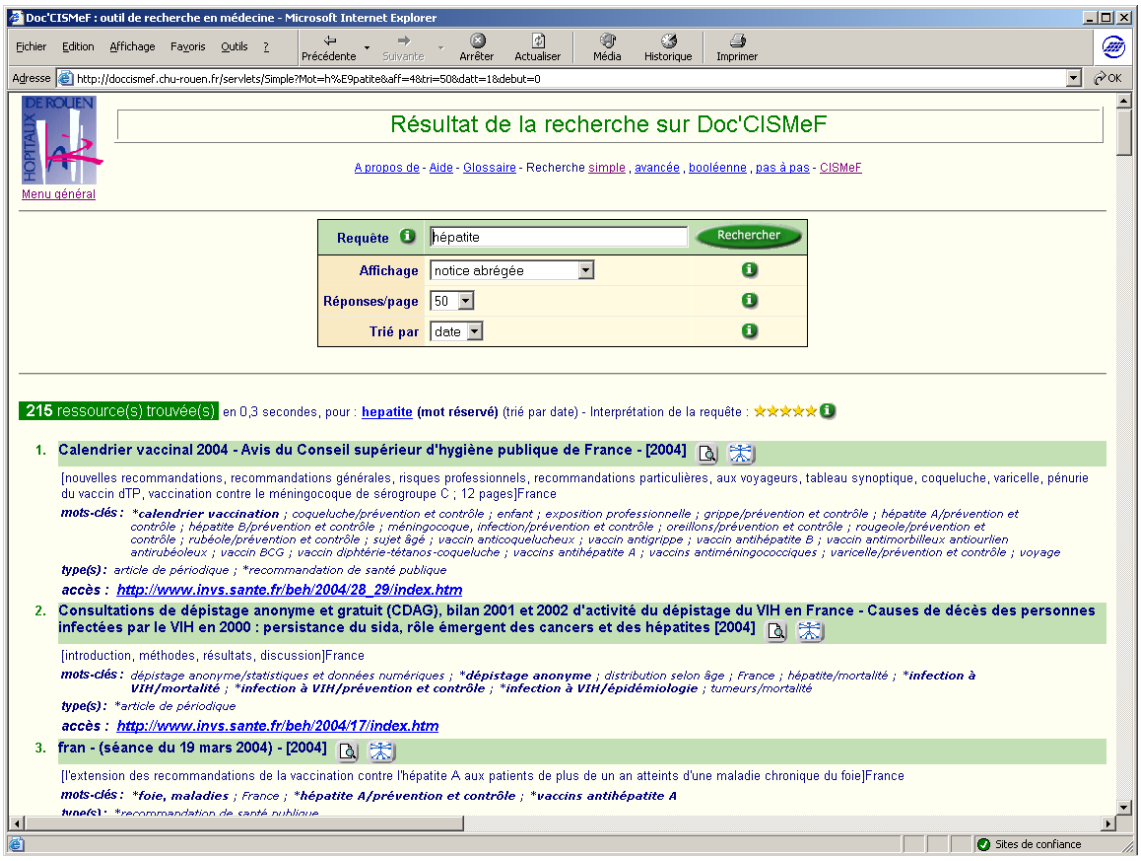


FIG.2.8.–Formulaire de recherche simple.

2.4.2.3 *La Recherche Avancée*

Afin de pouvoir mener des recherches plus pointues, le formulaire de recherche avancée permet d'utiliser des critères de recherches plus fins. La recherche peut porter sur les champs suivants (qui correspondent aux principaux champs des métadonnées) : *auteur, cible, date de publication, spécialité médicale, mots clés, pays, qualificatif, résumé, site éditeur, source, titre, type de ressource, URL* et *ville*. Il est possible de combiner les critères de recherche en utilisant les opérateurs Booléens ET, OU, SAUF. Ces opérateurs permettent des traitements ensemblistes des requêtes, les ensembles considérés étant les ensembles de documents indexés par les termes des requêtes.

Des listes sont disponibles pour aider à la recherche et permettent de prendre connaissance des termes utilisés. La liste des auteurs permet, par exemple, de connaître tous les auteurs référencés dans le CISMeF.

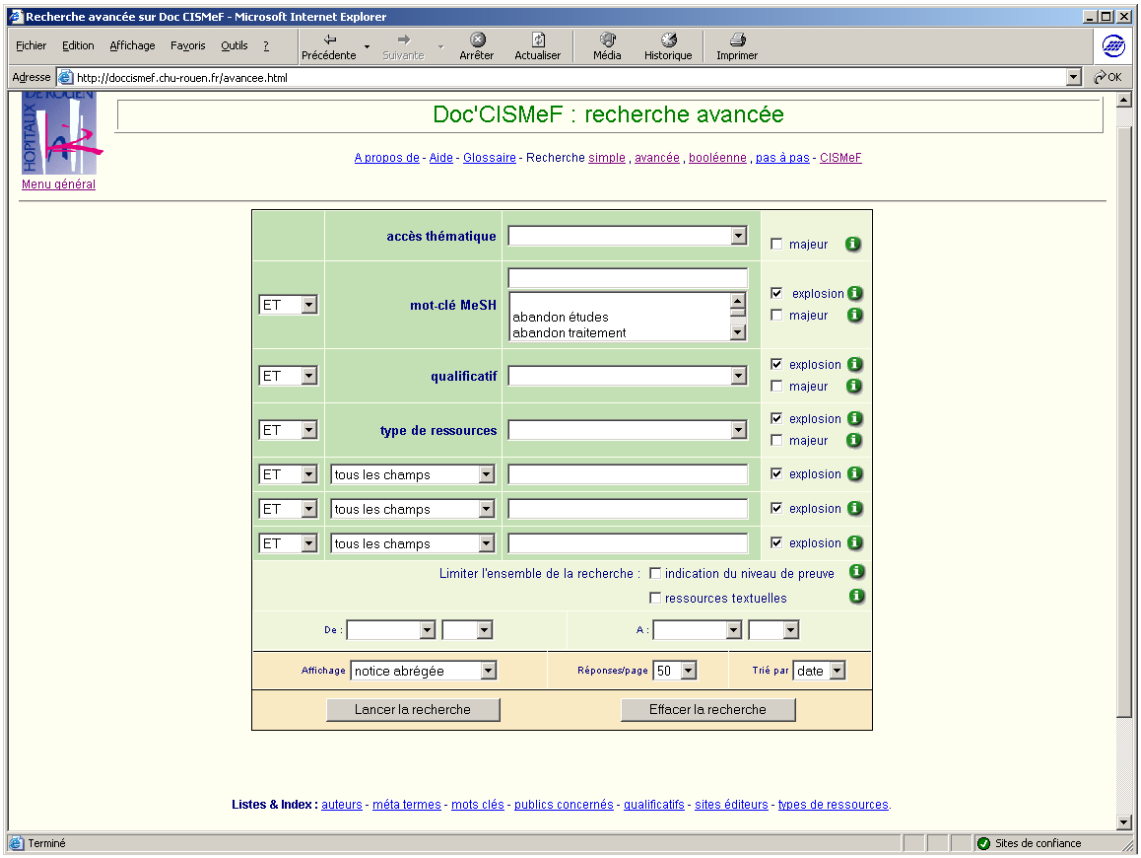


FIG.2.9.–Formulaire de recherche avancée.

2.4.2.4 *La Recherche Booléenne*

La recherche booléenne est une généralisation de la recherche avancée et permet de formuler soi-même la requête à l'aide d'un langage particulier (codes des champs et opérateurs booléens). Ce mode de recherche a été développé pour les professionnels de l'information et leur permet de formuler des requêtes complexes. La requête [*(lombalgie.mc ET diagnostic.qu) OU 2002.an]* permet d'obtenir les ressources concernant le diagnostic de la lombalgie ou bien toutes celles datant de l'année 2002 (*lombalgie* en mot clé, *diagnostic* en qualificatif, et *2002* en année de publication).

Cette recherche booléenne permet l'emploi de parenthèses, pour associer différents champs, et des caractères de troncature. Le caractère (?) permet une troncature d'une seule lettre en début, milieu ou fin de mot. Le caractère (*) permet d'inclure une troncature de une à plusieurs lettres. Ainsi la requête *an*ie.mc* inclut les mots *anémie, anatomie*... La requête **sang** inclut les termes *cellule sanguine, sang, maladie*, etc... La syntaxe *[nonexpl]* permet de désactiver l'option «explosion», et *[majeur]* permet de restreindre la recherche aux mots clés majeurs.

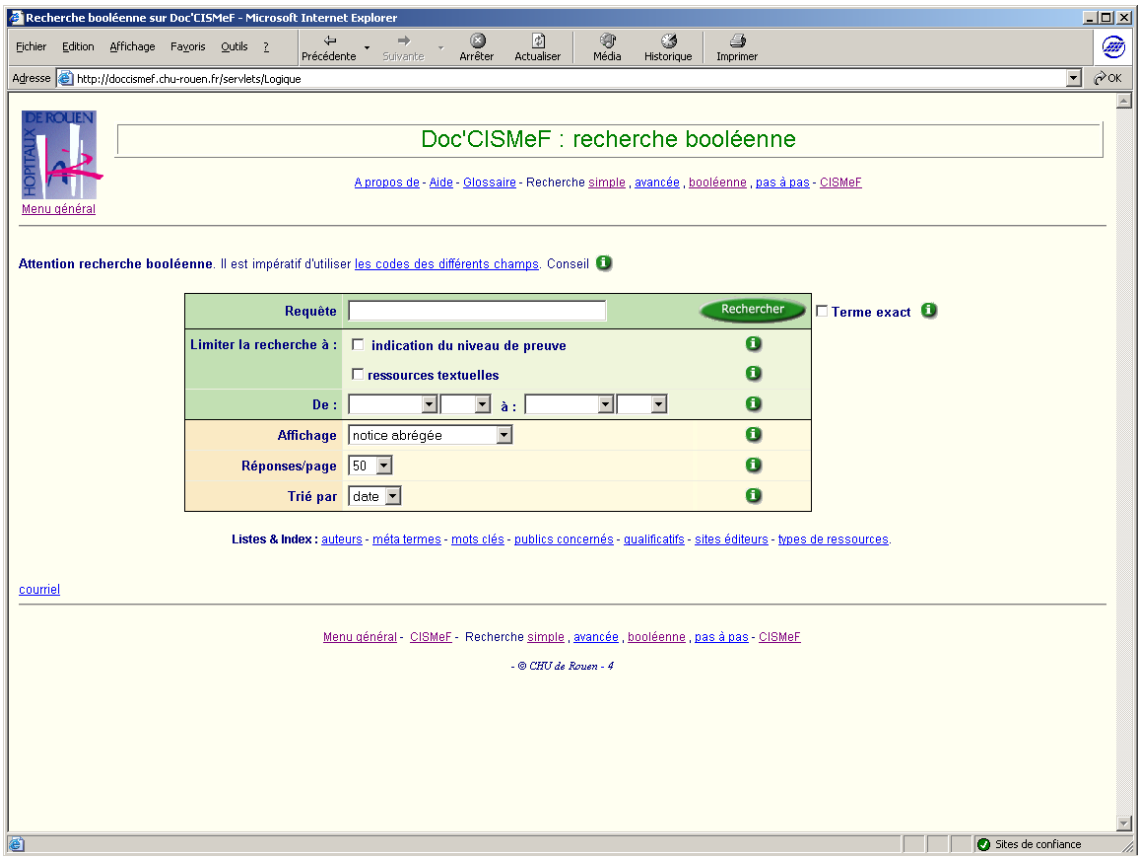


FIG.2.10.–Formulaire de recherche booléenne.

2.4.2.5 La Recherche Pas-à-Pas

Ce mode de recherche assiste l'utilisateur à chacune des étapes de sa démarche et s'apparente à une décomposition de la recherche avancée. Elle est destinée aux utilisateurs néophytes et permet, en utilisant les index proposés, d'avancer étape par étape en les reliant avec des opérateurs booléens. L'autre avantage de ce mode de recherche est d'obtenir ainsi un historique des requêtes et de réaliser des combinaisons entre elles.

2.4.2.6 Options de Recherche

2.4.2.6.1 Options par Défaut

Par défaut, les résultats se présentent sous la forme de notices descriptives abrégées comprenant : le titre, l'URL, les mots clés, les qualificatifs, le(s) type(s) de ressource et un bref résumé du contenu. Trois autres modes d'affichage des notices sont possibles (titre uniquement, titre et URL, et enfin titre, mots clés, type et URL). Quel que soit le mode d'affichage choisi, un

accès aux notices complètes des ressources est toujours proposé. Le nombre de notices par page est également paramétrable : 10, 20, 50 (choix par défaut) et 100. Les notices peuvent aussi apparaître triées par date de publication, par ordre alphabétique des titres, par type de ressource ou par pays et ville.

2.4.2.6.2 Option Arborescence

Doc'CISMeF permet également un accès selon les arborescences (paramètre affichage). De cette façon, les ressources sélectionnées ainsi que les différentes positions du mot clé demandé dans ses différentes arborescences sont présentées. Les liens vers le site de PubMed et vers les pages statiques des mots clés de CISMeF sont également présents. Doc'CISMeF indique également, pour chaque mot clé, le nombre de ressources disponibles. Il est à noter que cet affichage par arborescences présente la totalité des mots clés du thésaurus MeSH.

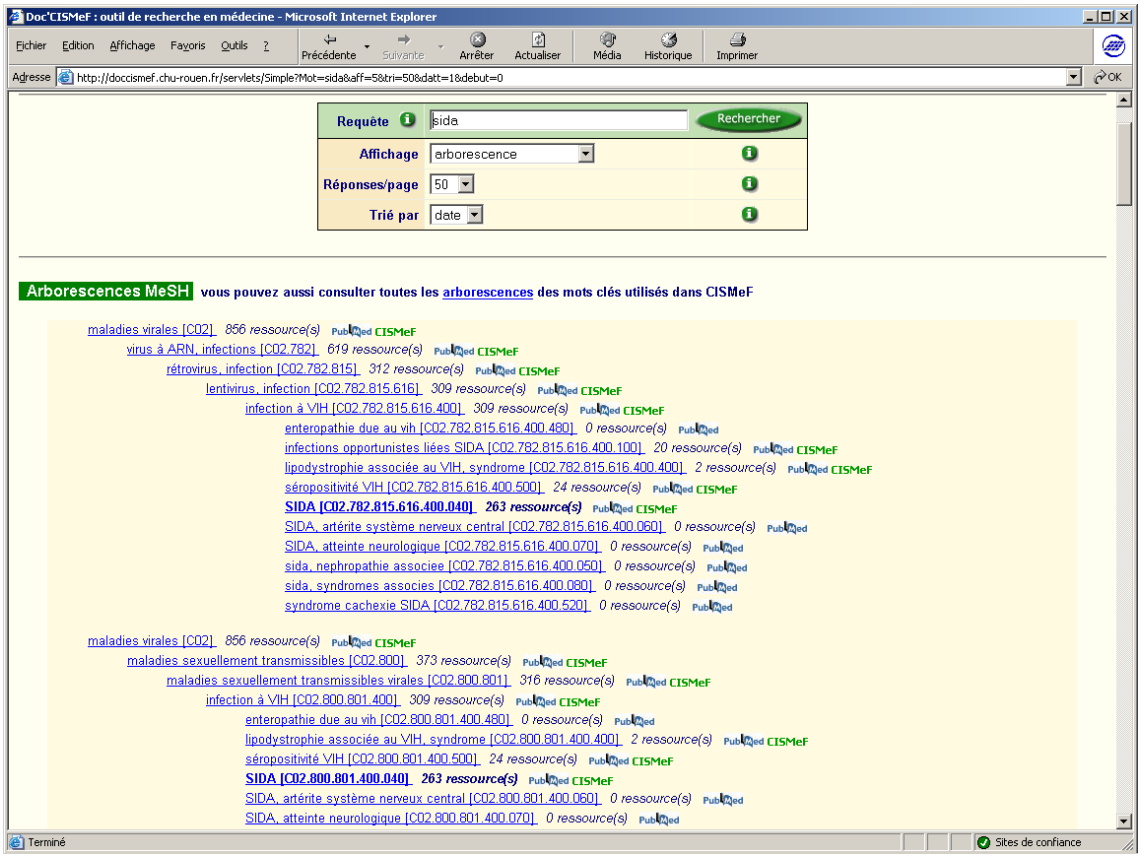


FIG.2.11.– Affichage par arborescence pour la requête *sida*.

2.4.2.6.3 Option Explosion

Le principal avantage de l'utilisation d'un thésaurus réside dans l'utilisation qui peut être faite de sa structure en arborescence dans le processus de recherche d'information et la gestion des relations hiérarchiques entre les termes. L'*explosion* permet dans le processus de recherche d'information d'étendre la recherche. Si le terme saisi fait partie de la terminologie (terme

réservé), le résultat de la recherche comprend l'ensemble des ressources indexées au terme en question, auxquelles sont ajoutées toutes les ressources indexées aux termes plus spécifiques (avec des liens directs et des liens indirects) dans toutes les arborescences des différentes hiérarchies. Par exemple, si le terme saisi est *embolie,* les résultats proposés par Doc'CISMeF comprendront les ressources indexées au mot clé *embolie* mais aussi les ressources indexées notamment aux mots clés *embolie pulmonaire, thromboembolie, embolie paradoxale*, etc... En effet, le mot clé *embolie* a pour termes spécifiques les termes suivants selon l'arborescence ci-dessous :

embolie

- *embolie amniotique*
- *embolie cholestérol*
  - *cyanose orteil, syndrome*
- *embolie gazeuse*
- *embolie graisseuse*
- *embolie pulmonaire*
- *thromboembolie*
  - *embolie et thrombose intracrânienne*
  - *embolie paradoxale*

Si le terme appartient à plusieurs niveaux du modèle d'information, Doc'CISMeF unifie les résultats des explosions de tous les niveaux. Par exemple, le terme *virologie* est un métaterme, un mot clé et un qualificatif et les résultats comprendront les explosions de toutes ces arborescences.

La politique d'indexation décidée au sein de l'équipe CISMeF consiste à indexer les ressources avec les mots clés les plus fins concernant le sujet traité. Ainsi, une recherche sans explosion sur un mot clé particulier engendrerait du silence dans les résultats. Par exemple, une recherche "non explosée" sur le terme *maladies virales* exclut les ressources concernant des maladies virales particulières pour ne renvoyer que les ressources concernant la notion générique de "maladies virales". Il peut être intéressant pour l'utilisateur, lorsqu'il formule une requête, d'obtenir toutes les ressources, incluant celles indexées avec les termes spécifiques.

Cependant, en cas d'un nombre de résultats trop important, la recherche peut être restreinte en désactivant l'option "explosion" afin d'obtenir seulement les ressources indexées spécifiquement au terme demandé.

2.4.2.6.4 Option Majeur/Mineur

Face au nombre parfois important de résultats pour une recherche, il est important de préciser la pertinence des résultats. Dans la base de données bibliographiques Medline, la notion de "MeSH Major Topic" permet de réduire une recherche aux références les plus pertinentes. La pondération des mots clés et des types de ressources permet donc d'introduire une finesse supplémentaire au cours de la recherche d'information. La pondération permet de limiter une recherche aux ressources où le terme MeSH recherché est le sujet principal de la

ressource. Par défaut, les recherches sont lancées sur tous les mots clés, qu'ils soient majeurs ou mineurs. Si le nombre de résultats est trop important, il est alors possible de restreindre les résultats aux thèmes traités de manière prépondérante dans le document, c'est-à-dire en menant la recherche sur les termes (mots clés et types de ressources) majeurs. Cette démarche permet de réduire d'un peu plus de 51 % le nombre de réponses obtenues.

2.4.3 Les Affiliations

2.4.3.1 Affiliation de Qualificatifs

Le thésaurus MeSH comprend des mots clés mais aussi 84 qualificatifs qu'il est possible d'affilier à ces mots clés. Les qualificatifs permettent de préciser le sens d'un mot clé pour en souligner un aspect spécifique. Ainsi, le qualificatif *thérapeutique* affilié au mot clé *tumeurs* permet d'indexer une ressource traitant des différentes thérapeutiques concernant le cancer. La syntaxe des couples (mot clé/qualificatif) se présente comme suit : *tumeurs/thérapeutique*. L'exploitation de cette caractéristique d'affiliation des qualificatifs à des mots clés, permet d'affiner les requêtes en ciblant précisément le thème recherché.

La hiérarchisation des qualificatifs permet d'inclure l'option d'explosion. Ainsi, le qualificatif *thérapeutique*, avec l'option explosion, permet de retrouver les ressources indexées aux qualificatifs *chirurgie*, *chimiothérapie*, *diétothérapie*, *radiothérapie*, etc…

Dans le processus de recherche, les qualificatifs n'étaient pas réellement affiliés, ils étaient 'flottants'. En effet, le traitement des requêtes ne distinguait pas le qualificatif du mot clé. Par exemple la requête *hépatite/diagnostic* était traduite en la requête booléenne *(hépatite.mr ET diagnostic.mr)*, avec *mr* mot réservé, sans faire de distinction entre le mot clé diagnostic et le qualificatif diagnostic.

Aujourd'hui, *(Soualmia & Darmoni, 2004a) (Douyère et al., 2004)*, la requête *hépatite/diagnostic* est interprétée plus précisément. La réponse à la requête regroupe tous les documents indexés au couple. La requête est traduite en requête booléenne *(hepatite.mc ET diagnostic.qu[affilié])* avec [affilié] la contrainte d'affiliation.

2.4.3.2 Affiliation de Types de Ressources

Dans la terminologie CISMeF, outre les mots clés et les qualificatifs qui décrivent le sujet de la ressource, les types de ressources concernent la nature des informations véhiculées. Ce concept s'applique à la ressource dans sa globalité et il est impossible, jusqu'à présent, d'appliquer un type de ressource à une partie de document. Il n'est pas possible d'utiliser les types de ressources pour spécifier un aspect particulier d'un mot clé ou d'un couple (mot clé/qualificatif). L'objectif de l'affiliation des types de ressources est de pouvoir faire une recherche plus précise permettant de trouver, par exemple, des images d'une pathologie particulière. Jusqu'à présent, la pathologie était représentée par une indexation par mots clés, et le type de la ressource concernait la ressource dans sa globalité, en ne tenant pas compte de la présence du type d'images dans le document (thèse de Filip Florea 2003-2006 concernant l'indexation bimodale texte-image).

91 types de ressources correspondant à des images médicales ont d'abord été créés. Cette liste est en partie dérivée de l'arborescence du mot clé MeSH *diagnostic par imagerie*, et elle a été revue et mise à jour par un expert en imagerie médicale *(Soualmia et al., 2004)*. L'utilisation de cette nouvelle liste de types de ressources pour l'indexation de documents médicaux contenant des images, permet de rendre le processus d'indexation plus précis. Ainsi, par exemple, une ressource pédagogique concernant la *lithiase cholédocienne* et contenant des images échographiques de cette pathologie sera indexée avec le mot clé *lithiase cholédocienne* et les types de ressources *échographie* et *matériel enseignement*. Si elle contient un paragraphe décrivant l'échographie de la lithiase cholédocienne, la ressource sera indexée avec le mot clé *lithiase cholédocienne* auquel le qualificatif *échographie* sera affilié (soit le couple *lithiase cholédocienne / échographie*) ainsi que le seul type de ressource *matériel enseignement*.

La lacune de ce système est que les types de ressources se trouvent toujours être "flottants" comme l'étaient les qualificatifs avant qu'existe la contrainte d'affiliation : ils concernent l'ensemble des mots clés sélectionnés pour l'indexation d'une ressource, un même document pouvant concerner plusieurs pathologies différentes, et donc être indexé avec des mots clés très divers. Par exemple, un cours est indexé aux mots clés *cardiomyopathie hypertrophique, cardiopathie congestive* et *cardiomyopathie restrictive* et aux types de ressource *radiographie* et *matériel enseignement*. Le type de ressource *radiographie* a été choisi parce que la ressource contient une image concernant la cardiopathie hypertrophique. Lors d'une recherche, un utilisateur peut vouloir trouver une ressource présentant une radiographie de la cardiopathie congestive et formuler la requête *cardiopathie congestive.mc ET radiographie.tr* qui renverra notre ressource précédemment décrite, alors que l'image contenue concerne la cardiopathie hypertrophique.

Pour affiner la procédure d'indexation et de recherche d'information, l'idéal est de pouvoir associer un type de ressource à un mot clé ou à un couple (mot clé/qualificatif), pour ainsi composer un triplet [(mot clé/qualificatif)\type de ressource] *(Soualmia et al., 2004)*. Ainsi, une ressource contenant des images radiographiques de la cardiopathie congestive sera indexée avec le couple *cardiopathie congestive/diagnostic*. Si l'image permet le diagnostic de la cardiopathie congestive, la ressource sera indexée avec le triplet [(*cardiopathie congestive/diagnostic)\radiographie*]. L'extension du couple (mot clé/qualificatif) au triplet [(mot clé/qualificatif)\type de ressource] améliore le processus de recherche d'information, la requête "(mot clé/qualificatif) ET type de ressource" étant moins pertinente que la requête "(mot clé/ qualificatif)\type de ressource"[18].

L'ambiguïté de certaines requêtes doit être gérée. Ainsi, si un utilisateur formule la requête *diagnostic de la lithiase cholédocienne par échographie*, il est impossible de savoir si l'utilisateur souhaite un texte expliquant l'échographie dans le but du diagnostic de la lithiase cholédocienne, s'il recherche ou une image échographique permettant le diagnostic de la lithiase cholédocienne. Dans le doute, pour l'instant, les résultats proposés englobent pour l'instant une union des deux, à savoir les textes explicatifs sur la technique d'imagerie ainsi que les images elles-mêmes.

2.4.4 Les Requêtes Préformatées

2.4.4.1 Les Stratégies de Recherche

[18] la gestion des affiliations des types de ressources a été mise en ligne début Octobre 2004

Le thesaurus MeSH ne peut prétendre à l'exhaustivité. En effet, il peut arriver qu'une notion ne soit pas représentée en tant que telle par un mot clé MeSH. Certains outils tels que Pubmed procèdent, lors de la saisie d'un terme quelconque dans l'interface de recherche, à un appariement d'un terme non-MeSH avec un terme MeSH à l'aide d'une table de correspondance.

Dans le CISMeF, des "stratégies de recherche" (n=22) *(Soualmia & Darmoni, 2004)* associent plusieurs termes pour représenter un concept médical. Par exemple, la notion de chirurgie urologique n'a pas de mot clé MeSH qui lui correspond. Si un utilisateur saisit l'expression "chirurgie urologique" dans l'interface de recherche Doc'CISMeF, la requête suivante est automatiquement générée : *[(intervention chirurgicale.mc OU chirurgie.qu) ET urologie.mt]* signifiant que nous recherchons les ressources indexées avec le mot clé *intervention chirurgicale* ou le qualificatif *chirurgie*, et avec un terme (mot clé, qualificatif ou type de ressource) appartenant au métaterme *urologie*.

2.4.4.2 CISMeF-Patients

CISMeF-*patients (Soualmia et al., 2002)(Soualmia et al., 2002d)(Soualmia et al., 2003b),* est un sous catalogue de CISMeF. Il a été crée pour répondre à un besoin de recherche d'information de santé pour un utilisateur lambda, comme les patients, leur famille et le grand public. Le modèle de CISMeF est difficile à comprendre pour les non professionnels de santé. De ce fait la recherche d'information, via CISMeF, doit être simplifiée pour ce type d'utilisateurs. CISMeF-*patients* est une vue particulière sur la terminologie : elle correspond au métaterme *patient*. Ce métaterme possède ses propres arborescences de mots clés. Pour permettre une navigation plus simple, certains termes de la terminologie CISMeF (métatermes, mots clés, qualificatifs et types de ressources) ont été synonymisés avec des termes employés de manière plus courante. Par exemple, *manie* est le synonyme courant de *trouble bipolaire*.

CISMeF-*patients* comporte un index thématique de spécialités médicales. Les arborescences sont visualisées de manière simplifiée. Le sous catalogue est composé de ressources d'information écrites à destination des patients par des professionnels de santé, des sociétés savantes ou des institutions médicales.

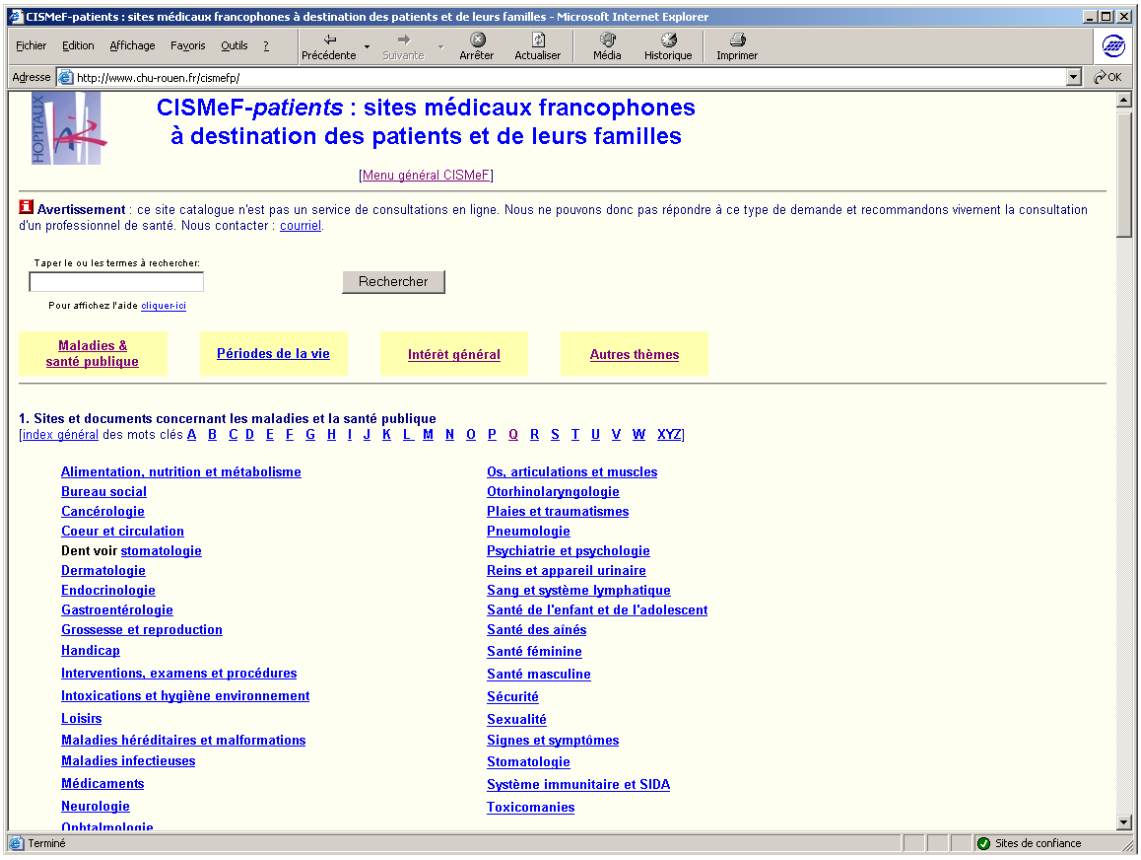


FIG.2.12.–Interface simplifiée de CISMeF-*patients*

Pour la recherche de ressources, des requêtes pré formatées sur Doc'CISMeF sont générées lorsqu'un utilisateur clique sur un des mots clés de CISMeF-*patients*. De ce fait, l'utilisateur n'est ni obligé de connaître le mot clé MeSH ni le moteur Doc'CISMeF. La requête correspondante qui est lancée est : (patient.mt ET [mot clé]). Il faut noter qu'il a été difficile pour l'équipe de choisir la requête pré formatée : l'alternative était soit une requête avec le métaterme *patient*, soit une requête plus précise limitée aux types de ressources *patient*. Le métaterme *patient* est en association avec des mots clés MeSH (exemple : *éducation patient*, *soutien patient*, *malade*...) et des types de ressources (exemple : *association patients*, *information patient*...etc).

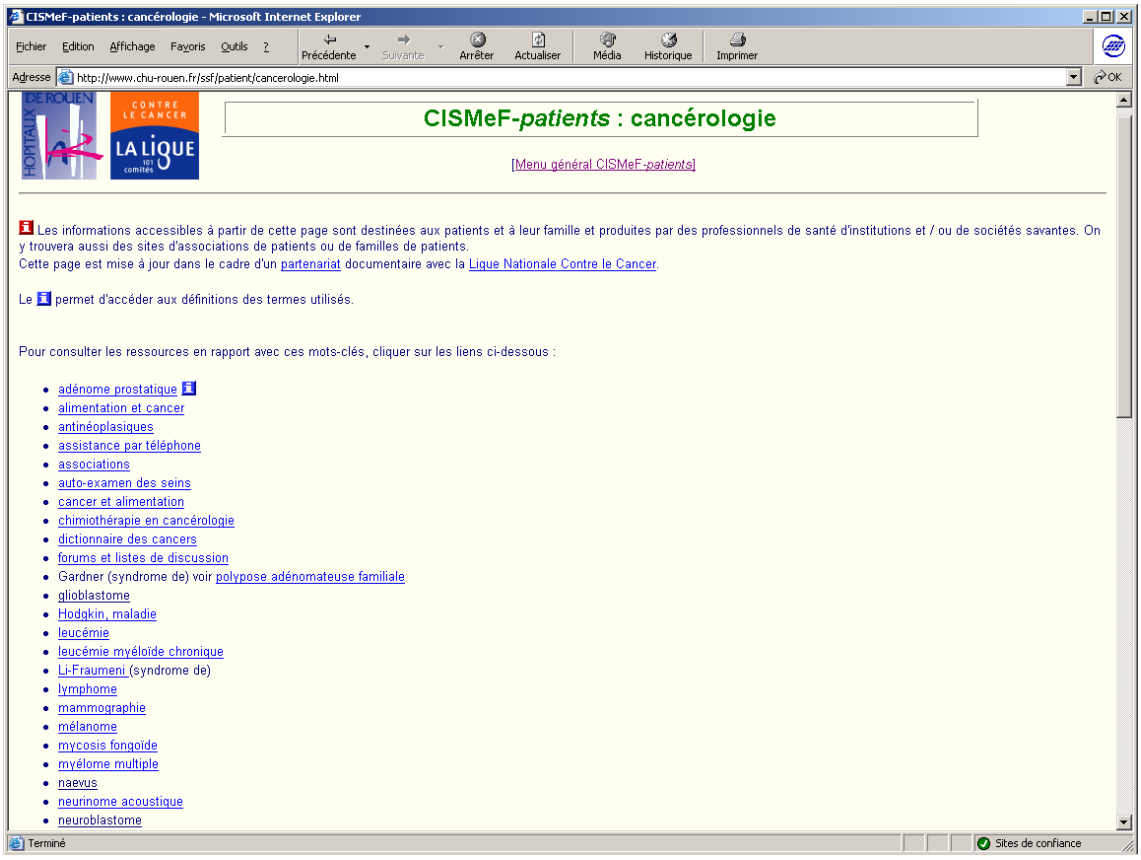


FIG.2.13.–Simplification d'une arborescence.

Un autre accès est possible via les périodes de la vie[19]. Les périodes sont : naissance, enfance, adolescence, age adulte et 3eme âge. L'accès est plus spécifique : il permet d'obtenir directement les ressources correspondantes par génération automatique de requête sur Doc'CISMeF. Les requêtes générées sont les suivantes :

nouveau-né : ((nouveau-ne.mc OU nourrisson.mc) ET patient.tr) ;

enfance : ((enfant.mc OU enfant age pre-scolaire.mc) ET patient.tr) ;

adolescence : (adolescence.mc ET patient.tr) ;

age adulte : (adulte.mc ET patient.tr)

3°age : (sujet age.mc ET patient.tr).

La principale majorité des ressources à destination des patients concerne la cancérologie. Pour indexer des ressources de ce thème, un partenariat existe avec la Ligue Nationale Contre le Cancer (LNCC)[20], une des principales associations de patients en France. La collaboration a été bénéfique pour les deux parties : d'une part l'indexation de ressources de la ligue, et d'autre part la détermination des synonymes utilisés dans la langue courante.

[19] à la manière des 'life events' de Healthinsite http://www.healthinsite.gv.au
[20] http://www.ligue-cancer.asso.fr

Les évolutions de CISMeF-*patients* se sont inspirées de Medline-*plus* (développé en 1998) qui est un catalogue anglophone similaire dédié aux patients, mis en place par la NLM. Il référence des sites de qualité concernant les maladies les plus courantes. En priorité sont recensées les ressources de la NLM et de la NIH (Nationale Institute of Health) et les ressources nationales des Etats-Unis. La principale différence entre Medline-*plus* et CISMeF *patient* c'est la structure de la terminologie. En effet, CISMeF et CISMeF-*patients* partagent la même structure et le même outil de recherche, alors que pour Medline-*plus*, une autre terminologie a été construite.

Pour toute requête de ce type, une recherche complémentaire sur le site de Medline-*plus*[21] est proposée. Elle permet aux utilisateurs d'élargir leur prospection aux sites non francophones.

L'intérêt d'une seule terminologie est de pouvoir modifier la requête en modifiant le type de ressource : par exemple enseignement ou encore ligne directrice. Ainsi, dans la même optique, lorsqu'une requête a pour but d'obtenir des recommandations destinées aux professionnels de santé, une recherche complémentaire est proposée vers le site de la National Guideline Clearing House[22].

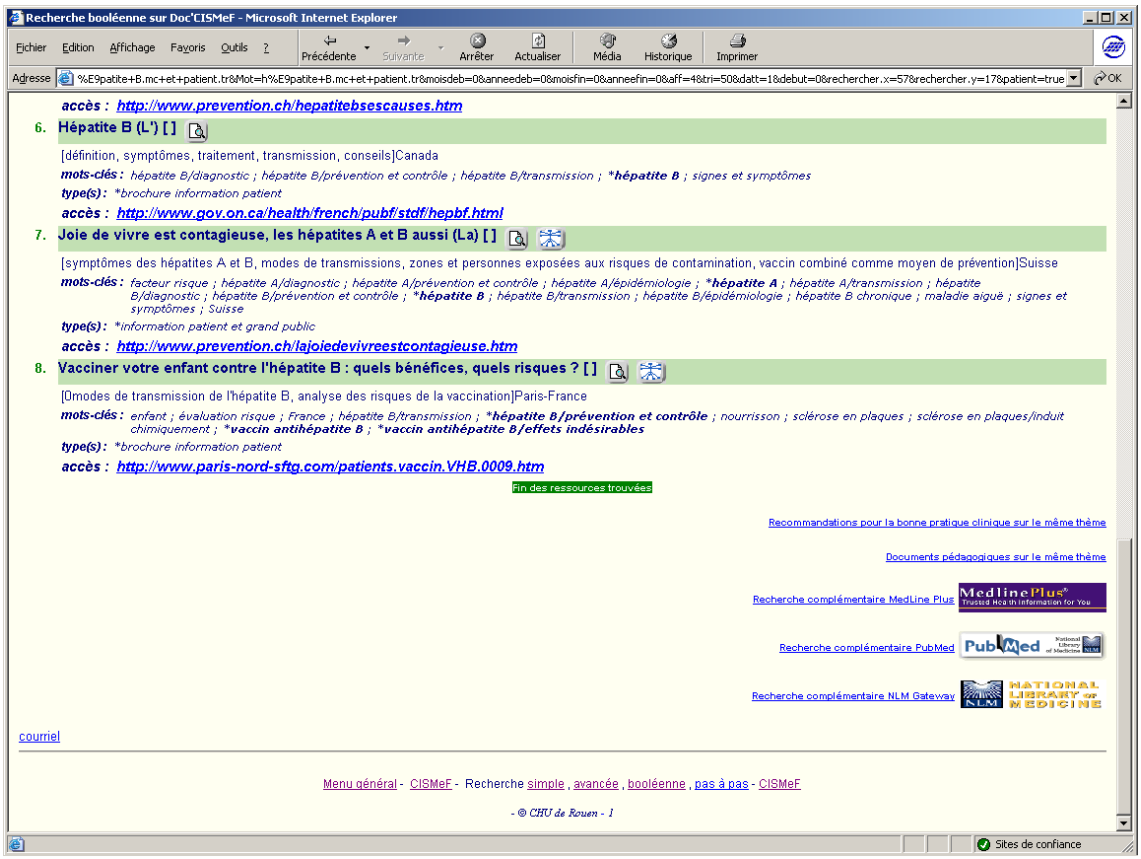


FIG.2.14.–Exemple de recherches complémentaires.

CISMeF ne proposant un accès qu'à des sites et documents francophones présents sur l'Internet, l'utilisateur peut avoir besoin de prolonger ses investigations vers d'autres types de sites-catalogues tout en gardant la même requête. L'avantage de l'utilisation du thésaurus MeSH

[21] http://www.nlm.nih.gov/medlineplus/medlineplus.html
[22] http://www.guideline.gov/

est que les requêtes formulées dans CISMeF peuvent l'être dans d'autres catalogues utilisant aussi le thésaurus MeSH. Ainsi, pour chaque requête formulée dans CISMeF et donnant lieu à des résultats, Doc'CISMeF propose de lancer une requête identique sur la base de données bibliographiques Medline via le site de Pubmed[23], ainsi que sur le site de la NLM Gateway[24], en adoptant automatiquement la syntaxe utilisée sur ces sites, et en utilisant le terme MeSH correspondant en anglais. L'utilisateur peut ainsi accéder aux résultats de sa recherche dans Medline et sur les sites de la NLM. Par exemple la requête *hépatite* est traduite dans la syntaxe de Medline par *hepatitis [MeSH Terms] or hepatitis [subheadings]* avec *hepatitis* le correspondant en anglais du terme *hépatite*.

2.4.4.3 Le Projet Cogni-CISMeF

L'analyse des requêtes des utilisateurs n'est pas contextualisée et aucune prise en compte de l'utilisateur n'est effectuée pour comprendre son raisonnement, son cheminement de pensée, et analyser sa démarche lors de la formulation de sa requête. L'interaction avec l'utilisateur est une direction à étudier pour tenter d'améliorer la recherche d'information sur l'Internet en assistant l'utilisateur dans sa quête d'information. Pour arriver à atteindre cet objectif une solution est de comprendre les mécanismes cognitifs sous-jacents.

Le but du projet Cogni-CISMeF[25] (2004-2007) est d'étudier les processus cognitifs mis en jeu lors de la construction d'une requête dans Doc'CISMeF afin d'intégrer un module de dialogue avec l'utilisateur qui l'amènera à préciser sa demande, dans le but d'identifier son intention et de la traduire en une requête utilisant la terminologie CISMeF.

L'expérimentation consiste à enregistrer des conversations téléphoniques entre un utilisateur (médecin ou patient) et un documentaliste (expert de la terminologie CISMeF) qui amène l'utilisateur à préciser sa demande pour construire la requête à formuler pour interroger CISMeF de façon pertinente. Le but est d'analyser la structure des conversations téléphoniques ainsi obtenues et d'étudier un certain nombre d'indices discursifs (vocabulaire employé, reformulations, commentaires, hésitations, expression implicite ou explicite des intentions des utilisateurs, protocoles et enchaînements conversationnels…) en vue de construire un profilage des utilisateurs. Cette analyse permettra à terme de proposer un ou plusieurs modèles computationnels afin de les implanter dans le système de recherche d'information de CISMeF.

[23] http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=PubMed
[24] http://gateway.nlm.nih.gov/gw/Cmd
[25] Projet en collaboration avec le laboratoire PSI et financé par le CNRS dans le cadre Programme Interdisciplinaire – "Traitement des Connaissances, Apprentissage et NTIC"

Grâce à la modélisation des interactions avec l'utilisateur, l'objectif opérationnel de ce projet est de créer la maquette d'un module de dialogue intégré dans CISMeF, qui sera mis à la disposition des utilisateurs sur l'Internet, si les résultats de l'évaluation du système sont satisfaisants.

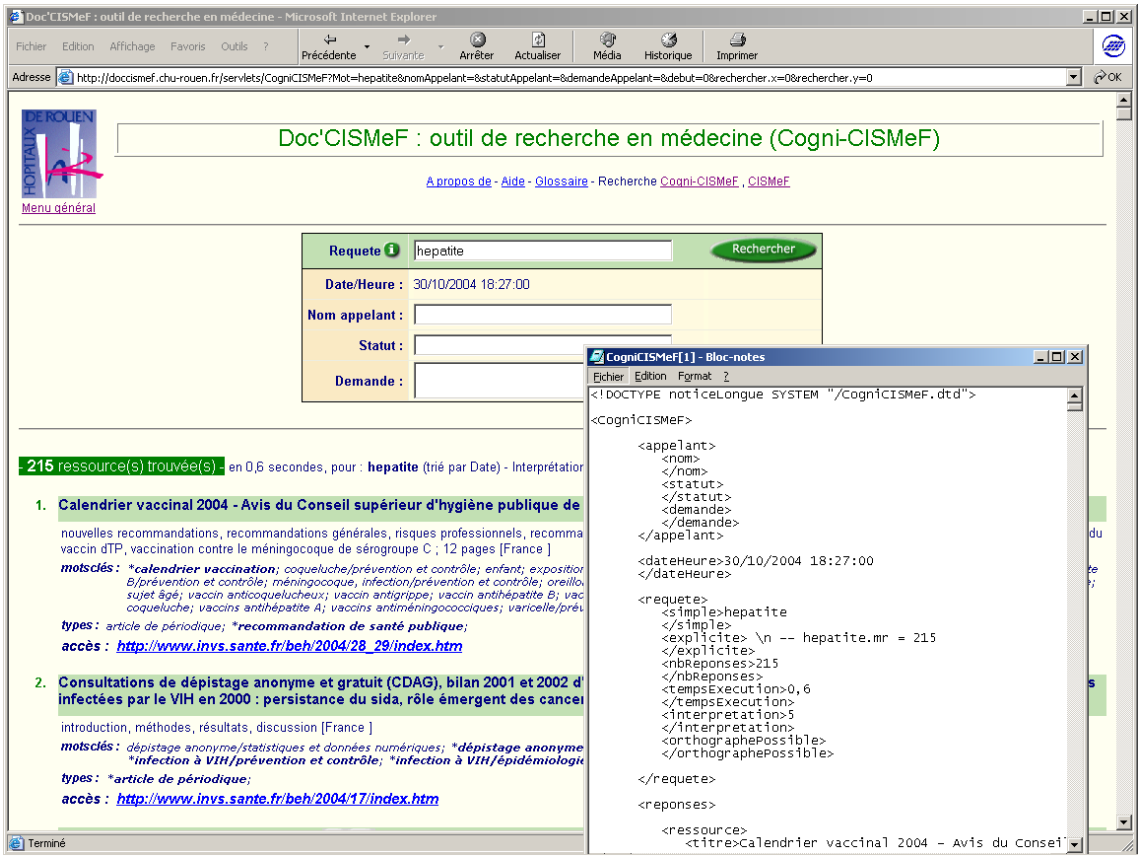


FIG.2.15.–Interface du module de Cogni-CISMeF permettant d'enregistrer les sessions sous la forme de fichiers XML qui seront étudiés pour construire le processus cognitif mis en jeu.

2.4.5 Catégorisation des Documents

L'évolution de la politique d'indexation vers une indexation fine a pour conséquence l'attachement d'un nombre parfois très important de mots clés à une ressource (jusqu'à 301 mots clés pour une seule et même ressource). Ces mots clés renvoient à des notions très spécifiques et souvent pointues. Nous avons mis en place un algorithme permettant d'obtenir une classification intuitive présentant la liste des domaines concernés dans un document par ordre d'importance, le but étant de dégager les principaux thèmes abordés.

L'algorithme de catégorisation se fonde sur l'expertise des documentalistes. Il utilise tous les liens sémantiques existants entre métatermes - mots clés - qualificatifs et types de ressources afin de dériver la liste des spécialités auxquelles appartient un document donné. La liste est ordonnée en allant de la spécialité la plus importante à la moins importante. Un document étant indexé par plusieurs mots clés, qualificatifs et types de ressources, plusieurs métatermes seront déduits. De plus, un terme peut être lié à plusieurs métatermes. Tous les métatermes sont

retenus. Par exemple, le mot clé *alcoolisme* permettra de faire apparaître les spécialités *psychiatrie* et *toxicologie*, tandis que le mot clé *main* fera apparaître la spécialité *anatomie*.

TAB.2.1.–Notations utilisées dans l'algorithme SPECIALITES-DOCUMENT

MC	Ensemble des mots clés du document D : $MC=\cup_{i=1..m} Mc_i / Mc_i \in D$;
QU	Ensemble des qualificatifs du document D : $QU=\cup_{i=1..q} Qu_i / Qu_i \in D$;
TR	Ensemble des types de ressource du document D : $TR=\cup_{i=1..t} Tr_i / Tr_i \in D$;

$$\text{Majeur}(t_i) = \begin{cases} 1 \text{ si majeur} \\ \\ 0 \text{ sinon} \end{cases} ; t_i \in \{MC \cup QU \cup TR\}$$

Spécialités_Terme (t_i) = liste des spécialités d'un terme ; $t_i \in \{MC \cup QU \cup TR\}$

Algorithme SPECIALITES-DOCUMENT

Entrée : Document D ; Ensembles MC, QU et TR ;

Sortie : Liste des spécialités $S = \cup_{i=1..k} S_k$;

Chaque spécialité S_k possède deux champs : Score_Majeur et Score_mineur.

Début

Pour ($i \leftarrow 1$; $i \le m$; i++) **faire** $S \leftarrow S \cup_{\text{distinct}}$ Spécialités-Terme (Mc_i) ;
Pour ($i \leftarrow 1$; $i \le q$; i++) **faire** $S \leftarrow S \cup_{\text{distinct}}$ Spécialités-Terme (Qu_i) ;
Pour ($i \leftarrow 1$; $i \le t$; i++) **faire** $S \leftarrow S \cup_{\text{distinct}}$ Spécialités-Terme (Tr_i) ;
Pour chaque $S_n \in S$ **faire** S_n.Score_Majeur $\leftarrow 0$; S_n.Score_mineur $\leftarrow 0$;
Pour chaque spécialité $S_n \in S$ **faire**
  Pour ($i \leftarrow 1$; $i \le m$; i++) **faire**
    Si $S_n \in$ Spécialités_Terme (Mc_i)
      Si Majeur (Mc_i) **alors** S_n.Score_Majeur $\leftarrow S_n$.Score_Majeur + 1 ;
      Sinon S_n.Score_mineur $\leftarrow S_n$.Score_mineur + 1 ;
    FinSi
  FinPour
Pour ($i \leftarrow 1$; $i \le q$; i++) **faire**
  Si $S_n \in$ Spécialités_Terme (Qu_i)
    Si Majeur (Qu_i) **alors** S_n.Score_Majeur $\leftarrow S_n$.Score_Majeur + 1 ;
    Sinon S_n.Score_mineur $\leftarrow S_n$.Score_mineur + 1 ;
  FinSi
FinPour
Pour ($i \leftarrow 1$; $i \le t$; i++) **faire**
  Si $S_n \in$ Spécialités_Terme (Tr_i)
    Si Majeur (Tr_i) **alors** S_n.Score_Majeur $\leftarrow S_n$.Score_Majeur + 1 ;
    Sinon S_n.Score_mineur $\leftarrow S_n$.Score_mineur + 1 ;
  FinSi
FinPour
Retourner S ;

Fin

Une évaluation a été menée *(Névéol et al., 2004)* pour établir une comparaison de la classification proposée par l'algorithme avec une classification établie manuellement par un

documentaliste de l'équipe, concernant un échantillon de 123 ressources sélectionnées au hasard dans le corpus de documents référencés par CISMeF. Les résultats de cette évaluation montrent un taux de précision de 80.75 % et un taux de rappel de 93.41 % (donc un bruit de 19.25 % et un silence de 6.59 %).

L'analyse des résultats permet de dégager une liste de termes de la terminologie CISMeF pour lesquels il est nécessaire d'instaurer des liens vers des métatermes existants voire même de créer des métatermes adaptés à cet effet. Dix-huit métatermes ont été créés à la suite de l'analyse des résultats. Ainsi, par exemple, les mots clés *voyage* et *médecine tropicale* ne sont liés à aucun métaterme et ne permettent donc de faire apparaître aucune spécialité médicale dans la classification de la ressource. La conséquence directe de l'analyse de ces résultats de l'évaluation a donc été la création du métaterme *médecine tropicale*. Pour tenter de réduire le silence de l'algorithme de classification, certains manques de la terminologie CISMeF ont été comblés en étendant sa couverture par le biais de la création de métatermes et de l'enrichissement de métatermes existants avec de nouvelles relations avec des termes MeSH ou des types de ressources.

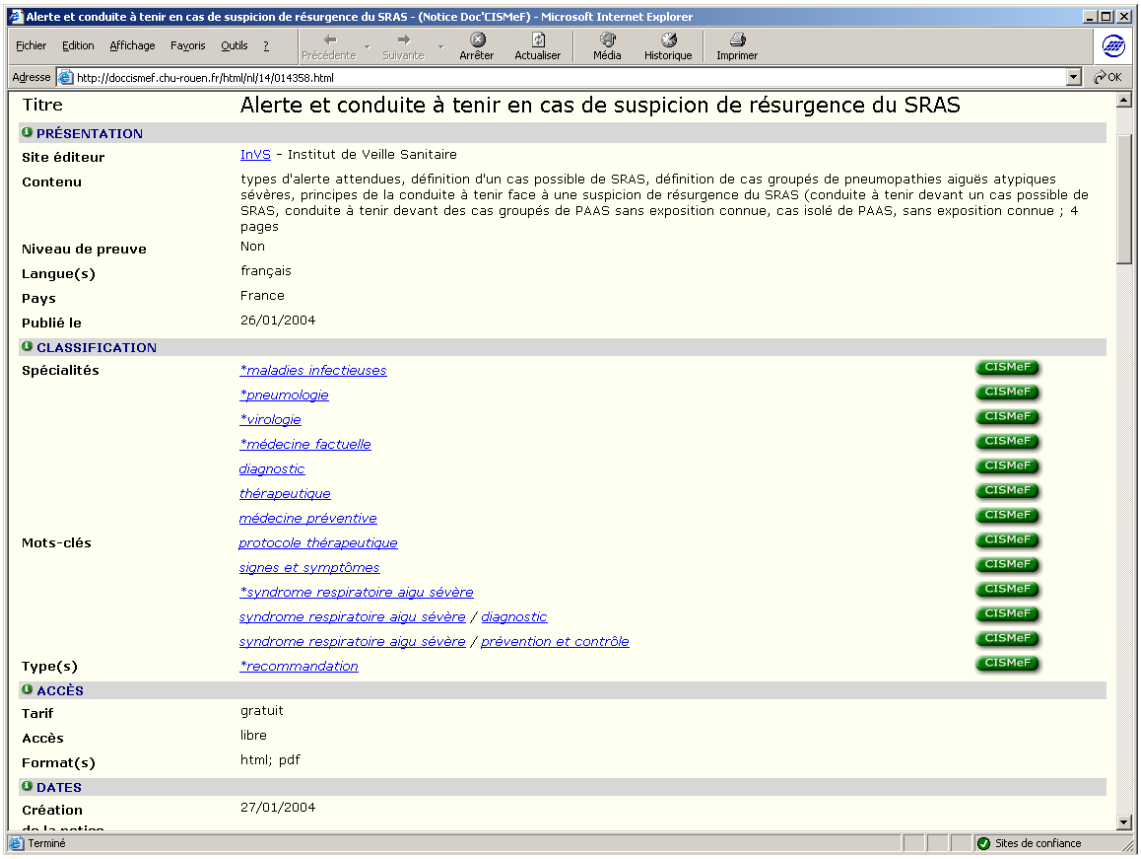


FIG.2.16.–Exemple de Notice avec la classification des spécialités.

2.5 Quelques Problèmes Rencontrés

L'utilisation d'une terminologie telle que celle du CISMeF est problématique lorsque l'utilisateur n'en a pas une connaissance approfondie. En effet, les utilisateurs potentiels de

CISMeF sont les étudiants et les professionnels mais aussi le grand public (rappelons que 40 à 45% des utilisateurs consultent les pages de CISMeF via le moteur de recherche Google). Ce type d'utilisateurs n'ont pas l'habitude de manipuler la terminologie MeSH, même si ce thésaurus est un des plus utilisés dans le monde professionnel médical.

La ou les requêtes saisies par l'utilisateur correspondent rarement à la formulation exacte effectivement utilisée pour l'indexation. Nous avons analysé les logs sur le moteur Doc'CISMeF avec le type de requête employé ainsi que le nombre de réponses obtenu entre le 15/08/2002 et le 06/02/2003. Le nombre de requêtes logiques est important vu que toutes les requêtes simples sont transformées en requêtes logiques explicites depuis octobre 2003[26]. Il en ressort que 33.5% des requêtes n'ont pas de réponse. Nous n'analyserons que les requêtes logiques et simples car elles seules emploient des interfaces de saisie des requêtes.

TAB.2.1.–Analyse des requêtes des utilisateurs du 15/08/2002 au 6/02/2003.

	Requêtes		**Requêtes Nulles**	
Type de Requête	**Nombre**	**Pourcentage**	**Nombre**	**Pourcentage**
Avancée	628 018	41,24 %	143 684	28,15 %
Logique	529 876	34,80 %	252 479	49,46 %
Simple	362 715	23,82 %	113 209	22,18 %
Autres	2 167	00,14 %	1 106	00,21 %
Total	1 522 776		510 478	

Une analyse plus fine des requêtes simples nous a permis de déduire que 12.01% des réponses sont nulles non pas parce que ce sont des requêtes erronées, mais car aucune ressource n'est rattachée au terme réservé et à ses fils directs et indirects.

TAB.2.2.–Répartition des requêtes simples à 0 réponse.

	Nombre	**Pourcentage**
Expression reconnue	13 597	12,01 %
Expression non reconnue	99 612	87,99%
Total	113 209	

C'est donc une difficulté réelle à faire correspondre des requêtes en langage naturel aux exigences de la terminologie. Le but est donc d'optimiser le processus de recherche d'information, notamment par le biais de l'amélioration du traitement des requêtes, dans le but de leur adéquation à la terminologie. Une solution peut être l'emploi de synonymes.

Dans ce contexte, *(Voorhees, 1994) (Mihalcea et al., 2000) (Baziz et al., 2003)* ont utilisé WordNet pour l'expansion des requêtes en ajoutant des termes reliés sémantiquement aux termes des requêtes d'origine. La relation sémantique de base utilisée est la synonymie. Cette

[26] En 2002, les statistiques d'utilisation de CISMeF, désignaient la recherche simple comme la technique de recherche d'information la plus employée : simple 75%, avancée 15%, booléenne 7% et pas à pas 3%.

technique nécessite de désambiguïser les termes dans les requêtes initiales. Ces auteurs s'accordent à dire que cette méthode peut être intéressante si la désambiguïsation s'avère performante. *(Guarino et al., 1999)* ont montré le rôle positif des ontologies linguistiques dans leur système OntoSeek, pour l'expansion de requêtes sur les catalogues de produits et les pages jaunes, en sélectionnant (manuellement) les synsets de WordNet appropriés et leurs catégories. *(Gonzalo, 1998)* a proposé une méthode d'indexation des documents s'appuyant sur les concepts d'une base de données sémantique. Elle améliore la précision de 25 %.

2.5.1.1 Ajout d'autres Types de Synonymes

Le thésaurus MeSH étant un produit américain, les synonymes créés ne sont pas toujours appropriés et adaptés au vocabulaire couramment employé en français. Ainsi, par exemple, selon le thésaurus MeSH, le mot clé *encéphalopathie bovine spongiforme* a pour synonyme MeSH uniquement *encéphalopathie spongiforme bovine*. Le synonyme CISMeF[27] *vache folle* a été ajouté à la terminologie. De même que *daltonisme* est le synonyme d'*achromatopsie* ou encore *fausse couche* est le synonyme d'avortement *spontané*.

A ce jour, 1702 synonymes de mots clés ont été créés. Cette opération a été possible grâce à l'analyse des requêtes fréquentes des utilisateurs qui restaient sans réponse. Un partenariat a été mis en place avec l'INSERM pour prévoir l'introduction éventuelle de ses synonymes dans la version française du thésaurus MeSH, dans le cadre du projet VUMeF (Vocabulaire Unifié Médical Français), un projet initié dans le cadre du RNTS (Réseau National des Technologies pour la Santé) *(Darmoni et al., 2003b)*. Il existe également des synonymes pour les qualificatifs. Par exemple, pour le qualificatif *thérapeutique,* le synonyme *traitement* a été créé. A ce jour, 6 synonymes de qualificatifs sont présents dans la terminologie CISMeF. Ces nouvelles relations de synonymie permettent d'élargir les possibilités au niveau de la recherche d'information en augmentant la pertinence des résultats lors de l'utilisation de l'interface de recherche de Doc'CISMeF.

2.5.1.2 Utilisation de Connaissances

Nous avons constaté dans des travaux récents *(Soualmia & Darmoni, 2004b) (Soualmia & Darmoni, 2004c)* que l'utilisation conjointe de connaissances morphologiques et de connaissances découvertes à partir des documents de la base de données du CISMeF permettent de corriger mais également de préciser et d'étendre les requêtes des utilisateurs. Le détail de nos approches fait l'objet des Chapitres 3 et 4.

BIBLIOGRAPHIE

(*Baker, 2000*) T. BAKER (2000). A Grammar of Dublin Core. *Digital-Library Magazine,* vol 6 n°10.

(*Baziz et al., 2003*) BAZIZ M., AUSSENAC-GILLES N., BOUGHANEM M. (2003) Exploitation des Liens Sémantiques pour l'Expansion de Requêtes dans un Système de Recherche d'Information. *Congrès INFORSID 2003*.

[27] synonymes créés par l'équipe CISMeF et utilisables uniquement dans le catalogue

(*Dahamna & Soualmia, 2002*) DAHAMNA B., SOUALMIA LF. (2002) Spécifications formelles du système d'information de CISMeF. Rapport interne.

(Darmoni et al., 1999) DARMONI SJ., LEROUX V., THIRION B., SANTAMARIA P., GEA M. (1999) Netscoring : critères de qualité de l'information de santé sur Internet. *Les enjeux des industries du savoir*, pp. 29-44.

(*Darmoni et al., 2001*) DARMONI, SJ., THIRION, B., LEROY, JP., DOUYERE, M., LACOSTE, B., GODARD, G., RIGOLLE I., BRISOU, M., VIDEAU, S., GOUPY, E., PIOT, J., QUERE, M., OUAZIR, S. AND ABDULRAB, H. (2001). A Search Tool based on 'Encapsulated' MeSH Thesaurus to Retrieve Quality Health Resources on the Internet. *Medical Informatics & the Internet in Medicine*, 26 (3) :165-178.

(*Darmoni et al., 2003 a*) DARMONI SJ., AMSALLEM E., HAUGH MC. LUKACS B., CHALHOUB C., LEROY JP. (2003) Level of evidence as a future gold standard for the content quality of health resources on the Internet. *Methods of Information in Medicine, 2003*, vol.42, n°3, pp.200-225.

(*Darmoni et al., 2003 b*) DARMONI SJ, JAROUSSE E, ZWEIGENBAUM P ET AL. (2003) VuMeF : extending the French involvement in the UMLS Methathesaurus : *AMIA 2003*, pp.824.

(*Douyère et al., 2004*) DOUYÈRE M., SOUALMIA LF., NÉVÉOL A., ROGOZAN A., DAHAMNA B., LEROY JP., THIRION B., DARMONI SJ. (2004) Enhancing the MeSH thesaurus to retrieve French online health resources in a quality-controlled gateway. *Health Information and Libraries Journal* 2004; in press.

(*Gonzalo, 1998*) GONZALO J., VERDEJO F., CHUGUR I., CIGARRAN J. (1998) Indexing with Wordnet synsets can improve text retrieval. *COLING/ACL '98 Workshop on Usage of WordNet for NLP,* pp.38-44.

(*Guarino et al., 1999*) GUARINO N., MASOLO C., VETERE G. (1999) OntoSeek : content-based access to the Web. *IEEE Intelligent Systems.*

(*Moldovan & Mihalcea, 2000*) MOLDOVAN DI., MIHALCEA R.(2000) Improving the search on the Internet by using WordNet and lexical operators. *IEEE Inter. Comp.* 4(1) 34-43.

(Névéol et al., 2004) NÉVÉOL A., SOUALMIA LF., DOUYÈRE M., ROGOZAN A., THIRION B., DARMONI SJ. (2004) Using CISMeF MeSH Encapsulated Terminology and a Categorization Algorithm for Health Resources. *International Journal of Medical Informatics*, Volume 73 Issue 1. pp 57-64.

(*Soualmia & Darmoni 2003 a*) SOUALMIA LF., DARMONI SJ. (2003) Une Terminologie Orientée Ontologie pour la Recherche d'Information sur la Toile. *Journées Francophones de la Toile*, pp. 185-194.

(*Soualmia et al., 2003 b*) SOUALMIA LF., DARMONI SJ., THIRION B., DOUYÈRE M. (2003 b) Modelisation of Health Consumer Information in a Quality -Controlled Gateway. *MIE, Medical Informatics Europe*, pp.701-706.

(*Soualmia & Darmoni 2004 b*) SOUALMIA LF., DARMONI SJ. (2004) Correcting and Refining Users Queries: the Contribution of Morphological Knowledge and Association rules. *Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU'2004*, pp 2059-2066.

(*Soualmia & Darmoni 2004 c*) SOUALMIA LF., DARMONI SJ. (2004) Combining Knowledge-based Methods to Refine and Expand Queries in Medicine. *FQAS'2004, Flexible Query Answering Systems. Lectures Notes in Artificial Intelligence # 3055* ; pp 243-255.

(*Soualmia & Darmoni, 2004 a*) SOUALMIA LF., DARMONI SJ. (2004 a) Coupling Different Standards and Different Approaches for Health Information Retrieval in a Quality-Controlled Gateway. *International Journal of Medical Informatics* ; à paraître.

(*Soualmia et al., 2002 a*) SOUALMIA LF., BARRY-GRÉBOVAL C., ABDULRAB H., DARMONI SJ. (2002 a) Modélisation et représentation des connaissances dans un catalogue de santé. *Journées Francophones d'Ingénierie des Connaissances,* pp 139-149.

(*Soualmia et al., 2002 b*) SOUALMIA LF., DARMONI SJ., LE DUFF F., DOUYÈRE M., THELWALL M. (2002 b) Web Impact Factor: a bibliometric criterion applied to medical informatics societies Web sites. In: Proceedings of the *International Congress of the European Federation for Medical Informatics*, Stud Health Technol Inform. 90; pp 178-183

(*Soualmia et al, 2002 c*) DOUYERE M., SOUALMIA LF., LE DUFF F., THELWALL M., DARMONI SJ. (2002 c) Web Impact Factor : un outil bibliométrique appliqué aux sites Web des facultés de médecine et des CHU français. In : L'informatique de la santé dans les soins intégrés : connaissances, application, évaluation. Actes des 9emes *Journées Francophones d'Informatique Médicale*, SoQibs, 2003(16); pp 496

(*Soualmia et al., 2002 d*) SOUALMIA LF., DARMONI SJ., THIRION B., DOUYÈRE M. (2002 d) "Modelization of Consumer Health Information in a Quality-Controlled Subject Gateway"; *American Medical Informatics Association Symposium*. pp.1168

(*Soualmia et al., 2004*) SOUALMIA LF., FLOREA FI., NÉVÉOL A., ROGOZAN A., THIRION B., DACHER JN., DARMONI SJ. (2004) Affiliation of a resource type to a MeSH keyword in the CISMeF health Internet gateway. Soumis à *Journal of Medical Libraries Association*.

(*Thirion et al., 2004*) THIRION B., DOUYÈRE M., SOUALMIA LF., DAHAMNA B., LEROY JP., DARMONI SJ. (2004) Metadata element sets in the CISMeF Quality-Controlled Health Gateway. *DC-2004, International Conference on Dublin Core and Metadata Applications*; in press.

(*Voorhees, 1994*) VOORHEES E. (1994) Query expansion using lexical-semantic relations. Proceedings of the *17th Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, pp.61-69.

CHAPITRE 3

TRAITEMENTS LINGUISTIQUES :

DE LA CHAINE DE CARACTERES A LA REQUETE

Sommaire

3.1 INTRODUCTION 109
3.2 PROBLEMES LIES AU TRAITEMENT DE LA LANGUE 111
3.3 LE TRAITEMENT MORPHOLOGIQUE 113
3.3.1 LE MODELE MORPHOLOGIQUE 114
3.3.2 APPLICATION DE LA MORPHOLOGIE A LA RECHERCHE D'INFORMATION 117
3.4 ACQUISITION DE RESSOURCES LINGUISTIQUES 118
3.4.1 LOGICIELS D'ACQUISITION DE TERMINOLOGIE 118
3.4.2 ACQUISITION DE CONNAISSANCES MORPHOLOGIQUES 119
3.4.3 DOMAINE MEDICAL 120
3.4.3.1 METHODE D'ACQUISITION DE RESSOURCES MORPHOLOGIQUES 121
3.4.3.2 LE PROJET UMLF 122
3.5 ACQUISITION DE CONNAISSANCES MORPHOLOGIQUES POUR LE MESH FRANÇAIS
3.5.1 DESCRIPTION DE 'LEXIQUE' 123
3.5.2 CONSTITUTION ET EVALUATION DES FAMILLES EXTRAITES 124
3.6 TRAITEMENTS LINGUISTIQUES POUR LA RECHERCHE D'INFORMATION 127
3.6.1 UTILISATION DE CONNAISSANCES MORPHOLOGIQUES 127
3.6.1.1 TRAVAUX INITIATEURS 128
3.6.1.2 ETUDE DU VOCABULAIRE DES UTILISATEURS 128
3.6.1.3 EXPERIENCES AVEC DES RESSOURCES ADAPTEES AU VOCABULAIRE 130
3.6.2 RESULTATS 132
3.6.2.1 DESCRIPTION DES REQUETES 132
3.6.2.2 RESULTATS AVEC LES UNITERMES 132
3.6.2.3 RESULTATS AVEC LES TERMES COMPOSES 133
3.6.3 EXPERIENCES DE PHONEMISATION 133
3.6.3.1 TRAVAUX ANTERIEURS : SOUNDEX / SOUNDEX 2 / PHONEX 134
3.6.3.2 PHONEMISATION DE TERMES MEDICAUX 135
3.6.3.3 APPLICATION A LA RECONNAISSANCE DE TERMES 138
3.6.4 CORRECTION ORTHOGRAPHIQUE 139

3.6.4.1 ALGORITHME 139
3.6.4.2 EVALUATION DES RESULTATS 140
3.7 TRAITEMENTS EN LIGNE 141
3.7.1 APPARIEMENT A BASE DE CONNAISSANCES MORPHOLOGIQUES 141
3.7.2 AUTRES TRAITEMENTS 142

Dans ce Chapitre 3 nous énumérons les problèmes liés au traitement linguistique pour la recherche d'information. Nous expliquons pourquoi nous avons choisi d'utiliser en premier lieu des connaissances sur les variations terminologiques afin d'apparier et d'étendre des requêtes à un vocabulaire contrôlé. Ce travail fait suite à de nombreuses expérimentations qui ont montré que les connaissances morphologiques étaient prépondérantes en matière de recherche d'information. Nous décrivons quelques travaux récents connus en la matière pour l'acquisition de telles connaissances et également pour la recherche d'information. Nous détaillons enfin les différents traitements que nous proposons d'effectuer afin de traduire la requête initiale d'un utilisateur, qui est assimilable à une chaîne de caractères, en une requête finale exploitable par tout système de recherche d'information.

3.1 Introduction

Le traitement automatique du langage naturel (TALN) est le domaine de l'ingénierie linguistique et a comme objectif la conception de logiciels ou de programmes, capables de traiter de façon automatique des données linguistiques. Ces données peuvent être des textes écrits ou bien des dialogues oraux ou encore des unités linguistiques comme des phrases, des énoncés, des groupes de mots ou simplement des mots. Il permet d'analyser et de représenter des données textuelles à un ou plusieurs niveaux de compréhension (morphologique, syntaxique, etc.). Nous donnons ci-après les définitions des notions que nous manipulons dans ce chapitre.

Définition 3.1 (TERME)

C'est une unité signifiante constituée d'un mot (terme simple ou uni-terme) de plusieurs mots (terme complexe ou terme composé) et qui désigne une notion de façon univoque dans un domaine de connaissance donné.

Définition 3.2 (MORPHEME)

C'est la plus petite unité supportant une signification. C'est un composant indivisible d'un terme qui ne peut plus être réduit sans perdre sa signification. Un morphème peut se trouver libre (racine) ou lié (affixe).

Définition 3.3 (MOT VIDE)

Un mot vide est un mot ayant un faible ou pas de contenu informatif, comme les déterminants.

Il existe six niveaux de compréhension de la langue pour l'analyse linguistique *(Jouis, 1993) (Arampatzis et al., 2000) (Amar, 2000) (De Saussure, 1972)* :

– Le niveau phonologique :

La phonétique est l'étude des sons du langage humain. Ce niveau fait référence à la façon dont les mots sont prononcés. Il est important pour la compréhension du langage oral notamment et dans les systèmes de reconnaissance vocale. A ce niveau, la plus petite unité de traitement est le *phonème*.

– Le niveau morphologique :

L'analyse morphologique permet de traiter les variations des mots du texte en prenant en compte leurs formes fléchies et leurs variations apparentes. La méthode consiste, soit à créer des dictionnaires de mots avec leurs formes fléchies, soit à créer un ensemble de règles morphologiques qui pourront dériver toutes les formes fléchies à partir des formes canoniques des mots contenus dans un dictionnaire. L'unité minimale d'une forme signifiante est le *morphème*.

– Le niveau lexical :

L'analyse lexicale permet de rechercher l'existence des mots et des expressions du texte dans un dictionnaire linguistique. Elle permet également de confirmer ou d'infirmer l'existence des morphèmes identifiés par l'analyse morphologique. Par opposition aux morphèmes, le *lexème* désigne un mot canonisé et signifiant.

– Le niveau syntaxique :

Un analyseur syntaxique analyse dans un premier temps les groupes de mots des phrases qui forment des unités fonctionnelles (principalement les syntagmes) et génère dans un deuxième temps un arbre syntaxique de la phrase. Une des difficultés de l'analyse syntaxique est la détection de syntagmes nominaux ou encore la désambiguïsation syntaxique d'un mot.

– Le niveau sémantique :

L'analyse sémantique est l'étude du sens et son objectif est de déterminer le sens des mots et des phrases.

– Le niveau discursif :

Le niveau discursif exploite la structure documentaire des différents types de documents et des requêtes en vue d'une extraction de thème(s).

– Le niveau pragmatique :

L'analyse pragmatique permet d'utiliser les connaissances pragmatiques (par exemple les connaissances communes d'un domaine) afin d'interpréter les situations du monde réel.

Le TALN peut être intégré à un ou plusieurs modules d'un SRI dans le but d'augmenter les performances de ce dernier. Ces performances peuvent être quantitatives ou qualitatives. Les performances quantitatives sont exprimées généralement par les mesures de rappel et de précision (§ Chapitre 1). Les études qualitatives quant à elles s'expriment souvent par une évaluation subjective qui est effectuée en parallèle de l'étude quantitative. Elle est généralement réalisée manuellement par un linguiste ou par un expert du domaine considéré.

Parmi les traitements linguistiques qui nous intéressent, nous détaillerons les approches que nous avons mises en œuvre et évaluées dans un système de recherche documentaire, notamment l'utilisation de connaissances morphologiques (spécifiquement la dérivation et la flexion), mais également l'utilisation d'une approche fondée sur les phonèmes qui permet d'effectuer une correction orthographique pondérée des requêtes des utilisateurs. Nos travaux font suite à *(Zweigenbaum et al., 2001a) (Zweigenbaum et al., 2001b)* que nous détaillerons en section 2.6.1.1. Nous considérons dans toute cette étude que nous sommes dans un domaine donné, notamment celui de la médecine et en particulier dans le contexte du système CISMeF qui dispose d'un vocabulaire contrôlé fondé sur la version en français du thésaurus MeSH. Les problèmes de polysémie ne seront pas abordés. L'approche que nous proposons ici peut en revanche s'adapter à d'autres domaines.

3.2 Problèmes Liés au Traitement de la Langue

La plupart des SRI actuels se basent toujours sur l'hypothèse initiale qu'un document doit partager les termes d'une requête pour être identifié comme pertinent *(Gaussier et al, 2000)*. De nombreux travaux ont été menés sur la variation terminologique. Elle consiste à identifier des expressions différentes de notions identiques ou proches. Cette identification peut se faire à l'aide de traitements au niveau des lettres *(Lovis & Baud, 2000)*(fautes de frappe, majuscules, accents), au niveau des mots et de leurs variantes morphologiques *(McCray, 1994) (Lovis et al., 1998) (Jacquemin & Tzoukermann, 1999)*, au niveau de la syntaxe des expressions *(Jacquemin & Tzoukermann, 1999)* ou en s'aidant de synonymes généraux *(Hamon et al., 1998)* ou spécifiques à un domaine comme le domaine médical *(Pouliquen et al., 2002)*, mais également utiliser des méthodes de correction phonémiques visant à corriger un mot en conservant sa prononciation.

Par ailleurs, dès l'origine, il a été proposé un minimum de traitements linguistiques simples dans un SRI qui se limitent à la troncature des mots et à l'élimination des mots vides. Ces traitements sont toujours appliqués car ils sont faciles à mettre en œuvre. Afin d'associer les mots de la requête de l'utilisateur avec des mots différents mais de sens voisin présents dans les textes, une méthode consiste à doter le système de recherche d'un réseau lexical de mots ou de termes décrivant le domaine visé ou encore d'une base de connaissances. Le système peut exploiter ce réseau pour préciser la requête de l'utilisateur quand celle-ci contient des mots polysémiques ou pour l'étendre en cas de synonymie par l'ajout de termes jugés voisins vis-à-vis du domaine de recherche. Les techniques employées pour la construction de ce type de réseau reposent sur des mesures de cooccurrences statistiques dans les documents et n'exploitent aucune information linguistique tels que la variation morphologique, lexicale, syntaxique ou sémantique *(Jacquemin, 1997a) (Chevallet & Haddad, 2001)*.

Les SRI doivent faire face à de nombreuses difficultés qui concernent le traitement de la langue *(De Loupy, 2000)* :

– La graphie :

Un mot peut s'écrire de plusieurs façons ou comporter des fautes de frappe ou d'orthographe ou s'écrire en majuscules. Cela diminue le rappel car si un mot est orthographié d'une certaine façon dans la requête, la simple recherche de ce mot ne permet pas de retrouver les documents qui le contiennent, ou bien qui sont indexés avec, mais sous une autre forme. La recherche systématique des variantes d'un mot (en utilisant les techniques propres aux correcteurs orthographiques, basées sur les phénomènes d'inversion, de répétition, de dédoublement, de phonétique, etc.) peut dans certains cas diminuer la précision. En effet, ce n'est pas parce qu'un mot inconnu est proche d'un autre mot, qu'il a forcément été mal orthographié.

– Les variantes grammaticales :

Une même forme peut être un verbe ou un adjectif. Identifier le terme en tant qu'adjectif dans une requête permet d'écarter les textes dans lesquels il apparaît en tant que verbe. Il y a donc diminution du bruit et augmentation de la précision *(Losee, 1996)*.

– Les variations morphologiques :

Les pluriels, les conjugaisons diminuent le rappel. Par exemple, il faut pouvoir retrouver la forme *chevaux* si la requête comporte le terme *cheval*. En revanche, la prise en compte des variations morphologiques peut impliquer une légère baisse de la précision *(Krovetz, 1993) (Riloff, 1995)*.

– Les expressions composées :

Dans les domaines spécialisés comme celui de la médecine, l'information pertinente est souvent contenue dans les groupes nominaux (par exemple *accident cérébral*, *rupture d'anévrysme*). De plus, le nombre de mots composés, pour le français, est largement supérieur à celui des mots simples *(Royauté et al., 1992)*. Les termes composés permettent généralement de limiter l'ambiguïté et d'augmenter la précision *(Faraj et al, 1996)*.

– La synonymie :

Par exemple *globule rouge* et *hématie* sont des termes synonymes. Lorsqu'une recherche est faite sur un terme comme *globule rouge*, il faut pouvoir retrouver des textes utilisant ses synonymes. Ce phénomène est également lié à la polysémie. En effet, il faut déterminer le sens dans lequel le mot est employé avant de rechercher ses synonymes (à l'aide d'un thesaurus). Par exemple, si l'on ne détermine pas le sens du terme circulation dans l'expression « circulation sanguine », le risque est d'enrichir la requête avec des termes comme « trafic », synonyme de « circulation », et donc d'augmenter considérablement le bruit.

– L'hyponymie (relation père/fils) :

Cette relation d'hyponymie (par exemple entre *cellule* et *cellule sanguine*) peut s'avérer très intéressante. En effet, si l'on recherche les textes concernant les *cellules*, il convient de savoir qu'une *cellule sanguine* est une *cellule*.

– La polysémie :

Un terme ou une expression peut avoir plusieurs sens en fonction des contextes dans lesquels on le retrouve.

– La terminologie :

Il est plus pertinent d'utiliser des connaissances spécifiques au domaine ciblé, plutôt que des données généralistes. Cependant, cela suppose de disposer d'un lexique spécialisé pour le domaine traité ou encore d'un thésaurus.

En résumé, les éléments présentés ici montrent qu'il est difficile de gérer la langue naturelle dans un système automatique de recherche d'information. En effet, les utilisateurs peuvent exprimer de multiples manières une même idée. Cependant, en l'absence d'une connaissance approfondie de la collection de documents (situation relativement fréquente) l'utilisateur risque de formuler sa requête en des termes proches mais non identiques à ceux employés dans un document sur le même sujet.

3.3 Le Traitement Morphologique

Les connaissances morphologiques sont intrinsèquement liées au traitement du langage et à la recherche d'information *(Grabar, 2004)*. Un programme de morphologie permet en général le découpage du mot en morphèmes et la génération de ses flexions à partir d'un mot lemmatisé.

On distingue traditionnellement trois types de variations morphologiques : la flexion, la dérivation et la composition.

Définition 3.4 (LA FLEXION)

> *La flexion permet de créer les différentes formes d'un même mot comme les pluriels, les féminins ou encore les différentes formes d'un verbe en fonction de la personne ou du temps. Par exemple : cœur/cœurs.*

Définition 3.5 (LA DERIVATION)

> *La dérivation ajoute des affixes (préfixes ou suffixes) autour d'une racine pour obtenir par exemple la forme adjectivale d'un nom. Par exemple : cœur/cardiaque.*

Définition 2.6 (COMPOSITION)

La composition permet de combiner plusieurs racines. Par exemple : cardiovasculaire.

Dans le cadre de la recherche documentaire, le courant le plus répandu et le plus simple à mettre en oeuvre consiste à créer un lexique qui contient une grande liste de mots accompagnés de toutes leurs flexions, sans associer les différents types d'informations linguistiques (morphologiques, syntaxiques ou sémantiques). Cette approche nécessite néanmoins une mise jour fastidieuse compte tenu de la masse d'informations concernée.

3.3.1 Le Modèle Morphologique

Un modèle *morphologique* permet d'effectuer les deux tâches d'analyse et de génération. L'analyse consiste à identifier les différents *morphèmes* qui composent les mots. La génération consiste à produire les formes correctes des mots à partir des différents morphèmes.

TAB.3.1.–Liste lexicale associant les formes des mots et leurs bases ainsi que les informations morphosyntaxiques.

Forme fléchie	Base	Catégorie	Morph.
Chiens	Chien	N	Mp
Portes	Porte	N	Fp
Chevaux	Cheval	N	Mp

Le traitement morphologique à l'aide de listes lexicales est très simple à mettre en oeuvre mais il ne permet pas de créer automatiquement les formes fléchies de nouvelles entrées. Cette organisation sous forme de tables est mal adaptée à la dérivation qui, contrairement à la flexion, permet de créer un ensemble de formes quasi infini qu'il est difficile de lister de façon exhaustive.

Un mot pouvant revêtir différentes formes au sein des différentes unités documentaires (adjectif, verbe…), la lemmatisation tend à les regrouper sous un même radical (forme neutre d'un mot). La racinisation vise à regrouper des mots de même famille afin de réduire la variabilité des formes sous lesquelles peut apparaître une notion donnée. Les méthodes de lemmatisation les plus courantes sont la troncature et l'extraction de radicaux par l'utilisation de règles. Le principe de la troncature consiste à ne conserver qu'un nombre fixe de caractères des termes. Une troncature à 7 caractères est communément effectuée pour le français *(Soulé-Dupuy, 1990)*. L'algorithme de Porter *(Porter, 1980)* qui fonctionne par règles est utilisé pour l'anglais.

Un lexique peut être spécifié de manière statique par une liste explicite de formes fléchies et de mots dérivés. Celle-ci peut être validée par des humains et peut permettre un temps de traitement plus rapide (accès direct dans une table). La spécification dynamique d'un lexique se fait par des règles et des outils de décomposition de mots. Les règles appliquées dynamiquement par des outils d'analyse morphologique peuvent traiter des mots inconnus et réduire les besoins en mémoire. Il existe des méthodes utilisant des règles générales complétées par des listes

d'exceptions *(Mc Cray et al., 1994) (Namer, 2000)*. La compilation de listes de mots sous forme d'automates *(Silberztein, 1993) (Lovis et al., 1998)* est une autre méthode générale.

/aux(nom) → /al(nom)

/e(verbe_1) → /er(verbe)

/eait(verbe_1) → /er(verbe)

FIG.3.1–Exemple de règles.

(Gaussier, 1999) construit une procédure de racinisation qui permet de regrouper des variantes dérivationnelles (de plusieurs langues). Cette procédure ne fournit pas une racine pour chaque mot mais un représentant choisi de façon arbitraire de la famille à laquelle appartient un mot. Ainsi l'ensemble des mots *produit, produire, production, producteur, productif, productivité, productivisme, productible* se trouve dans la même famille avec pour représentant *produit*.

La méthode à base de règles consiste à définir un ensemble de règles qui décrivent la morphologie flexionnelle du langage. La régularité de la structure interne des mots y est décrite à l'aide d'automates à états finis, qui définissent les chaînes acceptables de morphèmes ou de segments du langage étudié. Un automate à états finis est un système qui possède un nombre fini d'états avec des règles de transition pour passer d'un état à un autre. Pour le décrire, il peut être représenté à l'aide d'un graphe étiqueté où les états correspondent aux nœuds du graphe, les transitions aux arcs qui relient ces nœuds et les évènements qui déclenchent ces transitions à des étiquettes placées sur les arcs.

Les automates à états finis ont prouvé leur utilité dans une large variété d'applications en informatique linguistique entre autres pour la représentation compacte des dictionnaires électroniques *(Revuz, 1991)*. Ils sont à la base d'algorithmes efficaces à toutes les étapes du traitement des langues naturelles, de l'analyse phonologique et la reconnaissance de la parole *(Mohri, 1997)* jusqu'à l'analyse syntaxique de texte *(Roche & Schabes, 1997)*. Les logiciels INTEX *(Silberztein, 1993)* et Unitex *(Paumier, 2003)*, et les bibliothèques de manipulation d'expressions régulières et d'automates finis de Xerox *(Karttunen et al., 1997)* et AT&T *(Mohri et al., 2000)* sont fondés sur des automates à états finis. Les corpus de texte sont représentés par des automates, ou treillis de mots, dans lesquels chaque chemin correspond à une analyse lexicale. Les grammaires locales *(Nakamura, 2003)*, qui sont un moyen naturel de représenter des phénomènes linguistiques complexes, sont traduites en automates finis afin d'être aisément confrontées avec les corpus de texte.

Dans le cas de la morphologie, à chaque transition correspond un segment de mots et des algorithmes permettent de générer les chaînes du langage, en commençant par l'état initial et en concaténant au fur et à mesure les segments qui figurent sur les transitions.

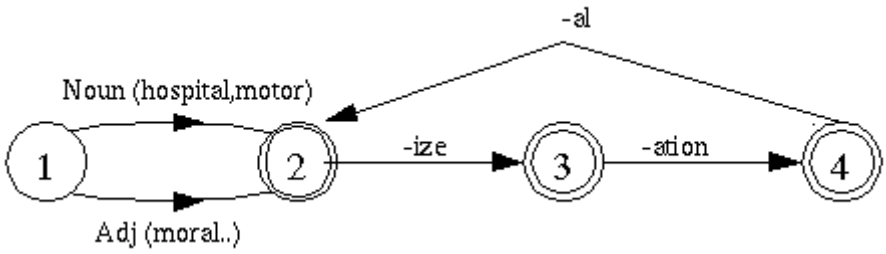


FIG.3.2.–Machine à états finis pour un exemple de la morphologie anglaise : hospital+ize+ation.

L'automate peut être utilisé en génération pour énumérer toutes les formes possibles ou bien en reconnaissance pour accepter les formes qui font partie du lexique et rejeter les autres formes. Certains états sont optionnellement finaux. L'automate peut être augmenté d'informations morphologiques de façon à servir pour les traitements nécessaires en analyse et en génération. L'automate peut également être généré à partir d'une la table lexicale. Cette équivalence est vraie que dans la mesure où l'automate ne comporte pas de boucles, sinon le nombre de chemins devient infini et cette même information ne peut plus être représentée dans une table. Ce gain d'expressivité peut être exploité pour traiter la dérivation qui, contrairement à la flexion, qui ne permet que des combinaisons limitées, elle se caractérise par la possibilité de combiner préfixes et suffixes de façon très libre et non limitée.

La morphologie à états finis décrite ne permet pas de représenter les phénomènes morpho-phonologiques qui interviennent lors de la concaténation de morphèmes. Par exemple le pluriel de *cheval* est *chevaux* et nécessite la transformation du *l* en *u*. Le formalisme morphologique *à deux niveaux (Koskenniemi, 1983)* a été développé pour représenter ces modifications systématiques.

Les modèles de linguistique computationnelle comme la morphologie à deux niveaux a formé la base pour des descriptions du langage général majoritairement acquis manuellement.

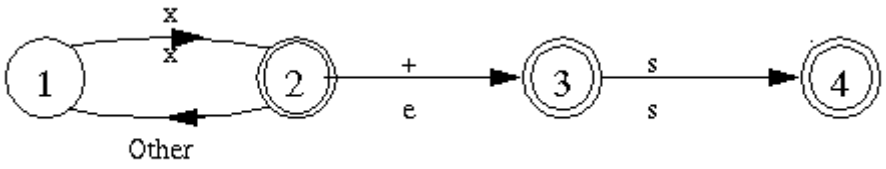


FIG.3.3.– Un transducteur à états finis qui accepte: *x:x +:e s:s*

Le premier symbole correspond au niveau lexical composé de morphèmes mis bout à bout et le second correspond au niveau de surface composé de la forme orthographique finale du mot. Il y a une distinction entre les mots de surface, tels qu'ils apparaissent dans les textes et les mots lexicaux, qui proviennent de la concaténation des segments contenus dans le dictionnaire. Si le dictionnaire contient par exemple les deux morphèmes *fox* et *+s*, *foxes* est une forme de surface, tandis que *fox+x* est une forme lexicale qui provient de la concaténation des deux morphèmes *fox* et *+s* (FIG.3.3). L'automate permet de faire le lien entre le niveau lexical et celui de surface. Lors de son parcours, deux symboles sont vérifiés en même temps. Avec ce type d'automate, le traitement des modifications phonétiques est moins contraignant et il est possible de rendre compte des différentes modifications comme les substitutions et les suppressions.

3.3.2 Application de la Morphologie à la Recherche d'Information

Le regroupement par lemmatisation ou racinisation, permet de retrouver des unités documentaires même si les termes de la requête n'ont pas la même forme que dans les unités documentaires. En effet, comme une requête peut employer des mots proches mais non nécessairement identiques à ceux des documents, la prise en compte de proximités morphologiques entre mots peut permettre d'obtenir de meilleurs résultats. En recherche d'information *(Tzoukermann et al., 1997*) *(Gaussier et al., 2000) (Zweigenbaum et al., 2001a) (Zweigenbaum et al., 2001b),* le but est de pouvoir étendre une requête à d'autres termes que celui fourni par l'utilisateur : les membres de sa famille dérivationnelle sont une piste possible, à condition qu'ils en soient suffisamment « proches » sémantiquement. Cela peut également se faire par extension de requête en ajoutant des formes fléchies ou des formes dérivées aux mots de la requête. Par exemple, la notion de *produire* peut se retrouver sous les formes *produisons, produira, produisent…etc.* Cette notion peut également apparaître sous la forme de *production*. Le premier ensemble de formes relève de la morphologie flexionnelle et concerne directement la lemmatisation. Le deuxième ensemble relève de la morphologie dérivationnelle et requiert des traitements spécifiques. *(Hull, 1996)(Gaussier et al., 2000)* ont évalué l'influence de la racinisation sur les performances d'un système de recherche d'information et ont abouti à la conclusion que la racinisation ne fournissait pas d'amélioration substantielle des résultats. Par ailleurs, l'apport de traitements morphologiques en recherche d'information en langue française a été mis en évidence dans plusieurs travaux récents *(Grabar et al., 2003) (Zweigenbaum et al., 2002)(Gaussier et al., 2000) (Savoy, 2002)*. L'observation générale est que la lemmatisation apporte une amélioration statistiquement significative et que la racinisation apporte une contribution supplémentaire, mais non significative.

L'étude de la similarité des distributions statistiques de mots en corpus peut aider à identifier des mots de sens proche. La base de données morphologiques CELEX *(Burnage, 1990)* a été utilisée avec succès en recherche d'information *(Jing & Tzoukerman, 1999)*. La méthode détermine une équivalence sémantique entre termes en recherche d'information. Elle intègre la racinisation et la recherche fondée sur la sémantique. L'algorithme proposé est constitué des étapes suivantes :

1. Construction de bases de données morphologiques en utilisant CELEX,
2. Pour chacun des mots du document calculer un vecteur de contexte,
3. Pour chaque couple de mots du corpus calculer leur cooccurrence lexicale dans le corpus et leur pertinence dans le corpus,
4. Indexation du corpus,
5. Pour chaque requête et chaque document :
    a. Calculer la distance moyenne en contexte entre le vecteur contexte d'un mot de la requête et les vecteurs de ses variantes morphologiques dans le document. Si cette distance moyenne est supérieure à un seuil donné : considérer les deux mots comme le même terme. Sinon, les deux mots sont des termes différents.
    b. Calculer la similarité entre la requête et le document.

L'utilisation de cette technique a montré des améliorations en recherche d'information sur le corpus TREC-4.

Les ressources de langue générale demeurent peu nombreuses pour le français et difficiles à acquérir.

3.4 Acquisition de Ressources Linguistiques

Les outils de gestion de l'information ont besoin de ressources terminologiques. Les termes représentant les concepts principaux d'un domaine technique ou scientifique, ils facilitent la gestion et la représentation de la connaissance en fournissant une description abrégée des documents *(Daille & Jacquemin, 1998)*.

Les ressources terminologiques sont construites pour un domaine donné et pour une application identifiée. Les ressources terminologiques sont nécessaires pour garantir l'efficacité du système de recherche d'information l'acquisition de ressources terminologiqiues exploite le corpus de documents disponibles pour l'application. Alors que les recherches d'associations entre mots avaient surtout été appliquées à la construction automatique de thésaurus pour la recherche d'information, elles ont été étendues à l'acquisition de données lexicales ou de familles morphologiques *(Xu & Croft, 1998)*.

Des outils d'analyse existent avec la connaissance flexionnelle correspondante (INTEX *(Silberztein, 1993)*, FLEMM *(Namer, 2000)*). En revanche, aucune description de la morphologie dérivationnelle et compositionnelle du français n'est disponible. Des méthodes automatisées pour l'apprentissage des variantes morphologiques ont été développées afin conçues pour rassembler des ressources pour la recherche d'information ou pour les systèmes de traitement du langage naturel. Des ressources dérivationnelles commencent à être constituées *(Jacquemin, 1997b) (Xu & Croft, 1998) (Dal et al., 1999) (Gaussier, 1999) (Daille, 1999) (Grabar & Zweigenbaum, 2000) (Hathout, 2001) (Hathout et al., 2001) (Hathout et al., 2002) (Tanguy & Hathout, 2002) (Namer, 2002)*.

L'acquisition de ressources linguistiques peut se faire à partir de corpus ou à base de connaissances.

3.4.1 Logiciels d'Acquisition de Terminologie

L'acquisition et la construction de ressources terminologiques ont certaines caractéristiques. La tâche d'analyse terminologique vise à construire une description des structures lexicales à l'œuvre dans une corpus textuel de référence. Cette tâche est réalisée par un expert du domaine et un analyste. La constitution du corpus de référence se fait par la collection d'un ensemble de textes jugés pertinents vis-à-vis du domaine de l'application. Grâce au développement du Web, l'accessibilité à des corpus spécialisés est plus grande et facilite ainsi l'acquisition de terminologie.

TERMINO *(David & Plante, 1990a) (David & Plante, 1990b)* est un des premiers progiciels d'acquisition de terminologie. Il effectue une analyse morpho-syntaxique du corpus fourni en entrée et repère des candidats termes (des mots ou des des séquences de mots) qui sont susceptibles d'être retenus comme termes par l'expert.

Les logiciels d'acquisition ANA, ACABIT et LEXTER sont relatifs à la langue française. ANA *(Enguehard, 1992) (Enguehard & Pantera, 1995)* a été développé pour l'enrichissement de réseaux lexicaux exploités par un système de gestion des connaissances. ANA extrait les candidats termes sans effectuer d'analyse linguistique. Les termes sont reconnus au moyen d'égalités approximatives entre mots et d'une observation de répétitions de patrons syntaxiques. ACABIT *(Daille, 1999)* a été développé pour la construction de lexiques terminologiques multilingues. ACABIT extrait des candidats termes à partir d'un corpus préalablement étiqueté et désambiguïsé. Il combine des traitements linguistiques et statistiques. L'exploitation des liens formes adjectivales et formes nominales permet d'augmenter les variantes prises en compte par ACABIT et d'améliorer le regroupement des termes en utilisant des liens morphologiques. LEXTER extrait également des candidats termes à partir d'un corpus préalablement étiqueté et désambiguïsé *(Bourigault, 1994) (Bourigault & Jacquemin, 1996)*. Il effectue une analyse syntaxique de surface dédiée au repérage et à l'analyse des syntagmes nominaux et organise l'ensemble des candidats termes extraits sous la forme d'un réseau.

On peut également citer l'outil FASTR qui est un analyseur syntaxique dédié à la reconnaissance en corpus de termes appartenant à une liste contrôlée fournie au système *(Jacquemin, 1997b) (Jacquemin, 1999)*. Les termes n'ayant pas toujours en corpus la même forme linguistique, l'idée est de pouvoir identifier leurs variantes. FASTR est doté d'un ensemble élaboré de méta-règles qui lui permettent de répérer différents types de variation : les variantes syntaxiques, les variantes morpho-syntaxiques et sémantico-syntaxiques.

3.4.2 Acquisition de Connaissances Morphologiques

Il existe deux approches pour le traitement automatique de la dérivation : les traitements basés sur dictionnaires et les traitements basés sur règles. Les traitements à base de connaissances ont pour objectif principal la recherche d'information. *(Savoy, 1993)* décrit un système complet qui propose une racine et effectue l'analyse flexionnelle et dérivationnelle de mots non étiquetés reçus en entrée. L'approche est basée sur la consultation d'un dictionnaire et les opérations de lemmatisation et de racinisation sont réalisées en une seule étape.

Les modèles à base de règles se fondent sur le modèle de la morphologie à deux niveaux *(Sproat, 1992)*, *(Fradin, 1994)*. *(Clémenceau, 1992)* implémente des automates à états finis *(Roche & Schabes, 1997)*. *(Clavier, 1996)* utilise des grammaires régulières pour analyser récursivement des mots suffixés à l'aide d'un lexique fortement structuré. Les méthodes à base de connaissances *(Lovis et al., 1998) (Namer, 2000)* supposent disponibles des connaissances à priori et les appliquent à une source donnée. Par exemple, le lemmatiseur FLEMM représente des connaissances linguistiques sur le calcul du lemme (forme non fléchie, *e.g.*, *abdominal*) d'une forme fléchie d'un mot (*e.g.*, féminin pluriel *abdominales*).

Par ailleurs, les méthodes de découverte comme celles de *(Xu & Croft, 1998) (Jacquemin, 1997a)* supposent que peu de connaissances sont disponibles, et mettent en jeu des processus d'apprentissage. Par exemple, *(Zweigenbaum & Grabar, 1999)* détectent des mots dérivés en relation avec des mots de base (par exemple l'adjectif *abdominal* avec le nom *abdomen*).

L'objectif du projet FRANLEX *(Dal et al, 1999)* est de construire semi-automatiquement une base de données morphologiques associant à chaque unité lexicale construite du français une description structurelle. Dans ce travail, sont examinés les mots construits par les suffixations par *-able* et *-ité* du français. Ce travail met en œuvre deux approches pour le traitement

automatique de la morphologie : des traitements à base de corpus avec l'analyseur DéCor et des traitements à base de connaissances avec l'analyseur DériF.

Le programme DériF effectue l'analyse dérivationnelle d'un mot étiqueté, au moyen d'un ensemble de règles, d'une liste d'exceptions non productives et du lexique TLFnome, qui est également utilisé comme corpus d'entrée. Les règles (i) découpent le mot selon sa suffixation (préfixation), (ii) représentent, dans un format crocheté la portée de l'affixe analysé sur le reste du mot, et (iii) rassemblent, au fur et à mesure de l'analyse du mot, les unités calculées en une famille. Dans sa version actuelle, DériF analyse les suffixes *-able, -ité, -et(te),* ainsi que quelques allomorphes du préfixe *in-*. Le programme analyse récursivement un mot jusqu'à l'obtention d'une unité lexicale indivisible.

DéCor permet de construire à partir du lexique TLFnome un graphe orienté et valué dont les noeuds sont des lemmes, et dont les arcs relient les dérivés à leur base. Chaque arc porte deux valeurs : les règles de préfixation et de suffixation qui ont permis de produire le dérivé à partir de sa base. Les règles ont été apprises à partir des corpus de mots et le lexique est construit de façon semi-automatique. L'option retenue consiste à dissocier totalement l'analyse des dérivés et la validation manuelle des résultats qui peut ainsi être réalisée de façon autonome. De ce fait, seuls les mots attestés dans le lexique de référence sont pris en compte.

L'objectif du projet MorTAL *(Hathout et al., 2002)* est de construire de manière automatique une base morphologique pour le français général. La méthode permet d'extraire automatiquement un lexique à partir d'un dictionnaire de synonymes et de construire automatiquement des liens morphologiques. Aucune connaissance linguistique n'est mise en œuvre, et elle est indépendante des langues. Elle est également indépendante des dictionnaires particuliers. *(Hamon et al. 1999)* se servent des renvois synonymiques du Robert pour enrichir la structure d'un thésaurus. Le système développé par *(Zweigenbaum & Grabar, 1999)* fonctionne par suppression/ajout de suffixes puis apprentissage, en vue de la construction d'une base de données morphologiques à partir de la nomenclature médicale SNOMED.

Les travaux de *(Klavans et al., 1997) (Woods, 2000)* sont destinés à l'extension de requêtes en Recherche d'Information à l'aide de variantes morphologiques de termes complexes. Aucune information linguistique n'est mise en oeuvre. *(Jacquemin, 1997a)* utilise une liste de termes et un corpus pour déterminer des variantes morphologiques pour les termes complexes de la liste. L'hypothèse initiale est que lorsque deux termes co-occurrents d'un corpus sont similaires (en partageant en commun un suffixe) aux mots d'un terme de référence, ils peuvent être morphologiquement et sémantiquement reliés. Cette méthode permet de trouver par exemple la relation entre *'expression de gène'* et *'expression génique'* où *'gène'* et *'génique'* sont reliés.

(Xu & Croft, 1998) recherchent des mots co-occurrents dans un corpus. Les auteurs se basent sur des statistiques d'information mutuelle afin de sélectionner les mots morphologiquement similaires qui pourraient être des variantes morphologiques. La méthode fonctionne sans terminologie ni règles a priori. Elle approxime la notion de continuité thématique à l'aide d'une fenêtre glissante de N mots. Les mots du corpus sont réduits par le raciniseur de *(Porter, 1980)*. Deux mots racinisés à la même forme réduite et qui co-occurrent significativement (variante de l'information mutuelle) sont considérés comme faisant partie de la même classe d'équivalence morphologique.

3.4.3 Domaine Médical

Relier les formes fléchies et les mots dérivés à leurs mots de base (*abdominaux – abdominal*, *diabétique – diabète*), accroît la puissance et la souplesse de l'appariement de termes *(Grabar, 2004)*. Cela peut également améliorer la recherche d'information, en particulier pour les langues morphologiquement riches comme le français.

3.4.3.1 Méthode d'Acquisition de Ressources Morphologiques

Le « Specialist Lexicon » de l'UMLS fournit des données et des outils pour la morphologie flexionnelle et dérivationnelle de l'anglais médical *(McCray et al., 1994)*.

Le but de *(Zweigenbaum & Grabar, 1999)* est de construire une base de connaissances morphologiques à partir du microglossaire médical SNODEM *(Côté, 1996)* et de la Classification Internationale des Maladies CIM-10 *(OMS, 1993)* dans l'objectif d'étendre des requêtes par mots clés. Ces ressources sont structurées et comportent des indications sémantiques comme la relation de synonymie. La flexion, la composition ou la dérivation ne sont pas distinguées. Le principe fondateur de la méthode est de (i) repérer des mots proches par la graphie et qui (ii) possèdent des liens sémantiques. La méthode de *(Xu & Croft, 1998)* procède de même, en raffinant le premier critère grâce au raciniseur de *(Porter, 1980)*. Dans une terminologie structurée, ce principe a été adapté en repérant (i) des mots qui partagent la même chaîne de caractères initiale et qui (ii) figurent dans des termes reliés par des liens sémantiques.

Les familles sont dérivées à partir de couples de synonymes qui ont en commun la même chaîne de caractères initiale de 3 caractères afin d'avoir moins de risques de trouver des couples de termes qui ne soient pas sémantiquement reliés. A partir des familles, des règles de génération sont déduites (566 règles à partir des 755 familles). Ces règles ont été ensuite appliquées à la CIM-10 et 1 304 nouvelles familles ont été obtenues. Ces familles ont été appliquées à 220 requêtes et ils ont observé une augmentation du rappel de 12% et une diminution de la précision de 2.5%. Les principaux problèmes remarqués sont dus aux suffixes présents. Par ailleurs, ce même travail a permis de découvrir 92% des formes fléchies et 79% des formes dérivées existantes dans l'UMLS pour l'anglais.

La taille nécessairement finie des terminologies disponibles limite le vocabulaire et la variation morphologique qui y sont effectivement présents. Le même travail a été réalisé à partir de la CIM-10 et de la SNOMED Internationale dans *(Grabar & Zweigenbaum, 2000)*. Cette extraction automatique repose sur les relations sémantiques entre termes comme la synonymie mais ici la notion de lien hiérarchique est également employée. Par exemple, la nomenclature SNOMED indique que *sinusite* est une sorte de *maladie du sinus paranasal* les mots {*sinus*, *sinusite*} sont alors en relation morphologique et la règle de substitution de suffixes -/*ite* est déduite et peut s'appliquer sur d'autres couples de mots attestés du domaine. Appliquée à ces terminologies, cette méthode génère très peu de bruit (3 à 5 %). Le même type de données a été obtenu sous forme lemmatisée (par FLEMM (Namer, 2000)) et étiquetée : 2 906 couples correspondant à 1 224 lemmes différents et à 4 125 formes en tout. Pour les ressources dérivationnelles, les 1 910 couples lemmatisés et étiquetés minimaux obtenus dans *(Grabar & Zweigenbaum, 2000)* ont été filtrés et lemmatisés (2 988 lemmes différents). Pour séparer dérivation et composition ces couples ont été divisés en deux ensembles selon les deux catégories syntaxiques en présence : catégories différentes (plutôt dérivation) *vs* catégories identiques (plutôt composition savante). Cette division a été ajustée manuellement et 1024

couples dérivationnels (794 familles) relatifs à 1 759 lemmes ont été collectés. L'union des couples flexionnels et dérivationnels constitue 1 600 familles et 5 462 formes.

3.4.3.2 Le Projet UMLF

Pour le français, des ressources lexicales existent, mais sont incomplètes et dispersées dans plusieurs équipes. Par exemple, des lexiques français ont été préparés pour divers projets de traitement automatique de la langue médicale *(Baud et al., 1992) (Menelas, 1994) (Lovis et al., 1998) (Bodenreider & Mc Cray, 1998) (Zweigenbaum, 2001)* et incluent des ressources morphosyntaxiques. Des méthodes ont été conçues pour apprendre des ressources lexicales à partir de terminologies *(Baud et al., 1997) (Grabar & Zweigenbaum, 2000),* et à partir de corpus *(Zweigenbaum et al., 2003) (Zweigenbaum & Grabar, 2003)*. L'objectif du travail *(Zweigenbaum et al., 2004)* est de rassembler et d'unifier ces ressources sous la forme d'une Union de Lexiques Médicaux Francophones (UMLF)[28] et des corpus diversifiés représentant des spécialités médicales, en compilant des vocabulaires médicaux contrôlés tels que les thésaurus et les classifications comme la CIM-10.

Une entrée dans le lexique associe des informations à un mot (*lexème*). Les termes complexes peuvent également être des entrées. Deux types d'entrées supplémentaires sont cconsidérés : des affixes (*-al*, *-ique*, *de-*, *in*, *hyper-*, *trans-*, *brady-*) et des éléments de composition « liés » (*adéno-*, *myo-*, *-carde*), qui ne peuvent apparaître seuls, mais constituent des éléments de base dans la formation de mots. Le lexique UMLF vise à associer à chaque mot des informations catégorielles (nom, adjectif, etc.) et morphosyntaxiques (genre, nombre, etc.). Chaque forme fléchie sera liée à sa ou ses formes canoniques et chaque mot dérivé à son mot de base.

L'étape initiale dans la compilation d'un lexique consiste à collecter des listes de mots à partir d'échantillons représentatifs de langue médicale. A partir des terminologies et des corpus médicaux on obtient des formes (potentiellement fléchies) de mots plutôt que des lemmes non fléchis. La collecte de 2 338 pages web indexées par CISMeF sous le terme MeSH « *Signes et symptômes, états pathologiques* », complétées par leurs voisins immédiats sur le Web (les pages situées un lien en dessous de chaque page), au total 9 787 pages ont été converties en texte brut. Le corpus a été étiqueté avec TreeTagger *(Schmid, 1994)* et le lemmatiseur FLEMM *(Namer, 2000)* donnant 5 204 901 occurrences de mots (180 000 formes différentes et 142 000 lemmes différents). Les mots pleins sont au nombre de 2 055 419. Après filtrage des mots contenant des caractères non-alphanumériques il reste 2 041 627 occurrences (54 324 lemmes différents).

Afin de relier deux termes en corpus, les liens sémantiques vont s'appuyer sur la notion de continuité thématique du texte. Cette continuité se traduit généralement par des liens thématiques lexicaux (redondance lexicale). Ces liens thématiques peuvent être instanciés par des mots d'une même famille morphologique. La proximité morphologique entre deux mots leur est donnée par le raciniseur de *(Porter, 1980)*. La méthode de *(Xu & Croft, 1998)* est également reprise ici avec une réduction aux 4 premiers caractères du mot. Les mots qui partagent la même chaîne de caractères initiale de longueur > 5 et qui se trouvent souvent dans une même fenêtre de M mots sont recensés.

[28] Projet subventionné par le ministère français pour la Recherche et l'Enseignement Supérieur (ACI UMLF, subvention #02C0163, 2002–2004). CISMeF est fournisseur de corpus.

La deuxième étape consiste à repérer les couples (nom, adjectif) dérivés par suffixation. Pour ce faire, des tests sont réalisés sur les couples. La dérivation ajoute un suffixe. Le système élimine un dérivé qui dépasse de plus de 5 lettres le mot de base. Un autre critère est appliqué ensuite : si plusieurs adjectifs sont proposés pour le même nom, celui correspondant à la règle d'application la plus fréquente est conservé. La force d'association est prise en considération pour départager deux couples produits par des règles de fréquence identique, mais également pour conserver les couples produits par une règle de faible fréquence mais ayant une forte association (seuil=50).

La méthode proposée repère un grand nombre de couples en relations morphologique, dont une part importante est correcte. Sur un échantillon de noms d'anatomie, dans une tâche de recherche d'adjectifs dérivés la précision est de 85 à 91 %. La proportion de noms pour lesquels un adjectif dérivé est trouvé est de 32 à 34 %.

3.5 Acquisition de Connaissances Morphologiques pour le MeSH français

Dans le système CISMeF, le vocabulaire utilisé est le MeSH français. Les familles obtenues dans *(Grabar & Zweigenbaum, 2000)* permettent de couvrir 707 unitermes (894 couples) du vocabulaire CISMeF avec une précision de 100%. Afin de construire une base de connaissances morphologiques relatives au CISMeF, nous avons utilisé un lexique du domaine général *Lexique*[29] *(New et al., 2001) (New et al., 2005)*.

3.5.1 Description de 'Lexique'

Le but de Lexique est de fournir une base de données lexicale avec des estimations de fréquences et comprenant des formes fléchies. Sa construction a été initiée suite aux travaux de *(Content et al., 1990)* qui ont développé *Brulex*, une base de données informatisée regroupant les 35 746 entrées lexicales du *Petit Robert* et leurs fréquences selon le *Trésor de la Langue Française (Imbs, 1971)*. Les fréquences de Brulex sont estimées sur un corpus de textes littéraires datant de 1919 à 1964 et comprenant 26 millions de mots mais des formes fléchies telles que les verbes conjugués ou certaines formes écrites plurielles ou féminines y sont absentes. *NOVLEX*, une base de données plus récente *(Lambert & Chesnet, 2001)*, fournit les formes fléchies mais se fonde sur un corpus spécialisé de textes pour enfants de 417 000 mots.

La base initiale des mots de Lexique a été constituée de textes publiés entre 1950 et 2000 sélectionnés dans la base *Frantext*[30], développée par l'ATILF[31]. Une liste de 246 000 items distincts (termes simples) ainsi que leur fréquence ont été extraits. Après filtrage des signes de ponctuation, des abréviations des mots étrangers et des noms propres, une liste de 130 000 items ayant des formes orthographiques distinctes.

Les fréquences sont calculées en fonction du corpus initial de *Frantext*, et en fonction du nombre exact de pages Web françaises contenant un mot donné, calculé à partir du moteur de

[29] http://www.lexique.org
[30] Cette base regroupe 3 200 textes représentatifs du français des XIXe et XXe siècles
[31] http://www.atilf.fr/

recherche FastSearch[32]. Pour obtenir la catégorie grammaticale, le genre, le nombre et le lemme des mots qui représentent toute une famille de formes apparentées les deux lemmatiseurs *Tree Tagger (Schmid, 1994)* et *Flemm 2.0 (Namer, 2000)* ont été utilisés.

Etant donné le grand nombre d'informations disponibles, la base est divisée en trois tables principales. Celle qui nous intéresse est *Lemmes* : une base organisée à partir des lemmes qui comprend environ 54 000 entrées (la forme "infinitif" pour les verbes ; la forme "masculin singulier" pour les participes passés, adjectifs et noms). Celle-ci est triée par ordre alphabétique et les différentes formes fléchies sont séparées par des ' ;'.

3.5.2 Constitution et Evaluation des Familles Extraites

Afin de constituer des familles relatives au vocabulaire que nous avons à notre disposition, la première étape a consisté à rassembler les différents lemmes qui peuvent faire partie de la même famille morphologique d'un radical. Nous considérons que les termes font partie de la même famille si un même terme est une forme fléchie de deux lemmes différents. Les autres conditions sont que ces lemmes partagent la même chaîne de caractères initiale de 5 caractères et qu'ils sont à un niveau +/- 2 avant ou après le terme dans la table Lemme. Toutes les formes fléchies ainsi que leurs lemmes sont récupérés pour former une même famille. Cela nous a également permis par effet de bord de récupérer les formes adjectivales des lemmes. Par exemple, *abdominale* est une forme fléchie d'*abdominal* mais en même temps une dérivation d'*abdomen*.

Afin d'associer les différentes familles aux termes de notre terminologie, nous avons d'abord considéré les unitermes. Nous n'avons traité que les termes qui ont en commun au moins 5 caractères en début de chaîne. De ce fait nous excluons de notre étude tous les termes de moins de 5 caractères du vocabulaire de CISMeF comme par exemple *sang, os, nez, sida*, mais également toutes les abréviations *bcg, hiv, irm…* (96 unitermes soit 0,95% du total).

Pour faciliter l'accès aux données qui seront exploitées pendant le processus de recherche d'information, nous avons créé une table avec un enregistrement pour chaque couple (mot du vocabulaire ; mot dérivé) avec *mot dérivé* désignant un terme de la famille précédemment constituée. Les termes du vocabulaire peuvent se trouver initialement dans une forme fléchie : par exemple *accidents* est un terme du vocabulaire. Le représentant de la famille est le terme du vocabulaire qu'il soit ou non sous sa forme fléchie ou dérivée. De ce fait, parmi les couples répertoriés, nous trouverons de « vrais » dérivés, mais également des mots d'une même famille morphologique (*articulation*/*articulaire*).

Par ailleurs, une liste de mots vides a pu être constituée à partir de *Lexique*. Les mots vides considérés sont tous les adjectifs possessifs (*mon*), les conjonctions (*mais*), les déterminants (*du*), les interjections (*diantre*), les prépositions (*durant*), les pronoms personnels (*il*), les pronoms possessifs (*leur*) et les pronoms relationnels (*auquel*) ainsi que les symboles et les locutions (*ainsi*). Nous avons déterminé ainsi une liste de 1 422 mots vides. Ce nombre est élevé vu que des termes comme *boum, bye, bravo* ou encore *sniff* sont considérés comme vides.

La première étape de génération des familles *(Soualmia & Darmoni, 2003a) (Soualmia & Darmoni 2003b)* d'unitermes nous a permis de constituer 2 883 familles (sur 3 485) soit 82.73% du total.

[32] http://www.alltheweb.com

Afin de compléter nos données, nous avons considéré dans une seconde étape tous les termes complexes. Notre méthode consiste à traiter séparément chaque segment d'un terme composé et de retrouver les familles de chaque segment *(Soualmia & Darmoni, 2004a) (Soualmia & Darmoni, 2004b)*.

TAB.3.2.– Couverture du vocabulaire.

	Nombre	**Total**	**%**
1-mot	2 883	3 485	82,73%
2-mots	4 160	4 242	98,07%
3-mots	1 540	1 573	97,90%
4-mots	477	480	99,38%
5-mots	134	134	100%
6-mots et +(12)	48	48	100%
Total	9 242	9 962	92,77%

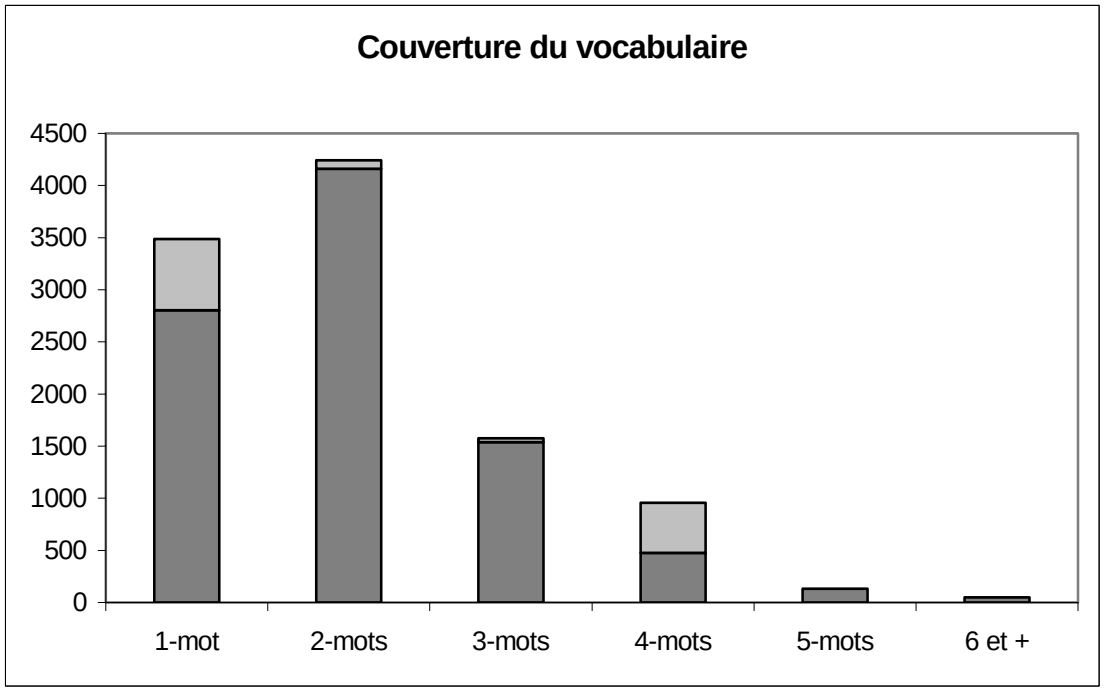


FIG.3.4.– Couverture du vocabulaire CISMeF en fonction de la structure des termes.

La qualité des termes dérivés extraits automatiquement peut être évaluée en utilisant les deux mesures classiques que sont la Précision et le Rappel.

$$Pr\acute{e}cision = \frac{Mots - Correctement - Fléchis - et - Dérivés}{Mots - Fléchis - et - Dérivés}$$

$$Rappel = \frac{Mots - Correctement - Fléchis - et - Dérivés}{Mots - à - Dériver - et - à - Fléchir}$$

Les résultats obtenus avec ces deux mesures peuvent être visualisés à l'aide de deux types de courbes : les courbes d'élévation pour la précision et les courbes ROC (Reciever Operating Characteristic) pour le rappel. Les courbes d'élévation consistent à donner la variation de la précision en fonction du nombre de termes trouvés par notre méthode. Les courbes ROC introduites permettent de visualiser le rappel des termes corrects relativement au rapport des termes incorrects. Le pourcentage de termes négatifs ou incorrects identifiés par le système est indiqué en abscisse et l'axe des ordonnées correspond au pourcentage de termes corrects (rappel).

Pour évaluer la qualité des familles de termes extraites, nous n'avons utilisé que la précision. Ce choix est contraint par le fait que notre approche est non supervisée et il est impossible d'avoir la liste des termes dérivés et fléchis de manière exhaustive. De plus, étant donné le nombre élevé de familles à évaluer, nous avons choisi d'évaluer un échantillon de chaque type de famille (uniterme, 2-mots, …etc.) avec un intervalle de confiance de 95,5%.

Pour un échantillon de taille N, l'intervalle de confiance IC à 95.5% est défini par :

$$IC = \pm 2\sqrt{\frac{p \times q}{N}}$$

Avec : p *pourcentage de la caractéristique au sein de l'échantillon* et $q = 100 - p$.

Dans notre cas p correspond au nombre de familles de l'échantillon qui sont correctes. Nous ne calculons pas la précision moyenne. En effet, dans une même famille, il peut y avoir des mots qui sont de vrais dérivés ou flexions et des faux dérivés ou flexions. On considère qu'une famille est correcte si tous les mots qu'elle contient sont de vraies flexions ou de vraies dérivations. Les résultats obtenus sont dans le tableau suivant :

TAB.3.4.– Evaluation de la précision des familles de termes obtenues.

	Echantillon	**Correct**	**Précision ± IC**
1-mot	282	262	0,93 ± 0,03
2-mots	326	267	0,82 ± 0,04
3-mots	150	112	0,75 ± 0,07
4-mots	50	46	0,92 ± 0,07
5-mots	50	48	0,95 ± 0,05
6-mots et +(12)	48*	47	0,98

* ici la taille de l'échantillon est la même que celle de l'ensemble

Cette méthode nous permet par exemple d'obtenir les familles suivantes :

Famille (Accidents) = {accident, accidenté, accidentées, accidentel, accidentels, accidentelle, accidentelles, accidentellement, accidenter…}

Famille (Abdomen) = {abdominal, abdominale, abdominales, abdominaux}

Famille (Genou) ={genou, genoux, genouillère, genouillères}

Famille (Albinisme Oculaire) = {albinisme, albinismes, albinos, oculaire, oculaires}

Famille (Accident Circulation) = Famille (Accident) + {circulant, circulants, circulantes, circulations, circulatoire, circulatoires, circuler…}

Famille (Infection Chirurgicale) = {infecte, infectant, infectes, infectants, infecter, infectant, infectantes, infections, infectieux, infectieuse, infectieuses} + {chirurgicale, chirurgicales, chirurgical, chirurgicaux, chirurgien, chirurgiens, chirurgie, chirurgies…}

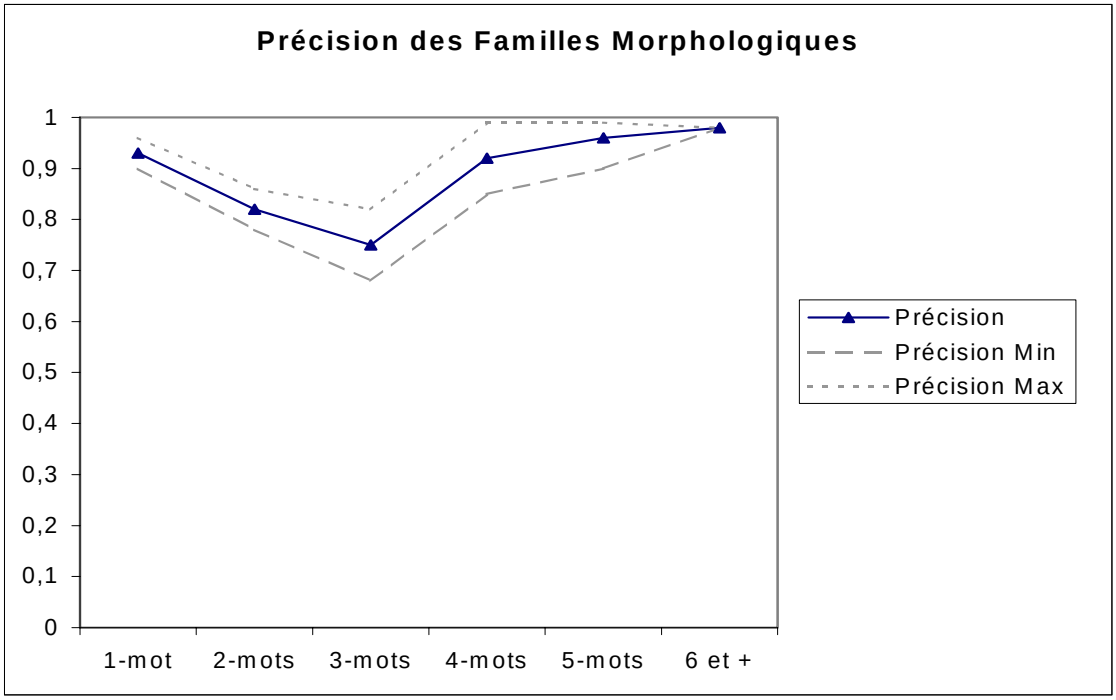


FIG.3.5.–Précision des familles du vocabulaire.

3.6 Traitements Linguistiques pour la Recherche d'Information

3.6.1 Utilisation de Connaissances Morphologiques

L'influence de connaissances morphologiques (flexion, dérivation) est étudiée sur les résultats d'une tâche spécifique de recherche d'information dans le cadre du CISMeF. Cette influence est étudiée à l'aide d'une liste de requêtes réelles des utilisateurs. Les traitements appliqués à l'origine au CISMeF (Novembre 2001) consistaient à comparer la requête de l'utilisateur avec le vocabulaire. Si aucun terme du vocabulaire ne correspond exactement à la requête, une troncature (droite et gauche) est effectuée sur la chaîne en entrée.

Le principe est que les deux formes d'un couple (lemme,variation) sont substituables en recherche d'information. Ainsi, si une forme fléchie est employée pour une requête, elle pourra être remplacée par la forme d'origine utilisée pour l'indexation (et inversement, quand c'est la forme fléchie qui est utilisée pour l'indexation). Comme dans tous les serveurs Web, les requêtes sont courtes. Ici, le fait que les «termes» cible aussi bien que les requêtes ne soient pas

nécessairement syntaxiquement bien formés réduit l'intérêt d'approches syntaxiques *(Grefenstette, 1997)*.

3.6.1.1 Travaux Initiateurs

L'appariement visé dans *(Zweigenbaum et al., 2001a)* repose sur la disponibilité de ressources morphologiques. Les ressources utilisées sont celles obtenues automatiquement dans *(Grabar & Zweigenbaum, 2000)*. L'expérimentation concerne les 6 469 requêtes différentes recueillies pour septembre 2000. Les termes cible sont pré-indexés : chaque terme est mis en minuscules, segmenté en mots et considéré comme un ensemble de mots (l'ordre n'y plus pertinent). La forme désaccentuée de ces termes est segmentée et indexée. Chaque requête est mise en minuscules et segmentée en mots, et les mots vides sont supprimés. Le principe de base de l'appariement est de proposer les termes qui contiennent le maximum de mots de la requête, sans tenir compte de leur position. Plusieurs termes cible peuvent être nécessaires pour « couvrir » les différents mots d'une requête. Tour à tour le terme cible couvrant le maximum des mots restants de la requête (méthode du sac de mots) est sélectionné. L'appariement est testé sur les mots, éventuellement augmentés de : (i) leur forme désaccentuée ; (ii) leurs formes fléchies ; (iii) leurs mots dérivés (et leurs formes fléchies). Les expériences pratiquées, comme dans *(Gaussier et al., 2000)*, consistent à ne réaliser qu'une partie des étapes pour étudier les différences de résultats.

Sur les 6 469 requêtes, 182 sont exactement des termes cible ; 1 315 sont identiques à des termes cible une fois mises en minuscules et segmentées ; 1 364 lorsqu'on les considère comme des sacs de mots ; et 1 679 une fois désaccentuées (26 %). 971 requêtes sur 6469 (15,0 %) ont des résultats différents une fois désaccentuées. La flexion permet de reconnaître 429 requêtes sur 6 469 (6,6 %). voient leur résultat changer. 199 séries différentes de formes fléchies ont été employées ainsi que des règles pour les féminins et les pluriels (403 occurrences ; 123 applications différentes du suffixe *-s*, et 48 du *-e*). Les quelques autres cas de flexion appliqués concernent des féminins : *if/ive*. L'apport supplémentaire de la dérivation concerne un nombre de 128 (2,0 %) requêtes. Les mots dérivés proposés sont uniquement ceux présents dans la base de *(Grabar & Zweigenbaum, 2000)*.

Une évaluation manuelle a été réalisée par les documentalistes de l'équipe CISMeF sur les résultats de 58 requêtes contenant au moins 2 mots, qui n'étaient pas exactement des termes cible et qui ne comportaient pas de faute d'orthographe. Chaque résultat a été noté de 0 (très mauvais) à 3 (très bon). La note moyenne a été de 1,7. Les requêtes dont les résultats ont changé avec l'apport de connaissances morphologiques ont été examinées. Une évaluation a porté sur un petit échantillon de 20 requêtes sur les 429 modifiées par la flexion. Leur note moyenne a augmenté de 0,53 à 1,07. Pour la dérivation, l'évaluation a porté sur 64 des 128 requêtes concernées. La note moyenne obtenue a augmenté de 1,14 à 1,91.

3.6.1.2 Etude du Vocabulaire des Utilisateurs

Dans *(Grabar et al., 2002) (Grabar et al., 2003)* nous avons réalisé des comparaisons mot à mot des requêtes et du vocabulaire. Les vocabulaires des requêtes et d'indexation ont été comparés sous leur forme initiale, puis sous une forme normalisée. Les normalisations

appliquées concernent des opérations au niveau des caractères : casse (majuscules/minuscules) et accentuation. Elles concernent également des opérations fondées sur des connaissances morpholexicales : lemmatisation (réduction des pluriels et des féminins) et racinisation (réduction d'un mot dérivé à un mot initial dont il est dérivé, directement ou indirectement). Les mots vides sont éliminés à l'étape initiale de la comparaison des deux vocabulaires.

L'étude réalisée traite les unitermes présents dans les requêtes et dans les termes du CISMeF pour un appariement mot simple à mot simple. La période de septembre 2000 à janvier 2001 correspond à 108 660 requêtes (29 092 requêtes différentes). Le vocabulaire cible est constitué de 29 035 termes différents.

La méthode de comparaison successive consiste à segmenter en mots les termes des requêtes et du vocabulaire et à les comparer après des normalisations successives. La segmentation en mots se fait en coupant la chaîne initiale aux espaces, ponctuations et autres caractères non alphanumériques. Les normalisations au niveau des caractères ont été conditionnées par la nature du MeSH (mots en majuscules non accentuées, hormis les termes en minuscules ajoutés par l'équipe CISMeF), de même que par la nature non prévisible des mots employés dans les requêtes par les utilisateurs de CISMeF (mots en minuscules ou en majuscules, accentués ou non). Il y a deux types de traitement pour la normalisation : la minusculisation et la désaccentuation.

Trois méthodes de lemmatisation sont utilisées. Une liste de 308 847 couples {lemme, forme} a été obtenue à partir de *(Grabar & Zweigenbaum, 2000)* et ont été compilées à partir de dictionnaires généraux (lexique de l'ABU[33]) et de différents corpus médicaux étiquetés (corpus MENELAS, parties du corpus CLEF). Une heuristique de suppression de la marque du pluriel qui est utilisée en recherche d'information avec de bons résultats *(Savoy, 2002)* consiste à supprimer les finales en -s, à réduire les -aux en -al et à supprimer les –x. La troisième méthode consiste à combiner les deux approches.

Suite à la lemmatisation, les mots qui ne sont pas reconnus subissent une racinisation. La racinisation est appliquée sur les mots du vocabulaire cible et sur les listes de mots qui sont toujours « inconnus » à la sortie des trois méthodes de lemmatisation. 1 041 couples pour la racinisation ont été générés à partir de terminologies *(Grabar & Zweigenbaum, 2000)*. Une correction orthographique sur les mots inconnus restants de longueur supérieure à 5 caractères est en fin réalisée avec l'outil ispell d'Unix, avec le vocabulaire comme dictionnaire de référence.

Les requêtes contiennent 29 092 termes différents. La segmentation permet d'en reconnaître 3 438 (11.82%). La normalisation permet de reconnaître 5 650 mots (2 437 par minusculisation et 3 213 par désaccentuation). Les mots vides sont au nombre de 85. Des traitements morpholexicaux permettent de reconnaître de 11,3 à 12,4 % des occurrences restantes selon la méthode de lemmatisation, plus moins de 1 % pour la racinisation. Il reste 9176 mots non reconnus. Enfin, la correction propose de corriger de l'ordre de 6,5 % des mots restants. Néanmoins, de nombreuses propositions sont erronées. La lemmatisation n'a pas été réalisée en fonction de la terminologie et la correction orthographique n'est qu'une proposition. La contribution totale des connaissances morphologiques est plus faible, comparable à celle de la désaccentuation (gain relatif de l'ordre de 9 %), alors qu'elle demande davantage de ressources initiales. Au final, 65,5 % des mots peuvent être mis en rapport avec des mots du CISMeF.

[33] abu.cnam.fr/DICO

3.6.1.3 Expériences avec des Ressources adaptées au Vocabulaire

Les travaux précédents ont montré que des traitements linguistiques de base étaient nécessaires. Afin d'intégrer les méthodes présentées à Doc'CISMeF, nous avons développé des servlets Java interrogeant la base de données qui contient le vocabulaire du CISMeF. Avec les connaissances de *(Grabar & Zweigenbaum, 2000)* nous avons pu obtenir 894 couples de mots pour 707 unitermes du vocabulaire. Soit en moyenne 1,26 termes par famille morphologique. Cela représente 7,13% du vocabulaire total. Avec un lexique général on arrive à couvrir près de 93% du vocabulaire avec une bonne précision des familles (entre 75 et 100%).

Cette expérience concerne un appariement mot à mot mais avec les connaissances morphologiques adaptées au vocabulaire du CISMeF. La seconde étude quant à elle concerne les requêtes elles-mêmes.

Nous détaillons les étapes que nous avons appliquées aux requêtes.

Segmentation : la requête est segmentée en mots à l'aide d'une liste de caractères de séparation et de *string tokenizers*. Cette liste de séparateurs a été constituée essentiellement avec tous les caractères qui ne sont pas alpha-numériques (par exemple : * *$,!§;*|@).

Fonction SEGMENTATION

Entrée: Chaîne de caractères *C*
Sortie: Liste de mots $C_Liste = \bigcup_{i=1}^{k} mot_i$
Début
StringTokenizer sTokens = new StringTokenizer(*C*," '\"*+()#~{}[]@<>=+°.;,:!§$£µ%?&¨^-_\\", false);
Tant Que (sTokens.hasMoreTokens())
C_Liste ← *C_Liste* ∪ {Tokens.nextToken().trim()};
FinTantQue
Retourner (*C_Liste*) ;
Fin

Normalisation *:* la normalisation se fait au niveau des caractères. Nous appliquons ici deux types de normalisation. Les termes du MeSH sont en majuscules non accentuées. Néanmoins, les termes employés dans CISMeF ont été mis en casse mixte (minuscules avec emploi « normal » des majuscules) et réaccentués. *(Zweigenbaum & Grabar, 2002)* ont travaillé sur l'accentuation du MeSH et ont obtenu de bons résultats.

(1) ***Minusculisation*** : tout caractère en majuscule est converti en son correspondant minuscule (par exemple le caractère *"A"* est remplacé par *"a"*). Cette étape est nécessaire car le vocabulaire contrôlé que nous avons à notre disposition est entièrement en minuscules.

(2) ***Désaccentuation*** *:* tous les caractères accentués sont remplacés par leur version non accentuée (par exemple les caractères *"éèêë"* sont remplacés par *"e"*). Le vocabulaire que nous avons à notre disposition est accentué mais cette étape est néanmoins nécessaire car elle permet de gérer les éventuelles erreurs orthographiques au niveau des accents contenus dans les mots des requêtes (par exemple *"h**è**patite"* au lieu de *"h**é**patite"*).

Fonction NORMALISATION

Entrée: Mot *M*

Sortie: Mot *mot_normalisé*

Début

M← Lower (*M*);

mot_normalisé ← *translate(*M, *'éèêëàâäïîûüùñôöç','eeeeaaaiiuuunooc') ";*

Retourner *(mot_normalisé) ;*

Fin

Mots Vides : tous les mots vides composant la requête initiale sont éliminés. Nous utilisons la liste constituée précédemment (1 422 mots vides).

(Riloff, 1995) a montré que la forme exacte des termes avait une importance pour la recherche d'information et la classification de textes. Nous avons ajouté l'étape intermédiaire suivante *(Soualmia & Darmoni, 2004a)* :

Mot exact : nous utilisons ici des expressions régulières afin d'apparier l'expression exacte de chaque mot de la requête. Cette étape est nécessaire car elle permet de prendre en compte les termes complexes du vocabulaire mais également d'éviter le bruit généré par des troncatures. Nous n'utilisons plus les troncatures des chaînes en entrée afin d'éviter le bruit qu'elles génèrent au détriment de certaines flexions et dérivations. Par exemple le mot "***accident***" sera apparié avec le terme *"circulation* ***accident****"* mais pas avec les termes *"****accident****s"* et *"chute* ***accident****elle"*. En revanche, De même le mot *"****sida****"* sera apparié avec les termes *"lymphome lié* ***sida****"* et *"****sida*** *atteinte neurologique"* mais pas avec les termes *"gluco****sida****ses"*, *"agra****sida****e"* ou encore *"bêta galacto****sida****se"*.

Fonction MOT_EXACT

Entrée: Mot *M* ; Terminologie *TB_vocabulaire* ;

Sortie: Liste de mots $ME_Liste = \bigcup_{i=1}^{l} mot_exact_i$

Début

ME_Liste = SELECT *TB_Vocabulaire.libelle* FROM *TB_vocabulaire*

WHERE *libelle* LIKE ('%*M*%') AND RE.get_match (*libelle*, '\<*M*\>') is not null

Retourner (*ME_Liste*) ;

Fin

Connaissances Morphologiques : nous remplaçons chaque mot de la requête par la *racine* de sa famille morphologique. La racine est généralement le mot de base qui a subi les différentes modifications, c'est à dire la forme simple du mot. Dans notre cas le mot de base est celui du vocabulaire même s'il n'est pas sous sa forme simple. Par exemple, le mot *"abdominal"* sera remplacé par *"abdomen"*, de même que le mot *"accident"* sera remplacé par *"accidents"*. Contrairement à *(Grabar et al., 2003)* nous n'utilisons pas de règles de racinisation. Cette étape concerne donc la lemmatisation.

Fonction MOT_MORPHO

Entrée: Mot *M* ; Base de Connaissances Morphologiques *TB_BCM* ;

Sortie: Liste de mots $MM_Liste = \bigcup_{i=1}^{m} mot_morpho_i$

Début

MM_Liste= SELECT *TB_vocabulaire.libelle* FROM *TB_vocabulaire, TB_BCM*
WHERE *TB_BCM.mot_derive* LIKE ('%*M*%')
AND RE.get_match(*TB_BCM.mot_derive*, '\<*M*\>') is not null
AND *TB_BCM.num_terme=TB_vocabulaire.num_terme*

Retourner (*MM_Liste*) ;

Fin

3.6.2 Résultats

3.6.2.1 Description des Requêtes

Nous avons testé les différentes étapes sur un ensemble de 77 382 requêtes n'ayant aucune réponse. Elles correspondent à 48 255 requêtes distinctes. 12 974 requêtes (26,89%) sont composées d'un seul mot ; 16 347 (33,88%) de 2 mots ; 10 972 (22,74%) de 3 mots ; 4 360 (9,03%) de 4 mots et 3 602 sont composées de plus de 4 mots (7,46%).

Le vocabulaire cible contient les différents termes du vocabulaire auxquels on a ajouté les différents synonymes et les termes anglais.

3.6.2.2 Résultats avec les Unitermes

TAB.3.4.–Reconnaissance des mots des requêtes.

	Termes Reconnus	**%**
Segmentation	12 579	10,31%
Normalisation	20 447	16,77%
Mots Vides	21 022	17,24%
Mot Exact	28 837	23,64%
Connaissances Morphologiques (1)	10 002	8,20%
TOTAL	**92 887**	**76,16%**

Les 48 255 requêtes n'ayant pas de réponse ont été segmentées en 121 958 mots. En appliquant les différentes étapes de traitement, un total de 92 887 (76,16%) mots ont été mappés avec la terminologie et la base morphologique. Cependant, 29 071 (23,84%) mots restent inconnus des vocabulaires. La plupart des mots inconnus sont des erreurs d'orthographe

ou encore des termes n'existant pas dans la base. Cependant, comme souligné dans le Chapitre 2, les connaissances sémantiques doivent être complètes.

3.6.2.3 Résultats avec les Termes Composés

Comme souligné dans *(Zweigenbaum et al., 2001a) (Zweigenbaum et al., 2001b)* l'appariement des termes complexes est plus exigeant.

TAB.3.5.–Reconnaissance des mots des requêtes.

	Termes Reconnus	**%**
Connaissances Morphologiques (1)	10 002	8,20%
Connaissances Morphologiques (2)	8 014	6,57%
TOTAL	**100 901**	**82,73%**

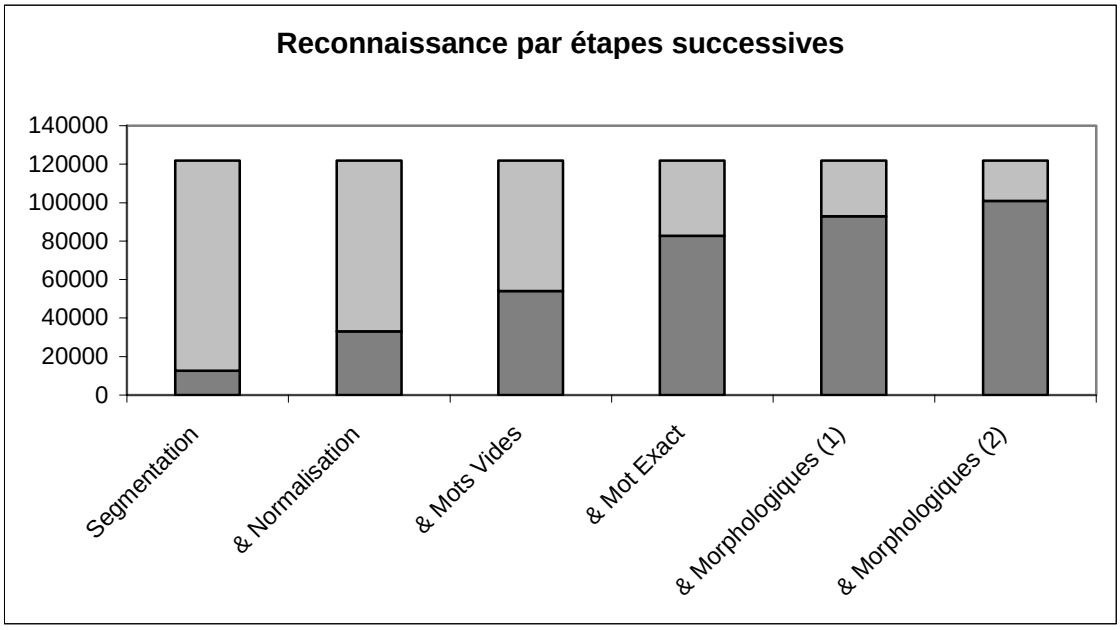


FIG.3.6.–Reconnaissance des mots des requêtes.

Nous avons utilisé ici la base de connaissances morphologique complétée avec les termes composés.

3.6.3 Expériences de Phonémisation

Une correction orthographique est en revanche un facteur d'amélioration important *(Grabar et al., 2003)*. Nous souhaitons ici réaliser des expériences de phonémisation afin de pouvoir corriger les requêtes des utilisateurs lorsqu'elles ont une mauvaise orthographe mais la bonne

sonorité. La fonction que nous proposons s'inspire de fonctions déjà existantes pour le français. Elle nous permet par exemple de retrouver « alzheimer » pour la requête « alzaymer ».

3.6.3.1 Travaux Antérieurs : Soundex / Soundex 2 / Phonex

Le terme SOUNDEX remonte à 1918. Le premier algorithme de ce type a été inventé par Margaret O'Dell et Robert C. Russell, lors des problèmes liés au recensement américain. En effet, de par leur constitution, les États Unis d'Amérique sont tenus à recenser leur population tous les 10 ans. A la fin du siècle dernier, le problème du recensement était devenu un casse tête majeur. Le fait de traiter des informations concernant une population de plusieurs dizaines de millions d'américains à la main demande un travail phénoménal. Le but du Soundex est de codifier les noms propres en éliminant les lettres non prononcées, par exemple le 'h'.

Le principe est le suivant :

Algorithme SOUNDEX

1. transformer en majuscule
2. conserver la première lettre du mot
3. supprimer les lettres non prononcées ('h' et 'w') et les voyelles
4. transcrire les lettres d'un mot en un code selon le tableau 2.11
5. suppression de lettres identiques limitrophes

Par exemple : SOUNDEX (MARTIN)= M635

TAB.3.6.–Correspondance des codes.

Lettre	Code
B F P V	1
C G J K Q S X Z	2
D T	3
L	4
M N	5
R	6

Cet algorithme a été conçu à l'origine pour les mots à consonance anglo-saxonne et s'adapte difficilement au français. Le principe de Soundex2 développé par Brouard en 1995) est quasiment le même avec des codes supplémentaires pour certaines combinaisons de lettres. Par exemple : Soundex2 (MARTIN) = MRTN

Le PHONEX développé en 1999 *(Brouard, 2004)* est un algorithme plus perfectionné que les Soundex car il permet de traiter plusieurs sons formés par des combinaisons de lettres et il est conçu pour la langue française. Le principe du Phonex est le suivant :

Algorithme PHONEX

1. 'y' = 'i'
2. Supprimer les 'h' qui ne sont pas précédés de 'c', de 's' ou de 'p'
3. 'ph'= f'
4. gan= kan; gam = kam; gain = kain; gaim= kaim
5. ain = yn; ein= yn; aim= yn; eim= yn ; si elles sont suivies par une lettre 'a', 'e', 'i', 'o', 'u'
6. eau=o ; oua=2 ; ein=4 ; ain=4 ; eim=4 ; aim=4
7. é/è/ ê/ai/ei =y ; er=yr ; ess=yss ; et=yt
8. an/am/en/em=1 ; in=4 ; sauf s'ils sont suivis par une lettre 'a', 'e', 'i', 'o', 'u' ou un son '1' à '4'
9. 's' = 'z' s'ils sont suivis et précédés des lettres 'a', 'e', 'i', 'o', 'u' ou d'un son '1' à '4'
10. oe/eu=e ; au=o ;oi/oy=2 ; ou=3 ;
11. ch/sch/sh=5 ; ss/sc=s
12. 'c'='s' s'il est suivi d'un 'e' ou d'un 'i'
13. c/q/qu/gu=k ; ga=ka ;go=ko ;gy=ky
14. a=o ;d/p=t ;j=g ;b/v=f ;m=n
15. Suppression des lettres dupliquées
16. Suppression des terminaisons suivantes : 't', 'x'
17. Affectation à chaque lettre du code numérique correspondant en partant de la dernière lettre : o=1 ; 1=2 ; 2= 3 ; 3=4 ; 4=5 ; 5=e ; 6=f ; 7=g ; 8=h ; 9=i ; 10=k ;11=l ; 12=n ; 13=o ; 14=r ; 15=s ; 16= t; 17= u; 18=w; 19=x ; 20= y; 21=z.
18. Conversion des codes numériques ainsi obtenus en un nombre de base 22 exprimé en virgule flottante.

3.6.3.2 Phonémisation de Termes Médicaux

L'avantage de PHONEX est qu'il est très performant sur les noms propres français et qu'il code selon le type de consonance. Les termes médicaux ont des prononciations très différentes des mots « classiques » et le fait de regrouper des lettres selon leur type de prononciation risque de provoquer des confusions entre deux mots ayant presque la même prononciation (mais ayant deux sens bien différents). Par exemple les mots « *androstènes* » et « *androstenols* » ont tous les deux le code 0,082050249 or ils ont deux sons et deux sens bien distincts.

De ce fait nous avons développé une fonction de phonémisation dans le but de retrouver un mot même s'il est écrit avec la mauvaise orthographe mais avec la bonne sonorité. Par exemple pour l'orthographe erronée « kollestérraulle » (à la place de « cholestérol ») la fonction renvoie la phonémisation « kolesterol » pour les deux orthographes. Cette fonction PHONEMISATION[34] ne gère qu'un seul mot en entrée et code les sons selon la table suivante.

[34] Algorithme implémenté en PL/SQL par N.Chmiel au cours de son stage de BTS.

TAB.3.7.–Correspondance des sons.

Code	Son [phonétique]	Exemple
1	"un" [œ]	Comm**un**
2	"oi" [wa]	F**oi**e
3	"ou" [u]	Gen**ou**
4	"en" [ã]	Sci**en**ce
5	"ch" [ʃ]	Bron**ch**e
6	"ill" [j]	Ore**ill**e
7	"gn" [ɲ]	Soi**gn**er
8	"é" [e] "è" [ɛ] "e" [ø]	Pr**é**l**è**v**e**ment
0	"oin" [wœ]	S**oin**

L'ordre des opérations suit celui de la prononciation française. Si celui-ci n'est pas respecté, la prononciation est faussée et de ce fait, la probabilité de retomber sur la bonne orthographe est fortement réduite.

Nous avons également constitué manuellement une liste de mots MOTS_ER qui se prononcent "é" mais dont la terminaison est "er" ou "ed" et ce afin de les différencier des termes comme "cancer". (Exemples : pied ; gaucher ; manger ; traiter ; informer ; accréditer…).

Algorithme PHONEMISATION

1. Suppression des accents
2. ed/er = 8 si le mot est dans la liste MOTS_ER (mots qui se prononcent é)
3. ç=ss ;ill=i6.
4. Suppression des 'h' en début de mot (ils ne sont jamais prononcés)
5. ees/ets/eds/ers/ez/et/ee/ed/ = 8 ; ier/iers=i8.
6. y=ii ;ph=f ;
7. 'c' = 'ss' lorsqu'il est suivi de 'e', 'i', ou '8'.
8. eau=o ; oua=2.
9. Modifications selon les conditions du tableau 13
10. Remplacement du 's' s'il est suivi d'une voyelle sinon la lettre est doublée
11. 'œ' = 'e'
12. remplacements selon le tableau 14.
13. er/ere/erss/eress =8r. Le 's' a été doublé lors de la 12ème étape ce qui explique la suppression des deux dernières terminaisons.
14. 'eu' = 'e'.
15. Suppression des lettres dupliquées
16. ess/tss/t/kss/ss/e=8r.
17. Retour définitif du résultat

TAB.3.8.–Tableau de modifications.

Combinaison	Conditions		Modif
Caractéristique	Précédentes	Suivantes	
Ii		‘a','e','o','u','1','2','3','4','8','0'	6
An		'a','e','i','o','u','n','1','2','3','4','6','8','0'	4
Am		'a','e','i','o','u','n','m','1','2','3','4','6','7','8','0'	4
Ein		'a','e','i','o','u','n','1','2','3','4','6','8','0'	1
Ain		'a','e','i','o','u','n','1','2','3','4','6','8','0'	1
Eim		'a','e','i','o','u','m','1','2','3','4','6','8','0'	1
Aim		'a','e','i','o','u','m','1','2','3','4','6','8','0'	1
En		'a','e','i','o','u','n','1','2','3','4','6','8','0'	4
Em		'a','e','i','o','u','m','1','2','3','4','6','8','0'	4
Oin		'a','e','i','o','u','n','1','2','3','4','6','8','0'	0
In	'o','e','a'	'a','e','i','o','u','n','1','2','3','4','6','8','0'	1
Im	'o','e','a'	'a','e','i','o','u','m','1','2','3','4','6','8','0'	1
Un		'a','e','i','o','u','n','1','2','3','4','6','8','0'	1
Ni		'a','e','o','u','1','2','3','4','6','7','8','0'	7
Ge		'a','o','2','3','4','0'	g
Gu		'e','i','1','2','4','6','8','0'	g

L’étape 9 permet de faire des modifications sur le mot mais avec certaines conditions qui sont fonction des lettres qui suivent ou qui précèdent le groupe de lettres caractéristique. Par exemple dans le mot "insomnie" le groupe de lettre caractéristique ‘in’ sera remplacé par ‘1’ donnant le mot "1somnie". En revanche dans le mot "inosine" on retrouve aussi la même combinaison de lettre ‘in’ mais comme la lettre suivante est une voyelle, il n’y a pas de modifications sur le mot.

TAB.3.9.–Tableau de remplacements.

Combinaison	Modification	Combinaison	Modification
sch	5	l1	l8n
Ch	5	74	71
Sh	5	ro	ro1
Ai	8	omac	oma
Xs	ks	8mm	am
o6	26	si5	sik
oeu	8	gn	7
5r	kr	tion	sion

Combinaison	Modification	Combinaison	Modification
5t	kt	ptio	psio
5l	kl	ati4	assi4
5o	ko	c	k
U	o	qu	k
Oz1	os1	q	k
irop	iro	j	g
irops	iro	s	ss
thm	m	h	Ø
stme	sm	31	o
Am7	ami	ei	8
Dom1	dom8n	oi	2

Dans beaucoup de cas des lettres voire même des combinaisons de lettres ne sont pas prononcées et souvent en fin de mot. Le tableau 3.9 permet de traiter des cas comme 'sirop', 'estomac'…

TAB.3.10.–Correspondance des sons.

Phonémisation	Correction	Orthographe
Akuponktur	Akup1ktur	Acupuncture
Tabak	Taba	Tabac
vi6	Vil	Ville
s4g	S4	Sang

Tout comme l'indexation et la représentation des documents et des requêtes, l'espace de représentation phonétique doit être le même. De ce fait, afin de pouvoir comparer le son de deux chaînes et proposer la bonne orthographe nous avons créé un dictionnaire de référence Vocabulaire. Chaque mot du vocabulaire est une entrée de ce dictionnaire. La fonction Phonémisation que nous avons développée ne prend en entrée qu'un seul mot. De ce fait, nous ne pouvons pas considérer chaque terme du vocabulaire comme entrée de ce dictionnaire phonémisé. Tous les termes du vocabulaire d'origine sont segmentés puis minusculisés et phonémisés, en évitant les doublons. Ce dictionnaire permet de mapper la requête phonémisée et le mot phonémisé. Cette segmentation est également nécessaire dans les cas où par exemple un utilisateur formule la requête « cretzvelt » à la place de du terme « creutzfeldt-jakob, maladie ».

3.6.3.3 Application à la Reconnaissance de Termes

La phonémisation avant l'étape de lemmatisation permet de reconnaître 11 453 termes (soit 9.4%). Après l'étape de lemmatisation des termes simples et des termes composés cette

phonémisation permet de mapper 4 421 mots supplémentaires, réduisant ainsi le nombre de mots inconnus à 16 603 (soit 13.61% du total).

TAB.3.11.– Appariement exact (même sonorité entre la requête et le vocabulaire).

	Termes Reconnus	%
Phonémisation	4 421	3,63%
TOTAL	**105 355**	**86,36%**

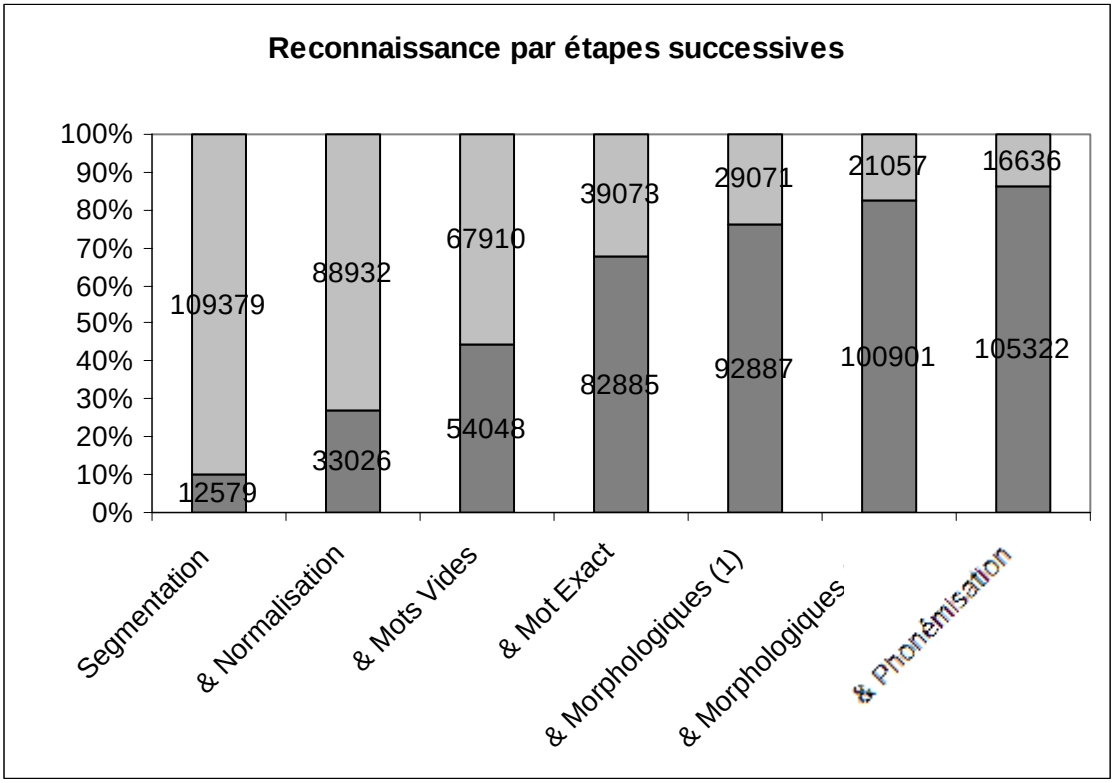


FIG.3.7.– Reconnaissance des mots par étapes successives (+ phonémisation).

3.6.4 Correction Orthographique

La correction orthographique des termes saisis par les utilisateurs, en fonction de la terminologie CISMeF, nous paraît une étape indispensable, surtout dans le domaine médical, où l'utilisation de termes parfois compliqués et de noms propres (comme pour les noms de syndromes par exemple) y est courante.

3.6.4.1 Algorithme

Dans le cas où il y a des erreurs de frappe le code final est complètement modifié et il devient quasiment impossible de retrouver la bonne orthographe du mot. Le but de la fonction CORRECTION est de rechercher l'orthographe la plus proche. Cette recherche est fondée sur un

système de scores et le mot qui a le meilleur score est retourné à l'utilisateur. Comme il est impossible à la machine de reconnaître automatiquement s'il y a une faute de frappe ou bien une inversion entre deux lettres ou encore un oubli ou un rajout d'une lettre, cette fonction doit traiter tous ces cas. Nous gérons ainsi plusieurs cas d'erreurs : inversion dans la chaîne, absence d'une lettre en début de chaîne, lettre en trop en début de chaîne, première lettre fausse, et seconde lettre fausse.

Tous les mots qui se rapprochent le plus du mot en paramètre sont sélectionnés du dictionnaire. Les critères de sélection sont la longueur de la chaîne ainsi que deux lettres communes de la phonémisation des mots du dictionnaire et de la requête. A chaque mot est associé un score afin de sélectionner le mot du vocabulaire qui se rapproche le plus du mot entré en paramètre. Cela permet également de diminuer les scores des mots dont la longueur diffère beaucoup du mot recherché.

Fonction CORRECTION (MOT DE LONGUEUR MAX 6 CARACTERES)

Début

Phonémiser (*mot*) ;

Enregistrer les permutations de lettres entre la 3ème et la 4ème, la 4ème et la 5ème et la 5ème et la 6ème.

score= $100/2^{\text{Length}(mot)}$

open c1(var, indexe_ph, length(ch_temp))

var : sous chaîne de deux lettres de la phonémisation du mot rentré en paramètre (mot)

indexe_ph : index des mots dans le vocabulaire (i)

length(ch_temp) : longueur de la chaîne phonémisée (longueur)

SELECT *Phonem*

FROM *TB_vocabulaire*

WHERE substr(*Phonem,* i, 2) like *mot* AND length(*Phonem*) BETWEEN longueur-2 and longueur+2

*score := score_initial - abs(length(Phonem) - length(ch_temp)) * 21;*

Poids = 20 lorsque qu'il n'y a aucune erreur sur la chaîne ou lorsqu'il y a les inversions entre la 4ème, la 5ème et la 6ème lettre.

Poids = 22 lorsque qu'il manque une lettre au début de la chaîne

Poids = 16 dans tous les autres cas

Si la combinaison de lettres appartenant à la phonémisation de *mot* appartient à un mot du dictionnaire à la bonne place **alors** le score = score + poids

Si la combinaison est décalée d'une lettre vers la droite ou la gauche **alors** le score = score + poids / 2

Si la combinaison n'est pas retrouvée dans le mot du dictionnaire **alors** le score reste inchangé.

Si la combinaison n'est pas retrouvée et la combinaison dépasse le 6ème caractère de *mot* **alors** score = score - 5

Le mot du vocabulaire possédant le plus grand score est proposé à l'utilisateur.

3.6.4.2 Evaluation des Résultats

Sur un échantillon de 100 requêtes, la correction orthographique opère avec une précision de 73,43%. En effet, certains termes se rapprochent du vocabulaire mais n'en font pas partie. (Par exemple '*colle*' se rapproche du mot clé réservé en anglais '*cold*').

Comme souligné dans *(Zweigenbaum et al., 2001a),* on entre dans un schéma classique dans la résolution de problèmes : au-delà d'un certain point, les efforts nécessaires pour améliorer les résultats sont de plus en plus importants pour un gain qui va en diminuant. Les mesures

effectuées concernent le recouvrement des mots individuels. La plupart des requêtes et des termes MeSH comprennent plusieurs mots, et leur appariement demande des conditions et des traitements supplémentaires.

3.7 Traitements en Ligne

Nous présentons ici les traitements qui ont été implémentées pour le traitement des requêtes dans le CISMeF. Les résultats obtenus grâce aux données morphologiques et à la phonémisation sont très encourageants. En revanche, les traitements étant plus exigeants, ils ne sont disponibles que sur un serveur en Intranet. D'autres traitements ont été mis en ligne et utilisent la technique du sac de mots ainsi que la notion d'adjacence.

3.7.1 Appariement à Base de Connaissances Morphologiques

L'algorithme consiste à segmenter les requêtes et à remplacer chaque mot par l'union des termes de sa famille morphologique. Il peut être testé sur l'Intranet du CHU.

Algorithme APPARIEMENT DE REQUETES A L'AIDE DE CONNAISSANCES MORPHOLOGIQUES

Entrée : Requête *R* ; Liste de Mots Vides *MV* ; Connaissances Morphologiques *BCM* ; Terminologie *T ;*

Sortie : Requête Booléenne *R_finale ;*

Début
liste_mot ← Segmentation(*R*) ;
Pour (*x* ← 1 ; *x* ≤ *k* ; *x*++) **faire**
mot_normalisé ← Normalisation(*R*)
Si *mot_normalisé* ∈ *MV* **alors** *liste_mot* ← *liste_mot* – {*mot_normalisé*} ;
Sinon
Si *mot_normalisé* ∈ *T* **alors** *R_final*← *R_final* ET *mot_normalisé*
Sinon
Si Mot_Exact (*mot_normalisé*)<>Ø **alors**
Pour (*y* ← 1 ; *y* ≤ *l* ;*y*++) **faire** *R_exact*← *R_exact* OU mot_exact_y ;
R_final ← *R_final* ET *R_exact*;
Sinon
Si Mot_Morpho (*mot_normalisé*)<>Ø **alors**
Pour (*z* ← 1 ; *z* ≤ *m* ; *z*++) **faire** *R_morpho*← *R_morpho* OU mot_morpho_z ;
R_final ← *R_final* ET *R_morpho* ;
FinSi
FinSi
FinSi
FSi
FinPour
Retourner (*R_final*);
Fin

Cet algorithme permet, par exemple, de remplacer la requête *complications de la prématurité* (ne donnant aucune réponse sur Doc'CIMSeF) par la requête booléenne :

(complications OU complications post-opératoires OU complications pré-opératoires OU complication grossesse OU accouchement compliqué OU grossesse compliquée) ET (prématuré OU prématuré maladies OU accouchement prématuré)

Le nombre de réponses passe de 0 à 18 avec une précision de 100%.

3.7.2 Autres Traitements

Les traitements actuellement mis en ligne concernent la méthode des sacs de mots ainsi que des tests sur l'adjacence des termes des requêtes en texte intégral. L'implémentation de la méthode des sacs de mots consiste à des listes linéaires chaînées. Nous donnons ci-après le pseudo algorithme correspondant qui consiste à segmenter, à éliminer les mots vides, à phonémiser et à classer par ordre alphabétique les mots des requtêtes. Le même traitement a été réalisé pour les termes du vocabulaire. Le terme du vocabulaire qui a en commun le maximum de mots en commun avec la requête est retourné. Le traitement est réalisé au fur et à mesure, jusqu'à ce que la requête soit appariée au maximum. Ces traitements donnent lieu à une requête booléenne. D'autres traitements sont réalisés pour les mots qui n'ont pas été reconnus dans la requête.

Algorithme IDENTIFICATION DE MOTS RESERVES (SAC DE MOTS)

Entrée : Requête *R* ; Liste de Mots Vides *MV* ; Terminologie *T*.
Sortie : Requête booléenne *R_finale* ;
Début
    Tant Que (¬ FinChaine(*R*))
        mot ← *R*.next ;
        Si *mot* ∉ *MV* **alors** *liste_mot* ← *liste_mot* + (PHONEMISATION(*mot*)) ;
    FinTantQue
    liste_mot.Trier() ;
    Tant Que (¬ *liste_mot.MotsNonTraites().EstVide()*)
        sous_liste_mot ← *liste_mot.MotsNonTraites()* ;
        Tant Que (*liste_mot_connue.EstVide()* **et** ¬FinListe())
            Si *sous_liste_mot ()* ∈ *T* **alors** *liste_mot_connue* ← *sous_liste_mot* ;
            sous_liste_mot ← *liste_mot* – {Element(*i*) };
        FinTantQue
        Si (¬ *liste_mot_connue.EstVide()*) **alors**
            R_final ← *R_final* ET *liste_mot_connue*
            liste_mot.MarquerCommeTraite (liste_mot_connue)
        Sinon
            Si (*liste_mot.MotsNonTraites().NbMots() = 1*) **alors**
            liste_mot.MarquerCommeTraite (mot_inconnu)
        FinSi
    FinTantQue
    Retourner (*R_final*) ;
Fin

L'appariement de termes à plusieurs mots est plus exigeant. La politique de l'équipe CISMeF concerant le traitement des requêtes consiste à limiter le silence de Doc'CISMeF au maximum.

Le processus de traitement des requêtes actuellement mis en place consiste à reconnaître si l'expression saisie est un terme réservé ou non. Si l'expression n'est pas un terme réservé, un découpage des termes est effectué pour rechercher si l'expression contient un ou plusieurs termes réservés. Par exemple, la requête *enfant asthme* est traduite par (*enfant.mr* ET *asthme.mr*), où *enfant* et *asthme* sont des termes réservés (mr).

Les termes réservés sont reconnus, quel que soit l'ordre des mots saisis grâce à la méthode des sacs de mots. Cela est utile si l'utilisateur n'a pas une connaissance approfondie de la terminologie. Ainsi, par exemple, si l'utilisateur saisit l'expression *aspergillose allergique bronchopulmonaire*, l'expression est reconnue automatiquement comme le terme réservé *aspergillose bronchopulmonaire allergique*.

Dans le cas où aucun résultat ne serait retourné, la recherche est lancée sur différents champs. Le champ *titre* des documents est considéré comme prioritaire. Les mots vides sont supprimés et la recherche est effectuée sur l'union des termes, incluant une troncature à droite, dans le champ titre (ti), selon la requête *terme1*.ti* ET *terme2*.ti* ET *terme3*.ti*.

Si aucun résultat n'est retourné, le système de traitement des requêtes recherche si certains termes sont des termes réservés et lance la recherche pour les autres termes sur le champ titre. Ainsi, par exemple, la saisie d'allergie *infantile* aboutit à la formulation de la requête suivante : *allergie.mr* ET *infantile.ti*.

Si cette requête n'aboutit pas, la recherche est lancée sur tous les champs (tc) des documents pour les termes non reconnus comme étant des termes réservés et également avec l'expression complète dans tous les champs avec une adjacence à *at* mots ($at = 5 \times (\text{nb mots} - 1)$). Par exemple, la requête *les problèmes respiratoires des enfants* est remplacée par la requête booléenne *(enfant.mr* ET *problemes.tc* ET *respiratoires.tc)* OU *(problemes respiratoires enfant.at)*. Dans cette requête, *enfant* est reconnu comme un terme réservé car il a la même sonorité que *enfants*, *problèmes* et *respiratoires* sont recherchés dans le contenu de tous les champs, et l'expression *problèmes respiratoires enfants* est recherché dans tous les champs avec une adjacence de 10 (ces 3 mots ne doivent pas se trouver éloignés de plus de 10 mots).

Si cette dernière étape n'aboutit à aucun résultat, la recherche est lancée sur le texte intégral des ressources avec une adjacence de *ap* mots ($ap = 10 \times (\text{nb mots} - 1)$). Par exemple, le traitement de la requête *bronchite asthmatiforme* permettra de lancer la recherche selon la requête *bronchite asthmatiforme.ap* où les termes *bronchite* et *asthmatiforme* ne devront pas être éloignés de plus de 10 mots dans les textes intégraux.

Enfin, la dernière étape du processus de traitement utilise le texte intégral des ressources lorsque le ou les terme(s) saisis ne font pas partie de la terminologie, ce qui permet de réduire considérablement le silence des résultats, mais en diminuant à priori la précision qui reste à évaluer.

Dans le but d'informer les utilisateurs du traitement qui a été effectué sur leurs requêtes, un système de score (de 1 à 5) intuitif révèle la pertinence de la requête transformée en fonction des traitements réalisés.

TAB.3.12.– Pertinence des requêtes.

Score	Interprétation de la requête	Exemple
★★★★★	requête composée de mots réservés	*asthme enfant*
★★★★☆	requête présente dans le titre de certaines ressources	*enfant asthmatique*
★★★☆☆	Requête composée de mots réservés et de mots présents dans le titre de certaines ressources	*allergie infantile*
★★☆☆☆	requête composée de mots réservés et/ou de mots présents (et proches les uns des autres) dans la description de certaines ressources	*problèmes respiratoires enfant*
★☆☆☆☆	requête composée de mots présents (et proches les uns des autres) dans le contenu de certaines ressources	*bronchite asthmatiforme*

Un échantillon de requêtes a été lancé automatiquement à l'aide de Servlets sur le serveur en intranet et le serveur en ligne. Les résultats ont été récupérés au niveau de fichier XML pour être analysés (selon le même principe de Cogni-CISMeF § Chapitre 2)

Des expérimentations il en ressort que la combinaison des traitements morphologiques et la méthode du sac de mots obtiennent les meilleurs résultats en terme de précision des documents retournés. Les autres traitements (adjacence et recherche sur les autres champs) permettent de réduire le silence du système en revanche la précision des documents retournés diminue de près de 30%.

Nous avons montré dans ce Chapitre que grâce à plusieurs séries d'expériences, que des traitements linguistiques de base, mais également à base de connaissances, permettent d'apparier au maximum les requêtes des utilisateurs avec le vocabulaire utilisé pour l'indexation. Dans le Chapitre 4 nous considérons que les requêtes ne comprennent aucune erreur d'orthographe. Nous y proposons d'enrichir les requêtes des utilisateurs (intéractivement) par termes qui sont en relation d'association. Ces relations d'association sont générées par un processus de fouille de données.

BIBLIOGRAPHIE

(*Amar, 2000*) M. AMAR. (2000) Les Fondements théoriques de l'indexation une approche linguistique. *ADBS éditions, Paris*.

(*Arampatzis et al., 2000*) ARAMPATZIS TH., VAN DER WEIDE P., KOSTER C., VAN BOMMEL P. (2000) Linguistically-motivated information retrieval. In *Encyclopedia of Library and Information Science*.

(*Baud et al., 1992*) BAUD R, RASSINOUX AM., SCHERRER JR. (1992) Natural language processing and semantical representation of medical texts. *Methods Inf Med* 31:117–25.

(*Baud et al., 1997*) BAUD R, LOVIS C, RASSINOUX AM, MICHEL PA, SCHERRER JR. (1997) Extracting linguistic knowledge from an international classification. Proceedings of *Medical Informatics Europe'97*.

(*Bodenreider & Mccray, 1998*) BODENREIDER O, MCCRAY AT. (1998) From French vocabulary to the Unified Medical Language System: A preliminary study. Proc *9th World Congress on Medical Informatics*.

(*Bourigault & Jacquemin, 1999*) BOURIGAULT D, JACQUEMIN C. (1999) Term extraction + term clustering : an integrated platform for computer-aided terminology. Actes *conference of the european association for computational linguistics EACL'99,* pp. 15-22.

(*Bourigault, 1994*) BOURIGAULT D. (1994) Lexter un logiciel d'extraction de terminologie application à l'extraction de connaissances à partir de textes. *Thèse de Doctorat, Paris.*

(*Brouard, 2004*) BROUARD F. (2004) L'art des Soundex. En ligne : *http://sqlpro.developpez.com/cours/soundex/*

(*Burnage, 1990*) BURNAGE G. (1990) CELEX - A Guide for Users. Nijmegen: Centre for Lexical Information, *University of Nijmegen.*

(*Chevallet & Haddad, 2001*) J.P. CHEVALLET, M. H. HADDAD. (2001) Proposition d'un modèle relationnel d'indexation syntagmatique : mise en oeuvre dans le système iota. In *Congrès Informatique des Organisations et Systèmes d'Information et de Décision (INFORSID'2001)*, pp.465–483.

(*Clavier, 1996*) CLAVIER V. (1996) Modélisation de la suffixation pour le traitement automatique du français. Application à la recherche d'information. *Thèse de Doctorat, Grenoble.*

(*Clémenceau, 1992*) CLEMENCEAU D. (1992) Structuration du lexique et reconnaissance des mots dérivés. *Thèse de doctorat, Université Paris 7.*

(*Content et al.,1990*) CONTENT A, MOUTSY P, RADEAU M. (1990) BRULEX une base de données lexicales informatisée pour le français écrit et parlé . *L'Année psychologique,* pp.551-556

(*Daille & Jacquemin, 1998*) DAILLE, B. ET JACQUEMIN, C. (1998) Lexical Database and Information Access: A Fruitful Association. In Proceedings of the *First International Conference on Language Resources and Evaluation*, pp. 669–673.

(*Daille, 1999*) DAILLE B. (1999) Identification des adjectifs relationnels en corpus. *TALN 1999* (Traitement automatique des langues naturelles), pp. 105–114.

(*Daille, 1999*) DAILLE B. (1999) Identification des adjectifs relationnels en corpus. Actes *TALN 1999.*

(*Dal et al., 1999*) DAL G., NAMER F., HATHOUT N. (1999) Construire un lexique dérivationnel : théorie et réalisations. *Actes de TALN 1999*

(*David & Plante, 1990*) DAVID S., PLANTE P. (1992) De la nécessité d'une approche morpho-syntaxique dans l'analyse des textes. *Intelligence artificielle et sciences cognitives au Québec*, 3(3) : 140-154

(*David & Plante, 1990*) DAVID S., PLANTE P. (1992) Le progiciel Termino : de la nécessité d'une analyse morphosyntaxique pour le dépouillement terminologique des textes. Actes du *colloque international sur les industries de la langue : perspectives des années 1990.* pp.71-88.

(*De Loupy, 2000*) DE LOUPY C (2000) Evaluation de l'apport de connaissances linguistiques en désambiguïsation sémantique et recherche d'information, *Thèse Doctorat, Université d'Avignon et des Pays de Vaucluse.*

(*De Saussure, 1972*) F. DE SAUSSURE. (1972) Cours de linguistique générale. *Ed. Critique de Tullio de Mauro.*

(*Enguehard & Pantera, 1995*) ENGUEHARD C., PANTERA L . (1995) Automatic natural acquisition of a terminology . *Journal of quantitative linguistics* 2(1) :27-32

(*Enguehard, 1992*) ENGUEHARD C. (1992) Acquisition naturelle automatique d'un réseau sémantique. *Thèse de doctorat, Université de Compiègne.*

(*Faraj et al., 1996*) FARAJ N., GODIN R., MISSAOUI R., DAVID S., PLANTE P. (1996) Analyse d'une méthode d'indexation automatique basée sur une analyse syntaxique de textes. *Canadian journal of information and library science. Revue information et bibliothéconomie* 21(1):1-21

(*Fradin, 1994*) FRADIN B. (1994) L'approche à deux niveaux en morphologie computationnelle et les développements récents de la morphologie, *T.A.L.* 35-2, pp.9-48.

(*Gaussier et al., 2000*) GAUSSIER E., GREFENSTETTE G., HULL D. & ROUX C. (2000) Recherche d'information en français et traitement automatique des langues. *Traitement automatique des langues*, 41(2) : 473–493.

(*Gaussier, 1999*) GAUSSIER E. (1999). Unsupervised learning of derivational morphology from inflectional lexicons. *ACL workshop on Unsupervised Methods in Natural Language Learning.*

(*Grabar & Zweigenbaum, 2000*) GRABAR N., ZWEIGENBAUM P. (2000) Automatic acquisition of domain-specific morphological resources from thesauri. In Proceedings of *RIAO 2000* : *Content-Based Multimedia Information Access,* pp. 765–784.

(*Grabar et al., 2002*) GRABAR N., ZWEIGENBAUM P., SOUALMIA LF., DARMONI SJ. (2002) Les utilisateurs de Doc'CISMeF peuvent-ils trouver ce qu'ils cherchent ? Une étude de l'adéquation du vocabulaire des requêtes des utilisateurs du MeSH; *Journées Francophones d'Informatique Médicale*, pp.158-169.

(*Grabar et al., 2003*) GRABAR N., ZWEIGENBAUM P., SOUALMIA LF., DARMONI SJ. (2003) Matching Controlled Vocabulary; *Medical Informatics Europe*, pp.445-450.

(*Grabar & Zweigenbaum, 2000*) GRABAR N., ZWEIGENBAUM P. (2000) A general method for sifting linguistic knowledge from structured terminologies. *J Am Med Inform Assoc*; 7(suppl):310–4.

(*Grabar, 2004*) GRABAR. (2004) Terminologie médicale et morphologie : acquisition de ressources morphologiques et leur utilisation pour le traitement de la variation terminologique. *Thèse de Doctorat, Paris 6.*

(*Grefenstette, 1997*) GREFENSTETTE G. (1997) Short query linguistic expansion techniques : palliating one-word queries by providing intermediate structure to texts. *Lecture Notes in Computer Science # 1299* pp.97-114

(*Hamon et al., 1998*) THIERRY HAMON, ADELINE NAZARENKO, AND CÉCILE GROS. (1998) A step towards the detection of semantic variants of terms in technical documents. Proceedings of the *17th COLING,* pp.498–504.

(*Hamon et al., 1999*) HAMON, T., GARCIA, D., ET NAZARENKO, A. (1999) Détection de liens de synonymie : complémentarité des ressources générales et spécialisées. In *Actes de Terminologie et Intelligence Artificielle TIA'99*, pp.45–58.

(*Hathout et al., 2002*) HATHOUT N, NAMER F, DAL G. (2002) An experimental constructional database: the MorTAL project. In: *Boucher P Ed., Many morphologies*. pp.178–209.

(*Hathout, 2001*) HATHOUT N. (2001) Analogies morpho-synonymiques. Actes de *TALN 2001*.

(*Hull, 1996*) HULL, D. A. (1996) Stemming algorithms : A case study for detailed evaluation. *Journal of the American Society of Information Science*, 47(1) :70–84.

(*Imbs, 1971*) Imbs. (1971) Etudes statistiques sur le vocabulaire français. dictionnaire des fréquences vocabulaire littéraire des XIX et XX siècles, centre de la recherche pour un trésor de la langue française (*CNRS*) Nancy

(*INSERM, 2000*) INSERM (2000). Thésaurus Biomédical Français/Anglais. *Institut National de la Santé et de la Recherche Médicale*, Paris.

(*Jacquemin & Tzoukermann, 1999*) JACQUEMIN C., TZOUKERMANN E. (1999) NLP for term variant extraction: A synergy of morphology, lexicon, and syntax.

(*Jacquemin, 1997 a*) JACQUEMIN CH. (1997) « Guessing morphology from terms and corpora », Proceedings, *20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'97).*

(*Jacquemin, 1997 b*) JACQUEMIN C. (1997b) Variation terminologique : Reconnaissance et acquisition automatique de termes et de leurs variantes en corpus. *Mémoire d'habilitation à diriger des recherches, Université de Nantes.*

(*Jacquemin, 1999*) JACQUEMIN C. (1999) Syntagmatic and pragmatic representations of term variation. *37 eme annual meeting of the association for computational linguistics (ACL'99)* pp.341-348.

(*Jing & Tzoukerman, 1999*) JING, H. & TZOUKERMAN, E. (1999) Information Retrieval based on Context Distance and Morphology. In Proceedings of *22th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'99)*, pp.90–96.

(*Jouis, 1993*) C. JOUIS. (1993) Contributions à la conceptualisation et à la modélisation des connaissances à partir d'une analyse linguistique de textes. réalisation d'un prototype : le système Seek. *Thèse de Doctorat, EHESS*, Paris.

(*Karttunen et al., 1997*) KARTTUNEN L., GAÁL T., KEMPE A. (1997) Xerox Finite-State Tool. *Technical Report. Xerox Research Centre Europe*, Grenoble.

(*Klavans et al., 1997*) KLAVANS, J.L. JACQUEMIN, C. & TZOUKERMANN, E. (1997) A Natural Language Approach to Multi-Word Term Conflation. In Proceedings, *DELOS Workshop on Cross-Language Information Retrieval.*

(*Koskenniemi, 1983*) KOSKENNIEMI, K. (1983) Two-level model for Morphological Analysis, *8th IJCAI Conference*.

(*Krovetz, 1993*) R. KROVETZ. (1993) Viewing Morphology as an Inference Process. *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*; pp.191-203.

(*Lambert & Chesnet. , 2001*) LAMBERT E, CHESNET D. (2001) Novlex : une base de données lexicale pour les élèves de primaire, *L'Année psychologique* 01,pp.277-288.

(*Losee, 1996*) R. M. LOSEE. (1996) How part-of-speech tags affect text retrieval and filtering performance.

(*Lovis & Baud, 2000*) LOVIS, C., BAUD, R. (2000) Fast Exact String Pattern-Matching Algorithms Adapted to the Characteristics of the Medical Language. *JAMIA*, 7(4):378–391.

(*Lovis et al., 1998*) LOVIS C, BAUD R, RASSINOUX AM, MICHEL PA, AND SCHERRER JR. (1998) Medical dictionaries for patient encoding systems: a methodology. *Artif Intell Med.* 14:201–14.

(*Mc Cray et al., 1994*) MC CRAY AT, SRINIVASAN S, AND BROWNE AC. (1994) Lexical methods for managing variation in biomedical terminologies. In Proc *18th Annu Symp Comput Appl Med Care,* pp.235–239.

(*Mohri et al., 2000*) MOHRI M., PEREIRA F., RILEY M. (2000) The design principles of a weighted finite-state transducer library. *Theoretical Computer Science*, 231:17-32.

(*Mooers, 1950*) MOOERS C. (1950) The theory of digital handling of non-numerical information and applications to machine economics; *Technical bulletin No. 48 ; Cambridge*, MA, USA.

(*Nakamura, 2003*) NAKAMURA T. (2003) Analysing texts in a specific domain with local grammars: The case of stock exchange market reports, In Proceedings of the *First International Conference on Linguistic Informatics*.

(*Namer, 2000*) NAMER F. (2000) FLEMM : un analyseur flexionnel du français à base de règles. *Traitement Automatique des Langues 2000*; 41(2):523–47.

(*Namer, 2002*) NAMER F. (2002) Acquisition automatique de sens à partir d'opérations morphologiques en français : études de cas. Actes de *TALN 2002*.

(*New et al., 2001*) NEW, B., PALLIER, C., FERRAND, L. AND MATOS R. (2001) Une Base de Données Lexicales du Français Contemporain sur Internet: lexique, *L'Année psychologique*, 447-462.

(*New et al., 2005*) NEW B, PALLIER C, BRYSBAERT M, FERRAND L. (2004) Lexique 2 a new French lexical database. *Behavior, research methods instruments and computers*. (in press)

(*OMS, 1993*) ORGANISATION MONDIALE DE LA SANTE. (1993) Classification statistique internationale des maladies et des problèmes de santé connexes — Dixième révision, Genève.

(*Paumier, 2003*) PAUMIER S. (2001). Some remarks on the application of a lexicon-grammar, Lingvisticæ Investigationes XXIV:2, *Amsterdam/Philadelphia, John Benjamins*, pp. 245-256.

(*Porter, 1980*) M.F. PORTER. (1980) An algorithm for suffix stripping; *Program 14 (3)* ; pp. 130-137.

(*Pouliquen et al., 2002*) POULIQUEN B, DELAMARRE D, ET LE BEUX P. (2002) Indexation de textes médicaux par extraction de concepts, et ses utilisations. In: *Actes de JADT, 2002*.

(*Revuz, 1991*) REVUZ D. (1991) Dictionnaires et lexiques : méthodes et algorithmes. *Thèse de Doctorat en Informatique*, Paris 7.

(*Riloff, 1995*) E. RILOFF. (1995) Little words can make big difference for text classification; Actes de *18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* ; pp.130-136.

(*Roche & Schabes, 1997*) ROCHE E., SCHABES Y. (Eds.) (1997) Finite-State Language Processing, *MIT Press*.

(*Royauté et al., 1992*) ROYAUTE J., SCHMIDT L., OLIVETAN E. (1992) Les expériences d'indexation à l'INIST; *COLING-92* ; pp. 1058-1063.

(*Savoy, 1993*) SAVOY J. (1993) Stemming of French Words Based on Grammatical Categories, *JASIS: Journal of the American Society for Information Sciences*, Vol. 44 :1, pp. 1-9.

(*Savoy, 2002*) SAVOY J. (2002) Morphologie et recherche d'information. Cahier de recherche en informatique CR-I-2002-01, *Université de Neuchatel, Division économique et sociale, Faculté de Droit et des Sciences Économiques*.

(*Schmid, 1994*) SCHMID H. (1994) Probabilistic part-of-speech tagging using decision trees. In: Proceedings of the *International Conference on New Methods in Language Processing*, pp.44–49.

(*Silberztein, 1993*) SILBERZTEIN M. (1993) Dictionnaires électroniques et analyse automatique de textes : le système INTEX. *Masson, Paris*.

(*Snomed, 1996*) SNOMED (1996) Côté RA. Répertoire d'anatomopathologie de la SNOMED internationale, v3.4. *Université de Sherbrooke, Sherbrooke*, Québec.

(*Soualmia & Darmoni, 2003 a*) SOUALMIA LF., DARMONI SJ. (2003 a) Une Terminologie Orientée Ontologie pour la Recherche d'Information sur la Toile; *Journées Francophones de la Toile*, pp.185-194.

(*Soualmia & Darmoni, 2003 b*) SOUALMIA LF., DARMONI SJ. (2003 b) Projection de Requêtes pour une Recherche d'Information Intelligente sur le Web. *Rencontres Jeunes Chercheurs en Intelligence Artificielle*

(*Soualmia & Darmoni, 2004 a*) SOUALMIA LF., DARMONI SJ. (2004 a) Combining Knowledge-based Methods to Refine and Expand Queries in Medicine; *FQAS, Flexible Query Answering Systems Lectures Notes in Artificial Intelligence # 3055* .pp 243-255

(*Soualmia & Darmoni, 2004 b*) SOUALMIA LF., DARMONI SJ. (2004 b) Correcting and Refining Users Queries: the Contribution of Morphological Knowledge and Association rules; *IPMU, Information Processing and Management of Uncertainty in Knowledge-Based Systems,* pp. 2059-2066.

(*Soulé-Dupuy, 1990*) SOULE-DUPUY C. (1990) Systèmes de recherche d'informations : mécanismes d'indexation et d'interrogation. *Thèse de Doctorat, Université Toulouse III*.

(*Sproat, 1992*) SPROAT, R.W. (1992) Morphology and Computation. *Cambridge: MIT Press*.

(*Tanguy & Hathout, 2002*) TANGUY L. & HATHOUT N. (2002) Webaffix : un outil d'acquisition morphologique dérivationnelle à partir du Web. *Actes de TALN 2002*, pp.245–254.

(*Woods, 2000*) WOODS, W.A. (2000). Aggressive Morphology for Robust lexical Coverage. *Sixth Applied Natural Language Processing Conference,* pp.218-223.

(*Xu & Croft, 1998*) XU J.,CROFT BW. (1998) Corpus-based stemming using co-occurrence of word variants. *ACM Transactions on Information Systems*; 16(1):61–81.

(*Zweigenbaum & Grabar, 1999*) ZWEIGENBAUM, GRABAR (1999) A contribution of medical terminology to medical language processing resources: experiments in morphological knowledge acquisition from thesauri. In working group 6 of *IMIA* pp. 155-167

(*Zweigenbaum & Grabar, 2002*) ZWEIGENBAUM, GRABAR. (2002) Restoring accents in unknown biomedical words: application to the French mesh thesaurus. *International Journal of Medical Informatics* pp.113-126.

(*Zweigenbaum & Grabar, 2003*) ZWEIGENBAUM P., GRABAR N. (2003) Corpus based associations provide additional morphological variants to medical terminologies. *American Medical Informatics Association* pp.768-772

(*Zweigenbaum & Menelas, 1994*) ZWEIGENBAUM P, CONSORTIUM MENELAS. (1994) Menelas: an access system for medical records using natural language. *Comput Methods Programs Biomed* 45:117–20.

(*Zweigenbaum et al., 2001 a*) ZWEIGENBAUM P, GRABAR N DARMONI SJ (2001 a) L'apport de connaissances morphologiques pour la projection de requêtes sur une terminologie normalisée. *Traitement Automatique du Langage Naturel 2001,* pp.403-408

(*Zweigenbaum et al., 2001 b*) ZWEIGENBAUM P, DARMONI SJ, ET GRABAR N. (2001 b) The contribution of morphological knowledge to French MeSH mapping for information retrieval. *Journal of the American Medical Informatics Association*; 8(suppl):796–800.

(*Zweigenbaum et al., 2003*) ZWEIGENBAUM P., HADOUCHE F., GRABAR N. (2003) Apprentissage de relations morphologiques en corpus. *TALN 2003*

(*Zweigenbaum et al., 2004*) ZWEIGENBAUM P, BAUD R BURGUN A NAMER F JARROUSSE E GRABAR N RUCH P LE DUFF F FORGET JF DOUYERE M DARMONI SJ. (2004) UMLF a unifed medical lexicon for French *International Jounal of Medical Informatics*. In press.

(*Zweigenbaum & Courtois, 1998*) ZWEIGENBAUM P, COURTOIS P. (1998) Acquisition of lexical resources from SNOMED for medical language processing. Proc *9th World Congress on Medical Informatics*, pp.586–90.

(*Zweigenbaum, 2001*) ZWEIGENBAUM P. (2001) Ressources pour le domaine médical : terminologies, lexiques et corpus médicaux. *Lettre de l'ELRA* 6(4):8–11.

CHAPITRE 4

DECOUVERTE DE CONNAISSANCES

DES TEXTES AUX REGLES D'ASSOCIATION

SOMMAIRE

4.1 INTRODUCTION 153
4.2 FOUILLE DE DONNEES 154
4.2.1 FOUILLE DE DONNEES TRANSACTIONNELLES 154
4.2.2 EXTRACTION D'ITEMSETS 158
4.2.3 EXTRACTION D'ITEMSETS FREQUENTS 159
4.2.4 PROCESSUS D'EXTRACTION DE REGLES D'ASSOCIATION 162
4.2.4.1 PREPARATION DES DONNEES 163
4.2.4.2 EXTRACTION DES ENSEMBLES FREQUENTS D'ATTRIBUTS 163
4.2.4.3 GENERATION DES REGLES D'ASSOCIATION 163
4.2.4.4 INTERPRETATION DES RESULTATS 163
4.2.5 AUTRES METHODES 164
4.2.5.1 DECOUVERTE DE PEPITES DE CONNAISSANCES 164
4.2.5.2 EXTRACTION D'ITEMSETS FREQUENTS MAXIMAUX 165
4.2.5.3 CONTRAINTES SUR LES ITEMS 165
4.2.5.4 EXTRACTION D'ITEMSETS FERMES FREQUENTS 165
4.2.6 ALGORITHMES CLOSE ET A-CLOSE 166
4.2.6.1 ALGORITHME CLOSE 167
4.2.6.2 ALGORITHME A-CLOSE 167
4.2.6.3 PROBLEME DE LA PERTINENCE ET DE L'UTILITE DES REGLES D'ASSOCIATION 172
4.2.7 BASES POUR LES REGLES D'ASSOCIATION 173
4.2.7.1 BASE GENERIQUE POUR LES REGLES D'ASSOCIATION EXACTES 174
4.2.7.2 BASE INFORMATIVE POUR LES REGLES D'ASSOCIATION APPROXIMATIVES 176
4.3 AUTRES MESURES STATISTIQUES 179
4.4 FOUILLE DE TEXTES ET FOUILLE DU WEB 182
4.4.1 PRINCIPE DE LA FOUILLE DE TEXTES 182
4.4.2 SIGNIFICATION DES REGLES D'ASSOCIATION 183
4.4.3 PROCESSUS D'EXTRACTION 184
4.4.4 SYSTEMES DE FOUILLE DE TEXTES 185
4.4.5 FOUILLE DU WEB 187

4.5 EVALUATION DES REGLES D'ASSOCIATION 188
4.6 EXPERIENCES D'EXTRACTION DE REGLES D'ASSOCIATION 189
4.6.1 FOUILLE DE DONNEES 189
4.6.1.1 MOTS CLES 191
4.6.1.2 MOTS CLES OU QUALIFICATIFS 191
4.6.1.3 ASSOCIATIONS MOT CLE/QUALIFICATIF 192
4.6.2 FOUILLE DE DONNEES AVEC CATEGORISATION 192
4.6.2.1 MOTS CLES 195
4.6.2.2 ASSOCIATION MOTS CLES/QUALIFICATIFS 195
4.6.3 FOUILLE DE TEXTES 196
4.6.4 FOUILLE DE DONNEES PONDEREES 197
4.6.4.1 MOTS CLES EN MAJEUR 197
4.6.4.2 ASSOCIATIONS MOTS CLES/QUALIFICATIFS EN MAJEUR 198
4.6.5 EVALUATION DES REGLES D'ASSOCIATION 200
4.7 EXPLOITATION POUR LA RECHERCHE D'INFORMATION 202
4.8 CREATION DE REGLES EXPERTES 204
4.9 QUELQUES PERSPECTIVES 206
C

Dans ce Chapitre 4, nous proposons d'adopter une nouvelle approche pour la recherche d'information, à savoir l'exploitation de règles d'association déduites à partir du contenu des documents. Pour cela, nous utilisons un processus d'extraction de connaissances 'enfouies' dans les données. Nous détaillons les algorithmes que nous avons appliqués. Ils se fondent sur une analyse formelle de treillis de concepts. Nous décrivons enfin comment ces règles d'association peuvent être exploitées dans la recherche d'information.

4.1 Introduction

L'extraction de connaissances ou encore fouille de données dans des bases de données est une activité qui consiste à analyser un ensemble de données brutes de façon à en extraire des connaissances exploitables. Ces *nouvelles* connaissances sont initialement sous la forme de motifs intéressants (non triviaux, implicites, précédemment inconnus et potentiellement utiles). Les connaissances extraites peuvent être utilisées pour faire des prédictions à propos de données nouvelles ou pour expliquer des données existantes. Un système d'extraction de connaissances à partir de bases de données peut s'appuyer sur des connaissances du domaine lors du processus d'extraction. Les applications de l'extraction de connaissances à partir de données vont de l'analyse de marché (problématique de l'analyse du « panier de la ménagère ») à la fouille de texte et du Web, en passant par la gestion des risques et l'analyse de données scientifiques. La finalité du processus d'extraction, outre la découverte de nouvelles connaissances, est la génération de règles d'association, la classification, l'estimation, la prédiction, le clustering…etc.

Un système de fouille de données peut générer plusieurs milliers voire des millions de motifs fréquents dont seulement quelques-uns sont intéressants pour un utilisateur donné. Un motif est intéressant s'il est facilement compréhensible par les utilisateurs, valide sur de nouvelles données avec un degré de certitude, potentiellement utile, nouveau, ou s'il confirme une hypothèse. Il existe des mesures d'intérêt objectives : basées sur des statistiques (support, confiance…etc.) et des mesures d'intérêt subjectives : basées sur les connaissances des utilisateurs à propos des données (motifs inattendus).

Dans notre étude nous nous intéressons à l'extraction de règles d'association. Les règles d'association ont été utilisées avec succès dans de nombreux domaines, parmi lesquels l'aide à la planification commerciale, l'aide au diagnostic et en recherche médicale, l'amélioration des processus de télécommunications, l'organisation et l'accès aux sites Internet, et l'analyse d'images, de données spatiales, de données géographiques et de données statistiques. Nous proposons ici de les exploiter dans un processus de recherche d'information afin d'enrichir les requêtes des utilisateurs avec de nouvelles connaissances issues des documents eux-mêmes. Pour nos expérimentations nous utilisons la base de données relationnelle du CISMeF. Nous adaptons ensuite le processus de fouille de données à la fouille de textes avec en entrée les textes intégraux du catalogue. L'extraction peut être vue comme le processus alimentant un système à base de connaissances : les connaissances extraites sont stockées dans la base pour être réutilisées lors du traitement des requêtes.

C'est souvent un expert du domaine relatif aux données « l'analyste » qui est chargé de diriger l'extraction en fonction de ses objectifs. Dans notre cas, l'expert interviendra en fin de processus, dans la phase d'évaluation des règles d'association extraites.

4.2 Fouille de Données

Un exemple de règle d'association extraite d'une base de données de ventes de supermarché est : « céréales, sucre → lait (support 7%, confiance 50%)». Cette règle indique que les clients qui achètent des céréales et du sucre ont également tendance à acheter du lait. La mesure de *support* définit la portée de la règle, c'est à dire la proportion de clients qui ont acheté les trois articles, et la mesure de *confiance* définit la précision de la règle, c'est à dire la proportion de clients qui ont acheté du lait parmi ceux qui ont acheté des céréales et du sucre. L'extraction de règles d'association consiste à extraire les règles dont le support et la confiance sont au moins égaux à des seuils minimaux de support et de confiance définis par l'utilisateur.

4.2.1 Fouille de Données Transactionnelles

Une base de données transactionnelles est composée d'un ensemble d'enregistrements. Chaque enregistrement est décrit par un ensemble d'attributs. Chaque attribut est à valeur booléenne ou discrète. Les exemples classiques de données transactionnelles sont les tickets de caisse d'un supermarché, les résultats des recensements, les réponses à un sondage d'opinions…etc.

TAB.4.1.– Exemples de transactions.

Identifiant	Transaction
1	Beurre, Fruits, Lait, Pain
2	Fruits, Lait, Pain
3	Beurre, Fromage, Pain, Pâtes, Viande, Soda
4	Fromage, Fruits, Lait, Légumes, Pain, Pâtes, Poisson
5	Beurre, Fruits, Lait, Légumes, Pain, Pâtes, Poisson, Viande
6	Beurre, Fromage, Légumes, Pain, Pâtes, Viande, Soda
7	Beurre, Fromage, Lait, Légumes, Pain, Pâtes, Viande, Soda
8	Fruits, Légumes, Poisson
9	Beurre, Fromage, Lait, Pain, Pâtes, Viande, Soda
10	Beurre, Fromage, Fruits, Lait, Légumes, Pain, Poisson, Viande

Le tableau 4.1 présente un exemple de données transactionnelles. La première mesure importante en fouille de données est le support (ou fréquence) des attributs. Sur l'exemple du tableau, le support de l'attribut lait est de 7/10 cad que 70% des transactions (enregistrements)

contiennent l'attribut lait. Le support d'un ensemble d'attributs peut également être exprimé : le support de l'ensemble des attributs {pain, lait, fromage} est de 4/10.

L'extraction de connaissances à partir de ces données peut prendre différentes formes :

– La recherche des attributs les plus fréquemment présents dans les transactions (pour l'exemple donné et par ordre décroissant de support nous obtenons pain, lait beurre, pâtes…etc.)

– Recherche des transactions contenant un ou plusieurs attributs donnés.

– Recherche de relations entre attributs permettant d'expliquer le comportement des acheteurs. Par exemple, tous les clients qui achètent du fromage achètent également du pain.

Chacune des formes d'analyse des données est pertinente. Le choix du type de connaissances recherchées est lié à l'utilisation que l'on souhaite en faire.

Nous nous sommes focalisés sur l'extraction de relations entre attributs dans le but d'essayer de trouver des lois permettant de mieux comprendre les données étudiées. Ces relations sont appelées règles d'association et peuvent s'exprimer sous la forme suivante :

$$attribut_1, attribut_4 \rightarrow attribut_{10}$$

Cette règle est composée d'une prémisse, $\{attribut_1, attribut_4\}$ et d'une conclusion $attribut_{10}$. La règle nous indique que lorsqu'une transaction contient les attributs $attribut_1$ et $attribut_4$, alors l'$attribut_{10}$ est souvent rencontré. Une nouvelle mesure, la confiance, nous permet d'apprécier la qualité de la règle trouvée. La confiance indique le pourcentage de transactions contenant la prémisse et la conclusion parmi celles contenant la prémisse. Une règle est toujours fournie avec les deux mesures de support et de confiance.

Le tableau 4.2 présente quelques-unes des 8 555 règles trouvées à partir de la base du magasin et vérifiant les contraintes de support ≥ 10% et de confiance ≥ 80%.

TAB.4.2.– Exemple de règles d'association.

Règle d'Association	**Support**	**Confiance**
Beurre → Pain	70%	100%
Pain → Beurre	70%	77.7%
Soda → Beurre	40%	100%
Poisson, Viande → Lait	20%	100%
Fromage, Pâtes → Soda	40%	80%

Les données transactionnelles peuvent être représentées sous la forme d'un tableau d'incidence (ou d'existence) à double entrée de dimension ($n \times p$) croisant un ensemble $O=\{o_i | 1 \leq i \leq n\}$ de n objets avec un ensemble $\mathcal{A} = \{a^j | 1 \leq j \leq p\}$ de p attributs booléens. Nous noterons $a_i^j = a^j(o_i)$, $1 \leq i \leq n$, $1 \leq j \leq p$.

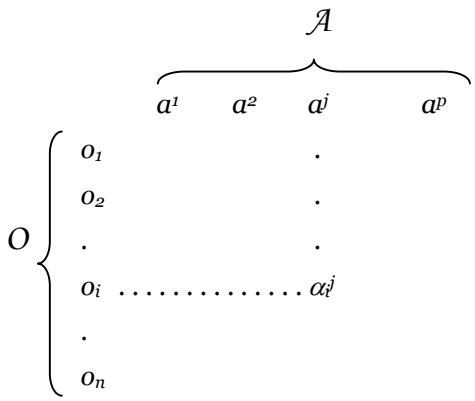


FIG.4.1.– Tableau d'incidence.

La figure 4.1 illustre les notations retenues pour les données manipulées. Les tableaux 3.3 et 3.4 présentent la correspondance entre la représentation transactionnelle d'une base $\mathcal{B}$ composée de 6 objets décrits par 5 attributs booléens et la représentation matricielle de cette même base. On peut supposer, sans restreindre la généralité, que la valeur *Vrai* d'un attribut est sémantiquement plus signifiante que la valeur *Faux* de cet attribut. Cela se traduit statistiquement par le fait que le nombre d'objets où l'attribut est à *Vrai* est inférieur au nombre d'objets où l'attribut est à *Faux*.

Pour éviter la surcharge d'indices les éléments de $\mathcal{A}$ sont désignés par des lettres capitales A, B, C qui sont couramment utilisées dans le domaine de la fouille de données *(Agrawal et al., 1993) (Guillaume, 2000) (Lehn, 2000) (Bastide et al., 2002)*. Les transactions sont classées par ordre alphabétique.

TAB.4.3.– Exemple de base de données transactionnelles $\mathcal{B}$

O	**Transaction**
O_1	A B C D E
O_2	A B C E
O_3	C
O_4	B C
O_5	A B C D
O_6	A B C

TAB.4.4.– Représentation matricielle de la base de données $\mathcal{B}$

	A	B	C	D	E
O_1	1	1	1	1	1
O_2	1	1	1	0	1
O_3	0	0	1	0	0
O_4	0	1	1	0	0
O_5	1	1	1	1	0
O_6	1	1	1	0	0

Notre principal objectif est la détection de relations entre les objets de O. Chaque objet étant décrit par un sous-ensemble des attributs booléens de $\mathcal{A}$, nous nous sommes intéressés aux relations pouvant exister entre les attributs de $\mathcal{A}$ et étant vérifiées par les objets de O.

Définition 4.1 (REGLE D'ASSOCIATION)

La relation $A \rightarrow B$ est appelée règle d'association entre A et B ($A \neq B$) avec $A \subseteq \mathcal{A}$, $B \subseteq \mathcal{A}$ et $A \cap B = \varnothing$. A est la prémisse de la règle et B la conclusion. Une règle est caractérisée par son support et sa confiance.

Notons n_A= *card* (O *(A))* (resp. *(n_B)*) le nombre d'objets de O où A (resp. *B)* est à *Vrai*. Notons n_{AB}= *card* (O *($A \cap B$))* le nombre d'objets de O où *A* et *B* sont tous les deux à *Vrai*. Cette valeur est appelée *indice brut d'association* entre *A* et *B*. Notons $\bar{A}$ la négation de *A*. Nous avons $O(A) \cup O(\bar{A}) = O$. $O(\bar{A})$ représente l'ensemble des objets de O où *A* est à *Faux*.

Définition 4.2 (SUPPORT)

Le support ou taux de couverture d'une règle d'association $A \rightarrow B$ est défini par :

Support ($A \rightarrow B$)= n_{AB} . Cette mesure peut être exprimée en fonction du nombre total d'objets n.

Définition 4.3 (CONFIANCE)

La confiance d'une règle d'association $A \rightarrow B$ est définie par :

Confiance ($A \rightarrow B$)= $n_{AB}/n_A \simeq P(B|A)$.

Cette mesure représente le pourcentage d'objets vérifiant la conclusion de la règle parmi celles qui vérifient la prémisse. P(B|A) représente la probabilité conditionnelle qu'une transaction contienne l'attribut B sachant qu'elle contient l'attribut A.

Deux types de règles d'association sont distingués :

• Les règles d'association exactes dont la confiance est égale à 100% qui sont vérifiées dans tous les objets du contexte.

• Les règles d'association approximatives dont la confiance est inférieure à 1 qui sont vérifiées dans une proportion d'objets du contexte égale à leur confiance.

Les deux mesures de support et de confiance, initialement utilisées par *(Agrawal et al., 1993)* dans un contexte d'extraction de règles d'association avaient déjà été introduites par *(Hajek et al., 1966) (Hajek, 2001)* dans un cadre de génération automatique d'hypothèses à partir de données. Les mesures de support et de confiance permettent retenir seulement les règles d'association vérifiant les conditions imposées par l'utilisateur. Ces conditions garantissent que chaque règle trouvée possède un support et une confiance supérieurs aux seuils *minsupport* et *minconfiance* fixés par l'utilisateur avant l'utilisation de l'algorithme d'extraction des règles.

4.2.2 Extraction d'Itemsets

Il existe plusieurs méthodes permettant de trouver toutes les règles d'association contenues dans une base de données transactionnelles. La plus simple d'entre elles consiste à énumérer tous les ensembles d'attributs appelés *itemsets* puis à partir de ces itemsets, de calculer toutes les règles d'association possibles. Le nombre total d'itemsets pour une base de données contenant n attributs booléens 2^n. Cette méthode très naïve est bien évidemment inapplicable aux données réelles car le nombre d'itemsets candidats devient très rapidement supérieur aux capacités de traitement d'un ordinateur.

Par exemple pour la base commerciale, le nombre total d'itemsets pouvant être obtenu est de 2^{10}=1 024. Dans un centre commercial, le nombre d'articles mis en rayon se rapproche plus de 1000 que de 10 et dans ce cas, le nombre d'itemsets pouvant être obtenu est proche de 10^{300}.

Les règles d'association recherchées sont obtenues à partir des itemsets fréquents trouvés. Pour un itemset de taille k appelé *k-itemset* le nombre possible de règles est $\sum_{i=2}^{k} C_k^i$. Ainsi pour un itemset de taille 6 cad un panier contenant seulement 6 articles, le nombre de règles possible est 57. Ce nombre de règles peut paraître faible mais le problème est que le nombre d'itemsets de taille 6 est de 10^{18}.

Les itemsets forment un treillis dont la représentation sous forme de diagramme de Hasse pour le contexte $\mathcal{B}$.

En résumé, cette première méthode est donc inexploitable pour des données transactionnelles réalistes. Un des défauts de cette méthode est lié au fait que de nombreux itemsets sont calculés pour finalement ne pas conduire à des règles intéressantes (support ou confiance inférieurs aux seuils choisis).

4.2.3 Extraction d'Itemsets Fréquents

Une méthode beaucoup moins naïve et nettement plus efficace consiste à calculer les itemsets ayant un support supérieur à un seuil fixé au départ par l'utilisateur. Ces itemsets sont appelés *itemsets fréquents*. Les temps de réponse de l'extraction des règles d'association dépendent principalement des temps d'extraction des itemsets fréquents car plusieurs balayages du contexte doivent être réalisés en comptant pour chaque itemset fréquent potentiel le nombre d'objets du contexte dans lesquels il est contenu. Le nombre d'itemsets à considérer et la taille des jeux de données (contexte d'extraction) étant importants (de plusieurs dizaines de milliers à plusieurs millions d'objets et de plusieurs centaines à plusieurs milliers d'items) des algorithmes permettant de minimiser le nombre d'itemsets candidats (itemsets potentiellement fréquents) considérés et le nombre de balayages du contexte ont été proposés.

Les algorithmes d'extraction des itemsets fréquents par niveaux considèrent un ensemble d'itemsets d'une taille donnée lors de chaque itération, c'est à dire un ensemble d'itemsets d'un « niveau » du treillis des itemsets. Ces algorithmes se basent sur les propriétés suivantes afin de limiter le nombre d'itemsets candidats considérés, en les générant à partir des itemsets fréquents de l'itération précédente : *tous les sur-ensembles d'un itemset infréquents sont infréquents et tous les sous-ensembles d'un itemset fréquent sont fréquents (Agrawal & Srikant, 1994) (Mannita et al., 1994)*. Nous pouvons citer les algorithmes APRIORI *(Agrawal & Srikant, 1994)* et OCD *(Mannita et al., 1994)* qui réalisent un nombre de balayages du contexte égal à la taille des *plus longs itemsets fréquents*. L'algorithme PARTITION *(Savasere et al., 1995)* autorise la parallélisation du processus d'extraction. L'algorithme DIC *(Brin et al., 1997a)* réduit le nombre de balayages du contexte en considérant les itemsets de plusieurs tailles différentes lors de chaque itération. Les algorithmes PARTITION et DIC entraînent un coût supplémentaire en temps par rapport aux algorithmes Apriori et OCD dû à l'augmentation du nombre d'itemsets candidats testés.

La classification fondée sur les treillis peut être considérée comme une technique symbolique de fouille de données qui peut être exploitée pour extraire, à partir d'une base de données, ou d'un ensemble de données régulières, un ensemble de concepts organisés en une hiérarchie (un ordre partiel), des itemsets fréquents (des ensembles de propriétés qui cooccurent ensemble avec une certaine fréquence). La classification basée sur des treillis repose sur l'analyse de tables booléennes qui relient un ensemble d'individus avec un ensemble de propriétés (ou de caractéristiques) où la valeur *Vrai* est utilisée pour indiquer que l'individu *i* possède la propriété *j*. Les relations entre les individus et les propriétés peuvent être lues comme suit : l'individu *i* possède ou ne possède pas la propriété *j*. Le treillis peut être construit en fonction de la correspondance de Galois, classifiant en un concept formel un ensemble d'individus (l'extension du concept) qui partagent un même ensemble de propriétés.

De manière parallèle, l'extension d'itemsets fréquents consiste en l'extraction de tables booléennes des ensembles de propriétés qui co-occurrent avec un support ou une fréquence supérieure à un seuil donné cad le nombre d'individus partageant les mêmes propriétés. A partir des itemsets fréquents, il est possible de générer des règles d'association de la forme $A \rightarrow B$ reliant le sous-ensemble de propriétés A avec le sous-ensemble de propriétés B et qui peut être interprétée comme suit : les individus incluant A incluent également B avec un certain support et une certaine confiance. Le nombre de règles qui peut être extrait est très grand et il y a donc

un besoin d'élaguer les ensembles de règles extraites pour mieux les interpréter (le plus souvent l'analyste est en charge d'interpréter les résultats du processus d'extraction de règles).

L'algorithme APRIORI *(Agrawal & Srikant, 1994)* est l'algorithme de référence en la matière. Il existe de nombreuses améliorations de cet algorithme mais leur principe général est le même et est fondé sur la génération des itemsets par niveaux :

- calcul des 1-itemset fréquents
- utilisation des (n-1)-itemsets fréquents pour calculer les n-itemsets fréquents candidats.

La réduction significative du nombre d'itemsets engendrés (par rapport à la méthode naïve) est due à la propriété mathématique d'anti-monotonie du support qui assure que si un itemset de taille *k* n'est pas fréquent alors tous les ensembles de taille *k+1* pouvant être obtenus à partir de cet itemset ne sont pas fréquents. Il n'est donc pas nécessaire d'engendrer les itemsets de taille *n* si ceux de taille *n-1* ne sont pas fréquents. Cette approche reste complète cad que tous les itemsets vérifiant la contrainte du support minimal sont trouvés.

Le pseudo-code d' APRIORI est donné dans l'algorithme 1. les notations utilisées sont définies dans le tableau 4.5.

TAB.4.5.– Notations utilisées dans l'algorithme APRIORI

Symbole	Signification
C_k	Ensemble des *k*- itemsets candidats
L_k	Ensemble des *k*-itemsets fréquents
Apriori-Gen	Fonction qui génère les itemsets candidats

Fonction Apriori-Gen

Entrée : C_k itemsets de taille *k*

Sortie : C_{k+1} itemsets de taille *k*+1

Début

```
INSERT INTO Ck+1
SELECT p[1], p[2], ..., p[k], q[k]
FROM Ck p, Ck q
WHERE p[1] = q[1] AND ... AND p[k-1] = q[k-1] AND p[k] < q[k]
```

Fin

La première phase de la procédure applique la phase de jointure de la procédure Apriori-Gen aux (*k*-1)-itemsets de L_{k-1} afin d'initialiser les (*k*)-itemsets candidats de C_k.

Algorithme APRIORI

Entrée : une base de données transactionnelle $\mathcal{B}$; seuil de fréquence *minsupport*.

Sortie : $\bigcup_k L_k$ l'ensemble de tous les items fréquents de $\mathcal{B}$.

Début

$L_1 \leftarrow$ {1-itemsets fréquents}

Pour ($k \leftarrow 2$; $L_{k-1} \neq \emptyset$; k++) **faire**

----------/Apriori-Gen calcule les itemsets candidats de taille k à partir de ceux de taille k-1/---------

$C_k \leftarrow$ Apriori-Gen (L_{k-1}) ;

Pour tout ($t \in \mathcal{B}$) **faire**

----------/Sélection des itemsets de C_k contenus dans la transaction t /----------

$C_t \leftarrow$ Subset (C_k, t) ;

Pour tout ($c \in C_t$) **faire** support(c) $\leftarrow$ support (c) +1 ;

FinPour

----------/ Sélection des itemsets vérifiant la contrainte de support /----------

$L_k \leftarrow \{c \in C_k \mid \text{support}(c) \geq minsupport\}$;

FinPour

Retourner $\bigcup_k L_k$;

Fin

TAB.4.6.– Exemple de base de données transactionnelle

Identifiant	Transaction
1	A B C D E
2	A B C E
3	C
4	B C
5	A B C D
6	A B C

Considérons la base de données $\mathcal{B}$ présentée dans le tableau 4.6. La recherche des itemsets fréquents sur ces données, avec l'algorithme APRIORI (seuil minimal de support fixé à 2/6) fournit 23 itemsets fréquents. Le nombre d'itemsets non calculés est égal à 7. A partir de ces itemsets fréquents, nous pouvons obtenir 32 règles d'association ayant une confiance supérieure ou égale à 80%.

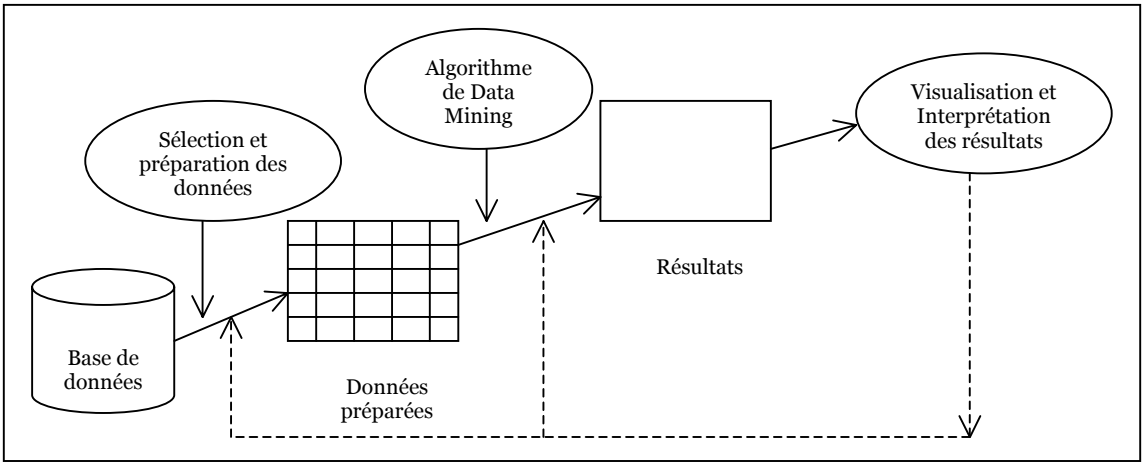
TAB.4.7.– Règles d'association obtenues à partir de la base $\mathcal{B}$

Règle	Support	Confiance
D → A	33.3 %	100 %
D → B	33.3 %	100 %
D → C	33.3 %	100 %
E → A	33.3 %	100 %
E →B	33.3 %	100 %
E → C	33.3 %	100 %
A → B	66.7 %	100 %
B → A	66.7 %	80 %
A → C	66.7 %	100 %
B → C	83.3 %	100 %
C → B	83.3 %	83.3 %
DA → B	33.3 %	100 %
DB → A	33.3 %	100 %
DA → C	33.3 %	100 %
DC → A	33.3 %	100 %
DB → C	33.3 %	100 %

Règle	Support	Confiance
DC → B	33.3 %	100 %
EA → B	33.3 %	100 %
EB → A	33.3 %	100 %
EA → C	33.3 %	100 %
EC → A	33.3 %	100 %
EB → C	33.3 %	100 %
EC → B	33.3 %	100 %
AB → C	66.7 %	100 %
AC → B	66.7 %	100 %
BC → A	66.7 %	80 %
DAB → C	33.3 %	100 %
DAC → B	33.3 %	100 %
DBC → A	33.3 %	100 %
EAB → C	33.3 %	100 %
EAC → B	33.3 %	100 %
EBC → A	33.3 %	100 %

4.2.4 Processus d'Extraction de Règles d'Association

L'extraction de règles d'association est un processus itératif et interactif constitué de plusieurs phases, allant de la sélection et la préparation des données jusqu'à l'interprétation des résultats, en passant par la phase de recherche des connaissances *(Fayyad et al., 1996)*.



TAB.4.3.– Processus d'extraction de connaissances et de règles d'association.

Le processus d'extraction s'effectue en plusieurs étapes :

- Pré-traitement : sélection et préparation des données, mise en forme pour l'extraction.
- Fouille de données : exécution de l'algorithme choisi.
- Visualisation : présentation des résultats à l'utilisateur pour leur interprétation.

4.2.4.1 Préparation des Données

La phase de préparation des données consiste à sélectionner les données (attributs et objets) de la base de données utiles à l'extraction des règles d'association et transformer ces données en un *contexte d'extraction*. Ce contexte ou jeu de données, est un triplet $\mathcal{B}=(O, I, \mathcal{R})$ dans lequel O est un ensemble d'objets, I est un ensemble d'attributs, également appelés *items*, et $\mathcal{R}$ est une relation binaire entre O et I. Cette phase est nécessaire afin qu'il soit possible d'appliquer les algorithmes d'extraction des règles d'association sur des données de natures différentes provenant de sources différentes, de concentrer la recherche sur les données utiles pour l'application et de minimiser les temps d'extraction.

4.2.4.2 Extraction des Ensembles Fréquents d'Attributs

La phase d'extraction des ensembles fréquents d'attributs consiste à extraire du contexte tous les ensembles d'attributs binaires $l \subseteq I$ appelés *itemsets*, qui sont fréquents dans le contexte $\mathcal{B}$. Un itemset l est fréquent si son support, qui correspond au nombre d'objets du contexte qui «contiennent» l, est supérieur ou égal au seuil minimal de support *minsupport* défini par l'utilisateur. L'ensemble des itemsets fréquents dans le contexte est noté $\mathcal{F}$. Le problème de l'extraction des itemsets fréquents est de complexité exponentielle dans la taille n de l'ensemble d'items puisque le nombre d'itemsets fréquents potentiels est 2^n. Ces itemsets forment un treillis. De plus, des balayages du contexte doivent être réalisés lors de cette phase, et il est donc nécessaire de développer des méthodes efficaces d'exploration de cet espace de recherche exponentiel.

4.2.4.3 Génération des Règles d'Association

Durant la phase de génération des règles d'association, les itemsets fréquents extraits durant la phase précédente sont utilisés afin de générer les règles, qui sont des implications entre deux itemsets fréquents I_1, $I_2 \subseteq \mathcal{F}$ tels que $I_1 \subseteq I_2$, de la forme $r : I_1 \rightarrow (I_2 \setminus I_1)$. Afin de limiter l'extraction aux règles d'association les plus informatives, seules celles qui possèdent une confiance supérieure ou égale au seuil minimal *minconfiance* défini par l'utilisateur sont générées. La confiance d'une règle $r : I_1 \rightarrow (I_2 \setminus I_1)$ est définie comme la proportion d'objets contenant la conséquence $(I_2 \setminus I_1)$ de r parmi ceux qui contiennent l'antécédent I_1 de r. Cette valeur est égale au rapport entre le support de l'itemset I_2 et le support de l'itemset I_1.

4.2.4.4 Interprétation des Résultats

La phase d'interprétation des résultats consiste en la visualisation par l'utilisateur des règles d'association extraites du contexte et leur interprétation afin d'en déduire des connaissances utiles pour l'amélioration de l'activité concernée. Dans notre cas, l'activité concernée est la recherche d'information. Le nombre important de règles d'association extraites en général impose le développement d'outils de classification des règles, de sélection par l'utilisateur de sous-ensembles de règles, et de leur visualisation sous une forme intelligible. De tels outils ont été proposés dans le système Rule Visualizer *(Klemettinen et al., 1994)*, qui utilise des *templates* de sélection des règles et permet de les visualiser sous forme textuelle ou bien sous forme de graphes orientés.

Les connaissances de l'utilisateur final, en l'occurrence l'expert, concernant le domaine d'application sont nécessaires lors des phases de pré-traitement, afin d'assister la sélection et la préparation des données, et de post-traitement, pour l'interprétation et l'évaluation des règles extraites. En fonction de l'évaluation des règles extraites, les paramètres utilisés lors des précédentes phases (critères de sélection et préparation des données et seuils minimaux de support et de confiance) peuvent être modifiés avant d'effectuer à nouveau l'extraction des règles d'association, ceci afin d'améliorer la qualité du résultat.

4.2.5 Autres Méthodes

4.2.5.1 Découverte de Pépites de Connaissances

L'objectif principal de *(Azé, 2003)* est de découvrir des *pépites de connaissance* dans les données. Une pépite de connaissance est caractérisée par rapport à un ensemble de règles et est définie simplement comme étant potentiellement utile et nouvelle pour l'expert. Il a choisi de minimiser l'interaction avec l'expert dans le processus d'extraction des pépites de connaissance. Ce choix est motivé par le fait que bien souvent l'expert peut difficilement fournir au système les informations nécessaires pour effectuer l'extraction des règles. Ces informations sont souvent réduites à la définition de bornes inférieures (ou supérieures) pour le support, la confiance et autres mesures de qualité.

(Azé, 2003) utilise d'autres propriétés que la contrainte de support pour limiter le nombre de règles obtenues qui augmente très vite dès que le support minimal diminue. Il utilise pour cela l'indice de *moindre contradiction* pour évaluer l'intérêt des règles et extraire des pépites de connaissance. L'algorithme proposé est indépendant de la mesure de qualité utilisée mais il faut qu'elle soit bornée. Il est difficile de rechercher de manière exhaustive l'intégralité des règles d'association les moins contradictoires. Cette contrainte l'a amené à réduire l'ensemble des règles recherchées en se limitant aux règles d'association ayant les propriétés suivantes :

– Les règles se distinguant le plus des autres règles.

– Les règles être telles que leurs prémisses ne contiennent pas plus de *Kmax* attributs et telles que la conclusion soit réduite à un seul attribut.

Ces règles appelées pépites de connaissance sont souvent inconnues de l'expert et sont soit indétectables par les systèmes classiques soit difficilement détectables dans l'ensemble des innombrables règles souvent trouvées par les systèmes classiques. Les expérimentations réalisées montrent que les pépites de connaissances obtenues sont effectivement peu nombreuses et qu'elles représentent relativement souvent des connaissances intéressantes. Il

prend cependant le risque de détecter des connaissances ayant un faible support donc non généralisables aux autres objets du contexte.

4.2.5.2 Extraction d'Itemsets Fréquents Maximaux

Les algorithmes d'extraction d'*itemsets fréquents maximaux* sont basés sur la propriété que les itemsets dont tous les sur-ensembles sont infréquents, forment une bordure en dessous de laquelle tous les itemsets sont fréquents. Leur extraction est réalisée par une exploration itérative du treillis des itemsets fréquents, en progressant d'un niveau allant du bas vers le haut et de un ou plusieurs niveaux du haut vers le bas lors de chaque itération. À partir des itemsets fréquents maximaux, tous les itemsets fréquents sont dérivés et leurs supports sont déterminés en réalisant un balayage du contexte. Quatre algorithmes basés sur cette approche ont été proposés, ce sont les algorithmes Pincer-Search *(Link & Kedem, 1998)*, MaxClique et MaxEclat *(Zaki et al., 1997)*, et Max-Miner *(Bayardo, 1998)*. Ces algorithmes permettent de réduire le nombre d'itérations, et donc de diminuer le nombre de balayages du contexte et d'opérations, réalisés.

4.2.5.3 Contraintes sur les Items

Les contraintes sur les items *(Bayardo et al., 1999) (Ng et al., 1998) (Srikant et al., 1997)* sont les expressions portant sur l'antécédent et la conséquence des règles, définies par l'utilisateur, qui spécifient la forme des règles d'association à extraire. Ces contraintes sont utilisées lors de la phase d'extraction des itemsets fréquents afin de limiter l'espace de recherche de cette phase aux itemsets permettant de générer les règles vérifiant les contraintes. Elles sont prises en compte lors de la génération des itemsets candidats afin de considérer seulement les candidats permettant de générer des règles satisfaisant les contraintes. En revanche cette approche ne permet pas d'éliminer les règles redondantes et ne fournit qu'un résultat partiel, tous les items du contexte n'étant pas considérés.

4.2.5.4 Extraction d'Itemsets Fermés Fréquents

Les *itemsets fermés fréquents (Pasquier et al., 1998) (Pasquier, 2000) (Pasquier et al., 2004)* sont définis en utilisant la fermeture de la connexion de Galois d'une relation binaire finie *(Davey & Priestley, 1994) (Ganter & Wille, 1999)*. Ces itemsets sont les itemsets fréquents qui sont fermés selon l'opérateur de fermeture γ de la connexion de Galois qui est la composition de l'application φ, qui associe à un ensemble $O \subseteq$ O les items communs à tous les objets $o \in O$, et de l'application ψ, qui associe à un itemset $l \subseteq$ I les objets en relation avec tous les items $i \in l$.

$$\varphi(O) : \mathrm{O} \rightarrow \mathrm{I} \qquad \psi(O) : \mathrm{I} \rightarrow \mathrm{O}$$

$$\varphi(\mathrm{O}) = \left\{ i \middle| \forall o, o \in O \rightarrow (o,i) \in R \right\} \qquad \psi(O) = \left\{ o \middle| \forall i, i \in l \rightarrow (o,i) \in R \right\}$$

L'opérateur $\gamma = \varphi \circ \psi$ associe à un itemset l l'ensemble maximal d'items communs à tous les objets contenant l, c'est-à-dire l'intersection de ces objets.

Par exemple, dans le contexte $\mathcal{B}$, l'itemset {A, B, C} est un itemset fermé. Il est l'ensemble maximal d'items communs aux objets {1, 2, 5, 6}. L'itemset {A, B} n'est pas un itemset fermé car il n'est pas un ensemble maximal d'items communs à certains objets : tous les objets contenant les items A et C (les objets 1, 2, 5 et 6) contiennent également l'item C. Dans le cas d'une base de données de ventes, cela signifie que les clients achètent *au plus* les articles A, B et C, et que tous les clients qui achètent les articles A et B achètent également l'article C. Les itemsets fermés fréquents, selon cet opérateur de fermeture, constituent un ensemble générateur non redondant minimal pour tous les itemsets fréquents et leurs supports. Tous les itemsets fréquents et leurs supports, et donc toutes les règles d'association ainsi que leurs supports et leurs confiances, peuvent donc être déduits efficacement, sans accéder au jeu de données, à partir des itemsets fermés fréquents et leurs supports. Cette propriété découle du fait que le support d'un itemset fréquent est égal au support de sa fermeture et que les itemsets fréquents maximaux sont des itemsets fermés fréquents maximaux *(Pasquier et al., 2004)*. Les itemsets fermés fréquents forment un treillis dont la taille est bornée par la taille du treillis des itemsets fréquents. Toutefois, en pratique, la taille de ce treillis est en moyenne bien inférieure à la taille du treillis des itemsets *(Godin & Missaoui, 1994)*.

Dans *(Pasquier, 2000) (Pasquier et al., 2004)*, les *générateurs* des itemsets fermés sont définis : les générateurs d'un itemset fermé f sont les itemsets minimaux g dont la fermeture est égale à f : $\gamma(g) = f$. Les algorithmes CLOSE et A-CLOSE sont des algorithmes d'extraction des itemsets fermés fréquents par niveaux : ils considèrent un ensemble de générateurs candidats d'une taille donnée, et déterminent leurs supports et leurs fermetures en réalisant un balayage du contexte lors de chaque itération. Les fermetures (fréquentes) des générateurs fréquents sont les itemsets fermés fréquents extraits lors de l'itération. Les générateurs candidats sont construits en combinant les générateurs fréquents extraits durant l'itération précédente.

4.2.6 Algorithmes Close et A-Close

Dans cette partie nous présentons les algorithmes CLOSE et A-CLOSE *(Pasquier, 2000) (Pasquier et al., 2004)* qui utilisent la sémantique basée sur la fermeture de la connexion de Galois. Les itemsets fermés fréquents et leurs générateurs ainsi que la base générique pour les règles d'association exactes et la réduction transitive de la base informative pour les règles d'association approximatives sont extraits. L'union de ces bases fournit un ensemble générateur non redondant pour toutes les règles d'association valides, leurs supports et leurs confiances. Elle est constituée des règles d'association non redondantes minimales (d'antécédent minimal et de conséquence maximale) et ne représente aucune perte d'information du point de vue de l'utilisateur : ce sont les règles d'association les plus utiles et les plus pertinentes. Toutes les informations véhiculées par l'ensemble des règles d'association valides sont également véhiculées par l'union de ces deux bases.

Nous nous attardons sur la description de l'algorithme A-CLOSE que nous avons implémenté et testé dans notre processus de découverte de connaissances à partir de la base de données CISMeF.

4.2.6.1 Algorithme Close

L'algorithme CLOSE est un algorithme itératif d'extraction des itemsets fermés fréquents qui parcourt l'ensemble des générateurs des itemsets fermés fréquents par niveaux. Durant chaque itération k de l'algorithme, un ensemble FFC_k de k-générateurs candidats est considéré. Chaque élément de cet ensemble est constitué de trois éléments : le k-générateur candidat, sa fermeture, qui est un itemset fermé candidat, et le support. À la fin de l'itération k, l'algorithme stocke un ensemble *FFk* contenant les k-générateurs fréquents, leurs fermetures, qui sont des itemsets fermés fréquents, et leurs supports.

L'algorithme commence par initialiser l'ensemble FFC_1 des 1-générateurs avec la liste des 1-itemsets du contexte et exécute ensuite un ensemble d'itérations. Durant chaque itération k :

• La fermeture de tous les k-générateurs ainsi que leur support sont calculés. La détermination des fermetures des générateurs est basée sur la propriété que la fermeture d'un itemset l est égale à l'intersection de tous les objets du contexte contenant l dont le décompte fournit le support du générateur qui est identique au support de sa fermeture. Un seul balayage du contexte est donc nécessaire pour déterminer les fermetures et les supports de tous les k-générateurs.

• Tous les k-générateurs fréquents, dont le support est supérieur ou égal au seuil minimal de support *minsupport*, ainsi que leur fermeture et leur support, sont insérés dans l'ensemble FF_k des itemsets fermés fréquents identifiés durant l'itération k.

• L'ensemble des (k+1)-générateurs candidats (utilisés durant l'itération suivante) est construit, en joignant les k-générateurs fréquents de l'ensemble FF_k.

Les itérations cessent lorsque aucun nouveau générateur candidat ne peut être créé et l'algorithme s'arrête alors.

L'algorithme A-CLOSE, développé pour améliorer l'algorithme ne calcule pas les fermetures des générateurs candidats durant les itérations, mais lors d'un ultime balayage réalisé après la fin de ces itérations.

4.2.6.2 Algorithme A-Close

L'algorithme A-CLOSE est un algorithme d'extraction des itemsets fermés fréquents utilisant les propriétés des supports des générateurs des itemsets fermés fréquents. Il génère itérativement les k-générateurs des k-groupes fréquents des ensembles FF_k. L'ensemble de 1-générateurs fréquents est initialisé avec les 1-itemsets fréquents du contexte et ensuite, durant une itération k *:*

• un ensemble de (k+1)-générateurs candidats est créé à partir des k-générateurs fréquents

• le support de tous les (k +1)-générateurs candidats est déterminé

• les (k+1)-générateurs candidats g dont le support est égal au support d'un k-générateur qui est un sous-ensemble de g sont supprimés.

Durant chaque itération, un balayage du contexte est réalisé afin de calculer le support des (k+1)-générateurs candidats. Lorsque tous les générateurs fréquents sont déterminés, la fermeture de chacun d'eux est calculée en réalisant un balayage du contexte.

TAB.4.8.– Notations utilisées dans l'algorithme

Symbole	Signification
FF_k	k-groupes fréquents des k-générateurs, Chaque élément de cet ensemble possède trois champs : *générateur*, *fermé* et *support*.
AC-Generator	Procédure qui détermine pour un ensemble de k-groupes fréquents retourne les (k+1)-générateurs fréquents.
AC-Closure	Procédure qui détermine pour un ensemble de groupes fréquents la fermeture de chaque générateur

Algorithme A-CLOSE.

Entrée : Contexte $\mathcal{B}$; seuil minimum de support *minsupport*;
Sortie : Ensemble FF_k des k-groupes fréquents;
Début
$FF_1.générateurs \leftarrow$ {1-itemsets} ;
Support-Count ($\mathcal{B}$, $FF_1.générateurs$)
Pour chaque générateur $g.générateur \in FF_1$ **faire**
Si ($g.support < minsupport$) **alors** $FF_1 \leftarrow FF_1 - \{g\}$
Finpour
Pour (k $\leftarrow$ 1 ; $FF_k.générateurs \neq \emptyset$; k++) **faire** $FF_{k+1} \leftarrow$AC-Generator (FF_k)
AC-Closure ($\bigcup$ FF_k);
Retourner ($\bigcup_k$ FF_k) ;
Fin

Durant la première itération, l'ensemble des 1-générateurs de FF_1 est initialisé avec la liste des 1-itemsets du contexte. La procédure Support-Count est ensuite appliquée afin de déterminer les supports de ces 1-itemsets générateurs en réalisant un balayage du contexte et les 1-générateurs infréquents sont supprimés de FF_1. Durant chaque itération k suivante, les (k+1)-générateurs de l'ensemble FF_{k+1} sont créés en utilisant les k-générateurs de l'ensemble FF_k et leurs supports respectifs. La procédure AC-Generator est pour cela appliquée à l'ensemble FF_k. Ces itérations cessent lorsque aucun nouvel itemset générateur ne peut être créé. Tous les générateurs des ensembles FF_k ont alors été créés et la procédure AC-Closure est appliquée à l'ensemble de ces générateurs afin de déterminer leurs fermetures qui constituent les itemsets fermés fréquents. L'algorithme retourne finalement la collection des ensembles FF_k qui contiennent chacun tous les k-générateurs et leurs fermetures.

Algorithme AC-GENERATOR.

Entrée : Ensemble FF_k de k-groupes des k-générateurs fréquents;

Sortie : Ensemble FFC_{k+1} avec les champs générateur des (k+1)-groupes candidats initialisés;

Début

----------*Phase 1*----------

INSERT INTO FF_{k+1}.*générateur*

SELECT p[1], p[2], ..., p[k], q[k]

FROM FF_k.*générateurs* p, FF_k.*générateurs* q

WHERE p[1] = q[1], ..., p[k-1] = q[k-1], p[k] < q[k]

----------*Phase 2*----------

Pour chaque générateur *g.générateur* ∈ FFC_{k+1} **faire**

Pour chaque *k-sous-ensemble s* ∈ *g.générateur* **faire**

Si *s* ∈ FF_k.*générateurs* ***alors*** $FFC_{k+1} \leftarrow FFC_{k+1} - \{g\}$

FinPour

FinPour

----------*Phase 3*----------

Support-Count ($\mathcal{B}$, FF_1.*générateurs*)

Pour chaque générateur *g.générateur* ∈ FFC_{k+1} **faire**

Si (*g.support* < *minsupport*) **alors** $FF_{k+1} \leftarrow FF_{k+1} - \{g\}$

Sinon faire

Pour chaque *k-sous-ensemble s.générateur* ∈ FFC_k de *g* **faire**

Si (*s.support* = *g.support*) **alors** $FFC_{k+1} \leftarrow FFC_{k+1} - \{g\}$

FinPour

FinSi

FinPour

Retourner FFC_{k+1}

Fin

La procédure AC-Generator reçoit un ensemble FF_k de k-groupes fréquents contenant les k-générateurs fréquents en paramètre. Elle retourne l'ensemble FF_{k+1} de (k+1)-groupes fréquents contenant les (k+1)-groupes fréquents contenant les (k+1)-générateurs fréquents.

La procédure AC-Generator est constituée de trois phases. Durant la première phase, les k-générateurs fréquents de FF_k sont combinés afin de créer les (k+1)-générateurs potentiels dans FF_{k+1}. La seconde phase supprime les (k+1)-générateurs potentiels infréquents ou qui ne sont pas minimaux. La troisième phase supprime les (k+1)-générateurs potentiels restants dont un sous-ensemble est un générateur du même itemset fermé fréquent.

La première phase de la procédure applique la phase de jointure de la procédure Apriori-Gen aux k générateurs FF_k afin d'initialiser les (k+1)-générateurs potentiels de FF_{k+1}. La seconde phase vérifie la présence dans FF_k de tous les k-générateurs qui sont des sous-ensembles de chaque (k+1)-générateurs potentiels dans FF_{k+1}. Durant la troisième phase, un balayage du contexte est réalisé afin de déterminer le support de chaque (k+1)-générateurs potentiels restants dans FF_{k+1} et tous les (k+1)-générateurs de FF_{k+1} sont examinés. Si un (k+1)-générateur g est infréquent, il est supprimé de FF_{k+1}. Sinon, s'il existe un générateur s de l'ensemble FF_k qui est un sous-ensemble de g et qui possède le même support que g, alors g est supprimé de FF_{k+1}.

Algorithme AC-CLOSURE.

Entrée : Ensembles FF_k des k-groupes des k-générateurs fréquents; contexte $\mathcal{B}$;
Sortie : Champs fermé des groupes de FF_k mis à jour ;
Début
  Pour chaque objet $o \in \mathcal{B}$ FFC_{k+1} **faire**
    $G_o \leftarrow$ Subset ($FF_k.générateurs$, $\varphi(\{o\})$) ;
    Pour chaque générateur $g.générateur \in G_o$ **faire**
      Si ($g.fermé = \emptyset$) **alors** $g.fermé \leftarrow \varphi(\{o\})$;
      Sinon $g.fermé \leftarrow g.fermé \cap \varphi(\{o\})$;
    FinPour
  FinPour
  Retourner $\bigcup \{ g \in FF_k \}$;
Fin

La procédure AC-Closure reçoit un ensemble $FF = \bigcup FF_k$ de groupes fréquents contenant tous les générateurs fréquents en argument. Elle détermine la fermeture de chaque générateur dans le champ fermé du groupe fréquent en réalisant un balayage du contexte. La méthode utilisée est identique à celle de la procédure Gen-Closure.

La procédure AC-Closure traite chaque objet du contexte successivement et crée pour chaque objet o un ensemble G_o contenant tous les générateurs de FF qui sont des sous-ensembles de l'itemset $\varphi(\{o\})$. Ensuite pour chaque générateur $g.générateur$ dans G_o la fermeture $g.fermé$ est mise à jour. Lorsque tous les objets du contexte ont été considérés, la procédure retourne l'ensemble FF_k dans lequel les champs fermé qui sont les fermetures de générateurs fréquents sont mis à jour.

Exemple avec le contexte $\mathcal{B}$ et *minsupport=2/6 :*

Identifiant	Transaction
1	A B C D E
2	A B C E
3	C
4	B C
5	A B C D
6	A B C

L'ensemble FF_1 est initialisé avec la liste des 1-itemsets du contexte et la procédure Support-Count détermine le support de chacun d'eux en réalisant un balayage du contexte. Les groupes candidats de FF_1 qui sont infréquents sont supprimés.

Balayage du Contexte

FF_1

Générateur	Fermé	Support
{ A }		4/6
{ B }		5/6
{ C }		6/6
{ D }		2/6
{ E }		2/6

La procédure AC-Generator aux générateurs de l'ensemble FF_1 génère des 2-générateurs potentiels qui sont insérés dans FF_2.

Balayage du Contexte

FF_2

Générateur	Fermé	Support
{ A, B }		4/6
{ A, C }		4/6
{ A, D }		2/6
{ A, E }		2/6
{ B, C }		5/6
{ B, D }		2/6
{ B, E }		2/6
{ C, D }		2/6
{ C, E }		2/6
{ D, E }		1/6

Tous les générateurs sont supprimés : l'itemset {D, E} est infréquent et tous les autres sont inutiles. En effet, le générateur {A, B} est supprimé car il a le même support que {A} et donc $\gamma(\{A,C\}) = \gamma(\{A\})$. De même, {C, D} est supprimé car il a le même support que {D}.

Suppression des générateurs inutiles et infréquents

FF_2

Générateur	Fermé	Support

Le processus s'arrête. Aucun 3-itemset ne peut être généré car $FF_2 = \varnothing$. Un dernier balayage du contexte est réalisé par la procédure AC-Close afin de calculer les fermetures des générateurs créés dans FF_1 qui sont les itemsets fermés fréquents du contexte.

Calcul des Fermés →

Générateur	Fermé	Support
{ A }	{A, B, C}	4/6
{ B }	{A, B, C}	5/6
{ C }	{C}	6/6
{ D }	{A, B, C, D}	2/6
{ E }	{A, B, C, E}	2/6

Le même principe d'extraction d'itemsets fréquents en utilisant l'opérateur de fermeture de la connexion de Galois a également été récemment proposé par *(Cherif-Latiri et al., 2003)*.

4.2.6.3 Problème de la Pertinence et de l'Utilité des Règles d'Association

Le problème de la pertinence et de l'utilité des règles d'association est lié au nombre de règles d'association extraites qui est en général très important et à la présence d'une forte proportion de règles redondantes, qui véhiculent la même information, parmi celles-ci. Si le problème de la visualisation d'un nombre relativement important de règles peut être simplifié par l'utilisation de systèmes de visualisation tels que le système Rule Visualizer proposé par *(Klemettinen et al., 1994)*, le problème de la suppression des règles d'association redondantes nécessite d'autres solutions. De plus, les règles d'association redondantes représentant pour certains type de données la majorité des règles extraites, leur suppression permet de réduire considérablement le nombre de règles à gérer lors de la visualisation.

Par exemple ces règles, extraites du tableau 4.7., possèdent un support et une confiance identiques :

1) E → A (sup : 33.3%; conf : 100%)
2) E → C (sup : 33.3%; conf : 100%)
3) EB → A (sup : 33.3%; conf : 100%)
4) EC → A (sup : 33.3%; conf : 100%)
5) EA → C (sup : 33.3%; conf : 100%)
6) EB → C (sup : 33.3%; conf : 100%)
7) EBC → A (sup : 33.3%; conf : 100%)
8) EAB → C (sup : 33.3%; conf : 100%)

Les règles 1, 3 et 4 sont redondantes par rapport à la règle 7. De même les règles 2, 5 et 6 sont redondantes avec la 8. En effet les règles redondantes n'apportent aucune information supplémentaire par rapport à la règle 7 (resp.8) qui est la plus générale. De ce fait, afin d'améliorer la pertinence et l'utilité des règles extraites, il est plus intéressant d'extraire cette règle et de la présenter à l'utilisateur.

La solution proposée par *(Bastide et al., 2002)* consiste à générer des bases, également appelées couvertures réduites, pour les règles d'association qui sont des ensembles de taille réduite ne contenant aucune règle redondante. Le but est de limiter l'extraction aux règles d'association les plus informatives, c'est-à-dire les plus générales et, éventuellement, dont les

mesures de précision sont les plus élevées parmi toutes les règles valides, du point de vue de l'utilisateur.

Utilisant la sémantique pour le problème de l'extraction de règles d'association basée sur la fermeture de la connexion de Galois, des bases pour les règles d'association sont caractérisées. Ces bases, qui sont la base générique pour les règles d'association exactes et la base informative pour les règles d'association approximatives, sont définies à partir des itemsets fermés fréquents et leurs générateurs. Ce sont des ensembles de taille réduite qui minimisent le nombre de règles d'association générées tout en maximisant la quantité et la qualité des informations véhiculées. Elles permettent :

• La génération des règles d'association non redondantes les plus informatives seulement, c'est à dire des règles les plus utiles et pertinentes : celles qui ont un antécédent (partie gauche) minimal et une conséquence (partie droite) maximale.

• La présentation à l'utilisateur d'un ensemble de règles couvrant tous les attributs de la base de données, c'est-à-dire contenant des règles dont l'union des conséquences est égale à l'union des conséquences de toutes les règles d'association valides dans le contexte. En effet, il est nécessaire de ne pas limiter la recherche à un seul sous-ensemble des attributs de la base de données car les règles « surprenantes » pour l'utilisateur constituent des informations utiles qu'il est nécessaire de considérer.

• L'extraction d'un ensemble de règles ne représentant aucune perte d'information, c'est à dire véhiculant toutes les informations convoyées par l'ensemble des règles d'association valides.

Il est possible de déduire de manière efficace, sans accès au jeu de données, toutes les règles d'association valides ainsi que leurs supports et leurs confiances à partir des règles d'association de ces bases.

4.2.7 Bases pour les Règles d'Association

La définition de bases pour les règles d'implication entre deux ensembles d'attributs binaires a été étudiée essentiellement dans les domaines de l'analyse de données et de l'analyse formelle de concepts. L'adaptation de la base de *(Duquenne & Guigues, 1986) (Ganter & Wille, 1999)* pour les implications totales, et la base de *(Luxenburger, 1991)* pour les implications partielles dans le cadre de l'extraction de règles d'association exactes et approximatives est présentée dans *(Pasquier et al., 1999) (Pasquier et al., 2004)*. Les bases obtenues sont des réductions de l'ensemble des règles d'association qui minimisent autant que possible le nombre de règles générées, sans tenir compte du support des règles. Cela signifie que les antécédents et les conséquences de toutes les règles d'association peuvent être déduits de l'union de ces bases, mais pas leurs supports. Le support et la confiance indiquent l'utilité et la pertinence de la règle et doivent être pris en considération lors de la définition des règles d'association redondantes. Une règle d'association $r : l_1 \rightarrow l_2$ de support s et de confiance c est notée $r\ (s,c)$: $l_1 \rightarrow l_2$.

Définition 4.4 (REGLES D'ASSOCIATION REDONDANTES)

Soit un ensemble $\mathcal{E}$ de règles d'association. Une règle d'association $r \in \mathcal{E}$ est redondante si la règle r peut être déduite ainsi que son support s et sa confiance c de l'ensemble $\mathcal{E} \setminus r$.

Il faut que seules les *règles d'association non redondantes minimales*, qui sont les règles les plus utiles et les plus pertinentes, soient extraites et présentées à l'utilisateur. Une règle d'association est redondante si elle véhicule la même information ou une information moins générale que l'information véhiculée par une autre règle de même utilité et de même pertinence. Une règle d'association r est non redondante minimale s'il n'existe pas une autre règle d'association r' possédant le même support et la même confiance, dont l'antécédent est un sous-ensemble de l'antécédent de r et la conséquence est un sur-ensemble de la conséquence de r.

Définition 4.5 (REGLES D'ASSOCIATION NON REDONDANTES MINIMALES)

Soit l'ensemble AR des règles d'association extraites du contexte. Une règle d'association $r : l_1 \rightarrow l_2 \in AR$ est non redondante minimale s'il n'existe pas de règle d'association $r' : l'_1 \rightarrow l'_2 \in AR$ telle que support(r)=support(r'), confiance(r)= confiance(r'), $l'_1 \subseteq l_1$ et $l_2 \subseteq l'_2$.

À partir de cette définition, les règles d'association exactes non redondantes minimales sont définies ainsi que les règles d'association approximatives non redondantes minimales. Ces règles constituent la *base générique* pour les règles d'association exactes et la *base informative* pour les règles d'association approximatives respectivement, et sont générées à partir des itemsets fermés fréquents et leurs générateurs.

4.2.7.1 Base Générique pour les Règles d'Association Exactes

Les règles d'association exactes, de la forme $r : l_1 \rightarrow (l_2 \backslash l_1)$, sont des règles entre deux itemsets fréquents l_1 et l_2 dont les fermetures sont identiques ($\gamma(l_1) = \gamma(l_2)$). En effet, de $\gamma(l_1) = \gamma(l_2)$ on peut déduire que $l_1 \subset l_2$ et $support(l_1) = support(l_2)$, et donc $confiance(r) = 1$. Puisque l'itemset maximal parmi ces itemsets (qui possèdent le même support) est l'itemset $\gamma(l_2)$, tous les sur-ensembles stricts de l_1 qui sont des sous-ensembles de $\gamma(l_2)$ possèdent le même support, et les règles entre deux de ces itemsets sont des règles exactes.

Soit l'ensemble $G_{\gamma(l2)}$ des générateurs de l'itemset fermé fréquent $\gamma(l_2)$. Par définition, les itemsets minimaux qui sont des sur-ensembles stricts de l_1 et des sous-ensembles de $\gamma(l_2)$ sont les générateurs $g \in G_{\gamma(l2)}$. Nous en concluons que les règles de la forme $g \rightarrow (\gamma(l_2) \setminus g)$ entre les générateurs $g \in G_{\gamma(l2)}$ et l'itemset fermé fréquent $\gamma(l_2)$ sont les règles d'antécédents minimaux et de conséquences maximales parmi les règles entre les sur-ensembles stricts de l_1 et les sous-ensembles de $\gamma(l_2)$. La généralisation de cette propriété à l'ensemble des itemsets fermés fréquents définit la base générique constituée de toutes les règles d'association exactes non redondantes d'antécédents minimaux et de conséquences maximales.

Définition 4.6 (BASE GENERIQUE POUR LES REGLES D'ASSOCIATION EXACTES)

Soit l'ensemble FF des itemsets fermés fréquents extraits du contexte et pour chaque itemset fermé fréquent f l'ensemble G_f des générateurs de f. La base générique pour les règles d'association exactes est :

$$BG = \{r : g \rightarrow (f \setminus g) \mid f \in FF \wedge \ g \in G_f \wedge g \neq f\}.$$

La condition $g \neq f$ est nécessaire car les règles entre un générateur g d'un itemset fermé fréquent f tel que $g = f$ sont de la forme $g \rightarrow \emptyset$ et n'appartiennent pas à l'ensemble des règles d'association valides (règles non informatives).

Le pseudo-code de l'algorithme Gen-BG *(Pasquier, 2000)* de construction de la base générique pour les règles d'association exactes à partir de l'ensemble des itemsets fermés fréquents et de leurs générateurs est présenté dans l'Algorithme suivant. Les notations utilisées sont présentées dans le tableau 4.9.

TAB.4.9.– Notations utilisées dans Gen-BG

Symbole	Signification
FF_k	k-groupes fréquents des k-générateurs, Chaque élément de cet ensemble possède trois champs : *générateur*, *fermé* et *support*.
BG	Ensemble des règles d'association exactes de la base générique

Algorithme GEN-BG.

Entrée Ensembles *FFk* des *k*-groupes fréquents des *k*-générateurs;

Sortie : Ensemble *BG* des règles d'association exactes de la base de générique;

Début

$BG \leftarrow \emptyset$;

Pour chaque ensemble FF_k **faire**

Pour chaque *k*-générateur $g \in FF_k$

Si $g \neq \gamma(g)$ **faire** $BG \leftarrow BG \cup \{(r : g \rightarrow (\gamma(g) \setminus g), \gamma(g).support)\}$;

FinPour

FinPour

Retourner *BG*;

Fin

L'algorithme commence par initialiser l'ensemble *BG* avec l'ensemble vide. Chaque ensemble FF_k de *k*-groupes fréquents est ensuite examiné successivement. Pour chaque *k*-générateur $g \in FF_k$ de l'itemset fermé fréquent $\gamma(g)$ pour lequel g est différent de sa fermeture $\gamma(g)$, la règle r : $g \rightarrow (\gamma(g) \setminus g)$, dont le support est égal au support de g et $\gamma(g)$, est insérée dans *BG*. L'algorithme retourne finalement l'ensemble *BG* qui contient toutes les règles exactes informatives entre un générateur et sa fermeture.

La base générique pour les règles d'association exactes extraite du contexte $\mathcal{B}$ pour *minsupport* = 2/6 est présentée dans le tableau 4.10. Cette base ne représente aucune perte d'information : toutes les règles d'association exactes valides dans le contexte peuvent être déduites ainsi que leurs supports (et leurs confiances qui sont égales à 1) à partir des règles de la base générique.

TAB.4.10.– Base générique extraite du contexte $\mathcal{B}$

Générateur	Fermé	Règle Exacte	Support
{ A }	{A, B, C}	A→ BC	4/6
{ B }	{A, B, C}	B→ AC	5/6
{ C }	{C}		
{ D }	{A, B, C, D}	D→ ABC	2/6
{ E }	{A, B, C, E}	E→ ABC	2/6

Le générateur étant {A} et son fermé {A, B, C}, la règle exacte extraite est A→ BC.

4.2.7.2 Base Informative pour les Règles d'Association Approximatives

Les règles d'association approximatives, de la forme $r : l_1 \rightarrow (l_2 \setminus l_1)$, sont des règles entre deux itemsets fréquents l_1 et l_2 tel que la fermeture de l_1 est un sous-ensemble de la fermeture de l_2 ($\gamma(l_1) \subset \gamma(l_2)$). Les règles d'association approximatives non redondantes d'antécédent l_1 minimal et de conséquence ($l_2 \setminus l_1$) maximale sont déduites de cette caractérisation. Soit l'itemset fermé fréquent f_1 qui est la fermeture de l_1 et le générateur unique g_1 de f_1 tels que $g_1 \subseteq l_1 \subseteq f_1$. Soit l'itemset fermé fréquent f_2 qui est la fermeture de l_2 et le générateur unique g_2 de f_2 tels que $g_2 \subseteq l_2 \subseteq f_2$. La règle $g_1 \rightarrow (f_2 \setminus g_1)$ entre le générateur g_1 et l'itemset fermé fréquent f_2 est la règle non redondante d'antécédent minimal et de conséquence maximale parmi les règles entre un itemset de l'intervalle $[g_1, f_1]$ et un itemset de l'intervalle $[g_2, f_2]$. En effet, le générateur g_1 est l'itemset minimal dont la fermeture est f_1, ce qui signifie que l'antécédent g_1 de cette règle est minimal, et la conséquence $(f_2 \setminus g_1)$ est maximale puisque f_2 est l'itemset maximal de l'intervalle $[g_2, f_2]$. En généralisant cette propriété à l'ensemble des règles entre deux itemsets l_1 et l_2 on peut définir la base informative constituée de toutes les règles d'association approximatives non redondantes d'antécédents minimaux et de conséquences maximales.

Définition 4.7 (BASE INFORMATIVE POUR LES REGLES APPROXIMATIVES)

Soit l'ensemble FF des itemsets fermés fréquents et l'ensemble G de leurs générateurs extraits du contexte. La base informative pour les règles d'association approximatives est : $BI = \{r : g \rightarrow (f \setminus g) \mid f \in FF \wedge g \in G \wedge \gamma(g) \subset f\}$.

La base informative ne représente aucune perte d'information car toutes les règles d'association approximatives valides dans le contexte peuvent être déduites ainsi que leurs supports et leurs confiances à partir des règles qui la constituent.

De la définition de la base informative il est possible de déduire la réduction transitive de celle-ci qui constitue également une base pour les règles d'association approximatives. Les règles transitives de la base informative sont de la forme $r : g \rightarrow (f \setminus g)$ pour un itemset fermé fréquent f et un générateur fréquent g tels que $\gamma(g) \subset f$ et $\gamma(g)$ n'est pas un prédécesseur immédiat de f : $\exists f' \in FF$ tel que $\gamma(g) \subset f' \subset f$, noté $\gamma(g) \lessdot f$.

Définition 4.8 (REDUCTION TRANSITIVE DE LA BASE INFORMATIVE)

Soit l'ensemble FF des itemsets fermés fréquents et l'ensemble G de leurs générateurs extraits du contexte. La réduction transitive de la base informative pour les règles d'association approximatives est :

$RI = \{r : g \rightarrow (f \setminus g) \mid f \in FF \wedge g \in G \wedge \gamma(g) \subset f\}$.

Il est possible de déduire toutes les règles d'association de la base informative ainsi que leurs supports et leurs confiances, et donc toutes les règles approximatives valides, à partir des règles de la réduction transitive. Cette réduction permet de diminuer le nombre de règles extraites en conservant les règles dont la confiance est la plus élevée puisque les règles transitives possèdent par construction des confiances inférieures aux règles non transitives.

Le pseudo-code de l'algorithme Gen-RI *(Pasquier, 2000)* de construction de la réduction transitive de la base informative pour les règles d'association approximatives à partir de l'ensemble des itemsets fermés fréquents et de leurs générateurs est présenté ci-après. Les notations utilisées sont présentées dans le tableau 4.11.

TAB.4.11.– Notations utilisées dans l'algorithme Gen-RI

Symbole	Signification
FF_k	*k*-groupes fréquents des *k*-générateurs, Chaque élément de cet ensemble possède trois champs : *générateur*, *fermé* et *support*.
$Succ_g$	Ensemble des itemsets fermés fréquents qui sont des successeurs immédiats de la fermeture du générateur *g* considéré.
RI	Ensemble des règles d'association approximatives de la réduction transitive de la base informative.

Algorithme Gen-RI.

Entrée : Ensembles *FFk* des *k*-groupes fréquents des *k*-générateurs;
Seuil minimal de confiance *minconfiance*;

Sortie : Ensemble *RI* des règles d'association approximatives de la réduction transitive de la base informative;

Début
RI ← ∅;
Pour (*k* ← 1; *k* ≤ μ-1; *k*++) **faire**
Pour chaque *k*-générateur $g \in FFk$ **faire**
$Succ_g \leftarrow \emptyset$;
Pour ($j = |\gamma(g)|; j \leq \mu; j$++) **faire**
$S_j \leftarrow \{f \in FF \mid f \supset \gamma(g) \wedge |f| = j\}$;
FinPour
Pour ($j = |\gamma(g)|; j \leq \mu; j$++) **faire**
Pour chaque itemset fermé fréquent $f \in Sj$ **faire**
Si ($\nexists\ s \in Succ_g \mid s \subset f$) **alors**
$Succ_g \leftarrow Succ_g \cup f$;
r.confiance ← *f.support* / *g.support*;
Si *r.confiance* ≥ *minconfiance* **alors** $RI \leftarrow RI \cup \{(r: g \rightarrow (f \setminus g), r.confiance, g.support\}$;
FinSi
FinPour
FinPour
FinPour
FinPour
Retourner *RI*;
Fin

L'algorithme commence par initialiser l'ensemble *RI* avec l'ensemble vide. Chaque ensemble FF_k de *k*-groupes fréquents est ensuite examiné successivement dans l'ordre des valeurs de *k* croissantes. Pour chaque *k*-générateur $g \in FF_k$ de l'itemset fermé fréquent $\gamma(g)$, l'ensemble $Succ_g$ des successeurs de la fermeture de $\gamma(g)$ est initialisé avec l'ensemble vide et les ensembles S_j des *j*-itemsets fermés fréquents qui sont des sur-ensembles de $\gamma(g)$ pour $|\gamma(g)| < j \leq \mu$ sont construits. Les ensembles S_j sont ensuite considérés dans l'ordre croissant des valeurs de *j*. Pour chaque itemset $f \in S_j$ dont aucun successeur immédiat de $\gamma(g)$ dans $Succ_g$ n'est un sous-ensemble, *f* est inséré dans $Succ_g$ et la confiance de la règle $r : g \rightarrow (f \setminus g)$ est calculée. Si la confiance de *r* est supérieure ou égale au seuil minimal de confiance *minconfiance*, la règle *r* est insérée dans *RI*. Lorsque tous les générateurs de taille inférieure à μ ont été considérés, l'algorithme retourne l'ensemble *RI*.

TAB.4.12.– La réduction transitive de la base informative extraite du contexte $\mathcal{B}$ pour *minsupport* = 2/6 et *minconfiance* = 2/6 :

Générateur	Fermé	Sur-ensemble Fermé	Règle Approximative	Support	Confiance
{ A }	{A, B, C}	{A, B, C, D}	A → BCD	4/6	2/4
{ A }	{A, B, C}	{A, B, C, E}	A → BCE	4/6	2/4
{ B }	{A, B, C}	{A, B, C, D}	B → ACD	5/6	2/5
{ B }	{A, B, C}	{A, B, C, E}	B → ACE	5/6	2/5
{ C }	{C}	{A, B, C }	C → AB	6/6	4/6
{ C }	{C}	{A, B, C, D}			
{ C }	{C}	{A, B, C, E}			
{ D }	{A, B, C, D}				
{ E }	{A ,B, C, E}				

Les bases générées présentent un fort intérêt pour la visualisation des règles extraites car le nombre réduit de règles dans ces bases ainsi que la distinction des règles exactes et des règles approximatives facilitent la présentation des règles à l'utilisateur. De plus, l'absence de règles redondantes dans les bases et la génération des règles non redondantes minimales seulement présentent un intérêt important du point de vue de l'utilisateur.

4.3 Autres Mesures Statistiques

L'utilisation de mesures de qualité permet de proposer à l'utilisateur des règles d'association les mieux adaptées à sa recherche. La condition nécessaire pour pouvoir ordonner et/ou filtrer les règles ne correspondant pas aux attentes de l'utilisateur, est de comprendre et de savoir ce que l'utilisateur recherche dans les données.

Définition 4.9 (EXEMPLE/CONTRE-EXEMPLE)

Etant donnée une règle d'association $A \rightarrow B$, les exemples de cette règle sont les transactions vérifiant A et B et les contre-exemples sont les transactions vérifiant A et $\overline{B}$.

Etant donnée une règle d'association $A \rightarrow B$ les observations dont nous disposons sont présentées ci-dessous.

TAB.4.13.– Données associées à une règle $A \rightarrow B$

	A	$\overline{A}$	Σ
B	n_{AB}	$n_{\overline{A}B}$	n_B
$\overline{B}$	$n_{A\overline{B}}$	$n_{\overline{A}\overline{B}}$	$n_{\overline{B}}$
Σ	n_A	$n_{\overline{A}}$	n

	A	$\overline{A}$	Σ
B	$P(AB)$	$P(\overline{A}B)$	$P(B)$
$\overline{B}$	$P(A\overline{B})$	$P(\overline{A}\overline{B})$	$P(\overline{B})$
Σ	$P(A)$	$P(\overline{A})$	1

A partir de ces observations on peut créer de nombreuses mesures de qualité permettant d'évaluer un ensemble de règles d'association. L'utilisation de ces mesures permet de proposer à l'utilisateur des règles d'association les mieux adaptées à sa recherche. La condition nécessaire pour pouvoir ordonner et/ou filtrer les règles ne correspondant pas aux attentes de l'utilisateur, est de comprendre et de savoir ce que l'utilisateur recherche dans les données.

Le support peut également être exprimé en fonction du nombre total d'objets du contexte. Cette mesure correspond alors à la probabilité d'apparition des objets correspondant à l'itemset A et l'itemset B.

$$P(A,B) = \frac{Support(A \rightarrow B)}{n} \in [0,1] \text{ avec } n \text{ le nombre d'objet du contexte } \mathcal{B}.$$

La confiance mesure le degré de validité d'une règle, c'est à dire lorsqu'il existe des contre-exemples d'objets qui vérifient A mais pas nécessairement tous les items de B.

$$Confiance(A \rightarrow B) = \frac{Support(A \rightarrow B)}{Support(B)} = P(B|A) \in [0,1]$$

Lorsque A et B sont indépendants, $P(A, B) = P(A) \times P(B)$. Une règle d'association générée entre deux itemsets indépendants indique seulement que A et B existent ensemble dans beaucoup d'objets. L'indice de dépendance est classiquement utilisé en probabilités et permet de calculer l'apport de A dans la règle.

L'indice de dépendance renforce une règle en mesurant le fait que A et B soient indépendants ou pas :

$$Dépendance(A \rightarrow B) = \left\| P(B \mid A) - P(B) \right\|$$

Les items très fréquents dans le contexte n'apportent pas d'information supplémentaire alors que les items rares, qui sont peut-être porteurs de connaissances, apparaissent dans des règles à faible support auxquelles on ne s'intéresse pas en premier lieu.

L'intérêt (ou *lift*) (*IBM, 1996)* mesure la dépendance entre A et B. Cet indice privilégie les items rares aux dépens d'items trop répandus dans le contexte. Il mesure la déviation du support de la règle par rapport au cas d'indépendance.

$$Intérêt(A \rightarrow B) = \frac{P(A,B)}{P(A) \times P(B)}$$

Cet indice dénote une indépendance de A et B s'il est = 1. Plus A et B sont incompatibles, plus P(A, B) tend vers 0, et donc l'intérêt est proche de 0. Plus A et B sont dépendants, plus l'intérêt se rapproche de 1. L'intérêt a un comportement symétrique pour A et B mais il ne reflète pas l'association A→B. En effet, $Intérêt(A \rightarrow B) = Intérêt(B \rightarrow A)$

La mesure de *conviction* a été introduite par *(Brin et al., 1997b)*. Elle permet de mesurer pour chaque règle la déviation de la dépendance entre la probabilité d'occurrence de l'antécédent et la probabilité de non-occurrence de la conséquence dans les objets. Elle mesure donc la dépendance mais pour les contre-exemples $A\overline{B}$.

$$Conviction(A \rightarrow B) = \frac{P(A) \times P(B)}{P\left(A, \overline{B}\right)}$$

Cet indice n'est calculable que pour les règles approximatives. En effet, si la règle est exacte, $P\left(A, \overline{B}\right) = 0$.

Toujours pour les règles approximatives, l'indice d'*étonnement* (également appelé *moindre* contradiction) a été introduit par *(Azé, 2003)*. Cet indice est défini pour mesurer l'*affirmation* : c'est à dire la différence entre la *confirmation P(AB)* et *l'infirmation P(A* $\overline{B}$ *)* d'une règle. Moins B est répandu, plus il est *étonnant* de trouver une bonne affirmation de la règle.

$$Etonnement(A \rightarrow B) = \frac{P(A,B) - P(A,\overline{B})}{P(B)}$$

La mesure du χ^2 spécifie le degré de dépendance entre les items d'un itemset en comparant la distribution réelle de leur occurrence avec la distribution attendue de leur occurrence sous l'assomption d'une complète indépendance et d'une distribution normale.

La confiance centrée *(Lallich & Teytaud, 2004)* permet de prendre en considération la taille de la conclusion de la règle $A \rightarrow B$. L'introduction de P(B) dans cette mesure permet de la rendre sensible à la taille des données.

$$Confiance\ Centrée\ (A \rightarrow B) = P(B|A) - P(B)$$

Lorsque la taille des données augmente et si les marges n_A, n_B et n_{AB} restent constantes, la confiance centrée de la règle $A \rightarrow B$ augmente et tend vers la confiance de la règle.

Les mesures de déviation sont des mesures de distance entre règles d'association définies en fonction de leurs supports et leurs confiances. Ces mesures peuvent être utilisées afin d'identifier les règles d'association fortement semblables, caractérisées par une faible distance entre elles, et ensuite regrouper ou supprimer certaines de ces règles *(Bayardo et al., 1999) (Toivonnet et al., 1995)*. Cette méthode entraîne une perte d'information et nécessite des traitements supplémentaires de comparaison des règles extraites deux à deux. Une autre utilisation consiste à identifier les règles d'association inattendues pour l'utilisateur qui apportent donc une connaissance importante car nouvelle *(Heckermann, 1996) (Piatetsky-Shapiro & Matheus, 1994) (Silberschatz & Tuzhilin, 1996)*. Les connaissances de l'utilisateur sont représentées en utilisant des modèles probabilistes auxquels sont confrontées les règles d'association extraites. La déviation d'une règle correspond à la différence entre la valeur attendue pour la règle dans le modèle probabiliste et la valeur réelle pour la règle dans le contexte. Cette méthode nécessite la définition par l'utilisateur de ces connaissances, ce qui dans de nombreux cas se révèle très complexe. Les temps des calculs des mesures de déviation sont très importants car ils requièrent de très nombreuses opérations.

Les templates *(Klemettinen et al., 1994)* sont des expressions booléennes permettant de sélectionner un sous-ensemble de l'ensemble des règles d'association valides. Ce sous-ensemble est construit en conservant les règles d'association valides qui vérifient les critères spécifiés par les templates parmi cet ensemble. Un template spécifie des contraintes d'occurrence ou de non occurrence des items dans l'antécédent et la conséquence des règles. Cette méthode de

traitement a posteriori des règles extraites ne permet pas de supprimer les règles redondantes, mais peut faciliter la visualisation de l'ensemble des règles extraites en visualisant les règles par groupes, chaque groupe correspondant à un ensemble de templates.

Nous présentons d'autres indices, dont une synthèse se trouve dans *(Lavrac et al., 1999)*, qui permettent différents classements des règles d'association.

TAB.4.14.– Autres mesures de qualité pour une règle $A \rightarrow B$

Mesure	Expression
Nouveauté	$P(A,B) - P(A) \times P(B)$
Satisfaction	$\frac{P(\overline{B}) - P(\overline{B}\|A)}{P(\overline{B})}$
Spécificité	$P(\overline{A}\|\overline{B})$
Fiabilité Négative	$P(\overline{B}\|\overline{A})$

L'utilisation de toutes ces mesures entraîne cependant des problèmes de performances car le calcul de leurs valeurs nécessite des temps d'exécution importants.

4.4 Fouille de Textes et Fouille du Web

4.4.1 Principe de la Fouille de Textes

L'extraction de connaissances à partir d'un corpus de textes peut prendre plusieurs formes : extraction de la terminologie asssociée au domaine, construction automatique d'une ontologie reliant les concepts découverts dans le corpus, découverte de formes syntaxiques spécifiques au domaine, extraction de règles d'association entre les concepts du domaine. La fouille de texte aide à acquérir la connaissance latente (*cachée*) à partir du contenu des documents telles que des relations d'association entre les termes d'un corpus. Cette fouille agit sur des textes individuels, des parties de textes ou des corpus.

Il n'existe pas à notre connaissance des travaux pour l'extraction de règles d'association à partir de textes en français. Dans la communauté francophone, *(Cherfi et al., 2003) (Azé & Roche, 2003)* travaillent sur des corpus de résumés de textes en anglais. Les règles d'association décrivent d'une façon symbolique les différentes corrélations entre mots dans les documents en se basant sur la notion d'ensemble fréquent qui sous-entend tout ensemble de mots apparaissant dans une base de documents avec une fréquence supérieure à un seuil fixé a priori. C'est donc le même principe que la fouille de données mais appliquée aux textes ou à des corpus de textes dans le but de trouver dans un grand ensemble de textes de nouvelles connaissances. La différence réside dans le fait que l'information traitée, les textes, sont écrits en langage naturel ce qui rend le processus de découverte plus complexe. En effet, le problème avec la langue naturelle est qu'elle n'est pas conçue pour être traitée par les ordinateurs, à la différence des données stockées dans des bases de données et l'information disponible à l'état brut est faiblement exploitable. Elle n'est pas explicite mais implicite, enterrée dans le texte.

Définition 4.10. (DECOUVERTE DE CONNAISSANCES A PARTIR DE TEXTES)

La Découverte de Connaissances à partir de Textes est le processus de découverte de modèles intéressants et utiles dans des corpus textuels son objectif est d'obtenir des connaissances précédemment inconnues et enfouies dans les textes.

4.4.2 Signification des Règles d'Association

Les mesures destinées à calculer les dépendances entre les termes (Information mutuelle, Coefficient de Dice, Coefficient de cohérence, etc.) prennent en compte les liaisons entre deux termes sans tenir compte du reste des termes dans le corpus. C'est généralement la fréquence des termes qui est utilisée pour calculer la dépendance. Ces mesures ne tiennent pas compte de la proportion que peut avoir cette dépendance par rapport à toutes les dépendances pouvant exister dans le corpus. En effet, l'objectif de la plupart de ces mesures est plutôt d'extraire des collocations ou des catégorisations des termes dans des ensembles homogènes alors que l'objectif des règles d'association est de trouver un lien implicite entre des termes sans se soucier de classer ces derniers ni chercher des collocations. La sémantique d'une règle d'association est alors différente de la sémantique d'une simple dépendance dans le sens où :

– elle exprime une probabilité de l'existence d'un terme par rapport à un autre.

– elle tient compte de l'ensemble des implications possibles dans le corpus. Un classement de toutes les implications possibles dans le corpus prend en compte la dispersion de l'antécédent et du conséquent dans le corpus mais aussi de la dispersion des autres termes.

Définition 4.11. (REGLE D'ASSOCIATION ENTRE TERMES)

Une règle d'association R entre termes ti est du type :

$$R : t_1 \wedge t_2 \wedge \ldots \wedge t_j \rightarrow t_{j+1} \wedge \ldots \wedge t_n$$

La partie gauche et la partie droite de la règle R sont constituées de conjonctions de termes. L'explication intuitive de la règle est que si des documents possèdent les termes $\{t_1,\ldots, t_j\}$ alors ils possèdent également les termes $\{t_{j+1},\ldots, t_n\}$.

Les mesures de support et de confiance se traduisent ainsi :

Définition 4.12. (SUPPORT D'UNE REGLE D'ASSOCIATION ENTRE TERMES)

Le support de la règle d'association R est l'ensemble des documents qui participent à sa génération. Cette mesure correspond au nombre de documents qui sont décrits par les termes $\{t_1, \ldots, t_n\}$

Définition 4.13. (CONFIANCE D'UNE REGLE D'ASSOCIATION ENTRE TERMES)

La confiance mesure le degré de validité de la règle c'est à dire lorsqu'il existe des contre-exemples de documents qui contiennent les termes $\{t_1,\ldots, t_j\}$ mais pas nécessairement les termes $\{t_{j+1},\ldots, t_n\}$.

4.4.3 Processus d'Extraction

Des systèmes appliquent l'analyse en texte intégral à des collections de documents. Deux types d'analyse sont possibles : l'analyse à but descriptif (fonctionnant en mode *non-supervisé* : l'outil analyse les documents sans référence à une classification prédéfinie) et l'analyse à but décisionnel (fonctionnant en mode *supervisé* : l'outil affecte automatiquement les documents selon une classification prédéfinie). Dans les deux cas, le couplage de techniques linguistiques (une analyse linguistique rudimentaire du texte qui consiste à distinguer les catégories grammaticales des termes et enlever les mots vides) et statistiques (une analyse statistique des données qui va permettre de corréler les données entre elles pour en saisir les invariants et les règles qui les régissent) est utilisé.

La fouille de textes commence par la sélection des textes et la représentation de leur contenu. La représentation doit refléter la sémantique. Cette représentation peut reposer sur un réseau terminologique et sur la liste de termes extraits à partir de textes.

Une architecture générale d'un système de fouille de données textuelles peut être la suivante :

– Sélection/Recherche : à partir d'une base de données textuelles, des informations sont sélectionnées suivant certains critères en fonction des besoins des utilisateurs.

– Prétraitement : un prétraitement suivant les objectifs du système est appliqué aux données sélectionnées (élimination de mots vides, sélections de mots-clés, etc.).

– Découverte de connaissances : il s'agit d'appliquer une des techniques de fouille : en l'occurrence l'extraction de règles associations…etc.

– Interprétation/Evaluation : il s'agit d'interpréter les connaissances découvertes et de représenter les résultats à l'utilisateur à travers dans une interface bien adaptée.

Pour éviter la dispersion de l'information dans le processus de fouille de textes les ambiguïtés du vocabulaire spécialisé doivent être prises en compte, par exemple en utilisant une terminologie. Lorsque celle-ci n'existe pas, les premières étapes de la fouille sont dédiées à l'acquisition de connaissances linguistiques : lexique, terminologie…etc.

On peut alors classer les techniques de fouille de données textuelles en deux catégories :

– Les techniques utilisant les mots-clés : les processus de découverte de connaissances sont appliqués à des mots-clés associés à des documents. Des thèmes sont établis a priori et chacun de ces thèmes est représenté par un ensemble de mots-clés. Selon les mots-clés qu'il contient, un document appartient à un thème donné.

Les techniques utilisant tout le texte : un document est représenté par tous les mots qu'il contient. L'application des processus de la fouille de données textuelles permet de découvrir des patrons de bas niveau c'est à dire au niveau des termes tels que les mots composés.

En fonction du type des transactions étudiées, on peut supposer les éléments suivants :

– Cas de la phrase : Dans le cas où une transaction est une phrase, les règles d'association permettent de découvrir des relations du niveau local (niveau de la phrase et de la construction de la phrase). Cela revient à redécouvrir la structure syntagmatique de la phrase c'est-à-dire des relations entre des constituants des syntagmes mais avec beaucoup moins de qualité que l'application de patrons syntaxiques.

– Cas du paragraphe : Un paragraphe peut être constitué d'une ou plusieurs phrases. Dans ce cas, nous retrouvons les règles d'association découvertes dans le cas de la phrase ainsi que des relations d'association interphrases. La plupart des relations d'association découvertes restent du niveau local mais on ne trouve pas les relations entre des termes appartenant à des paragraphes différents.

– Cas du document : D'autres cas sont aussi envisageables telle que une fenêtre de n termes ou une fenêtre de m phrases où m est déterminé par rapport à la taille du document et/ou la moyenne des tailles des documents dans le corpus. Dans le cas où la fenêtre est le document, le nombre des règles d'association découvertes est plus important que dans le cas de la fenêtre ou le paragraphe.

L'ensemble des règles d'association découvertes avec la phrase comme fenêtre est inclus dans l'ensemble des règles d'association découvertes avec le document comme fenêtre. De ce fait, toutes les règles d'association dans le cas de la phrase, seront également découvertes dans le cas du document. Souvent, dans le cas de la phrase, la plupart des associations découvertes sont des groupes nominaux ou des multi-termes (associations du niveau local). Ces associations du niveau local ne sont pas d'un apport du point de vue connaissances, puisque l'utilisation de la syntaxe pour extraire les groupes nominaux donnerait une connaissance plus intéressante.

Les conditions nécessaires pour l'extraction de règles d'association à partir de textes sont que :

– L'ensemble des textes doit refléter un contenu cohérent et homogène dans un domaine.

– Chaque texte doit être caractérisé par une forte densité de termes. Plus il y a de termes dans un texte plus le réseau sémantique reflétant le contenu du texte sera complet.

L'extraction de connaissances à partir de textes est très sensible à la phase d'indexation. En effet, si un terme est absent de l'indexation, aucune règle d'association contenant ce terme oublié ne sera extraite.

En terme d'utilisation des règles d'association ou des connaissances découvertes, de même que pour la fouille de données, l'application des techniques de fouille aux textes peut par exemple aider à explorer le contenu des corpus dans une tâche de recherche d'information.

Dans ce qui suit, nous présentons quelques systèmes de fouille de données textuelles.

4.4.4 Systèmes de Fouille de Textes

Le système d'extraction des textes décrit dans *(Lent et al., 1997)* traite le problème de l'identification de phrases (séquence de termes) comme un modèle séquentiel dans le but de découvrir des tendances dans des bases de données textuelles. Les auteurs définissent *une tendance* comme étant une sous-séquence d'une phrase. La méthodologie de cette approche consiste à identifier des phrases fréquentes à l'aide d'une mesure de support. La fenêtre de traitement est fixée à un paragraphe mais elle peut être plus petite pour préciser la distance entre les termes. La structure des documents est exploitée pour identifier les différentes sections.

La notion d'*épisode* et *règle d'épisode* est décrite par *(Ahonen et al., 1997)*. Elles sont inspirées de la notion des règles d'association mais qui sont appliquées aux données séquentielles (où un certain ordre des données est pris en compte). Dans le cas du texte, les

auteurs précisent que les épisodes sont des vecteurs composés de caractéristiques (sous la forme d'un ensemble de caractéristiques ordonnées) et un index (qui contient la position d'un mot dans la séquence). Une caractéristique peut être un mot, une expression, un signe de ponctuation ou une étiquette (par exemple un tag SGML). Un épisode est une séquence de texte qui apparaît dans une fenêtre donnée avec un ensemble de caractéristiques et l'apparition de ces caractéristiques dans la séquence selon un certain ordre. La taille d'une fenêtre est mesurée en nombre de mots non vides. Les auteurs soulignent l'importance de la phase de prétraitement des informations textuelles avant l'application des techniques de fouille. Ce prétraitement consiste à ne sélectionner que certaines catégories grammaticales (substantifs) ou à éliminer d'autres (préposition, articles, etc.) ou bien encore à lemmatiser les mots et il leur permet d'alléger le processus de traitement. Cette approche a été expérimentée sur 14 documents et non dans des corpus volumineux. Dans une deuxième expérience, les auteurs se sont orientés vers la découverte des dépendances structurelles en ne traitant que les Tag SGML des documents.

Fact est un système qui est basé sur les cooccurrences des mots dans les documents *(Feldman & Hirsh, 1996)*. La fouille de données textuelles est réalisée sur les mot-clés associés aux documents par les auteurs. Dans la collection utilisée, les mots-clés représentent 5 classes : les pays, les sujets des dépêches, les marchés, les personnes et les organisations. Le but du système est de trouver des règles d'association entre les termes, par exemple entre un pays et les organisations auxquelles il appartient. Les règles d'association extraites indiquent que la présence du terme (ou d'un ensemble de termes) X dans un document implique la présence du terme (ou d'un ensemble de termes) Y, avec deux probabilités calculées sur l'ensemble des documents du corpus.

(Roche et al., 2004) proposent une méthodologie de fouille de données textuelles selon les étapes suivantes :

1. Collecte d'un corpus homogène sur un thème donné : cette étape constitue la brique de base de l'ensemble du processus de fouille de textes. Le succès du processus est fonction de la qualité et de l'homogénéité du corpus collecté. La constitution du corpus est donc confiée explicitement à un expert.
2. Réaliser une détection des traces de concepts spécifique au corpus : La détection des concepts permet de réécrire le corpus de manière plus compacte. En effet chaque instance de concept détectée dans les textes est remplacée par le concept lui-même.
    a. Recherche de termes dans les textes : La recherche des termes consiste à isoler des collocations pour le domaine étudié. Cette première étape est réalisée de manière automatique.
    b. Utilisation des termes pour détecter les traces de concepts : La liste des termes obtenue permet à l'expert d'associer les termes à des concepts, c'est à dire des regroupements de termes ayant la même sémantique.
3. Extraire et valider les règles d'association à partir de corpus.
    a. Extraction des connaissances
    b. Validation

Un certain parallèle peut être fait avec les travaux de *(Cherfi et al., 2003)* qui présentent une méthodologie de fouille de textes scientifiques en anglais. La première étape, le prétraitement,

est chargée d'extraire dans les textes les parties textuelles intéressantes, en l'occurrence le titre et le résumé, et de les annoter pour que des outils des TAL puissent être mis en oeuvre dans la seconde étape. Chaque résumé est modélisé par l'ensemble des termes qu'il possède grâce à une indexation contrôlée à partir d'une liste de termes attestée. Cette seconde étape doit représenter le texte dans un système formel sur lequel les outils de fouille de données pourront être appliqués. Les règles d'association sont ensuite extraites et classées en fonction de plusieurs mesures de qualité et présentées à l'expert.

4.4.5 Fouille du Web

Une nouvelle tendance dans le domaine de recherche sur le Web est connue sous le nom de *découverte de connaissances du Web (Web mining)*. Le Web mining est l'application des techniques de fouille de données sur les ressources du Web.

Définition 4.14 (DECOUVERTE DE CONNAISSANCES A PARTIR DU WEB) *(Cooley et al., 1997)*

> *La Découverte de Connaissances à partir du Web est le processus de découverte et d'analyse d'informations intéressantes sur le Web incluant la recherche automatique de ressources d'information en ligne (la découverte de connaissances véhiculées par le contenu) et la découverte de modèles d'accès des utilisateur à partir des serveurs Web (la découverte de connaissances sur l'usage).*

La découverte de connaissances véhiculées par le contenu est le processus de découverte de connaissances dans le contenu des documents et des textes ou la découverte de leurs descriptions. La découverte de connaissances sur l'usage est le processus d'extraction de modèle ou de patrons intéressants dans les *Web access logs (Masseglia et al., 1999) (Masseglia, 2002)*. Les *Web access logs* sont les fichiers contenant l'activité des utilisateurs d'un siteWeb. Dans le cadre du CISMeF, ils nous ont permis d'analyser par exemple la structure des requêtes des utilisateurs, ou encore le nombre d'utilisateurs connectés par jour …etc.

La découverte de connaissances sur la structure du Web est un processus d'inférence de connaissances à partir de l'organisation du Web et des liens entre les référents et les référencés sur le Web. Les travaux sur la découverte de connaissance dans le contenu des pages Web sont généralement basés sur les agents logiciels. Plusieurs de ces travaux ont employé des techniques de clustering afin de filtrer, rechercher et classer des documents disponibles sur le Web par catégories. *(Theilmann & Rothermel, 1998)* proposent une approche appelée *domain expert*. Un domaine est un domaine de connaissances contenant des informations sémantiquement reliées. Un agent de filtrage mobile qui visite des sites spécifiques et examine la pertinence des documents qu'il contient par rapport au domaine. L'agent mobile utilise un ensemble de connaissances établies a priori et extraites des URL collectés par des moteurs de recherche existants. Les auteurs ont utilisé uniquement des connaissances qui décrivent les documents : mots-clés, métamots-clés, URL, date de modification, auteurs. Cette connaissance est représentée par des *facettes*.

Dans le cadre du CISMeF nous avons développé un *veilleur automatique* qui analyse périodiquement (quotidiennement, mensuellement ou par semaine) le contenu des sites Web producteurs de documents à indexer. Il se fonde sur l'analyse des liens à une profondeur

paramétrable et ce afin d'y sélectionner d'éventuels nouveaux documents intéressants d' indexer.

4.5 Evaluation des Règles d'Association

Que ce soit pour la fouille de données ou la fouille de textes, *(Sahar, 1999)* présente une méthode de filtrage de règles d'association fondée sur une interaction avec l'utilisateur. Cette méthode utilise un algorithme d'extraction de règles d'association classique tel que APRIORI pour obtenir un ensemble $\mathcal{E}$ des règles d'association vérifiant les contraintes de support et de confiance. Cet ensemble est souvent très volumineux et ne peut donc pas être proposé intégralement à l'expert. *(Sahar, 1999)* propose de présenter les règles une par une à l'expert selon un ordre particulier. Le travail de l'expert consiste alors à analyser la règle et à la valider selon des critères. Les informations obtenues sur la règle analysée permettent d'élaguer efficacement l'ensemble des règles restantes.

Lorsqu'une règle est proposée à l'expert, deux informations lui sont demandées :

– La règle est-elle vraie (V) ou fausse (F) ?

– La règle est-elle intéressante (I) ou inintéressante (NI) ?

Etant données ces deux informations, une règle appartient à l'une des quatre catégories suivantes :

– VI : règle considérée comme vraie et intéressante par l'expert. Ces règles ne permettent pas d'élaguer l'ensemble des règles

– FI : règle considérée comme fausse et intéressante par l'expert.

– VNI : règle considérée comme vraie et non intéressante par l'expert. Ce type de règles permet d'élaguer efficacement l'ensemble $\mathcal{E}$.

– FNI : règle considérée comme fausse et non intéressante par l'expert. Les règles appartenant à cette catégorie sont proches de celles appartenant à la catégorie FI. La différence majeure entre ces deux catégories est liée à la définition de l'intérêt de l'expert. L'élagage réalisé grâce à ce type de règles est identique à celui obtenu pour la catégorie VNI.

L'une des règles est associée à l'une de ces quatre catégories, le système proposé réalise deux opérations :

– Construction d'une base de connaissances : toutes les règles considérées comme vraies par l'expert sont ajoutées dans la base de connaissance.

– Filtrage des règles : les règles appartenant aux catégories VNI, FI et FNI participent à l'élagage de l'ensemble des règles.

Etant donné une règle $\mathcal{R} = a \rightarrow b$ associée aux catégories VNI ou FNI. L'élagage consiste à éliminer de $\mathcal{E}$ toutes les règles appartenant à la *famille*($\mathcal{R}|\mathcal{E}$) avec :

$$famille\ (\mathcal{R}|\mathcal{E})=\{ A \rightarrow B \mid A \rightarrow B \in \mathcal{E}, a \in A,\ b \in B\}$$

L'élagage réalisé pour les règles étiquetées FI est légèrement différent du précédent. Etant donné une règle $\mathcal{R} = a \rightarrow b$ étiquetée FI, seules les règles $\mathcal{R}' = a \rightarrow B$ avec $b \in B$ peuvent être supprimées car ces règles seront toutes fausses puisqu'elles possèdent exactement la même prémisse que la règle étiquetée FI.

En revanche, les règles appartenant à $\{ A \rightarrow B \mid a \in A, b \in B, A - \{a\} \neq \emptyset, B - \{b\} \neq \emptyset \}$ sont potentiellement intéressantes.

L'intérêt de cette méthode réside dans l'interaction mise en place avec l'expert pour élaguer un ensemble de règles d'association. Le temps de l'expert étant précieux, il convient d'optimiser l'interaction mise en place. *(Sahar, 1999)* propose également de sélectionner les règles pour lesquelles l'élagage sera maximal si l'expert les classe dans les catégories adaptées.

Ainsi pour chaque règle $\mathcal{R} = a \rightarrow b$ le système calcule $s_{\mathcal{R}} = card(famille\ (\mathcal{R} | \mathcal{E}))$ et les règles sont proposées à l'utilisateur par valeurs décroissantes de $s_{\mathcal{R}}$. A chaque itération de l'algorithme, ces valeurs sont remises à jour en fonction de l'élagage effectué. Lorsque les valeurs de $s_{\mathcal{R}}$ deviennent inférieures à un seuil prédéfini pendant deux itérations consécutives, l'expert est informé qu'il peut arrêter l'évaluation des règles s'il le désire car le gain potentiel (en terme de règles élaguées) devient minime. Il suffit en moyenne de cinq interventions de l'expert pour diviser par deux la taille de $\mathcal{E}$. Cette approche permet de rapidement diminuer le volume de règles à analyser et sollicitant peu l'expert.

4.6 Expériences d'Extraction de Règles d'Association

Nos expérimentations portent sur la base de données CISMeF, et en particulier sur les documents indexés du catalogue. Nous étudions pour cela différents contextes d'extraction de règles d'association en adaptant et en appliquant l'algorithme A-Close d'extraction d'itemsets fermés fréquents et les algorithmes de déduction de règles d'association exactes (ou totales) et des règles d'association approximatives. Pour cela nous distinguons deux cas de fouille :

– La technique fondée sur une indexation conceptuelle des documents.

– La technique utilisant le texte intégral, un document étant représenté par tous les termes qu'il contient.

Les approches s'appuient sur une description booléenne (présence vs absence) des items dans les objets, en l'occurrence la présence de concepts ou de termes.

4.6.1 Fouille de Données

En fonction de l'indexation utilisée dans le CISMeF, nous avons considéré différents contextes $\mathcal{B} = (O, I, \mathcal{R})$. Nous considérons que l'ensemble O des objets du contexte d'extraction sont les notices (ou fiches) décrivant les documents. Chaque notice a un identifiant unique. A chaque notice est associé un ensemble de descripteurs. La relation $\mathcal{R}$ entre les objets et les items est la relation d'indexation. Différents cas d'ensembles I d'items sont considérés. Il y a en moyenne 6,5 descripteurs par document avec au minimum 1 descripteur et au maximum 300 descripteurs.

Cette contrainte sur le nombre de descripteurs (items) a dû être résolue pendant la phase d'implémentation de l'algorithme A-Close. En effet, dans les structures de données proposées dans *(Bastide, 2000),* cette contrainte sur la taille des transactions n'est pas prise en compte. Les évaluations des algorithmes portent plutôt sur le temps d'extraction des règles d'association pour de grandes bases de données test, utilisées notamment dans les conférences Fast Itemset Mining Implementations. Ces bases de données sont des données de ventes et de recensements. Les données des bases de test contiennent en moyenne 20 items par objet et 10 000 objets. En revanche le nombre minimum d'items différents au total pour les différentes bases est de 127 items et le nombre maximum est de 2 178. Dans le cas du CISMeF, si l'on considère le nombre d'items différents possibles dans le cas où le descripteur est un mot clé, ce facteur est pratiquement de 4.

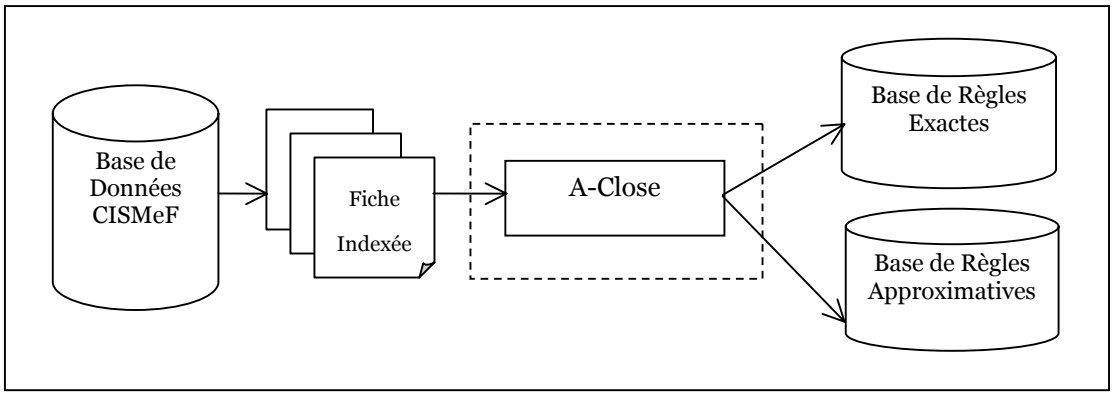


FIG.4.5.– Processus d'extraction de règles d'association du corpus indexé

Notre implémentation de l'algorithme A-Close reçoit en entrée un fichier généré (en PL/SQL) à partir de la base de données. Le fichier en entrée contient les différentes transactions. Les valeurs du support minimum et de la confiance minimum sont paramétrables en entrée de l'algorithme. Chaque transaction a un identifiant (le numéro unique de la notice) qui est suivi des différents items qui la composent, séparés par des tabulations. Les items sont classés par ordre alphabétique. En sortie de l'algorithme, les règles d'association peuvent au choix, soit être visualisées par l'utilisateur dans un fichier en sortie, soit être automatiquement ajoutées à la base de données CISMeF dans une table relationnelle créée à cet effet, afin d'être exploitées dans le processus de recherche d'information. Dans le cas de la visualisation, les règles d'association en sortie sont ordonnées en fonction de la taille de l'antécédent du support le plus élevé au moins élevé.

Un tuple de la table TB_KNOWQE_REGLES_ASSOC contenant les règles d'association extraites d'un contexte correspond a une règle d'association. Il possède les attributs suivants :

– L'attribut CONTEXTE : indique le nom du contexte qui a été utilisé pour extraire les règles d'association.

– L'attribut ANTECEDANTS : contient la liste des libellés des items antécédents de la règle séparés par une tabulation (la virgule ou le point virgule pouvant exister dans les libellés des items).

– L'attribut CONSEQUENTS : contient la liste des libellés des items conséquents de la règle

– L'attribut SUPPORT : indique la valeur du support de la règle (nombre entier) en l'occurrence le nombre de documents.

– L'attribut CONFIANCE : indique la valeur de la confiance de la règle (décimal).

Les autres indices de qualité présentés n'ont pas été considérés dans notre processus de fouille.

Une seconde condition, autre que les contraintes de support et de confiance, a été ajoutée dans l'implémentation de l'algorithme pour éviter des temps de génération trop longs : la taille maximum des générateurs d'itemsets fermés candidats a été fixée à 300 items. Cette valeur correspond au nombre maximum de descripteurs pour les notices existant dans la base de données CISMeF. En effet, la taille maximum d'items étant de 300, les itemsets fréquents maximaux potentiels ont une taille de 300.

4.6.1.1 Mots Clés

Dans ce premier cas, l'ensemble I correspond à l'ensemble des mots clés utilisés pour l'indexation des documents. Deux expérimentations ont été réalisées et les résultats obtenus sont résumés dans le tableau. Les notices sont sélectionnées de manière aléatoire. Nous avons fixé le support minimum à 20 documents et la confiance à 70% pour les règles approximatives.

TAB.4.15.– Nombre de règles extraites pour *minsup*=20 et *minconf*=70% et *I*={MC}

Nombre de Documents	Nombre de Règles Exactes	Nombre de Règles Approximatives	Total
9 811	2 906	12 149	15 055
11 373	2 438	9 381	11 819

Les règles sont trop nombreuses pour être analysées manuellement par l'expert (notre bibliothécaire médical en chef). En effet, comme l'ont indiqué *(Gras et al., 2001)* le nombre de règles calculé peut être très élevé et les tâches de dépouillement, d'interprétation et de synthèse des résultats peuvent devenir complexes, voire inextricables, pour l'utilisateur.

4.6.1.2 Mots Clés ou Qualificatifs

Dans ce cas, l'ensemble I correspond à l'ensemble des mots clés et des qualificatifs utilisés pour l'indexation des documents.

TAB.4.16.– Nombre de règles extraites pour *minsup*=20 et *minconf*=70% et *I*={MC}∪{Qu}

Nombre de Documents	Nombre de Règles Exactes	Nombre de Règles Approximatives	Total
11 373	5 241	11 738	16 979

4.6.1.3 Associations Mot Clé/Qualificatif

Dans ce troisième cas, l'ensemble I correspond à l'ensemble des associations (mots clés/qualificatifs) utilisés pour l'indexation des documents. Ceci nous permet d'obtenir des règles d'association entre couples de (Mot Clé/Qualificatif) plus fines que de simples associations entre mots clés ou entre mots clés et qualificatifs. Les mots clés qui ne sont associés à aucun qualificatif sont également considérés.

TAB.4.17.– Nombre de règles extraites pour *minsup*=20 et *minconf*=70% et *I*={(MC/Qu)}

Nombre de Documents	Nombre de Règles Exactes	Nombre de Règles Approximatives	Nombre de Règles
11 373	648	1917	2 565

On peut remarquer que pour ce contexte, avec les mêmes seuils de support et de confiance, le nombre de règles réduit considérablement. On passe de 11 819 règles à 2 565. En revanche cette réduction est profitable à la qualité des règles. En effet, on passe de simples règles d'association entre mots clés à des règles d'association plus précises et plus fines, les mots clés étant qualifiés.

4.6.2 Fouille de Données avec Catégorisation

Les règles extraites des expérimentations sont certes relatives au domaine de la santé, mais afin d'obtenir des règles plus précises, nous avons réalisé des expérimentations sur les notices catégorisées en fonction de spécialités, le domaine de la santé étant très large. Dans cette seconde série d'expériences nous avons catégorisé les différentes fiches en fonction de spécialités afin de comparer l'ensemble des règles d'association générées précédemment avec celles obtenues après catégorisation. Le but étant d'étudier l'influence de la catégorisation sur la génération des règles d'association.

Nous avons pour cela appliqué l'algorithme de catégorisation au corpus initial de 11 373 notices. Nous n'avons conservé que les spécialités considérées comme « majeures » en fonction de la pondération adoptée. La seconde condition pour la sélection des spécialités, est que le nombre de notices par spécialité soit supérieur à 500, cela sachant qu'une même notice peut être catégorisée par plusieurs spécialités, dont plusieurs en « majeur ». Les mesures de support et de confiance sont les mêmes que précédemment.

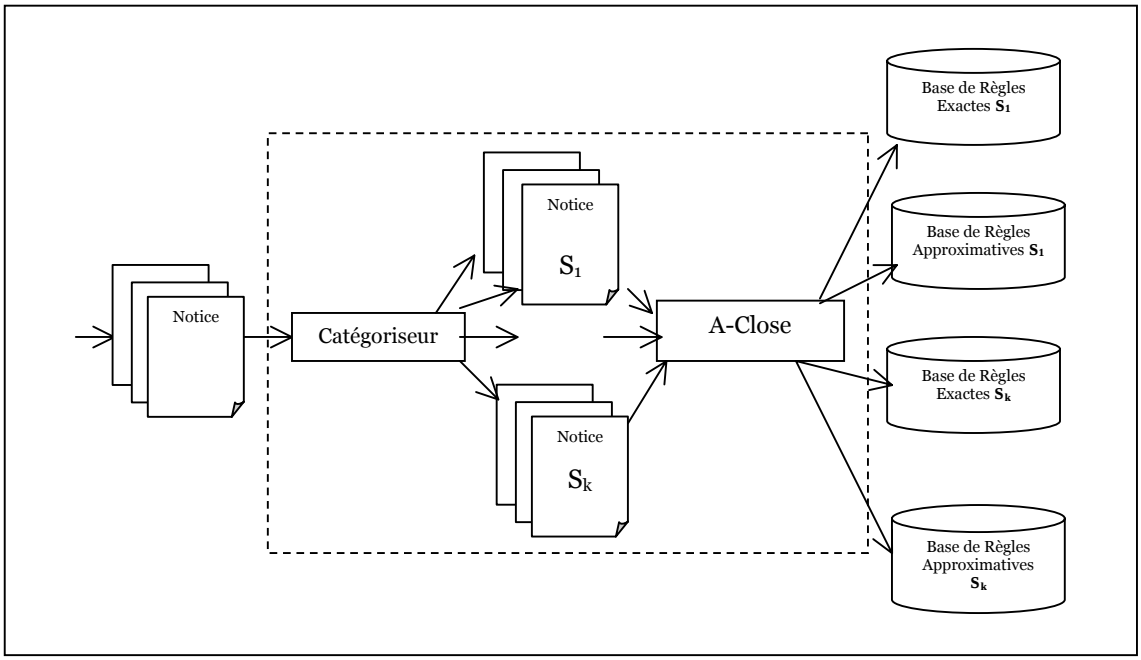
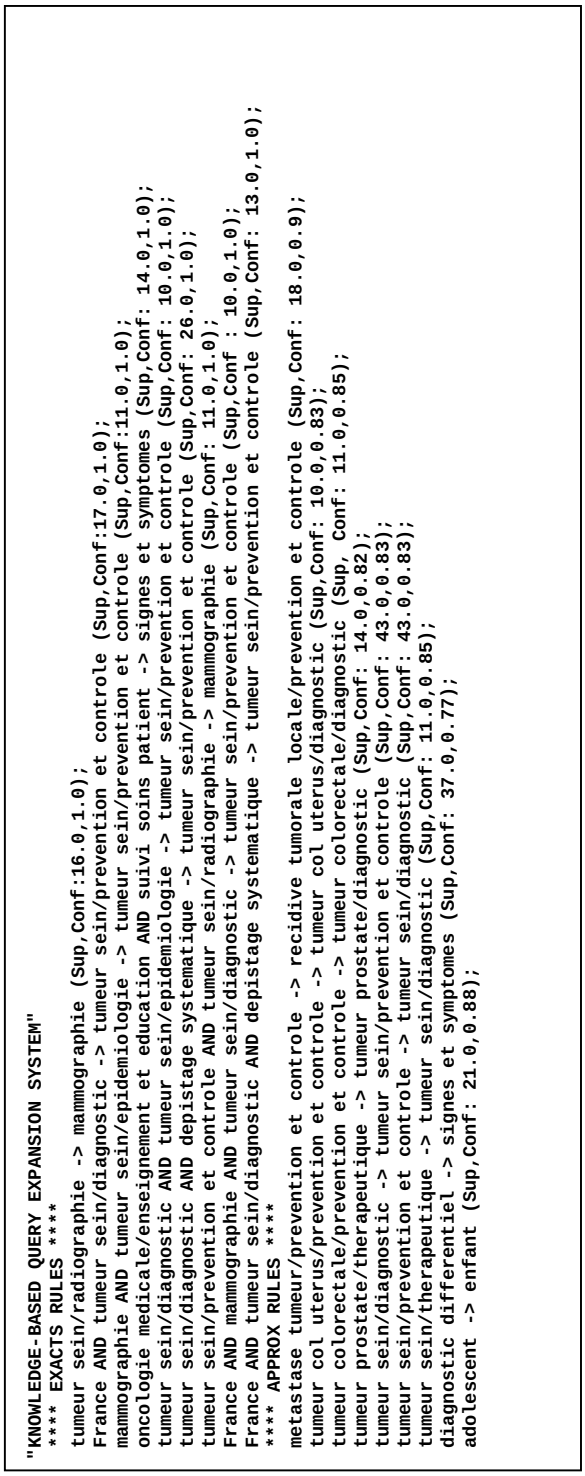


FIG.4.6.– Catégorisation de documents et extraction de règles d'association



```
"KNOWLEDGE-BASED QUERY EXPANSION SYSTEM"
**** EXACTS RULES ****
tumeur sein/radiographie -> mammographie (Sup,Conf:16.0,1.0);
France AND tumeur sein/diagnostic -> tumeur sein/prevention et controle (Sup,Conf:17.0,1.0);
mammographie AND tumeur sein/epidemiologie -> tumeur sein/prevention et controle (Sup,Conf:11.0,1.0);
oncologie medicale/enseignement et education AND suivi soins patient -> signes et symptomes (Sup,Conf: 14.0,1.0);
tumeur sein/diagnostic AND tumeur sein/epidemiologie -> tumeur sein/prevention et controle (Sup,Conf: 10.0,1.0);
tumeur sein/diagnostic AND depistage systematique -> tumeur sein/prevention et controle (Sup,Conf: 26.0,1.0);
tumeur sein/prevention et controle AND tumeur sein/radiographie -> mammographie (Sup,Conf: 11.0,1.0);
France AND mammographie AND tumeur sein/diagnostic -> tumeur sein/prevention et controle (Sup,Conf : 10.0,1.0);
France AND tumeur sein/diagnostic AND depistage systematique -> tumeur sein/prevention et controle (Sup,Conf: 13.0,1.0);
**** APPROX RULES ****
metastase tumeur/prevention et controle -> recidive tumorale locale/prevention et controle (Sup,Conf: 18.0,0.9);
tumeur col uterus/prevention et controle -> tumeur col uterus/diagnostic (Sup,Conf: 10.0,0.83);
tumeur colorectale/prevention et controle -> tumeur colorectale/diagnostic (Sup, Conf: 11.0,0.85);
tumeur prostate/therapeutique -> tumeur prostate/diagnostic (Sup,Conf: 14.0,0.82);
tumeur sein/diagnostic -> tumeur sein/prevention et controle (Sup,Conf: 43.0,0.83);
tumeur sein/prevention et controle -> tumeur sein/diagnostic (Sup,Conf: 43.0,0.83);
tumeur sein/therapeutique -> tumeur sein/diagnostic (Sup,Conf: 11.0,0.85);
diagnostic differentiel -> signes et symptomes (Sup,Conf: 37.0,0.77);
adolescent -> enfant (Sup,Conf: 21.0,0.88);
```

FIG.4.7.– Exemple de fichier en sortie de KnowQue-Dataminer

4.6.2.1 Mots Clés

TAB.4.18.– Nombre de règles extraites pour *minsup*=20 et *minconf*=70% et *I*={MC}

Spécialité	Nombre de Documents	Nombre de Règles Exactes	Nombre de Règles Approximatives	Total
Allergologie	509	101	231	332
Cardiologie	558	251	542	793
Oncologie	644	154	329	483
Psychiatrie	515	76	337	413
Gastroentérologie	501	85	300	385
Neurologie	1 137	169	520	689
Environnement	1 254	257	924	1 181
Diagnostic	883	465	1 218	1 683
Thérapeutique	782	555	2 010	2 565
Pédiatrie	906	1 116	5 629	6 745
Total		**3 229**	**12 040**	**15 269**

Le nombre de règles au total est sensiblement le même qu'avant catégorisation. En revanche, ce ne sont pas les mêmes règles. Elles n'ont pas les mêmes support et confiance.

4.6.2.2 Association Mots Clés/Qualificatifs

TAB.4.19.– Nombre de règles extraites pour *minsup*=20 et *minconf*=70% et *I*={(MC/Qu)}

Spécialité	Nombre de Documents	Nombre de Règles Exactes	Nombre de Règles Approximatives	Total
Allergologie	509	93	206	299
Cardiologie	558	151	332	483
Oncologie	644	119	358	477
Psychiatrie	515	57	155	212
Gastroentérologie	501	96	248	344
Neurologie	1 137	83	285	368
Environnement	1 254	148	584	732
Diagnostic	883	112	312	424
Thérapeutique	782	206	562	768
Pédiatrie	906	205	634	839
Total		**1 270**	**3 676**	**4 946**

En qualifiant les mots clés, les règles d'association sont plus précises et donc moins nombreuses.

4.6.3 Fouille de Textes

Dans ce type de fouille, nous appliquons la technique de fouille de données aux textes. On suppose que deux termes qui apparaissent dans le même document sont sémantiquement liés. Il peut être intéressant de découvrir des règles d'association entre termes « réservés » du domaine et termes « libres » n'appartenant pas à la terminologie. En effet les utilisateurs peuvent utiliser dans leur requête un terme sémantiquement équivalent à un terme réservé mais n'existant pas dans la terminologie. Nous nous basons sur l'hypothèse qu'un document représente un ensemble de termes sémantiquement cohérent et que tous les termes participent à la signification globale du document. Chaque terme a une signification dans le contexte où il est utilisé dans le document. Les termes sont alors des indicateurs d'une certaine connaissance qui révèle le contexte de l'utilisation des termes dans les documents.

L'indexation automatique est réalisée à partir du texte brut en utilisant comme identifiant les différentes URLs.

Les techniques traditionnelles supposent que l'extraction doit se faire sur des bases de données relationnelles structurées. Nos données brutes (textes intégraux) sont traitées par InterMediaText d'Oracle®. Cet outil permet d'indexer tout document ou contenu textuel pour améliorer la recherche d'information dans les applications du Web en évitant de stocker le contenu des pages Web. Les types indexés sont html, xml, pdf, doc, txt…etc. Une ressource peut avoir plusieurs URLs et plusieurs formats.

L'index est construit à l'aide de mesures de pertinence (ou score) en fonction de la fréquence d'apparition d'un terme. Nous y avons intégré notre liste de mots vides (§ Chapitre 3). Le corpus de documents non structurés est transformé en une base de données relationnelle disposant d'un index. Les objets sont les documents et les attributs sont les termes trouvés par InterMediaText. Le score d'un terme est converti en entier et est compris entre 0 et 100. Score = $3f(1+\log(N/n))$

f est la fréquence du terme dans le document, N est le nombre total de lignes d'un document et n est le nombre de lignes qui contiennent le terme.

Nous pouvons retenir, grâce à un script PL/SQL, tous les termes contenus dans un document à partir de l'index et dont le score est supérieur à 50. Un document est composé de 190 termes de score supérieur à 50, avec au minimum 73 termes par document et au maximum 693 termes différents ce qui est un nombre important.

Nous avons gardé les mêmes mesures de support et de confiance que précédemment (*minsup*=20 ; *minconf=70%*). Le processus d'indexation, de sélection des termes et d'extraction des règles d'association est assez long. Le tableau décrit les différents formats de document en entrée et le nombre de règles d'association extraites. Un document peut avoir plusieurs formats.

TAB.4.20.– Résultats pour la fouille de textes

	Format					*Nombre de Règles*	
	Html	Pdf	Doc	Zip	Autres	Exactes	Approximatives
5 128 Documents	4 647	1 689	99	172	135	1 250	3 836

TAB.4.21.– Fouille de textes à partir des documents catégorisés

Spécialité	Nombre de Documents	Nombre de Règles Exactes	Nombre de Règles Approximatives	Total
Neurologie	1 137	1 354	105 202	106 556
Environnement	1 254	2 815	397 073	399 888

Le nombre de règles obtenues est très important. Elles ne peuvent donc pas être évaluées par l'expert mais elles peuvent néanmoins être stockées dans la base de données et être exploitées dans la recherche d'information. L'expansion des requêtes dans ce cas précis sera interactive avec l'utilisateur qui pourra décider des autres termes à inclure dans sa nouvelle requête.

Le processus de fouille de textes est sensible à la phase d'indexation. Si un terme est absent de l'indexation d'un seul texte (i.e. silence), cela peut entraîner la disparition d'une règle du fait du seuil minimum du support.

4.6.4 Fouille de Données Pondérées

Dans les différentes expérimentations, nous avons exploré les relations item-objet classiques. Les seuls degrés figurant dans cette relation sont de type 0 ou 1 traduisant la présence ou l'absence d'un item dans un objet du contexte.

Dans le cadre de CISMeF, nous disposons de la notion de poids d'un descripteur dans un document indexé. Cette information peut être intéressante d'intégrer dans le processus de découverte de règles d'association entre descripteurs. Cependant le poids d'un descripteur n'est pas représenté par des valeurs décimales, mais représenté à l'aide des indicateurs « majeur » et « mineur ». Le poids « majeur » a un facteur de 2 par rapport au poids « mineur ». Nous avons extrait les règles entre descripteurs majeurs.

4.6.4.1 Mots Clés en Majeur

TAB.4.22.– Nombre de règles extraites pour *minsup*=20 et *minconf*=70% et *I*={MC*}

Spécialité	Nombre de Documents	Nombre de Règles Exactes	Nombre de Règles Approximatives	Total
Allergologie	509	4	12	16
Cardiologie	558	7	37	44
Cancérologie	644	2	13	15
Psychiatrie	515	1	8	9
Gastroentérologie	501	4	34	38
Neurologie	1 137	4	34	38
Environnement	1 254	6	85	91
Diagnostic	883	7	36	43
Thérapeutique	782	2	32	34
Pédiatrie	906	6	90	96
Total		**43**	**381**	**424**

4.6.4.2 Associations Mots Clés/Qualificatifs en Majeur

TAB.4.23.– Nombre de règles extraites pour *minsup*=20 et *minconf*=70% et *I*={(MC/Qu)*}

Spécialité	Nombre de Documents	Nombre de Règles Exactes	Nombre de Règles Approximatives	Total
Allergologie	509	2	12	14
Cardiologie	558	5	31	36
Oncologie	644	0	20	20
Psychiatrie	515	0	3	3
Gastroentérologie	501	2	12	14
Neurologie	1 137	0	25	25
Environnement	1 254	5	53	58
Diagnostic	883	4	36	40
Thérapeutique	782	2	18	20
Pédiatrie	906	4	61	65
Total		**24**	**271**	**295**

Les règles d'association obtenues sont relatives exclusivement la spécialité concernée. Ces mêmes spécialités sont déduites à partir des couples (MC/Qu) et (MC) en majeur.

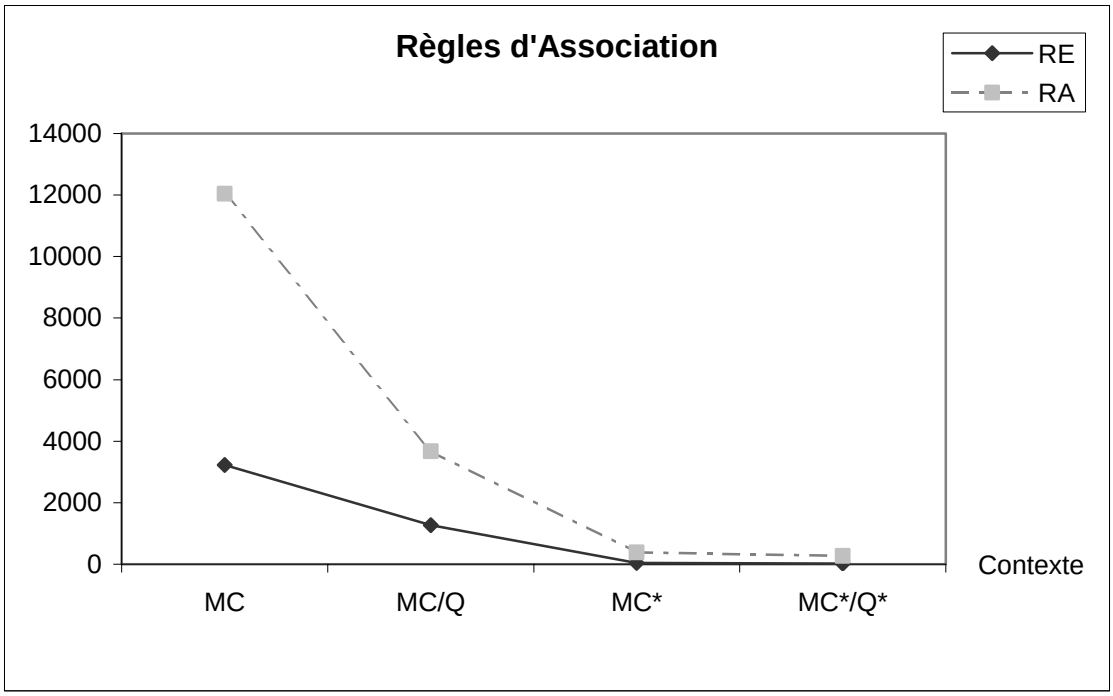


FIG.4.8.–Nombre de règles d'associations en fonction du contexte

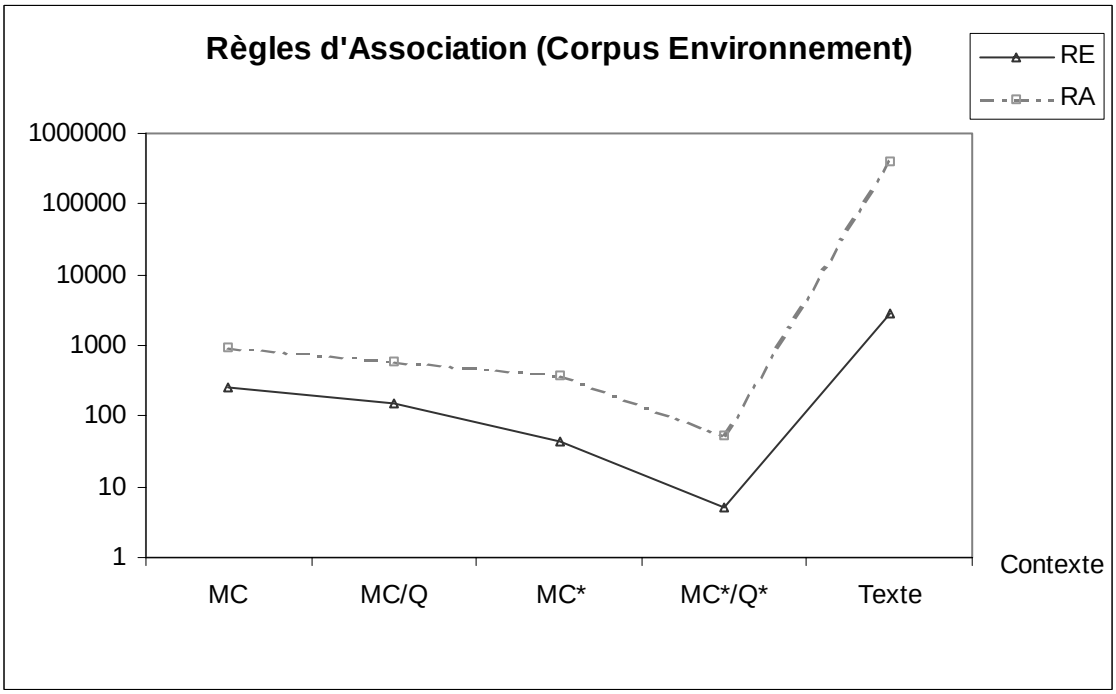


FIG.4.9.– Nombre de règles de la spécialité Environnement

4.6.5 Evaluation des Règles d'Association

Les mesures de support et de confiance permettent une évaluation objective des règles d'association. L'évaluation des règles d'association par l'expert en revanche est une évaluation subjective qui est fonction de ses connaissances du domaine. L'évaluation a porté sur les règles extraites du dernier contexte, en l'occurrence, les 295 règles d'association entre couples (Mot clé/ qualificatif) indexant les documents en « majeur » *(Soualmia & Darmoni, 2004a) (Soualmia & Darmoni, 2004b) (Soualmia & Darmoni, 2004c)*. Le but d'une telle évaluation est de déterminer si une règle d'association est intéressante ou pas. Une règle d'association est intéressante si elle confirme une hypothèse ou si elle établit une nouvelle hypothèse.

Dans le cas du CISMeF, nous avons distingué différents cas de règles intéressantes en fonction des relations qui peuvent exister entre les termes de la terminologie. Une règle d'association intéressante peut associer :

– Un fils (in)direct et son père : relation Père-Fils.

Exemple : *hépatite A → hépatite*. *hépatite A* est fils direct de *hépatite*.

– Deux termes appartenant à la même hiérarchie. Ils ont le même père direct ou indirect : relation Frère.

Exemple : *tumeur/prévention et contrôle → tumeur/diagnostic*. *diagnostic* et *prévention et contrôle* appartiennent à la même hiérarchie.

– Deux termes reliés par la relation Voir Aussi.

Exemple : *analgésiques → douleur*. *douleur* est un Voir Aussi de *analgésiques*.

– Deux termes par une nouvelle relation intéressante non formalisée dans la terminologie: relation Nouvelle. Par exemple : *tumeur sein/diagnostic → mammographie*. Cette règle indique une nouvelle association n'existant pas au préalable dans la terminologie. On peut interpréter qu'un *diagnostic* de *tumeur sein* peut se faire par une *mammographie*.

TAB.4.24.– Evaluation des règles d'association par l'expert

Spécialité	Nombre Nouvelle	Nombre Voir Aussi	Nombre Même Arbo	Nombre PèreFils	Nombre Autres
Allergologie	12	1	0	1	0
Cardiologie	26	3	2	1	4
Oncologie	17	1	1	0	1
Psychiatrie	0	1	0	0	2
Gastroentérologie	5	1	0	0	8
Neurologie	8	4	1	0	12
Environnement	18	4	7	5	24
Diagnostic	32	3	0	1	3
Thérapeutique	3	3	3	1	10
Pédiatrie	5	3	3	2	52
Total	**126**	**24**	**17**	**11**	**116**

Les règles décrivant des associations existantes dans la terminologie sont automatiquement classées. Au total plus que 242 règles à analyser sont proposées à l'expert. Les proportions des différentes règles intéressantes sont les suivantes :

- Nouvelles règles : 70,78 % (42,71% du total)
- Relation Voir Aussi : 13,48 % (08,13% du total)
- Relation Même Arborescence : 09,55 % (05,76% du total)
- Relation Père-Fils : 06,18% (03,73% du total)

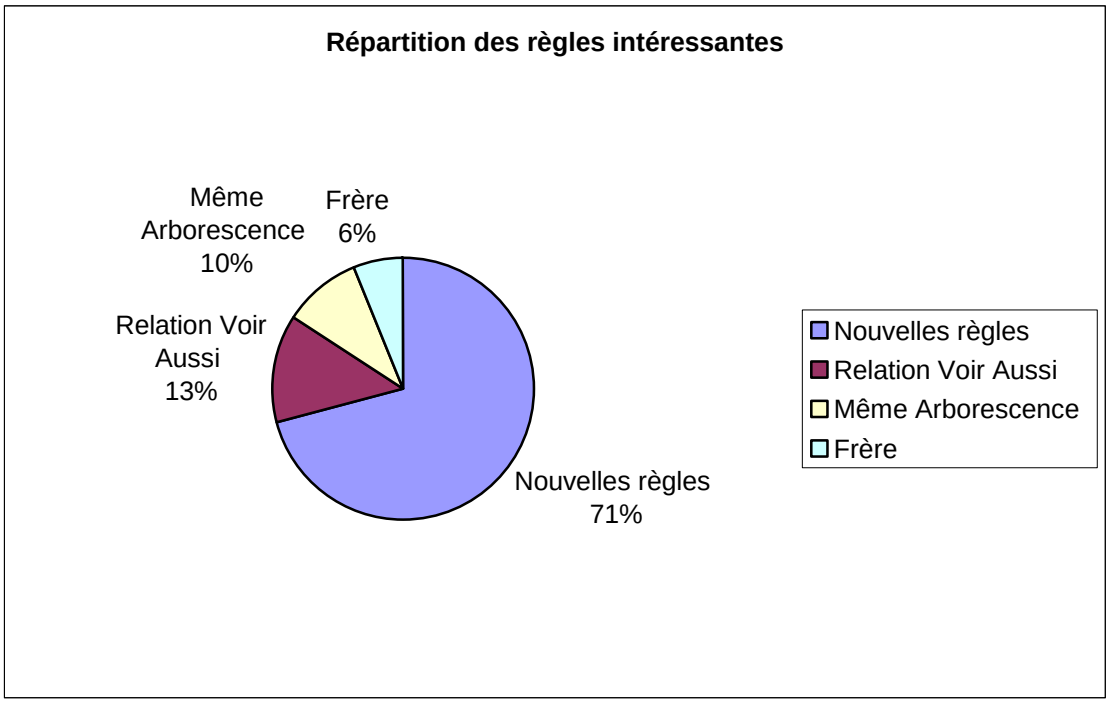


FIG.4.11.– Analyse des règles intéressantes

L'indexation étant réalisée par des experts, connaissant parfaitement la terminologie, on aurait pu penser que des termes qui ont une relation hiérarchique ne seraient pas utilisés conjointement pour indexer une ressource, l'indexation se faisant à l'aide du terme le plus spécifique. Cependant l'analyse des ressources indexées a montré qu'il en était tout autre : 1 466 ressources comportent des mots clés qui sont en relation hiérarchique. Par exemple, une ressource est indexée à l'aide des mots clés *aberration chromosomique* et *trisomie* alors que *trisomie* est fils d'*aberration chromosomique*. 478 ressources sont indexées en utilisant des qualificatifs qui ont une relation hiérarchique et qui sont associés à un même mot clé. Par exemple les associations *nez/anatomie et histoire* et *nez/anatomie pathologique* sont contenues dans la même ressource alors qu'*anatomie pathologique* est fils d'*anatomie et histoire* et enfin 106 ressources sont indexées avec des types de ressources qui sont reliés hiérarchiquement (exemple : *matériel enseignement* et *cours*). Cela se retrouve dans l'analyse des règles intéressantes avec une proportion de 29,22%.

4.7 Exploitation pour la Recherche d'Information

Les outils de cooccurrence développés dans le domaine de la recherche d'information rapprochent des termes qui apparaissent fréquemment dans les mêmes documents et qui possèdent donc sans doute une certaine proximité sémantique. La technique de recherche de cooccurrents est déjà très ancienne puisqu'elle a été promue très tôt en informatique documentaire pour permettre l'expansion de requêtes.

Les règles d'association peuvent être exploitées dans le système de recherche d'information en réalisant des expansions interactives des requêtes des utilisateurs. Une expansion interactive des requêtes (IQE) peut aider l'utilisateur à formuler sa requête *(Magennis & van Rijsbergen, 1997)* à l'image du module *Refine* d'*AltaVista (Bourdoncle, 1997)*, en utilisant le résultat d'une requête pour la reformuler, la filtrer et la réorienter en exploitant les termes liés aux termes de sa requête. En effet, l'utilisateur peut sélectionner des ensembles des termes ou des termes, suggérés, pour les ajouter à sa requête. Dans le cas d'un besoin d'information non précis c'est à dire que l'utilisateur a une idée vague de son besoin d'information, les contextes des termes de sa requête peuvent être suggérés et l'utilisateur choisit les termes à ajouter à sa requête. Dans le cas où le besoin d'information serait précis alors l'utilisateur peut choisir les termes d'un ensemble correspondant à son besoin d'information.

L'expansion d'une requête peut être vue comme un traitement pour élargir le champ de recherche pour cette requête. Une requête étendue va contenir plus de termes reliés. En utilisant le modèle vectoriel, par exemple, plus de documents seront repérés. Ainsi, ce traitement est un moyen d'augmenter le taux de rappel *(Bruandet & Chevallet, 2003)*.

L'expansion de requête peut être effectuée à l'aide des règles d'association extraites. Pour chaque terme d'une requête, son profil relationnel dans le corpus est ajouté dans la requête d'origine. Même si les liens ne sont pas typés, ils permettent de pouvoir préciser son besoin d'information en examinant le contexte d'utilisation des termes qu'il emploie. Le *contexte d'un terme* est une indication sur la définition possible d'un terme ou de son environnement d'usage.

En utilisant la base de connaissances constituée par les différentes règles d'association extraites l'expansion des requêtes peut être interactive ou automatique. Une expansion interactive, contrairement à l'expansion automatique, nécessite une implication de l'utilisateur

dans la phase d'expansion. Ce dernier, selon sa compréhension de la requête, ajoute un certain nombre de termes à la requête d'origine. Pour cela, un module d'expansion interactive a été développé. Les utilisateurs peuvent ajouter d'autres termes, extraits de la base de connaissances, à la requête originale.

Les relations Père-Fils entre les termes sont utilisées dans le processus de recherche d'information. En revanche, les autres types d'association sont proposés à l'utilisateur pour étendre sa requête. La figure est un exemple de propositions de couples (mot clé/qualificatifs) pour la requête *mammographie*. Cette requête peut être étendue en ajoutant par exemple le couple *tumeur sein/prévention et contrôle*. La requête initiale est alors transformée en requête booléenne :

Mammographie OU (tumeur sein/prévention et contrôle)

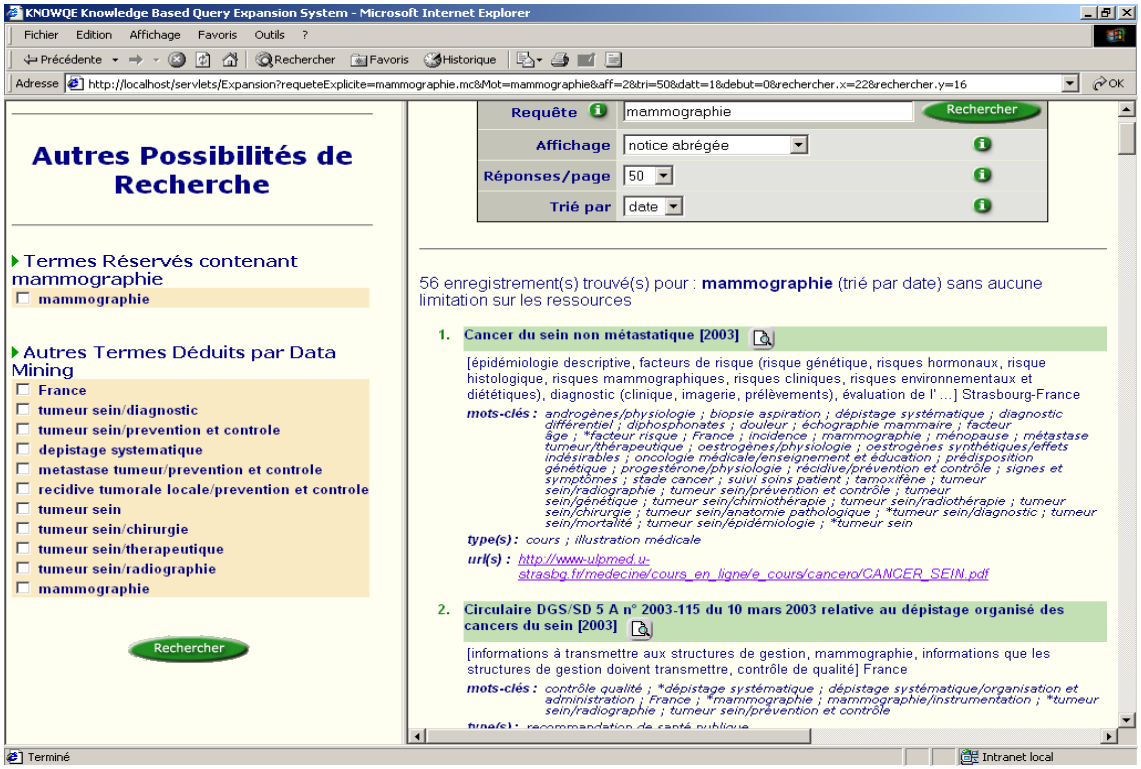


FIG.4.12.– Interface d'expansion de requêtes à l'aide de règles d'association

Au vu des différentes règles d'associations extraites à partir de la base de données, par un processus de fouille de données, les documentalistes ont jugé intéressant de formaliser et d'insérer dans la base de règles des associations entre couples (Mot Clé/Qualificatif) à l'aide de règles appelées *règles expertes*. Par exemple la règle découverte dans le contexte (Mot Clé/Qualificatif) : *hépatite B/prévention et contrôle* → *vaccins anti-hépatite B* a permis d'insérer les nouvelles règles suivantes par l'expert :

grippe/prévention et contrôle → *vaccins anti-grippe* ;

tuberculose/prevention et contrôle → *vaccin BCG ;*

oreillons/prévention et contrôle ET *rougeole/prévention et contrôle* ET *rubéole/prévention et contrôle* → *vaccins antimorbilleux, antiourlien, antirubéoleux.*

dysenterie bacillaire/prévention et contrôle → vaccins anti-shigella

Aujourd'hui cette table de règles *expertes* comporte 380 règles différentes.

4.8 Création de Règles Expertes

En plus d'une aide à la recherche d'information, ces règles peuvent être exploitées par les documentalistes pour une aide à l'indexation manuelle ou semi-automatique. Différents cas sont possibles et ont été modélisés (Octobre 2004) :

– La règle $MC_1/Q_1 \xrightarrow{--} MC_2$ indique qu'il convient de remplacer l'association MC_1/Q_1 par le mot clé MC_2. Par exemple, la règle *abdomen/radiographie* $\xrightarrow{--}$ *radiographie abdominale* indique que le couple (*abdomen/radiographie*) doit être remplacé par le mot clé *radiographie abdominale*.

– La règle $MC_1/Q_1 \xrightarrow{++} MC_2/Q_2$ indique qu'il convient d'ajouter la paire MC_2/Q_2 à l'indexation d'une ressource déjà indexée avec la paire MC_1/Q_1. Par exemple, la règle *appendicectomie* $\xrightarrow{++}$ *appendicite/chirurgie* indique que le couple (*appendicite/chirurgie*) peut être ajouté aux descripteurs d'une ressource indexée avec le mot clé *appendicectomie*.

Différentes catégories de descripteurs ont été définies pour ces règles :

– La catégorie MALADIE : elle comporte tous les concepts des arborescences C et F03.

– La catégorie ACTION (pour ACTION PHARMACOLOGIQUE) : elle comporte tous les concepts de l'arborescence D 27.505.

– La catégorie SUBSTANCE : elle comporte tous les concepts de l'arborescence D sauf les sous-arborescences D 05, D 12, D 13, D 25, D 27.505.

– La catégorie TECHNIQUE : elle comporte tous les concepts de l'arborescence E.

– La catégorie ORGANE : elle comporte tous les concepts de l'arborescence A.

– La catégorie VACCIN : elle comporte tous les concepts de l'arboresence D24.310.894.

Des règles générales ont été modélisées. Elles prennent en compte les termes contenus dans les requêtes ou dans les textes. Quelques-unes sont présentées dans le tableau suivant :

TAB.4.25.– Quelques règles expertes

Catégorie de Descripteurs	**Contenu de la requête ou du texte**	**Propositions**
MALADIE	Lutter contre la maladie Vaccin anti-maladie Immunisation contre maladie	Maladie/prévention et contrôle
ACTION	Traitement avec action	Action/usage thérapeutique
SUBSTANCE	Indiqué pour la substance	Substance/usage thérapeutique

Catégorie de Descripteurs	Contenu de la requête ou du texte	Propositions
TECHNIQUE	Indication de technique	Technique/utilisation
MALADIE (C 04)	Biopsie et maladie	Maladie/anatomie pathologique
MALADIE	Complications de la maladie	Maladie/complications
SUBSTANCE	Pharmacovigilence de la substance	Substance/effets indésirables
SUBSTANCE	la substance est contre-indiquée	Substance/contre-indications
TECHNIQUE	Contre-indications de la technique	Technique/contre-indications
TECHNIQUE	Complications de la technique	Technique/effets indésirables
SUBSTANCE	Sécurité d'emploi de la substance	Substance/contre-indications ET Substance/effets indésirables ET interaction médicamenteuse ET évaluation risque
SUBSTANCE	Analyse chimique de la substance	Substance/analyse
SUBSTANCE	synthèse chimique de la SUBSTANCE structure chimique de la SUBSTANCE propriété chimique de la SUBSTANCE contenu chimique de la SUBSTANCE caractérisation chimique de la SUBSTANCE composition chimique de la SUBSTANCE"	Substance/composition chimique
MALADIE	MALADIE due a une SUBSTANCE	maladie/induit chimiquement
ORGANE	effet de la SUBSTANCE sur un ORGANE	organe/action des produits chimiques
SUBSTANCE	métabolisme de la SUBSTANCE mécanisme d'action de la SUBSTANCE mode d'action de la SUBSTANCE action pharmacologique de la SUBSTANCE durée d'action de la SUBSTANCE	Substance/pharmacologie
ORGANE	Métabolisme de l'organe	organe/métabolisme
MALADIE	Métabolisme de la maladie	Maladie/métabolisme
VACCIN		Vaccination
SUBSTANCE	Toxicité pharmacologique	Substance/induit chimiquement Substance/effets indésirables
TECHNIQUE (E 02)	Efficacité de la technique	Evaluation résultat traitement
TECHNIQUE (sauf E 02)	Efficacité de la technique	Etude technologie biomédicale
MALADIE	Maladie chronique	Maladie et maladie chronique

Il existe également des cas plus complexes. Par exemple si le texte ou la requête comporte « *traitement de la maladie* » ou « *traiter la maladie* », il y a plusieurs cas possibles :

– S'il est également question d'une *action* et d'une *substance*, les couples (maladie/traitement) et (action/usage thérapeutique) et (substance/usage thérapeutique) sont proposés.

– S'il est également question dans la requête ou dans le document d'une *action* mais pas d'une *substance*, les couples (maladie/traitement) et (action/usage thérapeutique) sont proposés.

– S'il n'est question ni d'une *action* ni d'une *substance*, l'un des couples suivants peut être utilisé :

  – (maladie/thérapeutique) par défaut, si aucune précision n'est donnée sur le type de thérapie.

  – (maladie/chirurgie) s'il s'agit d'un « traitement chirurgical » ou autre type d'intervention de l'arborescence E4.

  – (maladie/radiothérapie) s'il s'agit d'un « traitement par rayons ».

4.9 Quelques Perspectives

La fouille de données que nous avons réalisé passe par une indexation conceptuelle de documents qui consiste à les indexer par des structures conceptuelles extraites des documents eux-mêmes. Même si les documents sont indexés par des descripteurs que nous appelons mot clé, ce type d'indexation dépasse la simple indexation au niveau du terme. En effet, les descripteurs utilisés sont des structures conceptuelles représentées par des concepts organisés en une hiérarchie sur lesquels il est possible de faire des calculs de spécialisation ou de généralisation. Cette hiérarchie peut être exploitée afin de générer automatiquement des règles d'associations généralisées en déterminant pour l'antécédent ou le conséquent de la règle, le concept généralisant dans la hiérarchie.

InterMediaText permet d'intégrer des dictionnaires et des thésaurus. On peut envisager d'intégrer le vocabulaire contrôlé du CISMeF afin d'indexer automatiquement les documents (*vs* indexation manuelle actuelle) et en extraire des règles d'association par un processus de fouille de textes. En déterminant les degrés d'appartenance des termes dans un document du corpus, calcul possible grâce à la fonction score de InterMediaText, on peut générer des règles d'association entre termes pondérés avec des poids compris entre [0,1], les règles d'association devenant *floues*.

Toujours concernant l'indexation, les règles d'association et les règles expertes proposées par l'expert, peuvent être traduites sous la forme d'automates ou de grammaires locales et intégrées à des outils comme INTEX, pour proposer une indexation automatique des documents.

Un seuil minimal de support a été fixé au préalable pour éliminer les règles très rares. On peut également envisager de fixer un seuil maximum de support afin d'éviter de générer des règles d'association très fréquentes qui ne sont pas utiles. En effet, si la mesure de support est très élevée cela veut dire que les deux ensembles d'items co-occurrent trop souvent ensemble dans les entités textuelles de la collection. Cette connaissance est donc explicite (elle n'est pas

nouvelle) et il n'est pas intéressant de l'extraire. On peut également envisager d'utiliser d'autres indices statistiques.

BIBLIOGRAPHIE

(*Agrawal & Srikant, 1994*) AGRAWAL R. & SRIKANT R (1994) Fast algorithms for mining association rules in large databases. *Proc. VLDB*. pp 478–499.

(*Agrawal et al., 1993*) AGRAWAL R., IMIELINSKI T., SWAMI A.N. (1993) Mining association rules between sets of items in large databases. *Proceedings of the 1993 ACM SIGMOD International conference on management of data,* pp. 207-216

(*Ahonen et al., 1997*) AHONEN H., HEINONEN O., KLEMETTINEN M. , A. VERKAMO. (1997) Applying data mining techniques in text analysis. *Technical report, Department of Computer Science, University of Helsinki.*

(*Azé, 2003*) AZE J. (2003) Extraction de connaissances à partir de données numériques et textuelles. *Thèse de doctorat Paris XI*.

(*Azé. & Roche, 2003*) AZE J. & ROCHE M. (2003) Une application de la fouille de textes : l'extraction des règles d'association à partir d'un corpus spécialisé. *In Proc. of Extraction et Gestion des Connaissances (EGC'03), volume 17 of RSTI/RIA-ECA*, pp. 283-294.

(*Bastide et al., 2002*) BASTIDE Y, TAOUIL R, PASQUIER N, STUMME G, LAKHAL L (2002) Pascal un algorithme d'extraction des motifs fréquents. *Techniques et sciences Informatiques 21*(1) :65-85.

(*Bastide, 2000*) BASTIDE Y. (2000) Algorithmique de data mining : techniques d'implémentation et négation. *Thèse de doctorat, Université de Clermont-Ferrand II*.

(*Bayardo et al., 1999*) BAYARDO RJ., AGRAWAL R., GUNOPULOS D.. (1999) Constraint-based rule mining in large, dense databases. Proceedings of the *ICDE conference*, pp 188–197.

(*Bayardo, 1998*) BAYARDO RJ. (1998) Efficiently mining long patterns from databases. Proceedings of the *SIGMOD conference*, pp 85–93

(*Bourdoncle, 1997*) BOURDONCLE F.(1997) Livetopics: Recherche visuelle d'information sur l'internet. *La Documentation Française*, (74):36–38.

(*Brin et al., 1997*) BRIN S., MOTWANI R., ULLMAN JD., TSUR S. (1997) Dynamic itemset counting and implication rules for market basket data. Proceedings of the *SIGMOD conference*, pp 255–264.

(*Brin et al., 1997*) BRIN S., MOTWANI R., SILVERSTEIN (1997) Beyond market baskets: generalizing association rules to correlations. Proceedings of the *ACM SIGMOD'97* pp.265-276.

(*Bruandet & Chevallet, 2003*) BRUANDET MF & CHEVALLET JP. (2003) Utilisation et construction de Bases de connaissances pour la recherche d'information. *Assistance Intelligente à la recherche d'information*, pp.85-118.

(*Cherfi et al., 2003*) CHERFI H, NAPOLI A, TOUSSAINT Y (2003) Vers une méthodologie de fouille de textes s'appuyant sur l'extraction de motifs fréquents et de règles d'association. *Conférence CAP'2003*

(*Cherif-Latiri et al., 2003*) CHERIF-LATIRI C., BENYAHIA S., MINEAU G., JAOUA A. (2003) Découverte de Règles Associatives non Redondantes : Application aux corpus textuels. *Revue Intelligence Artificielle,* Vol (17) N° 1, pp.131-144.

(*Cooley at al., 1997*) R. COOLEY, B. MOBASHER, AND J. SRIVASTAVA. (1997) Web mining: Information and pattern discovery on the world wide web. *IEEE International Conference on Tools with Artificial Intelligence (ICTAI'97),* pp.558–567.

(*Davey & Priestley, 1994*) DAVEY BA. & HA. PRIESTLEY. (1994) Introduction to lattices and order. *Cambridge University Press, Fourth edition*.

(*Duquenne & Guigues, 1986*) DUQUENNE V., GUIGUES JL. (1986) Famille minimale d'implications informatives résultant d'un tableau de données binaires. *Mathématiques et Sciences Humaines*, 24(95):5–18.

(*Fayyad et al., 1996*) UM. FAYYAD, G. PIATETSKY-SHAPIRO, P. SMYTH, R. UTHURUSAMY (1996) Advances in Knowledge Discovery and Data Mining. *American Association of Artificial Intelligence Press*.

(*Feldman & Hirsh, 1996*) R. FELDMAN & H. HIRSH. (1996) Mining associations in text in the presence of background knowledge. *2nd International Conference on Knowledge Discovery (KDD-96),* pp. 343–346.

(*Ganter & R. Wille, 1999*) B. GANTER & R. WILLE. (1999) Formal Concept Analysis : Mathematical foundations. *Springer 1999.*

(*Godin & Missaoui, 1994*) R. GODIN & R. MISSAOUI. (1994) An incremental concept formation approach for learning from databases. *Theoretical Computer Science: Special issue on formal methods in databases and software engineering*, 133(2):387–419.

(*Gras et al., 2001*) GRAS R, KUNTZ P, BRIAND H. (2001) Les fondements de l'analyse statistique implicative et quelques prolongements pour la fouille de données. *Revue Mathématiques et Sciences humaines* 154-155 :9-29

(*Guillaume, 2000*) GUILLAUME S. (2000) Traitement des données volumineuses : mesures et algorithmes d'extraction de règles d'association et règles ordinales. *Thèse Université de Nantes.*

(*Hajek et al., 1966*) HAJEK P, HAVEL I, CHYTIL M (1966). The GUHA method for Automatic hypotheses determination. *Computing 1* : 293-308

(*Hajek, 2001*) HAJEK P. (2001) The GUHA method and mining association rules. *Computational Intelligence: Method and Applications* pp.533-539.

(*Heckerman, 1996*) D. HECKERMAN. (1996) Bayesian networks for knowledge discovery. Advances in *Knowledge Discovery and Data Mining*, pp 273–305.

(*IBM, 1996*) IBM intelligent Miner User's Guide Version 1 Release 1.

(*Klemettinen et al., 1994*) M. KLEMETTINEN, H. MANNILA, P. RONKAINEN, H. TOIVONEN, A. I. VERKAMO. (1994) Finding interesting rules from large sets of discovered association rules. *CIKM conf.*, pp 401–407.

(*Lallich & teytaud, 2004*) LALLICH S. & TEYTAUD (2004) Evaluation et validation de l'intérêt des règles d'association. *Journée GafoDonnées.*

(*Lavrac et al., 1999*) LAVRAC N., FLACH P., ZUPAN B. (1999) Rule evaluation measures : a unifying view. Dzeroski &Flach (Eds) *Lecture Notes in Artificial Intelligence # 1634.* pp.174-185.

(*Lehn, 2000*) LEHN R (2000) Un système interactif de visualisation et de fouille de règles pour l'extraction de connaissances dans les bases de données. *Thèse Université de Nantes.*

(*Lent et al., 1997*) B. LENT, R. AGRAWAL, AND R. SRIKANT. (1997) Discovering trends in text databases. *Conference on Knowledge Discovery in Databases and Data Mining (KDD'1997),* pp. 227–230.

(*Lin & Kedem, 1998*) D. LIN & Z. M. KEDEM. (1998) Pincer-Search : A new algorithm for discovering the maximum frequent set. *EBDT conf., Lecture Notes in Computer Science #1377*, pp. 105–119.

(*Luxenburger, 1991*) M. LUXENBURGER. (1991) Implications partielles dans un contexte. *Mathématiques, Informatique et Sciences Humaines*, 29(113):35–55.

(*Magennis & Rijsbergen, 1997*) M. MAGENNIS & CJ. VAN RIJSBERGEN. (1997) The potential and actual effectiveness of interactive query expansion. In *International Conference on Research and Development in Information Retrieval (SIGIR'97)* pp.324–332.

(*Mannila et al., 1994*) H. MANNILA, H. TOIVONEN, A.I. VERKAMO. (1994) Efficient algorithms for discovering association rules. *AAAI KDD workshop*, pp 181–192.

(*Masseglia et al., 1999*) F. MASSEGLIA, P. PONCELET, R. CLICHETTI. (1999) Analyse du comportement des utilisateurs sur le web. *Congrès Informatique des Organisations et Systèmes d'Information et de Décision (INFORSID'1999),* pp.393–412.

(*Masseglia, 2002*) F. MASSEGLIA. (2002) Algorithmes et applications pour l'extraction de motifs séquentiels dans le domaine de la fouille de données : de l'incrémental au temps réel. *Thèse de doctorat, Université de Montpellier.*

(*Ng et al., 1998*) RT. NG, VS. LAKSHMANAN, J. HAN, A. PANG. (1998) Exploratory mining and pruning optimizations of constrained association rules. Proceedings of the *SIGMOD conference*, pp 13–24.

(*Pasquier et al., 1998*) N. PASQUIER, Y. BASTIDE, R. TAOUIL, L. LAKHAL. (1998) *Pruning closed itemset lattices for association rules. Bases de Données Avancées*, pp.177–196.

(*Pasquier et al., 1999*) N. PASQUIER, Y. BASTIDE, R. TAOUIL, L. LAKHAL. (1999) Closed set based discovery of small covers for association rules. *Bases de Données Avancées,* pp.361–381.

(*Pasquier et al., 2004*) PASQUIER N, TAOUIL R, BASTIDE Y, STUMME G, LAKHAL L. (2004) Generating a condensed representation of association rules, *Journal of intelligent information systems, Kluwer Academic Publisher,* à paraître.

(*Pasquier, 2000*) N. PASQUIER. (2000) Data mining : algorithmes d'extraction et de réduction des règles d'association dans les bases de données. *Thèse de Doctorat, Université de Clermont-Ferrand II*.

(*Piatetsky-Shapiro & Matheus, 1994)* G. PIATETSKY-SHAPIRO & CJ. MATHEUS. (1994) The interestingness of deviations. *AAAI Knowledge Data Discovery workshop*, pp 25–36.

(*Roche et al., 2004*) ROCHE M, AZÉ J, MATTE-TAILLIEZ O, KODRATOFF Y. (2004) Mining texts by association rules discovery in technical corpus. *Proceedings of Intelligent Information Processing and Web Mining 2004*, pp.89-98.

(*Sahar, 1999*) SAHAR S (1999) Interestingness via what is not interesting. Knowledge discovery and datamining. pp.332-336.

(*Savasere et al., 1995*) A. SAVASERE, E. OMIECINSKI, S. NAVATHE. (1995) An efficient algorithm for mining association rules in large databases. Proc. *Very Large Data Bases conference*, pp 432–444.

(*Silberschatz & Tuzhilin, 1996*) A. SILBERSCHATZ & A. TUZHILIN. (1996) What makes patterns interesting in knowledge discovery systems. IEEE *Transactions on Knowledge and Data Engineering*, 8(6):970–974.

(*Soualmia & Darmoni, 2004 a*) SOUALMIA LF.& DARMONI SJ. (2004) Correcting and Refining Users Queries: the Contribution of Morphological Knowledge and Association rules. IPMU, *Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pp. 2059-2066

(*Soualmia & Darmoni, 2004 b*) SOUALMIA LF. & DARMONI SJ. (2004) Combining Knowledge-based Methods to Refine and Expand Queries in Medicine. *FQAS, Flexible Query Answering Systems 2004, Lectures Notes in Artificial Intelligence # 3055,* pp.243-255.

(*Theilmann & Rothermel , 1998*) W. Theilmann & K. Rothermel. (1998) Domain experts for information retrieval. in the world wide web. *Lecture Notes in Artificial Intelligence # 1435,* pp.216–227.

(*Toivonen et al., 1995*) H. TOIVONEN, M. KLEMETTINEN, P. RONKAINEN, K. HATONEN, H. MANNILA. (1995) Pruning and grouping discovered association rules. *ECML MLnet workshop*, pp 47–52.

(*Zaki et al., 1997*) MJ. ZAKI, S. PARTHASARATHY, M. OGIHARA, W. LI. (1997) New algorithms for fast discovery of association rules. Proc. *Knowledge Data Discovery conference*, pp.283–286.

CHAPITRE 5

REPRESENTATION DES CONNAISSANCES : DU MODELE AU FORMEL

Sommaire

5.1 INTRODUCTION 212
5.2 LES ONTOLOGIES POUR LA RECHERCHE D'INFORMATION 213
5.3 LES MODELES ET LANGAGES DE REPRESENTATION 215
5.3.1 LES RESEAUX SEMANTIQUES 215
5.3.2 LES GRAPHES CONCEPTUELS 216
5.3.3 LES LANGAGES DE FRAMES 218
5.3.4 LA REPRESENTATION DES CONNAISSANCES PAR OBJET 219
5.4 QUELQUES PROJETS DU WEB SEMANTIQUE 220
5.4.1 LE LANGAGE SHOE 221
5.4.2 ONTOSEEK 221
5.4.3 WEBKB 222
5.4.4 CWEB 223
5.4.5 OCML 224
5.5 LE CISMEF DANS L'INFRASTRUCTURE DU WEB SEMANTIQUE 226
5.5.1 REPRESENTATION DES METADONNEES 226
5.5.2 LA TERMINOLOGIE CISMEF : ENTRE TERMINOLOGIE ET ONTOLOGIE 227
5.6 LES LOGIQUES DE DESCRIPTION 229
5.6.1 LES FORMALISMES TERMINOLOGIQUES 230
5.6.2 ASSERTIONS ET REGLES D'INFERENCE 231
5.6.3 LE RAISONNEMENT TERMINOLOGIQUE 232
5.6.3.1 LA RECONNAISSANCE D'INDIVIDUS 232
5.6.3.2 LE RAISONNEMENT SUR LES DESCRIPTIONS 232
5.6.4 EXPRESSIVITE ET COMPLEXITE 233
5.6.4.1 LES PROCEDURES DE CALCUL COMPLETES 233
5.6.4.2 PROCEDURES DE CALCUL INCOMPLETES 233
5.6.4.3 COMPLETUDE ET EFFICACITE 234
5.6.5 LES CONSTRUCTEURS 234
5.6.5.1 LES CONSTRUCTEURS DE CONCEPTS 234
5.6.5.2 LES CONSTRUCTEURS DE ROLES 235
5.6.5.3 LA SYNTAXE DES CONSTRUCTEURS 235
5.6.5.4 LA SEMANTIQUE DES CONSTRUCTEURS 236

5.6.5.5 Exemple 238
5.6.6 Raisonnement Taxinomique pour la Recherche d'Information 238
5.7 FORMALISATION DE LA TERMINOLOGIE 239
5.7.1 Experiences avec TRIPLE 239
5.7.2 De OIL a OWL 241
5.7.3 Travaux dans le Domaine de la Sante 242
5.7.4 Principes de Modelisation 243
5.7.5 Des Fichiers Textes a la Base de Donnees 244
5.7.6 De la Base de Donnees a la Base de Connaissances 244
5.7.6.1 Les Classes OWL 245
5.7.6.2 Les Proprietes OWL 246
5.7.6.3 La Propriete Part-Of 246
5.7.6.4 Les Restrictions sur les Domaines des Proprietes 247
5.7.6.5 Representation des Documents 247
5.7.7 Verification de la Consistance et Classification 248
5.7.7.1 Import sous Protege-2000 248
5.7.7.2 Verification de la Consistance 248
5.7.7.3 La Classification 250
5.7.7.4 Ameliorations Possibles 252
5.8 VERS UNE ONTOLOGIE ? LE PROJET ATONANT 254
5.8.1 Terminae 255
5.8.2 Resultats dans le Cadre du Projet ATONANT 255

Dans ce Chapitre 5, nous abordons la notion de Web Sémantique en général ainsi que les systèmes intelligents pour la recherche d'information. Ces systèmes ou bases de connaissances utilisent des langages formels de représentation des connaissances. Nous nous intéressons particulièrement aux logiques de description. Nous proposons une méthode de modélisation et de représentation automatique de la terminologie CISMeF en OWL-DL afin de pouvoir réaliser des inférences pendant la phase de recherche d'information.

5.1 Introduction

Une des préoccupations actuelles est de favoriser un accès intelligent aux données du Web en exploitant des connaissances relatives au domaine des données traitées. Dans ce cadre est proposée une approche générale qui couple l'exploitation de connaissances à un système de recherche d'information. En effet, la problématique qui se pose est celle d'une *recherche d'information intelligente* sur le Web. L'ambition du Web Sémantique *(Berners-Lee et al., 2001)* est qu'il soit exploité en priorité par des machines qui traitent des problèmes posés par des utilisateurs et qui délivrent les résultats. Une machine pourra fournir une aide aux utilisateurs si on la dote d'une certaine *intelligence*. Le Web Sémantique est un espace d'échange qui reste à construire. Un de ses intérêts est d'une part d'apporter suffisamment de renseignements sur les documents, en ajoutant des annotations sous la forme de *méta-données* et d'autre part, de décrire leur contenu de manière à la fois formelle et signifiante à l'aide d'une *ontologie* ou *d'un système à base de connaissances (Laublet et al., 2002)*.

Les systèmes à base de connaissances sont à la base des systèmes intelligents et du Web Sémantique. En effet, un système à base de connaissances repose sur une base de connaissances et un module de raisonnement. Les unités de connaissance sont représentées dans un formalisme de représentation des connaissances où ils ont une syntaxe et une sémantique associée. L'inférence peut être réalisée à partir des connaissances disponibles afin d'en dériver de nouvelles en s'appuyant sur la sémantique du formalisme de représentation et à l'aide de mécanismes de raisonnement.

En décrivant le contenu des documents à l'aide de connaissances dans un langage formel, cela permet à la machine de les reconnaître. De plus, ce genre de représentation documentaire à l'aide de connaissances a ouvert de nouvelles perspectives pour la recherche d'information sur le Web. En effet, les connaissances vont permettre d'organiser la masse d'information disponible et par conséquent en faciliter l'accès. Se pose alors la nécessité de disposer de langages pour exprimer le contenu des documents, d'une sémantique associée à ces langages et de moteurs d'inférences associés qui s'appuient sur cette sémantique pour raisonner. Cette nécessité est donc en relation avec les problématiques de représentation des connaissances. Les plus anciennes applications du Web Sémantique travaillent avec des documents HTML, le langage de balisage utilisé pour mettre en forme l'information mise à disposition sur le Web. Dès lors, les langages ont évolué en fonction des besoins.

Les ontologies et les méta-données sont deux éléments principaux pour la construction de l'infrastructure du Web Sémantique *(Laublet et al., 2002)*. Une ontologie est une modélisation partagée d'un domaine pour améliorer la communication et éliminer les ambiguïtés entre

personnes, entre personnes et applications ou entre applications. Elle est composée d'une hiérarchie de concepts, de relations entre concepts et d'un ensemble de règles ou de contraintes. Les méta-données font référence à une information descriptive des ressources du Web. Leur première utilité est la recherche d'information.

Nous décrivons dans ce chapitre des projets qui sont à la base du Web Sémantique, ainsi que quelques langages de représentation des connaissances. Nous nous attardons sur les logiques de description *(Baader et al., 2003)* qui sont des langages dotés de mécanismes de raisonnement puissants. Vu sa structure, le CISMeF se place à cheval entre le Web informel actuel et le Web Sémantique. En effet, il manque à la terminologie une dimension formelle. Nous proposons une méthode totalement automatique de formalisation de la terminologie à l'aide du langage OWL-DL *(Horrocks et al., 2003)*. Le but n'est pas d'obtenir une ontologie du domaine médical (tout ce qui est formel n'est pas forcément une ontologie…), mais de pouvoir utiliser les nouvelles technologies disponibles afin de les exploiter dans le processus de recherche d'information. Nous étudions en fin de chapitre quelques possibilités d'obtenir une ontologie au vrai sens du terme, notamment dans le cadre du projet Atonant en cours (2004-2005)

5.2 Les Ontologies pour la Recherche d'Information

D'après la définition[35] des connaissances du dictionnaire informatique Foldoc[36], les connaissances sont des informations sur lesquelles on peut raisonner. Dans le cadre d'un système de recherche d'information, le fait de caractériser les connaissances associées au document et de travailler sur des connaissances permet aux systèmes d'en inférer de nouvelles pour établir la correspondance entre un besoin d'information et un document.

L'ensemble des connaissances nécessaires aux raisonnements est relativement figé. Les composants (concepts, relations, individus etc.) du domaine sont identifiés par une terminologie fixée par un expert et partagée par l'ensemble des utilisateurs du système. De plus, pour pouvoir être manipulés par la machine, ces composants sont définis dans un langage formel. L'ensemble de ces composants et leur définition constituent ce que les chercheurs en IA appellent une ontologie, dont la définition la plus communément admise est une explicite spécification d'une conceptualisation *(Gruber, 1983)* ou plus sommaire, l'expression d'un point de vue sur les composants d'un monde restreint. Cette ontologie est connue et comprise par les utilisateurs du système. Les ontologies peuvent être définies par une commune compréhension des centres d'intérêt d'un domaine : "shared understanding of some domain of interest" *(Uschold & Gruninger, 1996)*. Il existe différentes sortes d'ontologies suivant le degré de formalisation utilisé pour définir les connaissances. A titre d'exemple, un thésaurus peut être considéré comme une ontologie non formelle.

En recherche d'information, une ontologie peut être utilisée par un moteur de recherche pour accéder à des documents. D'après *(Aussenac & Condamines, 2004)* l'apport de l'ontologie peut être appréhendé à trois niveaux dans un système de recherche d'information. Au niveau du processus d'indexation des documents : (1) en s'associant à des techniques de traitement automatique du langage naturel, les documents sont résumés puis reliés à des concepts de l'ontologie ; (2) Au niveau du processus de filtrage d'information ; (3) Au niveau de la

[35] "knowledge:…if information is data plus meaning then knowledge is information plus processing…"
[36] http://foldoc.doc.ic.ac.uk/foldoc/index.html

reformulation des requêtes, pour améliorer les requêtes utilisateurs. L'utilisation des ontologies pour l'expansion des requêtes utilisateurs peut être une solution *(Guarino et al., 1999) (Gandon et al., 2002)*. En effet, les ontologies fournissent les ressources généralement sous forme de relations sémantiques permettant de mieux identifier le sens des mots dans la requête et de la traiter pour élargir le champ de recherche. D'autre part, elles constituent un cadre partagé (un vocabulaire commun) que les différents acteurs peuvent mobiliser *(Bachimon, 2000) (Gandon, 2002)* pour rapprocher le langage des requêtes de celui dont les documents sont exprimés.

Le but du projet OntoBroker *(Fensel et al., 1998)* est de représenter de manière formelle la connaissance contenue dans des documents HTML en en fonction d'une ontologie de domaine qui définit le vocabulaire commun de la communauté des chercheurs en acquisition des connaissances. La finalité est de faciliter l'échange de publications, ces chercheurs se sont regroupés sous un projet commun "the (KA)² initiative" pour créer leur propre ontologie de leur domaine de recherche *(Benjamins & Fensel, 1998)*. La figure présente une partie de cette ontologie.

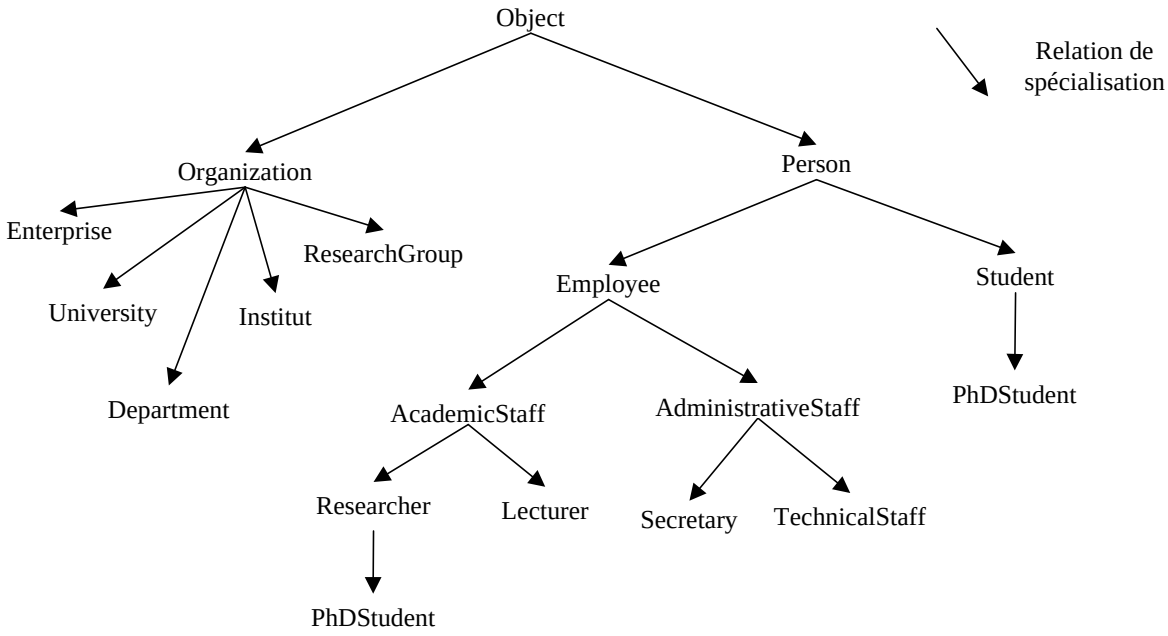


FIG.5.1.– Une partie de l'ontologie (KA)² initiative

D'autres projets plus ou moins récents se basent sur l'utilisation d'une ontologie pour décrire formellement des ensembles de documents ou de ressources. SHOE *(Heflin et al., 1999)* est l'un des précurseurs du Web Sémantique. Des ontologies et un langage basé sur HTML sont utilisés pour annoter sémantiquement des pages Web. Les outils de WebKB *(Martin & Eklund, 2001)* permettent aux utilisateurs de stocker organiser et retrouver la connaissance qui est contenue dans un ensemble de documents et formalisée dans un graphe conceptuel. OntoSeek *(Guarino et al., 1999)* est un moteur de recherche de pages Web qui utilise l'ontologie terminologique WordNet. Les documents et les requêtes sont représentés à l'aide de graphes conceptuels. Dans CoMMA *(Gandon et al., 2002)* des annotations sous forme de fichiers RDF *(Lassila & Swick, 1999)* sont associées à des documents d'entreprise en fonction des concepts d'une ontologie (O'CoMMA). Dans tous ces projets une ontologie formelle sert de pivot.

5.3 Les Modèles et Langages de Représentation

Même si les ontologies sont un premier pas vers la standardisation des structures d'indexation des pages Web du fait qu'elles définissent un vocabulaire standard, aucun standard sur la structuration à employer pour décrire des pages Web n'est communément admis. De nombreux systèmes de recherche sur le Web utilisent des connaissances pour décrire le contenu des documents et chacun de ces systèmes a sa propre structure d'indexation. Nous avons sélectionné certains projets du Web Sémantique qui se fondent sur différents langages et modèles de représentation des connaissances.

Deux grands courants ont fortement marqué le domaine de la représentation des connaissances : les réseaux sémantiques et les frames de Minsky. Dans ce paragraphe, nous présentons les modèles de représentation de connaissances les plus connus. Une étude comparative de ces modèles est le but du projet Escrire ou encore l'objet des travaux de *(Barry et al., 2001)*.

5.3.1 Les Réseaux Sémantiques

Les réseaux sémantiques sont le premier modèle à héritage utilisé en représentation des connaissances. Issus des travaux de psychologie cognitive sur l'organisation de la mémoire, ils ont donné lieu à de nombreux modèles de représentation des connaissances. On peut citer, entre autres, les réseaux sémantiques partitionnés *(Hendrix, 1978)*, les graphes conceptuels *(Sowa, 1984)*, le langage KL-ONE *(Brachman & Schmolze, 1985)*. En IA, on attribue souvent à *(Quillian, 1968)* l'origine de ce mode de représentation. Il fut le premier à formaliser les réseaux sémantiques, afin de représenter la signification des mots du langage sous forme de mémoire associative. Plus tard, *(Woods, 1975)* et *(Brachman, 1977)* en donnèrent un modèle rigoureux.

Un réseau sémantique est un graphe orienté et étiqueté ou plus précisément un multigraphe car deux nœuds du graphe peuvent être reliés par plusieurs arcs. Il est constitué d'un ensemble de nœuds typés, dénotant des concepts du domaine modélisé, et d'arcs orientés étiquetés représentant les relations sémantiques entre les concepts. Ainsi un concept est décrit par les autres concepts du réseau en lien avec lui. Certains nœuds correspondent intuitivement plus à des classes d'objets qu'à des individus. Représenter l'appartenance à une classe nécessite une relation d'appartenance. Les réseaux possèdent un nom réservé d'étiquette pour cette relation, parfois nommé "sorte de"[37]. Cette relation d'appartenance à une classe suppose que l'on sait différencier les nœuds qui représentent des classes de ceux qui dénotent des individus. De nombreux formalismes de représentation des connaissances dérivés des réseaux sémantiques imposent de noter différemment les deux types de nœuds (par exemple KL-ONE *(Brachman & Schmolze, 1985)*. La relation d'appartenance à une classe permet de reporter les connaissances de la classe sur un individu. Cette opération est appelée inférence par héritage.

[37] cette relation d'appartenance correspond à la relation d'instanciation des langages orienté objet.

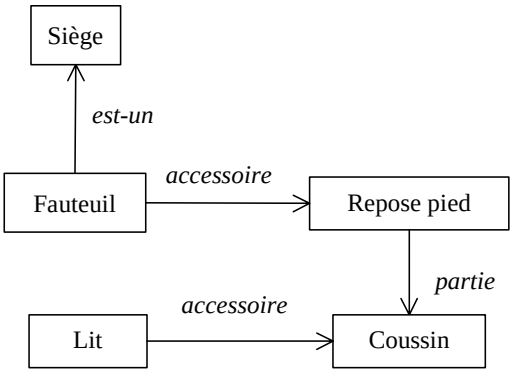


FIG.5.2.– Un exemple de réseau sémantique décrivant un mobilier

Le filtrage est un mécanisme de recherche de tous les sous-graphes du graphe qui ont une structure commune avec un graphe cible. Dans le cadre d'un système de recherche d'information, la requête est modélisée sous forme d'un réseau sémantique dans lequel les connaissances inconnues sont exprimées par des variables. Le système met en correspondance ce réseau requête et une partie du réseau contenant l'ensemble des connaissances afin de déduire les valeurs de ces variables. Il peut faire appel à l'héritage pour récupérer des informations des concepts généraux et les utiliser dans les nœuds plus spécifiques.

5.3.2 Les Graphes Conceptuels

Les deux familles de représentation par réseaux qui sont les plus utilisées à l'heure actuelle en IA sont les Graphes Conceptuels (CG), initialement proposés par *(Sowa, 1984)* et les nombreux dérivés de KL-ONE.

Pour présenter les graphes conceptuels, nous nous basons sur les détails donnés dans *(Chein & Mugnier, 1992)*. Le support gère l'ensemble des graphes conceptuels portant sur un même domaine de connaissance. Un support se compose :

– D'une hiérarchie *Tc* de types de concepts organisés par une relation de spécialisation notée $\preceq$, dans un treillis. Un exemple de treillis de types de concepts peut contenir le type 'Automobile' qui spécialise le type 'Véhicule' et inversement 'Véhicule' généralise 'Automobile' ('Automobile' $\preceq$ 'Véhicule'). Un treillis possède un plus grand élément appelé le type universel, noté T et un plus petit élément, le type absurde, noté $\perp$.

– D'un ensemble *Tr* de types de relations composé de plusieurs hiérarchies de types de relations de même arité (ayant le même nombre d'arguments). Chaque hiérarchie est organisée en treillis par la relation de spécialisation $\preceq$. Les arguments de chaque type de relations r_i de *Tr* sont numérotés et obéissent à des contraintes de typages représentées dans la signature de r_i.

– D'un ensemble *M* de marqueurs. Un marqueur identifie un individu de la base de connaissances. On dispose également d'un marqueur générique noté * qui représente un individu non spécifié.

– D'une relation de conformité entre les marqueurs et la hiérarchie des concepts. Tout marqueur est conforme au type universel et aucun au type absurde. Si un marqueur est

conforme à un type t, il est aussi conforme à tous les types généralisant t. Si un marqueur est conforme à deux types t et t', il est aussi conforme au type spécialisant t et t' ($t \cap t'$).

Un graphe conceptuel est un multigraphe, composé de deux sortes de nœuds : les nœuds concepts, aussi appelés sommets concepts ou plus sommairement concepts, et les nœuds relations ou relations. Chacun de ces nœuds a une étiquette. Un nœud concept est étiqueté par un type correspondant à une classe sémantique, et un marqueur précisant une instance particulière de la classe. Par exemple, le nœud concept étiqueté par 'Automobile : *' représente une automobile en général. Ce genre de nœud est qualifié de concept générique. Le nœud concept étiqueté par 'Automobile : 6793 SY 69' représente une voiture particulière dont le numéro d'immatriculation est 6793 SY 69. Ce nœud est qualifié de concept individuel. Les relations spécifient les rapports entre les concepts. Dans les graphes conceptuels, un concept est appelé un argument de la relation. Les nœuds relations sont aussi étiquetés par un type.

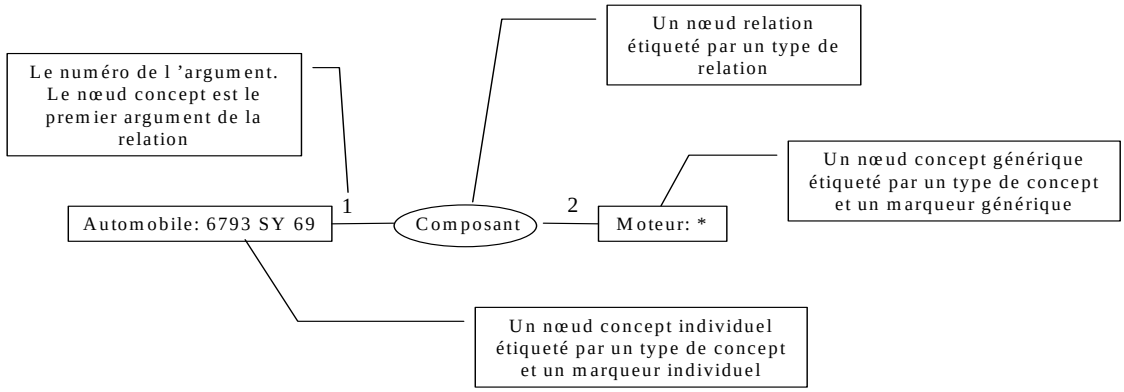


FIG.5.3.– Un graphe conceptuel décrivant une automobile

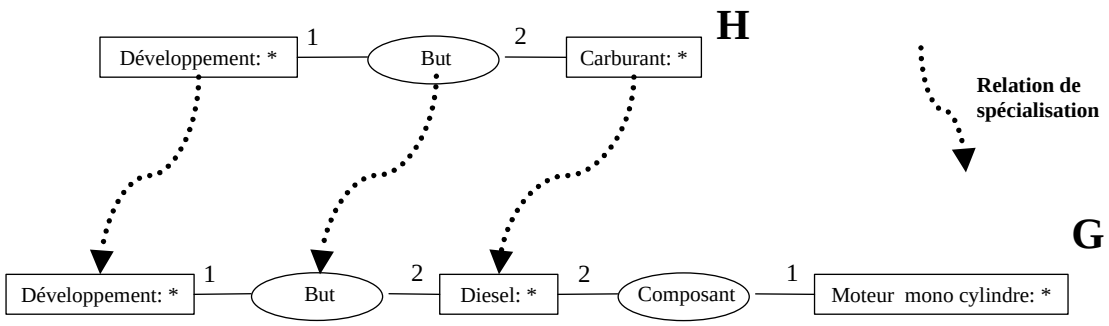


FIG.5.4.– Exemple de projection

Pour la recherche d'information, la relation de spécialisation présente un intérêt majeur car elle contribue à la comparaison de graphes. En effet, pour les comparer, Sowa a proposé un opérateur de projection qui s'appuie sur l'existence d'une relation de spécialisation entre deux graphes. Il existe une projection d'un graphe H dans un graphe G si le graphe G reprend de manière plus spécifique l'ensemble des concepts présents dans le graphe H. Dans ce cas, G est dit une spécialisation de H, $G \preceq H$.

De plus, Sowa a défini un opérateur ϕ qui permet de transformer un graphe conceptuel en une formule logique du premier ordre et il a montré que l'existence d'une projection d'un graphe H dans un graphe G était conditionnée par l'implication de leurs formules logiques associées :

$G \preceq H \Rightarrow \phi(G) \rightarrow \phi(H)$. Cette implication fait des graphes de Sowa un modèle adapté pour construire des systèmes de recherche d'information *(Chevallet, 1992)*. En effet, s'il existe une projection d'un graphe H dans un graphe G représentant respectivement une requête et l'index d'un document, elle se traduira par l'implication $\phi(G) \rightarrow \phi(H)$ qui rend le document pertinent pour la requête.

5.3.3 Les Langages de Frames

Dans les langages de frames, les connaissances sont regroupées en paquets. Ainsi les relations d'un concept du domaine avec les autres concepts font partie de la description de ce concept. Dans les réseaux sémantiques, il existe une unité sémantique dénotant un concept (le représentant), et un graphe composé d'autres unités sémantiques décrit la définition de ce concept. Dans un frame, l'unité sémantique, dénotant le concept, est confondue avec sa description, c'est à dire qu'une seule unité sémantique contient toute la description du concept et est aussi utilisée pour représenter ce concept.

Le premier formalisme informatique a été proposé par *(Minsky, 1975)*. Le frame correspond à une structure dynamique représentant des situations prototypiques. *(Schank & Abelson, 1977)*, qui s'intéressent à la représentation de séquence d'évènements et la compréhension d'histoires, aboutissent au même résultat avec la théorie des scripts.

Un frame est un objet typique d'une famille, représentant idéalement cette famille. Le frame contient donc des informations générales valides pour tous les membres de la famille, ainsi que des informations spécifiques à certains membres. Un frame est composé d'un ensemble d'attributs (ou slots) correspondant aux différentes propriétés du prototype. Un attribut est décrit par un certain nombre de facettes ou rôles possédant des valeurs. Par exemple, une facette définit le type de l'attribut, une autre la valeur courante ou par défaut de cet attribut. Les facettes procédurales permettent d'associer aux attributs des procédures appelées réflexes (par exemple une procédure peut être déclenchée pour calculer la valeur d'un attribut). Une des notions centrales des langages de frame est la notion de spécialisation, car elle permet non seulement l'ordonnancement de ces frames dans une hiérarchie mais aussi l'héritage des attributs et de leurs valeurs. Ainsi la description d'un frame peut être incomplète, les couples attribut-valeur hérités ne sont pas recopiés dans le frame. Un frame peut être spécialisé par enrichissement, en le dotant de nouveaux couples attributs valeurs, ou par substitution, en masquant certaines facettes des couples attribut-valeur hérités.

Une évolution des frames de Minsky a permis de distinguer deux types de frames : les frames classes et les frames instances. Les frames classes représentent les catégories d'objets du monde modélisé alors que les frames instances représentent des individus particuliers. La hiérarchie structurant les frames classes les uns par rapport aux autres, par la relation de spécialisation, est appelée une taxinomie. En plus de cette relation de spécialisation, des systèmes de frames proposent implicitement ou explicitement la relation d'instanciation (intitulée est-un dans le cas où elle existerait de façon explicite), qui relie une instance à sa classe d'appartenance, comme le montre la figure suivante :

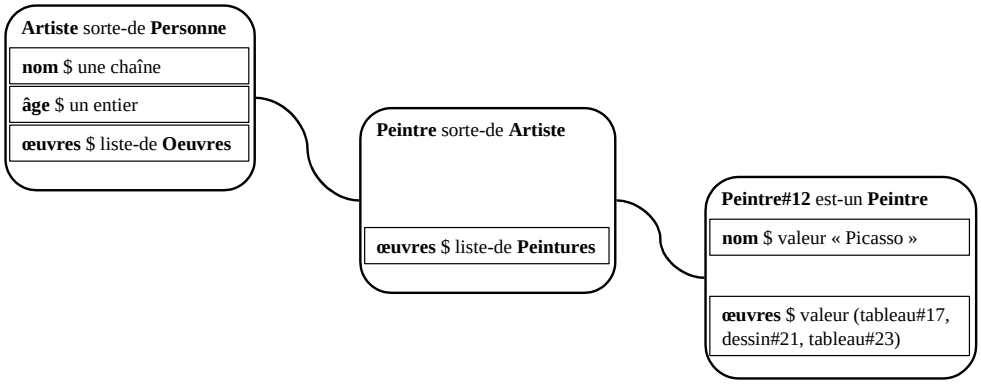


FIG.5.5.– Exemple de spécialisation de frames

Un système de frame sert à comparer des objets que l'on veut reconnaître ou classer. Deux mécanismes de raisonnement sont disponibles :

– Le filtrage consiste à rechercher, parmi un ensemble de frames ceux qui correspondent à des critères donnés. Il repose sur des mécanismes d'héritage et d'appariement.

– La classification consiste à intégrer un nouveau frame dans une hiérarchie établie. Le réajustement correspond à la modification de la position d'un frame mal placé.

Dans le cadre de la recherche d'information, un document est représenté par un frame, qui contient aussi une description de son contenu. La requête spécifie certaines caractéristiques que doit remplir le frame document. La fonction de comparaison effectue donc un filtrage sur l'ensemble des frames pour identifier tous les frames documents correspondant aux critères de la requête.

5.3.4 La Représentation des Connaissances par Objet

Dans le cadre de la représentation des connaissances par objets (RCO), la fonction d'un système de RCO est de stocker et d'organiser les connaissances autour de la notion d'objet et de fournir des services inférentiels destinés à compléter l'information disponible. Un système de RCO s'appuie sur une hiérarchie de classes liées entre elles par une relation de spécialisation. Une classe a une identité, un état et un comportement à la manière d'un type abstrait de données. Elle regroupe un ensemble d'instances qui ont chacune une identité et un état propres et un comportement décrit par la classe. La hiérarchie des classes est exploitée pour résoudre des problèmes par l'intermédiaire de procédures ou de mécanismes de raisonnement comme la classification de classes ou la classification d'instances (approche déclarative). La classification de classes consiste à placer une nouvelle classe dans une hiérarchie tandis que la classification d'instances cherche à déterminer les classes dont un objet donné peut être une instance. La classification s'appuie sur le texte de spécialisation qui consiste à vérifier qu'une classe donnée est plus générale qu'une autre classe.

Le processus de classification opère sur la hiérarchie de classes et cherche à mettre en évidence les dépendances implicites qui y existent. Le raisonnement par classification s'appréhende comme une procédure de déduction opérant sur cette hiérarchie en s'appuyant sur

les trois étapes d'initialisation de l'objet à classer, classification et exploitation de la classification.

Les systèmes de RCO partagent de nombreuses caractéristiques avec les logiques de description. Ces logiques s'appuient sur les notions de concepts (ils correspondent aux classes d'individus) de rôles (relations entre concepts) et d'individus (instances des concepts). Les concepts possèdent une syntaxe et une sémantique et sont organisés une hiérarchie par l'intermédiaire d'une relation de subsomption. La classification d'instances et la classification de classes sont à la base du raisonnement terminologique.

5.4 Quelques Projets du Web Sémantique

Les applications du Web sémantique se basent toujours sur un langage de représentation de connaissances pour manipuler les connaissances associées au document. Mais de manière générale, les chercheurs préfèrent souvent développer de nouveaux langages plus directement adaptés à leurs besoins, plutôt que d'utiliser des formalismes déjà existants mais qui ne correspondent pas exactement à leurs aspirations. Dans cette section nous en présentons quelques-uns. Une des caractéristiques de ces nouveaux langages est que leurs expressions puissent être directement intégrées à l'intérieur des documents du Web.

Les différentes approches du Web sémantique consistent à organiser le Web en domaines d'intérêts. Chacun de ces domaines est caractérisé par une ontologie qui permet de spécifier le vocabulaire de ce domaine. Les ontologies prennent différentes formes : de la base lexicale Wordnet *(Miller, 1990)* utilisée dans Ontoseek *(Guarino et al., 1999)* à une modélisation formelle du domaine comme dans WebKB *(Martin & Eklund, 2000)*. L'ontologie est utilisée par la communauté de personnes se reconnaissant du domaine d'intérêt pour échanger des connaissances et des documents.

Les approches du Web sémantique se différencient par :

– Le niveau de formalisation du langage de représentation des connaissances. Le langage peut être plus ou moins complet. RDF par exemple n'oblige pas à différencier une classe d'une instance. SHOE *(Heflin et al., 1999)* a la possibilité de décrire des règles d'inférences sous forme de clauses de Horn. L'inconvénient majeur des langages dits formels réside dans la complexité de l'interface à mettre en place pour composer une requête.

– La différenciation entre le symbole et la notion représentée par le symbole. Ontoseek *(Guarino et al., 1999)* indexe les documents par des synsets et non par des termes pour lever les ambiguïtés terminologiques.

– La déclaration des connaissances : les connaissances sont déclarées dans les documents *(Heflin et al., 1999) (Fensel et al., 1998)* ou dans des fichiers externes aux documents *(Michard, 2001) (Buckingham et al., 2000)*.

– L'objectif du système à base de connaissances : s'agit-il de déterminer des connaissances *(Martin & Eklund, 2000) (Fensel et al., 1998)* ou de rechercher des documents *(Guarino et al., 1999) (Heflin et al., 1999) (Buckingham et al., 2000)*.

– L'utilisation des connaissances : les connaissances expriment le contenu du document ou donnent un point de vue de lecture différent sur le document comme dans ScholOnto *(Buckingham et al., 2000)*.

– La prise en compte de l'existence de plusieurs ontologies pour un même domaine et la mise en place d'une solution pour interagir entre les différentes ontologies *(Heflin et al., 1999)*.

5.4.1 Le Langage SHOE

SHOE (Simple HTML Ontological Extensions) *(Heflin et al., 1999)*, un des projets précurseurs du Web sémantique, est une extension du langage HTML pour inclure des connaissances à l'intérieur des documents. SHOE contient deux types de balises, les balises définissant les ontologies et celles qui déclarent les instances. Une ontologie définit les éléments valides pour décrire les instances. Ces éléments sont :

– Les catégories organisées en hiérarchie par la relation de spécialisation.

– Les relations qui peuvent exister entre instances ou qui décrivent les propriétés d'une instance.

– Les règles permettant de renommer des éléments empruntés à d'autres ontologies.

– Les règles d'inférences définissant des inférences sous forme de clauses de Horn.

– Les constantes.

– Les types de données.

Pour SHOE, une page Web est considérée comme une instance. Afin de spécifier les relations entre instances contenues dans une page, chaque page Web est associée à une ontologie préalablement définie. Les pages Web définissent en fait des croyances et non des connaissances absolues ; par conséquent, un utilisateur peut spécifier son degré de confiance pour une source information.

Plusieurs technologies ont été mises en œuvre pour démontrer l'efficacité de ce langage :

– Un outil d'annotation permet aux auteurs de documents HTML d'enrichir leurs pages en incluant des connaissances exprimées à l'aide du langage SHOE.

– Un robot intitulé Exposé cherche sur le Web des pages annotées en langage SHOE. Ensuite, Exposé analyse les pages Web pour identifier les connaissances contenues dans les pages web. Ces connaissances, qui peuvent être des ontologies ou des instances, sont enregistrées dans un système à base de frames, intitulé PARKA *(Evett et al., 1993)*.

– Un outil interrogant la base de connaissances construite par Exposé et visualisant les pages Web résultats. Les requêtes spécifient les attributs ou propriétés que doit avoir un frame instance. La page Web identifiant le frame résultat est ensuite affiché.

SHOE fournit un mécanisme permettant de modifier des ontologies déjà existantes en spécialisant par exemple sa hiérarchie, ce qui veut dire que des parties des ontologies sont dispersées sur le web. Cette dispersion peut aussi bien entraîner une redondance qu'une absence d'information. Plusieurs applications ont été construites dont une ontologie modélisant le domaine des départements d'informatique dans les universités. Elle est cependant très simple.

5.4.2 Ontoseek

Ontoseek *(Guarino et al., 1999)* est un système de recherche de pages Web utilisant le modèle des graphes conceptuels de Sowa. Le contenu des pages Web et les requêtes est

représentés sous forme de graphes conceptuels. La fonction de comparaison est basée sur l'opérateur de projection de Sowa et recherche les spécialisations des nœuds de la requête. Ontoseek exploite la base lexicale Wordnet *(Miller, 1990)* pour lever les ambiguïtés des termes utilisés comme étiquette des nœuds des graphes. Ainsi, les étiquettes des nœuds ne sont plus des termes mais des synsets donnant lieu au modèle des Graphes Conceptuels Lexicaux (GCL). Une des applications de ce système a été de travailler sur l'annuaire des pages jaunes et pour l'expansion de requêtes sur des catalogues de produits. Les graphes indexant ces pages Web ainsi que les requêtes sont construits manuellement. L'utilisateur peut choisir les composants de son graphe en naviguant dans l'ontologie ou, à partir d'un terme, Ontoseek lui présente différents synsets et l'utilisateur sélectionne celui qui correspondant à son besoin. Des heuristiques sont utilisées pour valider la cohérence des relations. Cependant, Wordnet est une base lexicale et non une base de connaissances. Ce projet a montré l'intérêt d'utiliser une base lexicale pour lever les ambiguïtés terminologiques et ainsi améliorer la précision et le rappel du système de recherche. Cette méthode d'indexation des documents s'appuyant sur les concepts d'une base de données sémantique améliore la précision de 25% lors de la recherche.

5.4.3 WebKB

L'ensemble des outils de WebKB (WebKB set of tools) permet à ses utilisateurs de stocker, d'organiser et de retrouver des connaissances formalisées à l'aide du modèle des Graphes Conceptuels *(Martin & Eklund, 2001)*. Ces outils ne sont pas dédiés à la recherche documentaire car leur but n'est pas de retrouver des documents mais de retrouver à partir d'une requête l'ensemble des connaissances répondant à cette requête, stockées dans une base de connaissances. WebKB propose une série d'outils pour l'acquisition de connaissances, dont le but est de créer une base de connaissances illustrée par une documentation (l'ensemble des textes au format HTML) qui ont permis de valider les connaissances de la base. Les connaissances et les parties de documents associées sont toutes les deux représentées sous forme d'Elément de Document (ED). Un ED doit être accessible par le Web, en utilisant par exemple son URL. De plus, un ED est caractérisé par un contexte, constitué par son créateur et sa date de création. Les ED sont liés les uns aux autres par différents liens :

– Un lien hypertexte permet d'indexer une ED documentation à une ED connaissance représentant les connaissances déduites de l'ED documentation.

– Des liens sémantiques entre ED contenant des connaissances formelles (par exemple des liens de spécialisation entre deux graphes conceptuels).

WebKB comprend :

– Un outil de recherche sur la base de connaissances. Ainsi à partir d'une requête sous forme de graphe conceptuel sont retrouvés tous les graphes de la base répondant en partie à cette requête. Le graphe réponse peut être une spécialisation ou une généralisation d'un sous-graphe du graphe requête. D'autres relations sémantiques entre types sont prises en compte pour établir la pertinence d'un graphe par rapport à la requête (par exemple la relation "partie de"). Plusieurs contraintes sur la recherche peuvent être spécifiées, entre autres l'affichage des ED documentations associés aux ED connaissances retrouvés.

– Un outil d'indexation des ED permet d'indexer des ED documentation par des ED connaissance et d'associer des ED connaissances par des liens sémantiques qui expriment la relation de subsomption "sorte de" ou la relation d'appartenance "partie de".

– Un éditeur de connaissances permet de créer des graphes conceptuels. Ces graphes peuvent être représentés à l'aide de plusieurs langages : le langage défini par Sowa, un langage proche du langage naturel, le langage Frame-CG qui permet une représentation des graphes sous forme de frames. WebKB a pour but de représenter le plus précisément possible les connaissances, c'est pourquoi le langage de Sowa a été amélioré par l'ajout de plusieurs quantificateurs (tout, un, quelque, 96%, etc.)

– Un navigateur de hiérarchie permet de rechercher différentes catégories. Une catégorie peut être un type de concepts, un type de relations, un marqueur. Ainsi une différence est faite entre le niveau lexical (terminologique) et le niveau conceptuel. WebKB, afin de guider le cogniticien dans sa phase de modélisation du domaine, propose plusieurs hiérarchies de base (hiérarchie de concept, de relation) construites à partir de la base lexicale Wordnet.

Ces outils intègrent dans des documents HTML des commandes permettant de construire des connaissances mais non destinés à la recherche documentaire sur le Web, car les requêtes portent sur des connaissances précises et non sur le contenu global des documents.

5.4.4 CWeb

Dans le cadre du Web sémantique, le Consortium du World Wide Web (W3C) chargé de développer des technologies pour le Web, a validé une application du format XML [XML] pour la description du contenu sémantique, appelé RDF pour Ressource Description Framework *(Lassila & Swick, 1999)*. RDF est un formalisme basé sur un modèle de graphes étiquetés orientés. RDF décrit le graphe sous la forme d'un ensemble de triplet {Propriété, Ressource, Valeur}. Les ressources sont des entités d'informations pouvant être référencées par un nom symbolique ou un identifiant, par exemple un URI (Unique Ressource Identifier) *(Berners-Lee et al., 1998)*. Les Propriétés sont les étiquettes des arcs orientés reliant un premier nœud étiqueté par une Ressource à un second qui peut être soit une valeur atomique soit une autre ressource.

Par exemple, considérons les triplets suivants :

ftp://foo.bar.com/srdf.XML → dc:Creator → "John Smith"

ftp://foo.bar.com/srdf.XML → dc:Date → "1998-06-18"

ftp://foo.bar.com/srdf.XML → ExempleOf → [http://loc/xmlex23.RDF]

Cet ensemble de triplets peut être spécifié dans un même élément descriptif RDF :

```
<rdf:Description about="ftp://foo.bar.com/srdf.XML" >
   <dc:Creator> John Smith </dc:Creator>
   <dc:Date > 1998-06-18</dc:Date>
< ExempleOf  resource=" http://loc/xmlex23.RDF "/>
</rdf:Description>
```

RDF définit uniquement une syntaxe particulière pour représenter un graphe. Par conséquent, différents vocabulaires peuvent être associés à cette syntaxe pour définir des langages de description de pages web. De ce fait, un vocabulaire comprend l'ensemble des propriétés, ressources ou valeurs qui peuvent être utilisés pour décrire le contenu des documents. Par exemple, le Dublin Core *(Baker, 2000)* a proposé un ensemble de 15 propriétés (Title, Creator etc.) permettant de décrire les documents accessibles sur le Web par des métadonnées. Le vocabulaire d'un domaine spécifique est décrit à l'aide d'un Schéma RDF

[RDFS]. Entre autres, RDFS permet aussi de différencier les ressources représentant les classes de celles représentant les individus.

Plusieurs projets de recherche documentaire sur le Web utilisant le formalisme RDF/RDFS pour décrire le contenu du document. Le projet européen Community-Webs[38] propose une architecture pour interroger des ressources distribuées et hétérogènes décrites dans une base indépendante de documents RDF. La description d'un document XML en RDF peut tout aussi bien être incluse dans le document que contenue dans un autre document XML. Cette base RDF est interrogée à l'aide du langage de requête RDF Query Language (RQL) *(Michard, 2001)* proche de SQL. Par conséquent, les inférences sur les connaissances sont totalement décrites dans la requête RQL, donc la personne chargée d'interroger la base doit au préalable connaître le modèle des connaissances sous jacent à la base RDF pour composer des requêtes valides.

5.4.5 OCML

Les travaux de *(Motta et al., 2000)* concernent la gestion des connaissances. Le but est d'améliorer la recherche d'information en modélisant formellement un point de vue sur le domaine considéré partagé par les intervenants dans ce domaine. Les connaissances permettent de : définir le contexte dans lequel les documents ont été écrits (projet PlanetOnto *(Domingue & Motta, 2000)*), vérifier la complétude des informations décrites dans des documents médicaux (projet PatMan *(Motta et al., 2000)*), ou établir des liens non explicites entre différents documents (projet ScholOnto *(Buckingham et al., 2000)*. Le but n'est pas d'exprimer le contenu des documents mais de localiser les points d'intérêt du document dans un modèle de connaissances représentant un point de vue partagé sur le domaine. Le modèle est indépendant de la base documentaire et les connaissances ne sont pas incluses dans des documents. Leur interprétation n'est pas figée et dépend du modèle de connaissances considéré.

Le point de vue d'un domaine d'étude est formalisé dans une ontologie. Cette ontologie est instanciée pour construire un modèle conceptuel du domaine et la recherche documentaire s'effectuera en interrogeant ce modèle.

Plusieurs technologies ont été développées pour la mise en œuvre d'un système complet.

– Un langage de modélisation des connaissances intitulé OCML a été défini *(Motta, 1998)*. C'est un langage de frames qui permet de spécifier des relations, des fonctions, des classes, des instances, des règles de contrôle sur les structures de représentation. Il permet d'évaluer ou d'inférer des connaissances.

– WebOnto est un outil d'édition et de gestion, à partir du Web, des ontologies et des modèles conceptuels basés sur OCML. WebOnto permet une gestion collaborative des modèles conceptuels.

– Lois est une interface d'interrogation des modèles conceptuels basée sur les formulaires. Lois s'adapte au modèle et permet de parcourir les connaissances décrites dans chaque modèle pour établir une requête.

– Knote est un outil permettant d'instancier les connaissances du modèle pour enregistrer un document, mais aussi pour compléter le modèle conceptuel avec de nouvelles classes. Knote présente, sous forme de formulaire, les classes à instancier.

[38] http://cweb.inria.fr/

5.4.6 SyDOM

Il est à noter que tous les projets décrits se sont concentrés sur la prise en compte des connaissances pour améliorer la recherche sur le Web mais ils ne se sont pas encore intéressés à l'aspect multilingue du Web *(Roussey et al., 2001)*. Ils prennent en considération la présence de plusieurs langues sur le Web, en séparant le niveau terminologique du niveau sémantique dans un langage de représentation des connaissances. Ils proposent un nouveau langage de représentation des connaissances, intitulé Graphe Sémantique (GS), pour mettre en oeuvre cette différenciation dans le modèle des Graphes Conceptuels.

Les Graphes Sémantiques sont utilisés pour représenter les connaissances indexant les documents. Pour faciliter la maintenance de leur base de connaissances et sa cohérence avec la base documentaire, les connaissances sont intégrées aux documents. Les connaissances, incluses dans un document, peuvent être relatives à son contenu ou bien exprimer un point de vue sur ce document.

Le but des systèmes basés sur leur méthode d'indexation est de rechercher des documents appartenant à un corpus multilingue. C'est pourquoi ils proposent également un nouveau modèle d'ontologie intitulé thésaurus sémantique, permettant à des utilisateurs de langues différentes d'appréhender la même conceptualisation du domaine. Dans le but d'améliorer la qualité de leur thésaurus sémantique, leur méthode d'indexation met à jour le vocabulaire du thésaurus, à partir du vocabulaire du corpus de recherche. L'indexation est cependant manuelle.

5.4.7 De Ontobroker à OIL

Le système Ontobroker *(Fensel et al., 1998)* pour objectif de représenter formellement les connaissances contenues dans les documents HTML afin de les interroger de manière précise. Dans cette approche, le Web est considéré comme une base de connaissances distribuée, non structurée, et non exprimée car implicite. Leur but est de réutiliser et partager ces connaissances. Tout d'abord, un groupe d'utilisateurs, appelé ontogroup, voulant mettre en commun leurs connaissances et leurs informations s'accorde sur la définition d'une ontologie. L'ontologie définit non seulement leur vocabulaire commun mais stocke et organise les connaissances à l'aide du langage Frame Logic *(Kifer et al., 1995)* sous la forme de frames (les classes = [classe, attribut, type de valeurs] et les instances = [instance, attribut, valeur]) et de règles.

Dans ce système, les documents sont considérés comme des instances de classe. Les connaissances sont explicitement représentées dans les pages HTML. Ainsi, à une partie de texte, est ajoutée la description formelle de sa sémantique. Les connaissances exprimées sont les valeurs des attributs de l'instance considérée (la page Web). Le texte des documents ou les liens peuvent être réutilisés comme valeur d'un attribut. L'explicitation de ces connaissances peut être faite automatiquement dans le cas de document HTML fortement structuré à l'aide d'un wrapper, un médiateur entre la structure du document et l'ontologie ; ou manuellement en annotant le document. Un langage d'annotation a été défini. Il s'agit d'une extension des ancres HTML définissant un nouvel attribut, intitulé onto. Le fait d'expliciter les connaissances près de leur source d'information dans le document lui-même, permet de faciliter la maintenance de la base de connaissances, seule la mise à jour du document est nécessaire.

Ontobroker se décompose en trois éléments :

– Un outil d'interrogation de la base de connaissances. Les requêtes sont des expressions du langage Frame Logic. Les requêtes sont une conjonction d'expressions du type : une instance O d'une classe C a pour attribut A, la valeur V. Les variables O, C, A, V peuvent être remplacées par des constantes ou des expressions. Une interface sous forme de frame a été développée pour générer des requêtes de manière plus conviviale. Le résultat de ces requêtes sont des valeurs qui ne sont pas forcement des références à des documents.

– Un moteur d'inférence. Ce moteur traduit les expressions Frame Logique en formule logique du premier ordre pour pouvoir faire des inférences. Les inférences permettent d'améliorer la cohérence de l'ontologie.

– Un webcrawler parcourant le Web pour intégrer de nouvelles connaissances. Celles-ci étant stockées dans la même base.

Pour rendre le système Ontobroker compatible avec de nouvelles approches, un traducteur RDF a été développé pour représenter les connaissances liées au document sous forme d'une description RDF.

Ontobroker est à la base de deux projets : Knowledge Acquisition initiative (KA)² *(Benjamins & Fensel, 1998)* et Ontoknowledge[39]. Le projet (KA)² développe une ontologie propre aux chercheurs de la communauté de l'acquisition des connaissances. Cette ontologie est utilisée pour stocker toutes les connaissances relatives aux laboratoires, projets, chercheurs, publications du domaine. Elle est utilisée en exemple dans Ontobroker. Ontoknowledge est un projet soutenu par la commission européenne visant à regrouper des chercheurs travaillant sur le Web sémantique. Un des objectifs de ce projet consiste à définir un nouveau langage de représentation des ontologies Ontology Interchange Layer (OIL) *(Fensel et al., 2000)* basé sur les schémas RDF.

Ontobroker n'est pas un outil de recherche documentaire, mais un outil de recherche de connaissances. Son but n'est pas de représenter le contenu des documents pour améliorer la recherche de document, mais de représenter les connaissances décrites dans les documents pour améliorer la recherche de connaissances.

5.5 Le CISMeF dans l'Infrastructure du Web Sémantique

5.5.1 Représentation des Métadonnées

Certaines ontologies, dites ontologies de représentation (Schreiber *et al.*, 1997) et constituant des métadonnées c'est-à-dire des données sur les données, tendent déjà à devenir des standards pour la description de documents. Le concept de métadonnées est apparu bien avant l'avènement de l'Internet et son intérêt a augmenté avec le nombre de documents électroniques. L'utilisation de métadonnées est d'ailleurs une solution recommandée par le W3C pour la description et la publication des documents sur le Web. Parmi les standards utilisés, le Dublin Core, constitué de quinze éléments de description de ressources en ligne, et l'IEEE LOM, qui reprend quelques primitives de description proches de celles du Dublin Core mais qui vise la description explicite de documents pédagogiques.

[39] http://www.ontoknowledge.org/

Dans le cadre du CISMeF, l'approche ontologique est utilisée pour définir l'ensemble des informations relatives aux documents : informations sur le type du document, sur son auteur, sur ses conditions d'usage, sur l'intérêt de sa consultation, sur le contexte dans lequel il peut être employé, mais également sur sa qualité, s'il provient d'un site producteur agréé…etc. Pour ce faire, plusieurs ensembles de métadonnées sont utilisés, l'amélioration de la recherche d'information étant visée. Les documents répertoriés dans CISMeF sont décrites par onze des éléments du Dublin Core (*auteur*, *date*, *description, format, identifiant, langage, éditeur, type de ressource, droits, sujet* et *titre*) ainsi que huit autres éléments spécifiques à CISMeF (*institution*, *ville*, *province ou département*, *pays*, *public ciblé*, *type d'accès*, *coût* et *parrainage* de la ressource). (§Chapitre 2)

Le type d'utilisateur est également pris en compte. Pour les ressources destinées aux professionnels de la santé (les lignes directrices et consensus de bonne pratique clinique) deux champs supplémentaires sont définis par CISMeF : *indication du niveau de preuve*[40] et la *méthode* utilisée pour le déterminer. Pour les ressources pédagogiques ce sont onze éléments de la catégorie « Educational » du standard IEEE 1484 qui sont rajoutés. Le format de ces méta-données est passé du langage HTML en 1995 à XML en 2000 pour des soucis d'interopérabilité dans la plateforme Archimède dans le cadre du projet UMVF (Université Médicale Virtuelle Francophone) et depuis décembre 2002 à RDF dans le cadre du projet européen MedCIRCLE *(Mayer et al., 2003)*. Ce projet a été initié afin de qualifier la qualité de l'information de santé, et de guider les utilisateurs vers des sites d'information en santé de confiance. Le vocabulaire des méta-données HIDDEL[41] est contenu dans une ontologie (représentée à l'aide du langage RDFS) et les documents sont décrits en RDF à partir des concepts de l'ontologie HIDDEL. Par exemple l'item *Infoprovider_feedback_email_technical* désigne l'email du fournisseur de l'information pour les questions techniques et l'item *Infoprovider_feedback_email_content* pour les questions de contenu.

5.5.2 La Terminologie CISMeF : entre Terminologie et Ontologie

La terminologie CISMeF a été construite à partir des concepts du thésaurus MeSH et de sa traduction en français fournie par l'INSERM. Le MeSH dans sa version 2004 est composé d'environ 22,000 *mots clés* et 84 *qualificatifs* regroupés sous la forme d'arborescences. Les mots clés correspondent à des concepts médicaux et sont organisés sous la forme de hiérarchie à 11 niveaux allant du terme le plus général en haut de la hiérarchie aux termes les plus spécifiques en bas de la hiérarchie. Les qualificatifs, organisés également en hiérarchie, permettent de préciser le sens des mots clés en limitant leur étendue à certains aspects. Les fichiers MeSH sont traités automatiquement *(Dahamna & Soualmia, 2002)* pour renseigner la terminologie CISMeF afin qu'elle soit exploitable au niveau du site.

Bien qu'il existe des ontologies médicales générales, comme GALEN *(Rodrigues et al. 1998)*, ou spécifiques à un domaine comme MENELAS *(Bouaud et al., 1995)*, c'est le MeSH qui a été choisi car il correspond aux attentes des documentalistes et il est connu des professionnels de santé. Les mots clés ont été regroupés dans CISMeF en fonction de spécialités médicales intitulés *métatermes*. Ce sont des super-concepts qui permettent une vision plus globale concernant une spécialité. Les métatermes permettent de connaître l'ensemble des termes

[40] Les Guides de bonne pratique, l'evidence based medicine (medecine factuelle) consistent à baser des décisions cliniques sur les connaissances théoriques mais également des preuves scientifiques déduites par des recherches cliniques qui fournissent des résultats valides.
[41] High Information Description Disclosure Evaluation Language, http://www.medcircle.org

MeSH qui sont répartis dans plusieurs arborescences mais qui concernent une même spécialité. Une hiérarchie de *types de ressources* a été modélisée et elle permet de décrire la nature de la ressource.

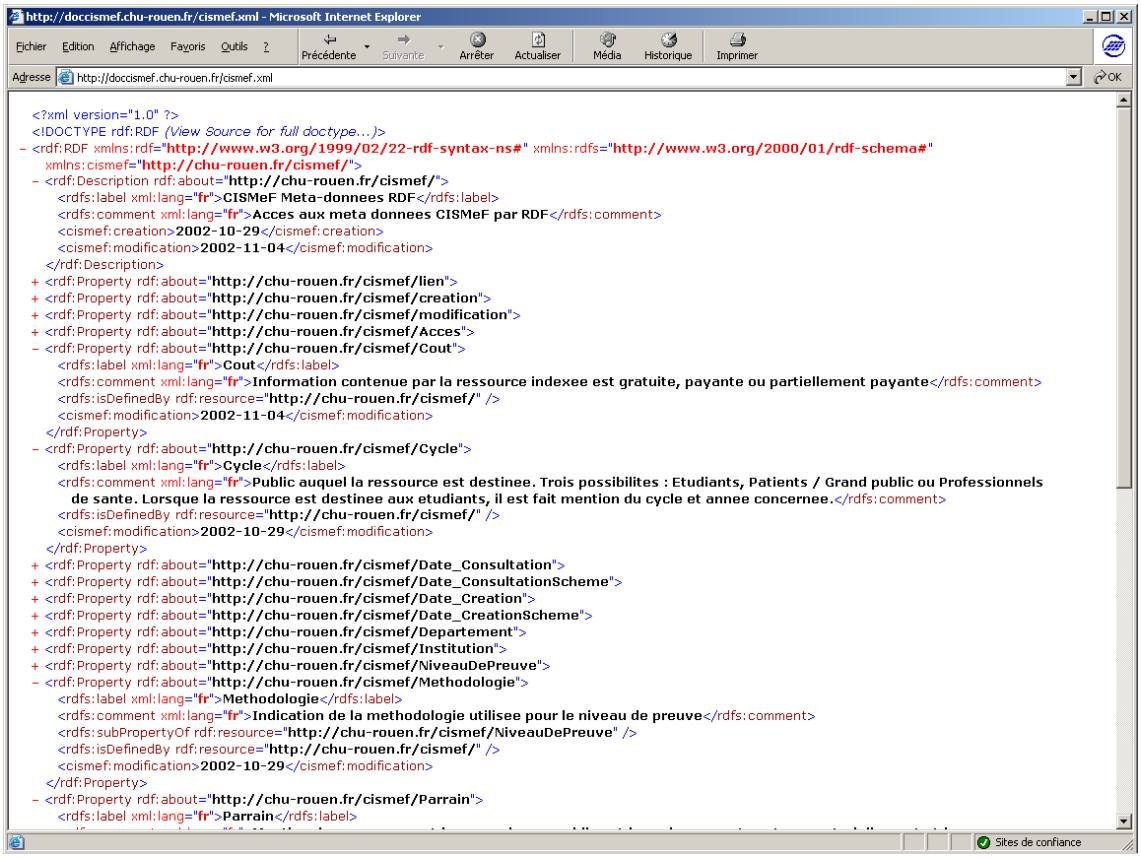


FIG.5.6.– Métadonnées CISMeF sous le format RDF

La terminologie CISMeF a une structure semblable une ontologie terminologique *(Sowa, 2000) (Desmontils & Jaquin, 2002)* :

– Le vocabulaire est bien connu des documentalistes et des professionnels de la santé et il correspond à celui du domaine médical.

– Chaque concept a :

  – un terme préférentiel (Descripteur) pour l'exprimer en langue naturelle

  – un ensemble de propriétés

  – une définition en langage naturel pour quelquefois le différencier des concepts le subsumant et de ceux qu'il subsume (principe de *(Bachimon, 2000)*)

  – un ensemble de synonymes

  – un ensemble de règles et de contraintes

Les concepts sont organisés selon une relation de subsomption allant du concept le plus général au plus spécifique.

La Figure 5.7 est un exemple de contraintes sous la forme de règles à appliquer sur les concepts. Par exemple l'association *hépatite/induit chimiquement* est équivalente (=) au

concept hépatite, toxique. En revanche, ces informations ne sont pas exploitées car elles ne sont pas disponibles sur un support électronique.

```
Hepatitis : C06.552.380+
Viral Hepatitis = Hepatitis, Viral Human and Hepatitis, Viral Animal
/chemically induced = Hepatitis, Toxic
/veterinary = Hepatitis, Animal or Hepatitis, Viral, Animal
hepatitis parenterally transmitted= Hepatitis C
hepatitis enterally transmitted = Hepatitis E
not specified as parenteral or enteral = probably Hepatitis, Viral, Human
Non-A, Non-B hepatitis = probably Hepatitis C
```

FIG.5.7.– Exemple de règles et de contraintes

Il manque à cette terminologie une dimension formelle. En effet, des thésaurus comme le MeSH et l'UMLS *(Lindberg et al., 1993)* ont été développés pour la recherche d'information mais leur numérisation ainsi que la disponibilité de ressources documentaires a fait naître de nouveaux besoins en recherche d'information avec des requêtes plus complexes demandant des inférences plus complexes. De ce fait, on envisage le remplacement de ces thésaurus par des ontologies ou des bases de connaissances formelles, les seules permettant de supporter des inférences *(Charlet, 2002)*.

Nous avons choisi de travailler sur les logiques de description, d'une part du fait du bon compromis entre le pouvoir d'expressivité (sa capacité à représenter les connaissances) et son efficacité. Par ailleurs, nous les avons déjà utilisées pour des applications comme la découverte de schéma dans des bases de données semi-structurées *(Hacid et al., 2000)* et les données XML, mais également pour le catalogage automatique de produits dans le cadre du commerce électronique *(Soualmia, 2001a) (Soualmia, 2001b)*.

Notre ambition n'est pas de construire une ontologie du domaine médical mais en revanche de formaliser la terminologie à notre disposition à des fins de recherche d'information. En effet, étant donné un domaine d'activité, il n'y a pas une terminologie, mais autant de ressources terminologiques ou ontologiques que d'applications qui les utilisent *(Aussenac-Gilles & Condamines, 2004)*. Un récapitulatif des ressources disponibles pour le domaine médical est dressé dans *(Charlet, 2003)*.

5.6 Les Logiques de Description

Les logiques de description, issues des recherches sur la représentation des connaissances, tiennent leur origine du langage KL-ONE. L'approche terminologique est caractérisée par la distinction entre les descriptions en intension et celles en extension. En effet, un système terminologique comporte au moins deux composants. Le premier appelé TBox, supporte le formalisme terminologique qui est utilisé pour décrire l'univers du discours en intension. Le deuxième composant, appelé ABox, relève du niveau factuel. Il permet de spécifier l'univers du discours à l'aide d'un ensemble d'assertions. Certains systèmes terminologiques supportent un troisième composant appelé IBox, qui est constitué d'un ensemble de règles d'inférences simples.

Au niveau intensionnel, l'univers du discours est décrit à l'aide d'un ensemble de termes et de restrictions sur ces termes. Un terme est une description structurée qui dénote un ensemble d'individus partageant des propriétés communes, ou des relations entre les individus. Dans le

premier cas, il est appelé concept et dans le deuxième cas rôle. Les restrictions sur les termes sont définies à l'aide de différents types de constructeurs. La description d'un univers de discours à l'aide d'un formalisme terminologique est appelée une terminologie.

Dans une terminologie, les concepts sont ordonnés selon une relation d'implication, appelée subsomption. Dans une relation de subsomption entre deux concepts, on appelle subsumant (super-concept ou super-rôle) le concept le plus général et subsumé (sous-concept ou sous-rôle) le concept le plus spécifique. Les logiques terminologiques disposent de deux types particuliers de concepts, le concept universel (Top) et le concept inconsistant (Bottom). De manière analogue, les rôles peuvent être organisés selon la relation de subsomption dans une hiérarchie de rôles délimitée par le rôle universel et le rôle inconsistant.

La structure hiérarchique des terminologies constitue la base des raisonnements disponibles dans les systèmes terminologiques. L'insertion par exemple d'un nouveau terme dans la hiérarchie des concepts ou des rôles, de manière à ce qu'il soit placé au-dessus de tous les termes qu'il subsume et en dessous de ceux qui le subsument est appelé classification. Ce type de raisonnement est spécifique aux systèmes terminologiques, d'où l'utilisation de l'appellation classifieurs pour les désigner *(Schmolze & Lipkis, 1983)*.

5.6.1 Les Formalismes Terminologiques

Les formalismes terminologiques sont utilisés pour construire les expressions qui décrivent les termes de l'univers du discours. Ces expressions apparaissent sous la forme d'équations dans lesquelles les noms des termes apparaissent en partie gauche et des descriptions complexes, formées à l'aide de constructeurs de termes, apparaissent en partie droite. De telles équations constituent les axiomes terminologiques de la TBox.

Deux types de termes sont distingués : les termes *atomiques* et les termes *composés*. Les premiers ne possèdent pas de description propre et apparaissent uniquement en partie droite des axiomes terminologiques. Les seconds sont associés à des descriptions conceptuelles définies à l'aide des constructeurs de termes. Ils apparaissent au moins une fois en partie gauche d'un axiome terminologique.

L'association des noms de termes aux descriptions peut être réalisée de deux manières telles que présentées ci-dessous :

– Les spécifications *primitives* introduites à l'aide de l'opérateur $\dot{\preceq}$, indiquent uniquement les conditions nécessaires d'appartenance aux extensions des termes : toute instance d'un terme doit vérifier la description de ce terme, en revanche, tout individu (ou couple d'individus) qui vérifie la description du terme considéré n'est pas obligatoirement instance de ce terme. Les termes ainsi introduits sont appelés termes primitifs.

– Les *définitions* introduites à l'aide de l'opérateur $\doteq$ permettent de spécifier les conditions nécessaires et suffisantes d'appartenance aux extensions des termes : chaque individu ou couple d'individus qui satisfait à la description du terme est une instance de ce terme. Les termes ainsi introduits sont appelés termes définis.

Par exemple : soit deux concepts Enseignant et Moniteur décrits à l'aide du constructeur de conjonction de concepts (noté $\wedge$)

$$\text{Enseignant} \doteq \text{Mâle} \wedge \text{Universitaire}$$

$$\text{Moniteur} \doteq \text{Etudiant} \wedge \text{Enseignant}$$

Le premier axiome spécifie que tout Enseignant doit être un Universitaire de sexe Masculin. Cependant tout individu qui vérifie cette condition n'est par obligatoirement un enseignant. Dans le deuxième axiome, le concept Moniteur est défini comme étant exactement l'ensemble des individus qui sont à la fois des étudiants et des enseignants. Cette condition est donc suffisante pour rattacher un individu donné à l'extension du concept Moniteur.

Les systèmes terminologiques supportent des formalismes terminologiques qui varient selon le domaine d'application et les objectifs assignés à ces systèmes.

Le développement des procédures de raisonnement dans les systèmes terminologiques exige l'assignation d'une sémantique formelle aux logiques terminologiques. La sémantique dénotationnelle est la plus utilisée dans ce domaine. Etant donné un ensemble d'objets (individus) dit univers du discours, les concepts sont alors interprétés comme des sous-ensembles de l'univers du discours et les rôles comme des relations entre ces sous-ensembles. Les ensembles ainsi définis constituent les extensions des concepts et des rôles respectivement. Les extensions sont assignées aux termes à l'aide d'une fonction d'interprétation qui fait correspondre à chaque terme son extension.

5.6.2 Assertions et Règles d'Inférence

Les connaissances factuelles sont exprimées dans la ABox du système terminologique à l'aide d'un ensemble d'axiomes appelés assertions. Elles sont introduites à l'aide d'un langage assertionnel. Ce type de langage permet par exemple de spécifier qu'un individu donné est une instance d'un concept ou que deux individus donnés sont reliés par un rôle. On appelle cette opération l'instanciation d'un concept ou d'un rôle.

Certains systèmes terminologiques utilisent la IBox pour exprimer des liens entre les concepts à l'aide de règles d'implication qui sont de la forme : antécédent $\Rightarrow$ conséquent. Généralement, l'antécédent et le conséquent sont constitués de descriptions quelconques de concepts, comme illustré par l'exemple :

$$\text{Moniteur} \wedge \text{Féminin} \Rightarrow \text{Etudiante}$$

La règle précédente considère tous les moniteurs de sexe féminin comme étant des instances du concept étudiante.

La IBox est complémentaire de la TBox dans le sens où elle permet d'exprimer des informations qui ne peuvent pas être représentées dans cette dernière. En effet, les informations représentées dans une IBox n'expriment pas des liens de définitions entre les concepts qui apparaissent dans les deux parties d'une règle : *le concept conséquent ne contribue pas à la définition du concept antécédent*. Dans l'exemple précédent, être moniteur et féminin ne constitue pas une condition obligatoire (nécessaire) pour l'appartenance à l'extension du concept étudiante.

5.6.3 Le Raisonnement Terminologique

Généralement, une grande partie des connaissances d'une base de connaissances est implicite : elle n'a pas été explicitement introduite dans la TBox ou dans la ABox. D'où la nécessité pour un système terminologique de fournir pour un système terminologique de fournir des mécanismes de raisonnement qui permettent d'inférer (expliciter) ces connaissances.

Deux types de raisonnements sont disponibles dans les systèmes terminologiques : la reconnaissance d'individu et le raisonnement sur les descriptions.

5.6.3.1 La Reconnaissance d'Individus

Il s'agit de vérifier si un individu donné est une instance d'un concept donné. Cette opération constitue la base du processus de réalisation qui consiste à retrouver pour chaque individu le concept le plus spécifique auquel il se rattache. La reconnaissance d'individus est un raisonnement hybride qui implique une interaction entre la TBox et la ABox.

5.6.3.2 Le Raisonnement sur les Descriptions

Il s'agit de détecter les différentes relations, telle que la subsomption qui existent entre terme. Deux types de raisonnements sont fondamentaux pour la construction et l'utilisation des terminologies :

– La subsomption qui consiste à vérifier si un terme est plus spécifique qu'un autre terme.

– La satisfiabilité d'un terme qui consiste à vérifier si la description d'un terme donné ne contient pas d'informations contradictoires.

Le test de subsomption est à la base du processus de classification qui consiste à déterminer automatiquement la position d'un concept (d'un rôle), dans la hiérarchie des concepts (des rôles). La classification revient à répondre à la question suivante : étant donné un nouveau terme introduit dans une terminologie quels sont les subsumant et ses subsumés les plus immédiats ? Elle permet donc de calculer toutes les relations de subsomption induites par l'introduction d'un nouveau terme dans une terminologie.

L'efficacité des procédures de raisonnement a été une des préoccupations majeures des travaux de recherche concernant les logiques terminologiques. Cet axe de recherche a été initié par Brachman et Levesque qui se sont intéressés à la complexité du calcul de la subsomption entre des expressions formées uniquement de concepts. Ce type d'inférence, connu sous le nom de subsomption pure est fondamental dans la majorité des systèmes terminologiques. Brachman et Levesque ont montré qu'il existe des algorithmes efficaces (polynomial en temps) pour vérifier le test de subsomption pure pour des formalismes simples. Ces algorithmes deviennent vite très coûteux (exponentiel en temps) pour de faibles extensions de ces formalismes.

Un résultat important a été la mise en évidence du rapport entre l'expressivité des formalismes terminologiques et la complexité des procédures de raisonnement : plus un formalisme terminologique est riche plus le raisonnement est complexe. De nombreux travaux sont venus confirmer ce résultat.

Indépendamment de la logique terminologique utilisée, *(Nebel, 1990)* a montré qu'il existe une autre source de complexité qui est inhérente à toutes les terminologies : l'opération d'expansion d'une terminologie, qui consiste à remplacer dans une description donnée les noms des termes composés par leurs descriptions ne peut pas être réalisée de manière efficace. Cette complexité est cependant purement théorique car les définitions qui peuvent augmenter la complexité du raisonnement apparaissent que très rarement dans des cas pratiques.

5.6.4 Expressivité et Complexité

Généralement, un compromis entre la puissance d'expression du formalisme et la complexité des mécanismes de raisonnement est recherché. Les résultats des travaux concernant la complexité de la subsomption sont établis par rapport à des familles de formalismes qui sont caractérisées par un formalisme de base constituant le noyau de la famille, par exemple le formalisme $\mathcal{AL}$. Le noyau consiste en un ensemble réduit de constructeurs choisis parmi ceux qui sont les plus utilisés dans le domaine des logiques terminologiques. Les formalismes appartenant à la famille considérée sont ensuite obtenus en étendant le formalisme de base par d'autres constructeurs (par exemple les langages $\mathcal{ALC}$, $\mathcal{ALCN}$, $\mathcal{ALCNR}$). La complexité de subsomption est étudiée pour chaque formalisme de la famille considérée. Cette approche permet d'analyser le rapport entre le coût de la subsomption et le pouvoir d'expression du formalisme.

Les résultats issus des travaux sur la complexité des procédures de calcul ont fait émerger deux familles de systèmes terminologiques.

5.6.4.1 Les Procédures de Calcul Complètes

Cette approche s'appuie sur le principe que le raisonnement dans un système terminologique doit être efficace et complet (retrouve l'ensemble des inférences valides). Pour atteindre ces objectifs, des systèmes terminologiques au pouvoir d'expression limité ont été développés, comme Classic *(Brachman et al., 1999)*. Dans cette approche, l'écart entre ce qui peut être exprimé par le système et ce qui doit être exprimé est comblé par l'utilisation de règles d'inférences, quand ces dernières sont supportées par le système terminologique.

5.6.4.2 Procédures de Calcul Incomplètes

Les systèmes se réclamant de cette approche comme Loom *(Mc Gregor, 1988)* supportent des logiques terminologiques au pouvoir d'expression élevé et utilisent des procédures de calcul efficaces mais incomplètes. Ils s'appuient sur l'idée que n'importe quelle logique terminologique ayant une puissance d'expression raisonnable implique un coût de raisonnement élevé. Les limites pratiques de l'expérience de KLONE ont favorisé le développement de ce type d'approche. En effet, la faible puissance d'expression de ce formalisme l'a rendu peu utilisable dans des cas réels. Cette expérience a eu pour effet de mettre l'accent sur la nécessité des systèmes supportant des formalismes puissants pour être utilisables en pratique. Dans cette approche également, les règles d'inférence peuvent s'avérer utiles pour compléter le comportement du système terminologique par rapport aux informations qu'il contient.

Notons enfin qu'un troisième type d'approche a été proposé par Patel- Schneider. Elle consiste à abandonner la sémantique standard des logiques terminologiques au profit d'une sémantique plus faible basée sur une logique à quatre valeurs. Il a montré qu'il existe dans cette sémantique un algorithme de calcul de la subsomption complet et efficace pour un formalisme puissant. Cependant, dans cette sémantique les seules relations de subsomption valides sont des relations simples. Par conséquent, de nombreuses relations de subsomption qui sont valides dans la sémantique standard ne peuvent pas être détectées bien qu'elles soient parfois très intuitives comme les inférences basées sur la règle du modus ponens.

5.6.4.3 Complétude et Efficacité

Les deux premières approches présentent l'inconvénient d'avoir recours à l'utilisateur pour combler les spécifications manquantes dans la première approche et pallier l'incomplétude du raisonnement dans la deuxième. Cependant, d'un point de vue pragmatique, la deuxième approche s'avère la plus intéressante. En effet, dans les expériences pratiques réalisées avec cette approche, les inférences non effectuées par le système terminologique ne semblent pas avoir posé de problèmes particuliers. Ces inférences sont en général loin d'être intuitives contrairement à celles qui sont perdues lors de l'utilisation d'une sémantique faible.

5.6.5 Les Constructeurs

5.6.5.1 Les Constructeurs de Concepts

Ils permettent de construire les expressions complexes qui constituent les descriptions associées aux noms de concepts.

– La conjonction de concepts : spécifiée à l'aide de l'opérateur ($\wedge$). Par exemple l'axiome terminologique : Etudiante $\doteq$ Etudiant $\wedge$ Féminin définit le concept étudiante comme étant l'ensemble des individus qui sont à la fois des femmes et qui ont un statut d'étudiant.

– Les restrictions sur les nombres de valeurs pour un rôle appelées restrictions *nr*, sont introduites à l'aide des deux opérateurs ATLEAST et ATMOST. Ils définissent respectivement le nombre minimal et maximal de valeurs autorisées pour un rôle donné.

Par exemple : soit l'axiome terminologique suivant :

Enseignant $\doteq$ personne $\wedge$ ATLEAST (1, enseigne) $\wedge$ ATMOST (1, rattaché-à)

Si les rôles (enseigne) et (rattaché-à) ont pour domaines de valeurs les concepts (cours) et (département) respectivement, le concept *enseignant* représente alors toutes les personnes qui *enseignent* au moins un *cours* et qui sont au plus *rattachés* à un seul *département*.

La restriction EXACTLY(n, R) peut être utilisée pour imposer un nombre de valeurs exact pour un rôle (EXACTLY (n,R) = ALEAST (n, R) $\wedge$ ATMOST (n, R)).

– La disjonction entre concepts est spécifiée à l'aide du constructeur *not*. Elle indique que deux concepts donnés ne peuvent pas avoir d'instances en commun. La description

étudiant $\doteq$ personne $\wedge$ not(enseignant) indique qu'un étudiant est une personne qui ne peut pas être enseignant.

– L'union entre concepts est spécifiée à l'aide du constructeur $\vee$.

5.6.5.2 Les Constructeurs de Rôles

Introduisons les constructeurs qui permettent de spécifier les descriptions des rôles.

– Les restrictions sur les domaines des rôles sont définies à l'aide des constructeurs DOMAIN(nom de concept) et RANGE(nom de concept). Ils permettent respectivement de restreindre le domaine de définition et le domaine de valeurs d'un rôle. Par exemple la déclaration : travailler $\doteq$ DOMAIN(employé) indique que le rôle travailler est défini uniquement pour les instances du concept employé.

– La conjonction de rôles également spécifiée à l'aide du constructeur $\wedge$. par exemple la déclaration : enseigner $\doteq$ travailler $\wedge$ RANGE(université) indique que le rôle enseigner est un rôle travailler particulier qui prend obligatoirement ses valeurs parmi les instances du concept université.

– Les rôles inverses sont définis à l'aide du constructeur *inv*(nom de rôle).

5.6.5.3 La Syntaxe des Constructeurs

Nous nous limitons dans cette section à introduire les règles syntaxiques qui indiquent la manière dont les concepts et les rôles sont spécifiés en utilisant les constructeurs précédents.

Les concepts (notés *C*, *D*) sont construits autour des concepts atomiques (notés *A*) et des rôles (notés *R*, *R'*) selon la règle suivante :

$$C,D \rightarrow Top \mid A \mid C \wedge D \mid C \vee D \mid not(C) \mid atleast(n,R) \mid atmost(n,R) \mid exactly(n,R) \mid Bottom$$

Les expressions des rôles sont spécifiées selon la règle suivante.

$$R,R' \rightarrow Domain(C) \mid Range(C) \mid R \wedge R' \mid inv(R) \mid \exists R.C \mid \forall R.C$$

Le raisonnement sur les descriptions est basé sur la comparaison des descriptions de termes. Cette comparaison est possible grâce à une hypothèse importante qui sous-tend tous les travaux concernant les logiques terminologiques : la sémantique d'un terme est dérivée de manière complète à partir de ses composants et de la manière dont il est construit. En conséquence, les deux conditions suivantes sont généralement requises lors de la définition d'une terminologie :

– Chaque terme doit apparaître une seule fois comme premier argument d'un axiome.

– Les introductions des termes sont acycliques

5.6.5.4 La Sémantique des Constructeurs

La sémantique associée au formalisme est définie par une fonction d'interprétation qui associe à chaque description dans une terminologie un ensemble d'individus ou de couples d'individus de l'univers du discours. Il est possible d'associer la terminologie et les éléments de l'univers du discours de plusieurs manières ; chacune de ces correspondances constitue une interprétation de la terminologie.

Définition 5.1 : (INTERPRETATION)

Une interprétation $i = (\Delta^i, \cdot^i)$ *consiste en :*

– *un ensemble* Δ^i *(le domaine de l'interprétation i)*

– *une fonction* $(\cdot^i)$ *appelée fonction d'interprétation qui associe à chaque concept C un sous-ensemble de* Δ^i *et à chaque rôle R un sous-ensemble de* $\Delta^i \times \Delta^i$*, de manière à ce que les équations suivantes dans lesquelles le symbole* $\| \ \|$ *dénote la cardinalité d'un ensemble soient satisfaites :*

$$Top^i = \Delta^i$$
$$Bottom^i = \varnothing$$
$$(C \wedge D)^i = C^i \cap D^i$$
$$(C \vee D)^i = C^i \cup D^i$$
$$(atleast(n,R))^i = \left\{ a \in \Delta^i \ \middle| \ \left\| \left\{ b \in \Delta^i \ \middle| \ (a,b) \in R^i \right\} \right\| \geq n \right\}$$
$$(atmost(n,R))^i = \left\{ a \in \Delta^i \ \middle| \ \left\| \left\{ b \in \Delta^i \ \middle| \ (a,b) \in R^i \right\} \right\| \leq n \right\}$$
$$(exactly(n,R))^i = \left\{ a \in \Delta^i \ \middle| \ \left\| \left\{ b \in \Delta^i \ \middle| \ (a,b) \in R^i \right\} \right\| = n \right\}$$
$$(inv(R))^i = \left\{ (a,a') \middle| (a',b) \in R^i \right\}$$
$$(not(C))^i = \Delta^i \setminus C^i$$
$$(R \wedge R')^i = R^i \cap R'^i$$
$$(\forall R.C)^i = \left\{ a \in \Delta^i \middle| \forall b, (a,b) \in R^i \Rightarrow b \in C^i \right\}$$
$$(\exists R.C)^i = \left\{ a \in \Delta^i \middle| \exists b, (a,b) \in R^i \right\}$$
$$(Domain(C))^i = C^i \times \Delta^i$$
$$(Range(C))^i = \Delta^i \times C^i$$

L'interprétation d'un terme quelconque du formalisme est déterminée de manière unique en appliquant les équations précédentes. La sémantique des axiomes terminologiques est définie ci-dessous.

Définition 5.2 : (SEMANTIQUE DES AXIOMES TERMINOLOGIQUES)

Soient t_n *un nom de terme et t' une description quelconque dans une terminologie* $\mathcal{L}$*. Une interprétation i de la terminologie* $\mathcal{L}$ *satisfait l'axiome terminologique suivant :*

$$t_n \doteq t' \text{ si et seulement si } t_n^i \subseteq t'^i$$
$$t_n \doteq t' \text{ si et seulement si } t_n^i = t'^i$$

Définition 5.3 : (INTERPRETATION VALIDE)

Soit $\mathcal{L}$ une terminologie. $i = (\Delta^i, \cdot^i)$ est une interprétation valide de la terminologie $\mathcal{L}$ si et seulement si la fonction $(\cdot^i)$ satisfait tous les axiomes terminologiques de $\mathcal{L}$.

Une interprétation valide d'une terminologie est appelée modèle de cette terminologie.

Définition 5.4 (SUBSOMPTION)

Soient t et t' deux termes d'une terminologie $\mathcal{L}$. Le terme t' subsume le terme t si et seulement si toute interprétation de t est incluse dans toute interprétation de t' :

$$t' \text{ subsume } t \Leftrightarrow t_n^i \subseteq t'^i \text{ pour toute interprétation valide i de } \mathcal{L}.$$

Définition 5.5 (SATISFIABILITE D'UN TERME) :

Une interprétation $i = (\Delta^i, \cdot^i)$ est un modèle d'un terme t si $t^i \neq \emptyset$. Un terme est satisfiable s'il a un modèle, il est insatisfiable sinon. Un terme insatisfiable est un terme incohérent.

La notion de satisfiabilité est étendue aux terminologies par la définition ci-dessous :

Définition 5.6 (SATISFIABILITE (CONSISTANCE) D'UNE TERMINOLOGIE)

Une terminologie est dite satisfiable (consistante) si elle ne contient pas de concept incohérent. Elle est dite insatisfiable (inconsistante) sinon.

Cette définition réduit la satisfiabilité d'une base de connaissances à la satisfiabilité des concepts qui la composent. Ceci est possible en remplaçant dans les descriptions des concepts les noms de rôles par leurs descriptions.

Généralement, la réalisation d'un seul des tests de subsomption ou de satisfiabilité est suffisante car la subsomption peut être réduite à la satisfiabilité et inversement. Ceci est exprimé par les deux règles d'équivalence suivantes :

– Tester la subsomption entre deux concepts *C* et *C'* équivaut à tester la satisfiabilité du concept *C'*

C' subsume $C \Leftrightarrow C \wedge not(C')$ est insatisfiable

– Le test de satisfiabilité d'un concept *C* peut se réduire à un test de subsomption entre le concept *C* et le concept vide.

C est insatisfiable $\Leftrightarrow$ Bottom subsume *C*.

De manière générale le test de subsomption constitue le mécanisme raisonnement de base des systèmes terminologiques existants.

5.6.5.5 Exemple

Soit la terminologie suivante :

$$\text{Humain} \doteq \text{Mammifère}$$

$$\text{Femme} \doteq \text{Humain}$$

$$\text{Employé} \doteq \text{Humain} \wedge \text{atleast (1, Travaille-dans)}$$

$$\text{Directrice} \doteq \text{Femme} \wedge \text{atleast (1, Supervise)}$$

$$\text{Supervise} \doteq \text{Travaille-dans}$$

Cette terminologie contient deux termes atomiques, le concept *mammifère* et le rôle *travaille-dans* (ils apparaissent uniquement en partie droite des axiomes terminologiques). Nous ne disposons pas des définitions complètes des concepts Humain, Femme et du rôle Supervise : ils sont donc introduits en tant que concepts primitifs. En revanche, les concepts Employé et Directrice sont définis exactement comme étant respectivement des humains qui travaillent et des femmes qui exercent au moins un rôle de supervision.

Lors du processus de classification, la position d'un terme primitif dans la hiérarchie de termes est donnée de manière explicite dans sa description, par exemple le concept Femme est directement placé sous le concept Humain. En revanche la position d'un terme défini est inférée à partir de sa définition. Les trois cas suivants peuvent se présenter :

1- La relation de subsomption est spécifiée explicitement dans la description du terme défini. Par exemple la relation de subsomption entre le concept Employé et le concept Humain apparaît de manière explicite dans la description du premier concept.

2- La subsomption est spécifiée indirectement. Dans l'exemple précédent, la propriété de transitivité de la relation de subsomption permet de déduire que le concept Femme est sous-concept de Mammifère.

3- Certaines relations de subsomption, inférées par le système terminologique, ne se déduisent pas par transitivité. Dans l'exemple précédent, le concept Directrice est défini comme étant une Femme, donc par transitivité un Humain, qui a au moins un rôle de supervision. Etant donné que le rôle supervise est une spécialisation du rôle travaille-dans, une Directrice est également un Humain qui travaille, donc un Employé. Par conséquent, lors de la classification du concept Directrice, le système terminologique explicite une relation de subsomption entre ce concept et le concept Employé.

5.6.6 Raisonnement Taxinomique pour la Recherche d'Information

Une des applications les plus importantes du raisonnement taxinomique concerne le traitement des requêtes et l'indexation. En effet, l'expression des connaissances dans un formalisme formel permet un traitement uniforme des classes, des requêtes ou encore des vues sur la terminologie. Les requêtes sont considérées comme des nouveaux concepts à intégrer dans la taxinomie et le mécanisme de classification permet de positionner automatiquement la requête cette taxinomie. La combinaison de ces deux possibilités offre un moyen puissant pour le traitement des requêtes en exploitant les connaissances disponibles *(Meghini et al,*

1997)(Horrocks & Tessaris, 2000)(Andreasen et al., 2004). Cette technique est à la base des travaux concernant l'optimisation sémantique des requêtes dans les bases de données ou dans les ontologies. Dans cette approche, l'évaluation d'une requête consiste à récupérer toutes les instances de ses sous-concepts et à filtrer parmi les instances de ses sous-concepts directs celles qui vérifient les conditions de la requête. En particulier dans les systèmes de recherche d'information, si la requête est exprimée à l'aide du concept *Q* on recherche les instances *D* du concept *Q*, telles que la requête *Q* subsume le concept décrivant le document *D*. Ainsi les documents jugés pertinents par le système correspondent aux instances du concept représentant la requête et de ses subsumés. Cette phase peut être améliorée en utilisant la technique d'indexation sémantique qui consiste à définir des index conformément à l'organisation de la terminologie (utiliser les concepts comme des index et stocker la relation entre chaque concept et ses instances).

Le raisonnement taxinomique, notamment la vérification de la consistance, peut être également utilisé dans les tâches de maintenance des bases de connaissances, par la découverte de cycles dans la taxinomie. Dans le processus de recherche d'information, si une requête est considérée comme une description de concept, celui-ci peut être comparé au concept inconsistant. S'il l'est, la requête de l'utilisateur est fausse et ne renverra aucune réponse. Par rapport aux systèmes actuels qui ne renvoient pas de réponses, une explication pourra être donnée à l'utilisateur, à savoir que la requête est sémantiquement incohérente.

5.7 Formalisation de la Terminologie

5.7.1 Expériences avec TRIPLE

Dans cette première expérience *(Soualmia et al., 2003)(Soualmia & Darmoni, 2003a) (Soualmia & Darmoni, 2003)*, nous avons défini une base de connaissances reposant sur un schéma RDFS qui définit les concepts (classes) qui sont les mots clés et types de ressources, les rôles (relations entre concepts) qui sont les qualificatifs et une relation de subsomption pour organiser les classes en hiérarchie. Les ressources seront annotées en fonction des concepts et des rôles de l'ontologie sous le format RDF. Nous avons ajouté à cela 50 règles métier définies par un médecin généraliste (A.Soualmia) concernant essentiellement les complications des maladies. Les règles métier ne sont pas utilisées pour annoter les ressources et elles ne contribuent pas à la définition de concepts. Elles ont été traduites sous la forme de règles d'inférence et utilisées pour un raisonnement sur le contenu des ressources dans le processus de recherche d'information.

RDFS permet de définir des hiérarchies de classes et des propriétés mais il n'intègre pas de capacités de raisonnement, comme ceux qu'offrent les systèmes basés sur des langages formels comme les Logiques de Description. L'écriture des règles d'inférence n'étant pas possible en RDFS, nous utilisons les fonctionnalités de l'outil TRIPLE *(Sintek & Decker, 2001)* qui a été développé pour une recherche information intelligente basée sur les connaissances. Il permet de réaliser des raisonnements complexes sur des ressources RDF instances de concepts en traduisant RDF en Horn-Logic mais aussi en DAML+OIL[42]. Un accès à des éléments externes

[42] http://www.daml.org/2001/03/daml+oil-index.html.

comme le classifieur RACER *(Haarslev & Möller, 2001)* (basé sur les Logiques de Description) offre la possibilité de bénéficier de ses mécanismes de raisonnement. Pour la recherche de ressources c'est particulièrement utile. Par exemple si une ressource *a* est instance du concept $C := \exists$ *Hepatite.Complications* et qu'un utilisateur recherche des ressources associées à *Cirrhose*, c'est à dire instances du concept *Cirrhose*, le système déduira que la ressource *a* est également une réponse à la requête grâce à la règle d'inférence $\exists$*Hepatite.Complications* $\Rightarrow$ *Cirrhose*.

```
// collection of resources
@cismef:resources {
  cismef:doc1[
    meta:title->"Document 1 is related to Hepatitis and Aids";
    meta:author->"Lina Soualmia";
    meta:keyword->HEPATITIS;
    meta:keyword->AIDS;
    meta:qualifier->COMPLICATIONS].
 cismef:doc2[
    meta:title->"Document 2 is related to abdomen";
    meta:author->"Doctor D  in Gastrology";
    meta:keyword->ABDOMEN;
    meta:qualifier->COMPLICATIONS;
    meta:qualifier->RADIOGRAPHY].
cismef:doc3[
    meta:title->"Document 3 is related to Accidents";
    meta:author->"Doctor E  in Risks";
    meta:keyword->ACCIDENT;
    meta:qualifier->RISKS].
cismef:doc4[
    meta:title->"Document 4 is related to Cirrhosis";
    meta:author->"SJ Darmoni";
    meta:keyword->LIVERCIRRHOSIS].
}
// domain ontology
@cismefOntology {
  HEPATITIS[subClassOf -> LIVERDISEASES].
     LIVERCIRRHOSIS[subClassOf -> LIVERDISEASES].
  HEPATITIS[Complications->LIVERCIRRHOSIS].
  // ...
}
// query : all the resources and their author related to
LIVERCIRRHOSIS//
FORALL Resource, Author <-
search(Resource,LIVERCIRRHOSIS)@search(cismef:resources,
cismefOntology) AND  Resource[meta:author -> Author].
```

FIG.5.8.– Exemple de déclaration de connaissances, métadonnées règle d'inférence et requête sous TRIPLE

```
compiling cismef.triple
running cismef.triple
***

   Resource = cismef:doc1, Author = 'Lina Soualmia'
   Resource = cismef:doc4, Author = 'SJ Darmoni'

   done.
```

FIG.5.9.– Fichier résultat

L'outil TRIPLE peut être utilisé pour un ensemble restreint de déclarations de concepts et de documents, et au-delà d'un certain nombre les capacités de calcul sont très longues. De ce fait, pour une terminologie de la taille de celle du CISMeF, mais également vu le nombre important de documents indexés, nous avons ensuite opté pour le langage de représentation OWL-DL *(Horrocks et al., 2003)*, en laissant de côté le règles d'inférence. Ce langage est un nouveau standard du W3C mais également car il se fonde sur une logique de description avec ce que cela comporte comme capacités de raisonnement et d'inférences. Des systèmes de raisonnement comme Racer existent et peuvent être utilisés pour vérifier et interroger des bases de connaissances terminologiques. Nous utilisons Racer *(Haarslev & Möller, 2003)* plutôt que Fact *(Horrocks, 1998)* en raison de la possibilité de raisonnement sur les individus (ABox) alors que Fact ne permet que le raisonnement sur la TBox. La disponibilité d'éditeurs comme Protégé-2000 *(Noy et al., 2001)* a également orienté notre choix.

Nous présentons tout d'abord les caractéristiques du langage OWL puis une solution de modélisation que nous avons retenue.

5.7.2 De OIL à OWL

Le projet Ontobroker a proposé le langage OIL *(Fensel et al., 2000)*qui combine les frames pour la modélisation et les logiques de description pour le raisonnement. Le langage permet la distinction entre les concepts définis et les concepts primitifs. Les propriétés sont des rôles organisés sous la forme d'une taxinomie. Ils possèdent les propriétés de négation, symétrie ou transitivité. Les descriptions de concepts sont des expressions booléennes faisant référence à d'autres concepts avec ou sans rôles.

Par ailleurs, le programme Daml (DARPA Agent Markup Language) a proposé les langages Daml-Ont et Daml-Logic *(Hendler & Mc Guiness, 2000)*. Le premier langage permet d'exprimer des ontologies et le Daml-Logic permet d'ajouter des règles d'inférence. La fusion de Daml-Ont et OIL a donné lieu au langage Daml+Oil qui est équivalent à la logique $\mathcal{SHIQ}$ qui est très expressive.

Daml+oil a ensuite donné lieu en 2001 au langage OWL (Ontology Web Language) par le groupe de travail du W3C sur le Web Sémantique. Le but d'un tel langage est son utilisation dans les cas où les informations contenues dans les documents doivent être traitées par des applications logicielles. La sémantique de OWL se fonde sur un domaine d'interprétation ouvert permettant aux systèmes d'effectuer des raisonnements même s'il n'y a pas une connaissance complète du monde.

Le langage OWL fournit trois sous-langages : OWL-Lite, OWL-DL et OWL-Full :

– OWL-Lite est une extension du langage RDFS permettant la modélisation d'ontologies simples. Il est d'une complexité peu élevée *(Horrocks & Patel-Schneider, 2003)*.

– Le langage OWL-DL contient des constructeurs supplémentaires et le langage est plus expressif mais décidable.

– Le langage OWL-Full est caractérisé par les mêmes constructeurs que OWL-DL. En revanche l'interprétation des concepts est plus large et de ce fait, le langage n'est pas décidable.

Abstract Syntax	DL Syntax	Semantics
Descriptions (C)		
A (URI reference)	A	$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
`owl:Thing`	$\top$	$\texttt{owl:Thing}^{\mathcal{I}} = \Delta^{\mathcal{I}}$
`owl:Nothing`	$\bot$	$\texttt{owl:Nothing}^{\mathcal{I}} = \{\}$
`intersectionOf`(C_1 C_2 ...)	$C_1 \sqcap C_2$	$(C_1 \sqcap D_1)^{\mathcal{I}} = C_1^{\mathcal{I}} \cap D_2^{\mathcal{I}}$
`unionOf`(C_1 C_2 ...)	$C_1 \sqcup C_2$	$(C_1 \sqcup C_2)^{\mathcal{I}} = C_1^{\mathcal{I}} \cup C_2^{\mathcal{I}}$
`complementOf`(C)	$\neg C$	$(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
`oneOf`(o_1 ...)	$\{o_1, \ldots\}$	$\{o_1, \ldots\}^{\mathcal{I}} = \{o_1^{\mathcal{I}}, \ldots\}$
`restriction`(R `someValuesFrom`(C))	$\exists R.C$	$(\exists R.C)^{\mathcal{I}} = \{x \mid \exists y.\langle x, y\rangle \in R^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\}$
`restriction`(R `allValuesFrom`(C))	$\forall R.C$	$(\forall R.C)^{\mathcal{I}} = \{x \mid \forall y.\langle x, y\rangle \in R^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}}\}$
`restriction`(R `hasValue`(o))	$R : o$	$(\forall R.o)^{\mathcal{I}} = \{x \mid \langle x, o^{\mathcal{I}}\rangle \in R^{\mathcal{I}}\}$
`restriction`(R `minCardinality`(n))	$\geqslant n\, R$	$(\geqslant n\, R)^{\mathcal{I}} = \{x \mid \sharp(\{y.\langle x, y\rangle \in R^{\mathcal{I}}\}) \geqslant n\}$
`restriction`(R `minCardinality`(n))	$\leqslant n\, R$	$(\geqslant n\, R)^{\mathcal{I}} = \{x \mid \sharp(\{y.\langle x, y\rangle \in R^{\mathcal{I}}\}) \leqslant n\}$
`restriction`(U `someValuesFrom`(D))	$\exists U.D$	$(\exists U.D)^{\mathcal{I}} = \{x \mid \exists y.\langle x, y\rangle \in U^{\mathcal{I}} \text{ and } y \in D^{\mathbf{D}}\}$
`restriction`(U `allValuesFrom`(D))	$\forall U.D$	$(\forall U.D)^{\mathcal{I}} = \{x \mid \forall y.\langle x, y\rangle \in U^{\mathcal{I}} \rightarrow y \in D^{\mathbf{D}}\}$
`restriction`(U `hasValue`(v))	$U : v$	$(U : v)^{\mathcal{I}} = \{x \mid \langle x, v^{\mathcal{I}}\rangle \in U^{\mathcal{I}}\}$
`restriction`(U `minCardinality`(n))	$\geqslant n\, U$	$(\geqslant n\, U)^{\mathcal{I}} = \{x \mid \sharp(\{y.\langle x, y\rangle \in U^{\mathcal{I}}\}) \geqslant n\}$
`restriction`(U `maxCardinality`(n))	$\leqslant n\, U$	$(\leqslant n\, U)^{\mathcal{I}} = \{x \mid \sharp(\{y.\langle x, y\rangle \in U^{\mathcal{I}}\}) \leqslant n\}$
Data Ranges (D)		
D (URI reference)	D	$D^{\mathbf{D}} \subseteq \Delta_{\mathbf{D}}^{\mathcal{I}}$
`oneOf`(v_1 ...)	$\{v_1, \ldots\}$	$\{v_1, \ldots\}^{\mathcal{I}} = \{v_1^{\mathcal{I}}, \ldots\}$
Object Properties (R)		
R (URI reference)	R	$R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
	R^-	$(R^-)^{\mathcal{I}} = (R^{\mathcal{I}})^-$
Datatype Properties (U)		
U (URI reference)	U	$U^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta_{\mathbf{D}}^{\mathcal{I}}$
Individuals (o)		
o (URI reference)	o	$o^{\mathcal{I}} \in \Delta^{\mathcal{I}}$
Data Values (v)		
v (RDF literal)	v	$v^{\mathcal{I}} = v^{\mathbf{D}}$

FIG.5.10.–. Liste des Constructeurs de OWL-DL *(Horrocks et al, 2003)*

5.7.3 Travaux dans le Domaine de la Santé

Des travaux récents se sont intéressés à la représentation de l'UMLS et de son Réseau Sémantique à l'aide d'un langage formel. On peut citer le travail de *(Schulz & Hahn, 2001)* qui ont traduit en langage Classic les concepts de l'UMLS étant en relation 'Part-of', le travail de *(Cornet & Abu-Hanna, 2002)* qui ont proposé de représenter l'UMLS avec la logique $\mathcal{ALC}$, ou encore le travail de *(Kashyap & Bordiga, 2003)* qui proposent de représenter le Réseau Sémantique de l'UMLS en utilisant OWL.

(Wroe et al., 2003) proposent une méthodologie pour traduire la Gene Ontology en DAML+Oil. Par ailleurs, le thésaurus du National Cancer Institute (NCI) a été traduit dans le langage OWL-Lite *(Golbeck et al., 2003)*. L'ontologie qui en découle est fondée sur l'UMLS et elle sera exploitée pour l'indexation et la recherche de données relatives au NCI. D'autres travaux concernent la modélisation d'ontologies comme *(Hu et al., 2003)(Horrocks & Rector, 1997)(Aitken et al, 2004)*.

En revanche, à notre connaissance il n'existe pas de travaux sur la 'formalisation' du MeSH. Le MeSH souffre en effet de sa taille mais également de ses ambiguïtés. Par exemple *'diagnostic'*

est défini comme une spécialité médicale mais également comme qualificatif, de même que *'virologie'* est un mot clé, une spécialité et un qualificatif. Ces aspects ont été traités dans la modélisation de la terminologie CISMeF par l'introduction de nouveaux concepts comme les métatermes, mais cela ne nous apparaît pas être suffisant puisque l'ambiguïté persiste, notamment à cause des liens (spécialité-mot clé) et (spécialité-qualificatif). L'avantage d'utiliser une logique de description est de bénéficier de mécanismes de raisonnement qui leur sont associés (satisfiabilité, subsumption, classification, vérification de consistance, instanciation, réalisation et recherche de concepts), permettant entre autres de maintenir un système terminologique et d'améliorer les résultats de requêtes grâce aux inférences.

Cette partie décrit la première étape de formalisation du MeSH et de la terminologie CISMeF *(Soualmia et al., 2004a) (Soualmia et al., 2004b).*

Abstract Syntax	DL Syntax	Semantics
Class(A partial C_1 ...C_n)	$A \sqsubseteq C_1 \sqcap \ldots \sqcap C_n$	$A^{\mathcal{I}} \subseteq C_1^{\mathcal{I}} \cap \ldots \cap C_n^{\mathcal{I}}$
Class(A complete C_1 ...C_n)	$A = C_1 \sqcap \ldots \sqcap C_n$	$A^{\mathcal{I}} = C_1^{\mathcal{I}} \cap \ldots \cap C_n^{\mathcal{I}}$
EnumeratedClass(A o_1 ...o_n)	$A = \{o_1, \ldots, o_n\}$	$A^{\mathcal{I}} = \{o_1^{\mathcal{I}}, \ldots, o_n^{\mathcal{I}}\}$
SubClassOf(C_1 C_2)	$C_1 \sqsubseteq C_2$	$C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}$
EquivalentClasses(C_1 ...C_n)	$C_1 = \ldots = C_n$	$C_1^{\mathcal{I}} = \ldots = C_n^{\mathcal{I}}$
DisjointClasses(C_1 ...C_n)	$C_i \sqcap C_j = \bot, i \neq j$	$C_i^{\mathcal{I}} \cap C_j^{\mathcal{I}}\{\}, i \neq j$
Datatype(D)		$D^{\mathcal{I}} \subseteq \Delta_{\mathbf{D}}^{\mathcal{I}}$
DatatypeProperty(U super(U_1)...super(U_n)	$U \sqsubseteq U_i$	$U^{\mathcal{I}} \subseteq U_i^{\mathcal{I}}$
domain(C_1) ...domain(C_m)	$\geqslant 1\, U \sqsubseteq C_i$	$U^{\mathcal{I}} \subseteq C_i^{\mathcal{I}} \times \Delta_{\mathbf{D}}^{\mathcal{I}}$
range(D_1) ...range(D_l)	$\top \sqsubseteq \forall U.D_i$	$U^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times D_i^{\mathcal{I}}$
[Functional])	$\top \sqsubseteq \leqslant 1\, U$	$U^{\mathcal{I}}$ is functional
SubPropertyOf(U_1 U_2)	$U_1 \sqsubseteq U_2$	$U_1^{\mathcal{I}} \subseteq U_2^{\mathcal{I}}$
EquivalentProperties(U_1 ...U_n)	$U_1 = \ldots = U_n$	$U_1^{\mathcal{I}} = \ldots = U_n^{\mathcal{I}}$
ObjectProperty(R super(R_1)...super(R_n)	$R \sqsubseteq R_i$	$R^{\mathcal{I}} \subseteq R_i^{\mathcal{I}}$
domain(C_1) ...domain(C_m)	$\geqslant 1\, R \sqsubseteq C_i$	$R^{\mathcal{I}} \subseteq C_i^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
range(C_1) ...range(C_l)	$\top \sqsubseteq \forall R.C_i$	$R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times C_i^{\mathcal{I}}$
[inverseOf(R_0]	$R = ({}^{-}R_0)$	$R^{\mathcal{I}} = (R_0^{\mathcal{I}})^{-}$
[Symmetric]	$R = ({}^{-}R)$	$R^{\mathcal{I}} = (R^{\mathcal{I}})^{-}$
[Functional]	$\top \sqsubseteq \leqslant 1\, R$	$R^{\mathcal{I}}$ is functional
[InverseFunctional]	$\top \sqsubseteq \leqslant 1\, R^{-}$	$(R^{\mathcal{I}})^{-}$ is functional
[Transitive])	$Tr(R)$	$R^{\mathcal{I}} = (R^{\mathcal{I}})^{+}$
SubPropertyOf(R_1 R_2)	$R_1 \sqsubseteq R_2$	$R_1^{\mathcal{I}} \subseteq R_2^{\mathcal{I}}$
EquivalentProperties(R_1 ...R_n)	$R_1 = \ldots = R_n$	$R_1^{\mathcal{I}} = \ldots = R_n^{\mathcal{I}}$
AnnotationProperty(S)		
Individual(o type(C_1) ...type(C_n)	$o \in C_i$	$o^{\mathcal{I}} \in C_i^{\mathcal{I}}$
value(R_1 o_1)...value(R_n o_n)	$\langle o, o_i \rangle \in R_i$	$\langle o^{\mathcal{I}}, o_i^{\mathcal{I}} \rangle \in R_i^{\mathcal{I}}$
value(U_1 v_1)...value(U_n v_n))	$\langle o, v_i \rangle \in U_i$	$\langle o^{\mathcal{I}}, v_i^{\mathcal{I}} \rangle \in U_i^{\mathcal{I}}$
SameIndividual(o_1 ...o_n)	$o_1 = \ldots = o_n$	$o_i^{\mathcal{I}} = o_j^{\mathcal{I}}$
DifferentIndividuals(o_1 ...o_n)	$o_i \neq o_j, i \neq j$	$o_i^{\mathcal{I}} \neq o_j^{\mathcal{I}}, i \neq j$

FIG.5.11.– .Syntaxe de construction des descriptions de concepts *(Horrocks et al., 2003)*

5.7.4 Principes de Modélisation

Le premier principe de modélisation consiste à "nettoyer" la taxinomie du MeSH en distinguant les relations *'pat-of'* et *'is-a'*. Pour ce faire, les arborescences Anatomie, Sciences Biologiques et Régions Géographiques sont traitées séparément.

Le second principe consiste à distinguer clairement entre les différentes notions (les spécialités, les mots clés et les qualificatifs). Par exemple, la spécialité *"diagnostic"* est distinguée du qualificatif *"diagnostic"* car ils désignent des notions différentes (il en est de même pour "*virologie"*).

Les contraintes sur les qualificatifs ont été définies pour vérifier si l'association d'un qualificatif à un mot clé est valide. Ces contraintes ont été définies comme étant des restrictions du domaine des qualificatifs, chaque domaine étant un ensemble de mots clés. Par exemple, le qualificatif "*diagnostic*" peut être associé au mot clé "*maladies*" (et de ce fait il peut être associé à tous les descendants de "*maladies*"). En revanche, il ne peut pas être associé aux mots clés de l'arborescence "*régions géographiques*".

Enfin, l'ensemble des concepts utilisés pour l'indexation d'un document constitue une nouvelle description de concept à ajouter dans la terminologie. Cette nouvelle description correspond à la conjonction des concepts utilisés pour l'indexation. Cette description est nécessaire pour la définition des individus de la base de connaissances terminologique.

5.7.5 Des Fichiers Textes à la Base de Données

Les fichiers textes du MeSH sont traités automatiquement depuis 2002 avec un script a sur une plateforme Unix afin de renseigner la table TB_MeSH dans la base de données CISMeF avec les attributs suivants : *Descripteur Français, Code Cat MeSH* et *NLM*. Les autres champs des fichiers MeSH ne sont pas pris en compte (par exemple le champ *MeSH definition*) car ils sont en anglais. En pratique, une jointure sur les deux tables TB_MeSH et TB_MC, qui contient tous les descripteurs utilisés dans le catalogue (n = 9 861 en Mars 2004), afin de mettre à jour la terminologie mais également de calculer tous les liens qui existent entre les descripteurs et les niveaux dans les hiérarchies.

5.7.6 De la Base de Données à la Base de Connaissances

Le langage OWL-DL possède les caractéristiques d'une logique de description. Comme décrit plus haut (en 5.6) les logiques de description structurent les connaissances du domaine en deux, voire trois niveaux: un niveau terminologique (la TBox ou l'ontologie) dans lequel se trouvent les classes des objets du domaine (concepts), avec leurs propriétés (rôles) et un niveau d'assertions (ABox), dans lequel se trouvent les individus (instances) et enfin un niveau de règles d'inférences (IBox) venant compléter la TBox. Dans notre cas, nous considérons que la TBox contient la terminologie (les spécialités, les mots clés, les qualificatifs et les types de ressources) en OWL-DL et que les instances sont les documents indexés (à inclure dans la ABox). Une requête peut être alors considérée comme une description de concept et grâce au processus de classification, les instances (les documents) correspondant à ce concept sont récupérées car elles satisfont la requête *(Horrocks & Tessaris, 2000) (Stuckenschmidt & van Harmelen, 2002) (Andreasen et al., 2004)*.

La terminologie, stockée dans une base de données relationnelle est automatiquement représentée en OWL-DL grâce à un algorithme en Java couplé avec des requêtes SQL (§). La

construction est descendante (Top-Down), allant du concept Top (Thing en OWL) aux spécialités et ensuite des têtes d'arborescence aux mots clés et aux types de ressources. La hiérarchie de qualificatifs est gérée séparément. L'objectif est d'automatiser au maximum tout le processus.

D'un point de vue syntaxique, comme dans *(Golbeck et al., 2003)*, tous les caractères illégaux (*-:,&*) et les espaces contenus dans les noms des descripteurs sont remplacés par des underscores. Tous les caractères accentués (*"éèêë"*) sont désaccentués (*"e"*). Les libellés des descripteurs qui commencent avec un caractère numérique sont préfixés par des underscores. Le libellé "*11-hydroxycorticostéroïdes*" est remplacé par "*_11_hydroxycorticosteroides*".

D'un point de vue modélisation, les choix retenus sont décrits dans la section suivante.

5.7.6.1 Les Classes OWL

Les mots clés, les spécialités et les types de ressources sont représentés par des concepts primitifs. Lorsque deux concepts ont le même libellé mais désignent deux notions distinctes, leur nom en OWL est préfixé par *mt_,* lorsque c'est une spécialité, et par *tr_* lorsque c'est un type de ressources.

Les spécialités sont décrites comme des concepts primitifs. Ils n'ont pas de description associée. Toutes les spécialités sont stockées dans une même table, de ce fait, pour chaque spécialité une description est ajoutée simplement dans le fichier OWL en sortie de l'algorithme. Par exemple, nous représentons la spécialité *cardiologie* de la manière suivante en OWL:

```
<owl:ClassOf rdf:ID="mt_cardiologie">
</owl:Class>
```

Les relations *'is-a'* entre concepts sont déclarées en OWL comme étant des relations de subsomption. En premier lieu, seuls les mots clés et les types de ressources qui sont des fils directs des spécialités sont décrits. Ensuite leurs descendants sont ajoutés niveau par niveau (nous considérons un concept et son super-concept qui ont une différence de niveau égale à 1). Par exemple *'accident domestique'* est un sous-concept de *'accident'*, nous l'exprimons en OWL par :

```
<owl:Class rdf:ID="accident_domestique">
   <rdfs:subClassOf>
       <owl:Class rdf:about="#accidents" />
   </rdfs:subClassOf>
</owl:Class>
```

Si un concept possède plus d'un super-concept (héritage multiple), nous le représentons comme un concept primitif est spécifié par l'intersection de ses super-concepts. Par exemple, *hépatite C chronique* est à la fois une *hépatite C* et une *hépatite chronique*. La syntaxe correspondante en OWL est la suivante :

```
<owl:Class rdf:ID="hepatite_C_chronique">
   <rdfs:subClassOf>
      <owl:Class>
         <owl:intersectionOf rdf:parseType="Collection">
              <owl:Class rdf:about="#hepatite_C" />
              <owl:Class rdf:about="#hepatite_chronique" />
         </owl:intersectionOf>
      </owl:Class>
   </rdfs:subClassOf>
</owl:Class>
```

5.7.6.2 Les Propriétés OWL

Les qualificatifs sont organisés en une relation hiérarchique. Nous les représentons par des rôles. Tous les qualificatifs sont regroupés dans une table dans la base de données CISMeF. Afin de les représenter en OWL, nous préfixons leur libellé par *qu_* . Chaque qualificatif possède un domaine que nous désignons par *domain_qu_*. En revanche nous n'avons pas d'information pour l'instant concernant le range des rôles.

```
<owl:ObjectProperty rdf:ID="qu_contre_indication">
   <rdfs:domain rdf:resource="#domain_qu_contre_indication" />
      <rdfs:subPropertyOf>
         <intersectionOf rdf:parseType="Collection">
            <owl:ObjectProperty rdf:about="#qu_action_pharmacologique" />
            <owl:ObjectProperty rdf:about="#qu_usage_therapeutique" />
         </intersectionOf>
   </rdfs:subPropertyOf>
</owl:ObjectProperty>
```

5.7.6.3 La Propriété Part-Of

Les mots clés des arborescences *Anatomie, Sciences Biologiques* et *Régions Géographiques* sont organisés selon la relation *part-of*. Nous les traitons séparément. Le rôle *partOf* est défini en OWL par :

```
<owl:ObjectProperty rdf:ID="partOf">
</owl:ObjectProperty>
```

Dans la terminologie, le mot clé *"doigt"* est positionné sous le mot clé "*main*" dans l'arborescence *Anatomie*. Comme cette relation hiérarchique correspond en fait à la relation "*partOf",* nous définissons le concept *"main"* en OWL par :

```
<owl:Class rdf:ID="doigt">
   <rdfs:subClassOf>
       <owl:Restriction>
             <owl:onProperty rdf:resource="#partOf" />
             <owl:hasValue rdf:resource="#main" />
       </owl:Restriction>
   </rdfs:subClassOf>
</owl:Class>
```

5.7.6.4 Les Restrictions sur les Domaines des Propriétés

Un qualificatif peut être associé et appliqué à plusieurs arborescences de mots clés. Nous spécifions les domaines des qualificatifs par des unions de concepts. Cette information est disponible sous forme de chaîne de caractères dans la table des qualificatifs au niveau de l'attribut *Restriction*. Les têtes d'arborescence sont séparées par des virgules. Cette information a été renseignée manuellement par le bibliothécaire médical. Par exemple, la restriction "C01-C05, D, G" indique en fait que le qualificatif concerné ne peut être appliqué qu'aux mots clés qui appartiennent aux arborescences C01 à C05, D01 à D27, et G01 à G14. La difficulté que nous avons rencontrée est de traiter chaque restriction (au nombre de 83) afin de déterminer toutes les têtes d'arborescence les mots clés correspondants. En OWL nous définissons le domaine de *"qu_contre_indication"* par :

```
<owl:Class rdf:ID="domain_qu_contre_indication">
   <owl:unionOf rdf:parseType="Collection">
      <owl:Class rdf:about="#anesthesie_et_analgesie" />
      <owl:Class rdf:about="#intervention_chirurgicale" />
      <owl:Class rdf:about="#produits_chimiques_inorganiques" />
…
   </owl:unionOf>
</owl:Class>
```

5.7.6.5 Représentation des Documents

Les concepts représentant les documents (leurs instances au nombre de 13 198) doivent également être inclus. Les documents sont traités comme étant des définitions de concept. Pour chaque document, un nouveau concept est créé. Par exemple, le document 112, qui traite du

diagnostic de l'hépatite et de ses complications est indexé avec *'hepatite/diagnostic'* et '*hepatite/complications'*. Nous considérons ce document comme étant une instance du concept défini

R_112 ≐ ∃ diagnostic.hepatite ∧ ∃ complications.hepatite. Cette description en OWL est décrite par :

```
<owl:Class rdf:ID="R_112">
      <owl:intersectionOf rdf:parseType="Collection">
            <owl:Restriction>
                  <owl:onProperty rdf:resource="#qu_diagnostic" />
                  <owl:someValuesFrom rdf:resource="#hepatite" />
            </owl:Restriction>
            <owl:Restriction>
                  <owl:onProperty rdf:resource="#qu_complications" />
                  <owl:someValuesFrom rdf:resource="#hepatite" />
            </owl:Restriction>
      </owl:intersectionOf>
</owl:Class>
```

5.7.7 Vérification de la Consistance et Classification

5.7.7.1 Import sous Protégé-2000

La TBox obtenue est de grande taille (23 420 concepts (9 861 mots clés ; 104 spécialités; 144 types de ressources; 84 domaines; 13 227 concepts représentant les documents) et 85 relations (84 qualificatifs et 1 relation *partOf*). Nous avons essayé d'importer le fichier OWL résultant (20,75 MB) sous l'éditeur d'ontologies Protégé-2000 grâce à son plug-in OWL, mais la mémoire de la machine virtuelle Java est arrivée très vite à saturation à cause de la taille du fichier. Nous avons choisi de réduire le nombre de concepts relatifs aux documents pour n'en garder que 3 000 (sélectionnés aléatoirement). Le fichier a pu être traité avec succès en ~ 30 minutes dans la version bêta 2.0 (traitement réduit à 5 min dans la version actuelle). Le langage a été validé par Protégé comme étant le langage OWL-DL.

5.7.7.2 Vérification de la Consistance

La vérification, à l'aide du classifieur Racer, de la consistance de la terminologie augmentée de 3 000 concepts décrivant les documents prend environ 3 heures de temps (Protégé 2.0 bêta et le plug-in OWL 119). Aucune description inconsistante n'est trouvée. Cela peut s'expliquer par différentes raisons :

- Le prétraitement des fichiers MeSH en une base de données structurée
- La distinction entre les différentes notions permet d'éviter les cycles dans la terminologie (entre spécialités, mots clés, qualificatifs et types de ressources)

- L'utilisation de l'opérateur d'intersection lorsqu'un concept (ou un rôle) appartient à plusieurs super-concepts (super-rôle)
- Tous les concepts, mis à a part ceux décrivant les documents et ceux relatifs aux contraintes sur les domaines des qualificatifs, n'ont pas de propriétés
- Les concepts décrivant les documents ont des propriétés, mais du fait qu'ils ont été indexés manuellement par l'équipe des documentalistes, leurs descriptions ne peuvent pas être inconsistantes.

Le test de consistance peut en revanche être utilisé pour vérifier la validité des requêtes, ou encore vérifier qu'une indexation automatique est consistante.

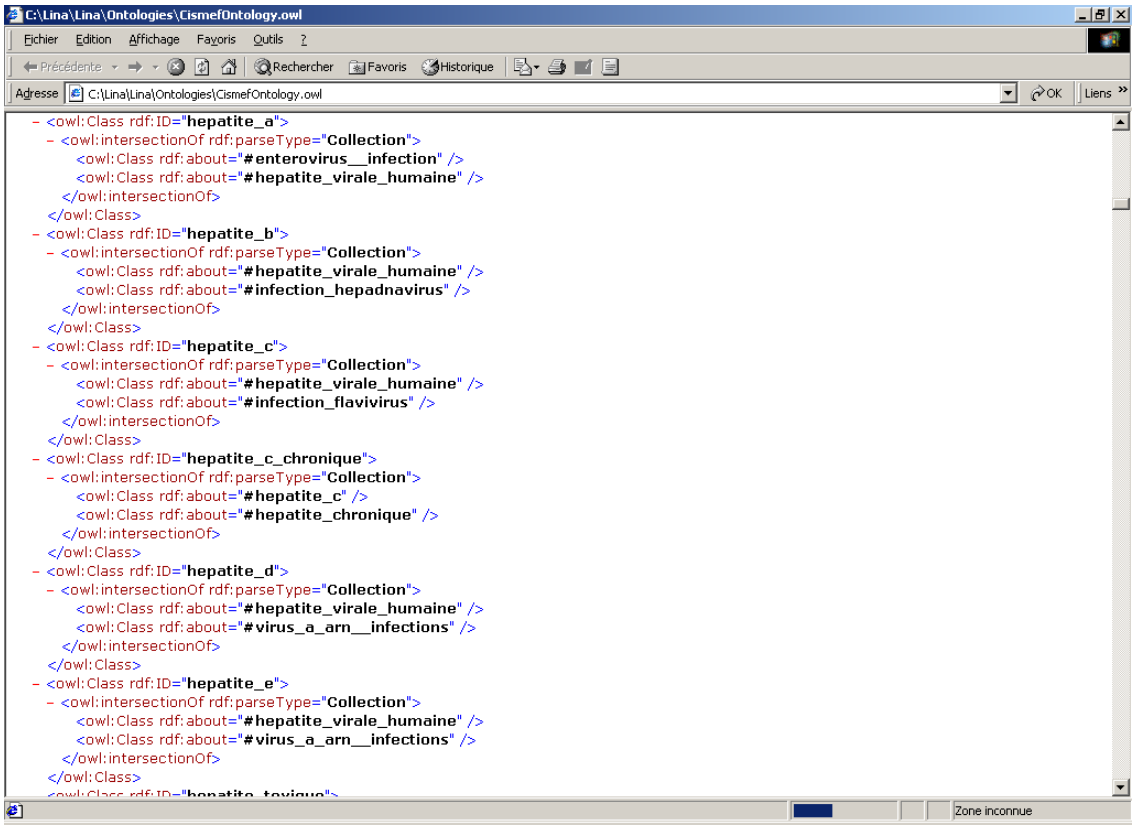


FIG.5.12. – Fichier OWL-DL en sortie du processus

5.7.7.3 La Classification

Le processus de classification prend également beaucoup de temps. En effet, le processus de classification fait appel à la vérification de consistance. Une nouvelle hiérarchie est inférée et tous les domaines et les concepts décrivant les documents sont classés en fonction de leur description. En revanche, il n'existe pas d'utilitaire dans Protégé 2000 qui permette de stocker cette nouvelle hiérarchie. En exemple, le concept *domain_qu_biosynthese* qui décrit le domaine du rôle *qu_biosynthese,* a comme description:

```
<owl:Class rdf:ID="domain_qu_biosynthese">
   <owl:unionOf rdf:parseType="Collection">
       <owl:Class rdf:about="#hormones_substituts_hormones" />
       <owl:Class rdf:about="#enzymes__coenzymes__anti_enzymes" />
       <owl:Class rdf:about="#glucides_et_hypoglycemiants" />
       <owl:Class rdf:about="#acides_amines__peptides_et_proteines" />
       <owl:Class rdf:about="#nucleosides_et_nucleotides" />
       <owl:Class rdf:about="#substances_biologiques_immunologiques" />
   </owl:unionOf>
</owl:Class>
```

Par sa description, le concept *domain_qu_biosynthese* est classé sous le concept *domain_qu_analyse* car il est plus spécifique. En gras on retrouve la description de *domain_qu_biosynthese* incluse dans celle de *domain_qu_analyse.*

```
<owl:Class rdf:ID="domain_qu_analyse">
    <owl:unionOf rdf:parseType="Collection">
       <owl:Class rdf:about="#produits_chimiques_inorganiques" />
       <owl:Class rdf:about="#composes_chimiques_organiques" />
       <owl:Class rdf:about="#composes_heterocycliques" />
       <owl:Class rdf:about="#hydrocarbures_polycycliques" />
       <owl:Class rdf:about="#hormones__substituts_hormones" />
       <owl:Class rdf:about="#agents_regulateurs_reproduction" />
       <owl:Class rdf:about="#enzymes__coenzymes__anti_enzymes" />
       <owl:Class rdf:about="#glucides_et_hypoglycemiants" />
       <owl:Class rdf:about="#lipides_et_hypolipemiants" />
       <owl:Class rdf:about="#acides_amines__peptides_et_proteines" />
       <owl:Class rdf:about="#nucleosides_et_nucleotides" />
       <owl:Class rdf:about="#agents_systeme_nerveux_central" />
       <owl:Class rdf:about="#agents_systeme_nerveux_peripherique" />
       <owl:Class rdf:about="#agents_cardiovasculaires" />
       <owl:Class rdf:about="#antiinfectieux" />
       <owl:Class rdf:about="#antineoplasiques_et_immunodepresseurs" />
       <owl:Class rdf:about="#produits_dermatologiques" />
       <owl:Class rdf:about="#substances_biologiques_immunologiques" />
       <owl:Class rdf:about="#materiaux_biomedicaux_et_dentaires" />
       <owl:Class rdf:about="#drogues_et_agents_divers" />
       <owl:Class rdf:about="#actions_chimiques_et_utilisations" />
    </owl:unionOf>
</owl:Class>
```

Le service de classification peut être utilisé lors de la mise à jour de la terminologie et des documents du catalogue.

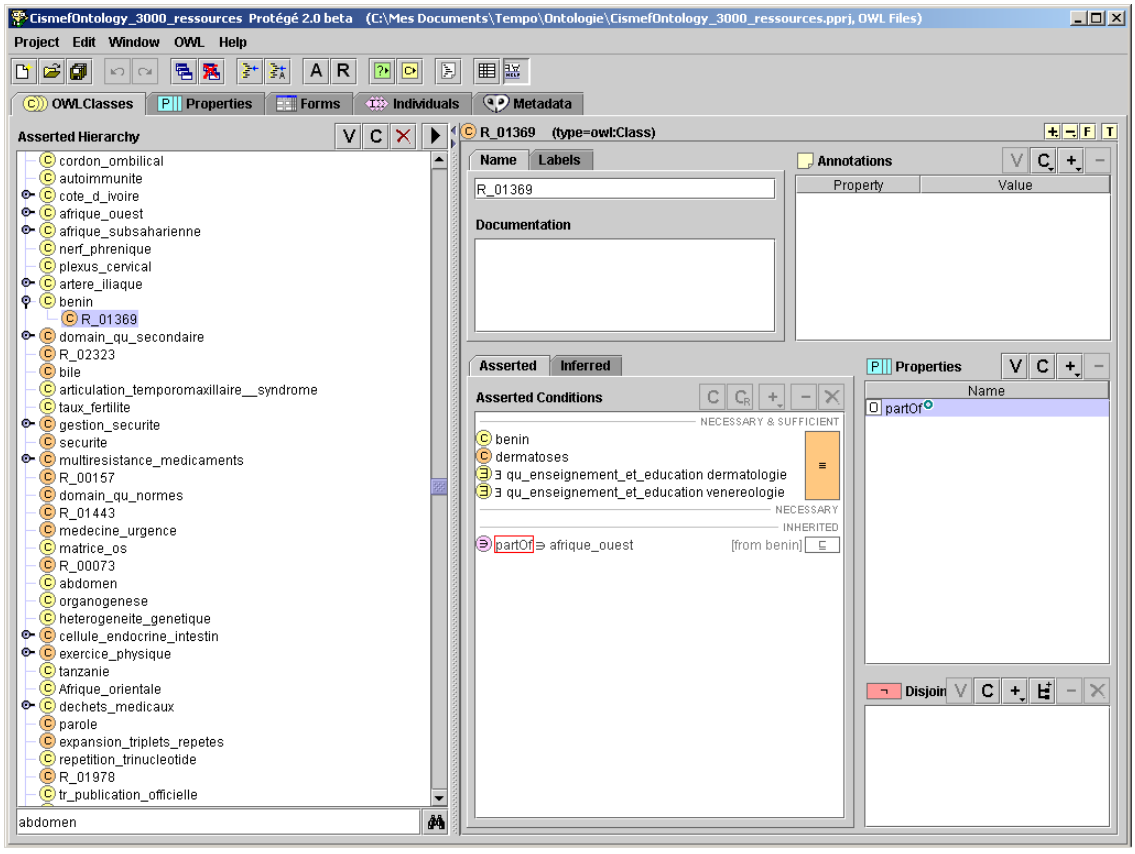


FIG.5.13. – Import sous Protégé : Hiérarchie de concepts

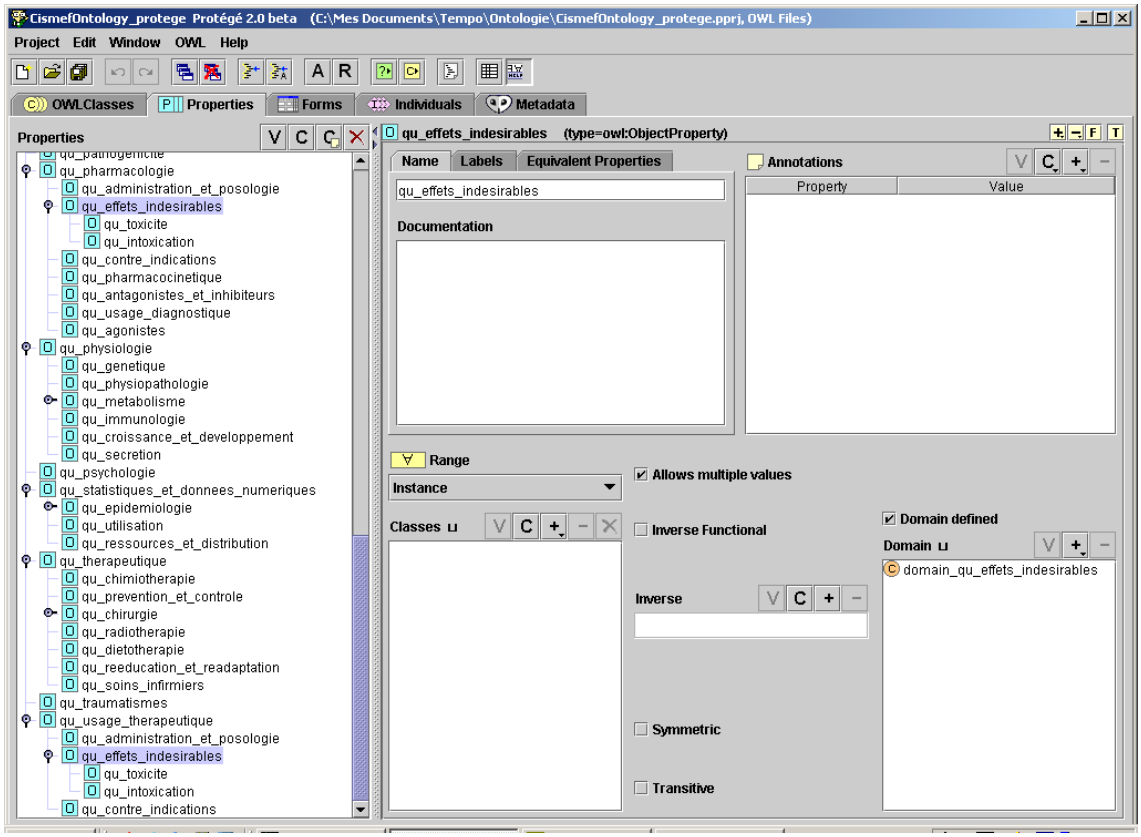


FIG.5.14. – Import sous Protégé : Hiérarchie de Rôles

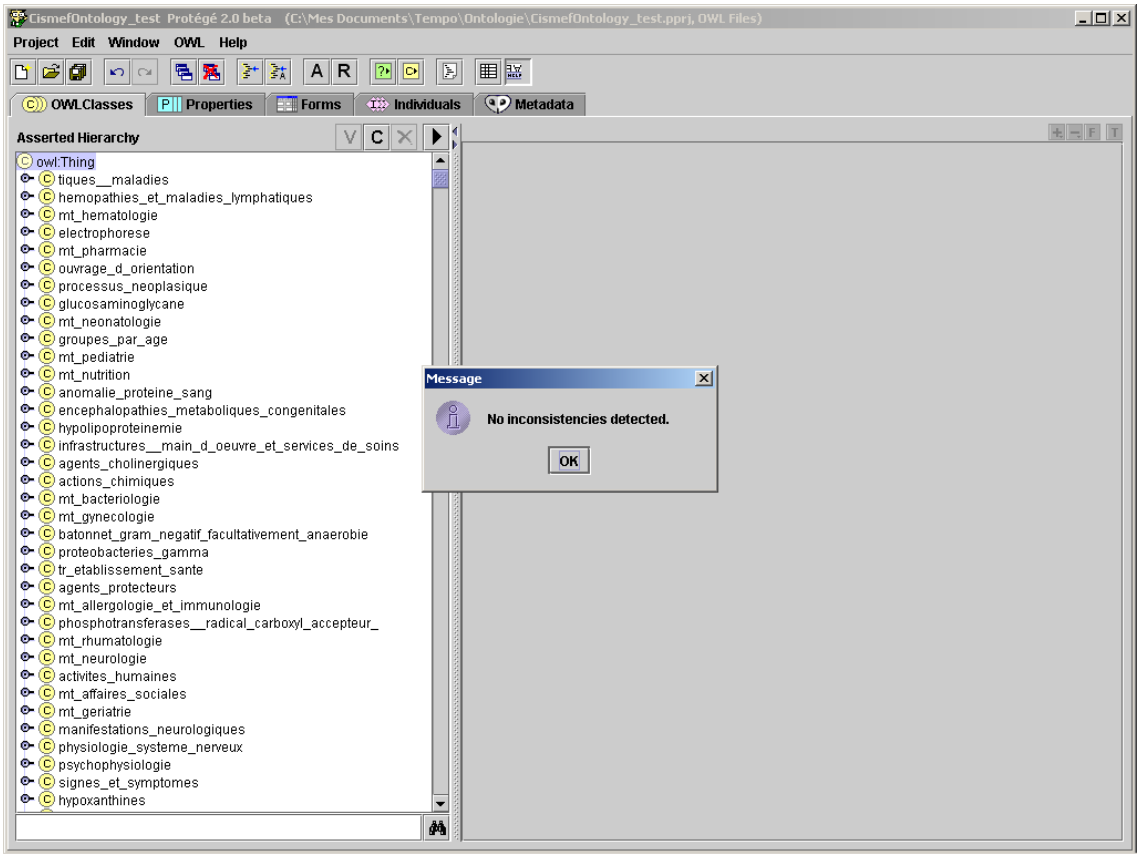


FIG.5.15. – Vérification de la Consistance

5.7.7.4 Améliorations Possibles

Les principales contributions à la traduction de la terminologie en une base de connaissances est la distinction entre les relations *is-a* et *part-of* ainsi qu'entre les concepts qui correspondent à des notions différentes. Cela permet d'éviter les ambiguïtés et les cycles dans la terminologie, notamment lorsqu'un même libellé est utilisé pour deux notions différentes.

Des améliorations sont possibles au niveau de la base de connaissances obtenue automatiquement. Par exemple, toutes les relations dans l'arborescence *Anatomie* sont considérées des relations "part of". Cependant, les sous-arborescences A11 (cellules), A12 (liquides et sécrétions biologiques) et A15 (Systèmes sanguins et immunitaires) correspondent en fait à des hiérarchies *is-a*. Par exemple une "*cellule sanguine*" [A11.118] est une "*cellule*" [A11]. Ces améliorations doivent se faire au cas par cas manuellement…

Un autre problème provient des hiérarchies en elles-mêmes qui ne correspondent pas à de vraies hiérarchies *is-a*. Par exemple, *erreur_de_diagnostic* est défini dans le MeSH, et donc dans notre base de connaissances comme un sous-concept de *diagnostic* et de *erreur_médicale* ce qui sémantiquement n'est pas correct. Le concept *erreur_de_diagnostic* devrait être défini comme une *erreur_médicale* due à un *diagnostic* ($erreur_de_diagnostic \doteq erreur_medicale \wedge \forall origine.diagnostic$).

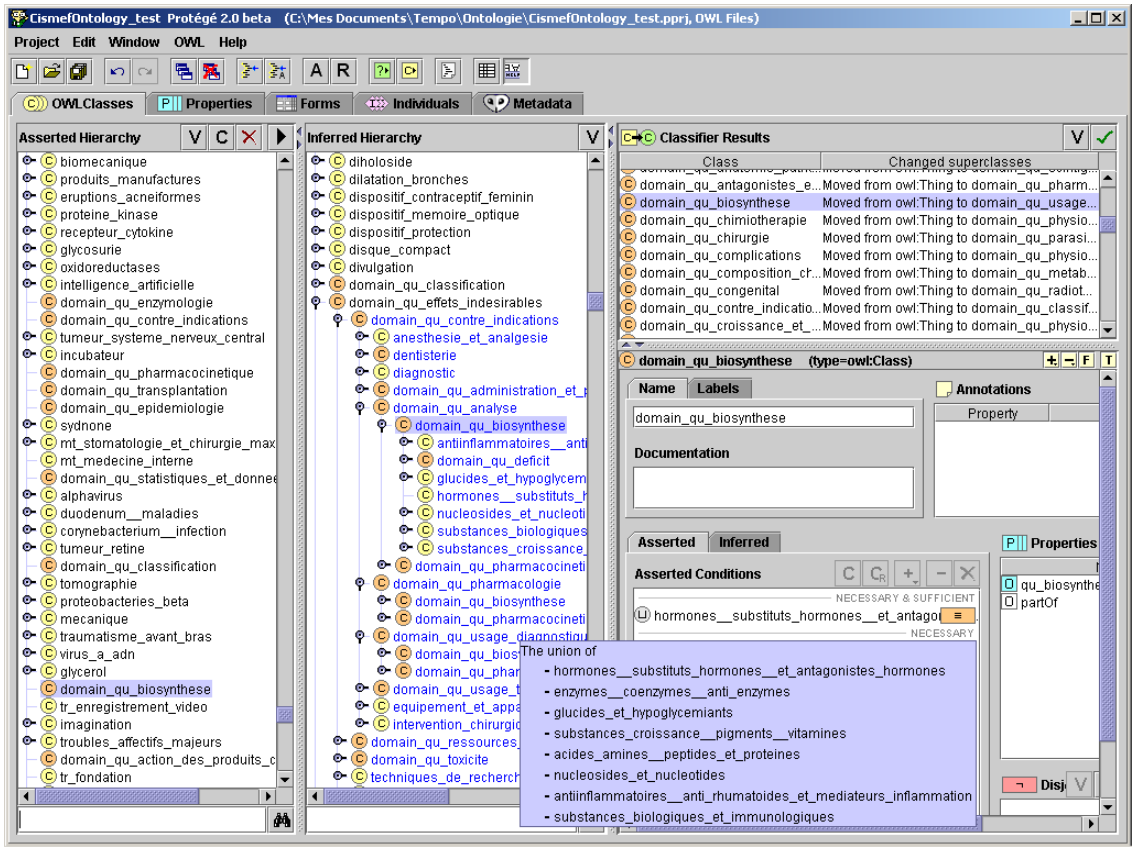


FIG.5.16. – Classification de concepts (seuls les concepts définis peuvent être classés)

D'autres propriétés, notamment les concepts décrivant l'ontologie des métadonnées relatives aux documents médicaux peuvent être ajoutés aux descriptions de concepts décrivant les documents. La description conceptuelle des documents se ferait alors à deux niveaux comme dans *(Troncy, 2004)*.

Concernant les concepts issus du MeSH, d'autres propriétés comme par exemple l'ID (numéro unique d'identification) peuvent être ajoutés aux descriptions. Nous envisageons également d'exploiter le réseau sémantique de l'UMLS. Le réseau sémantique en anglais est composé de 134 relations. Elles sont de la forme *is_complicated_by (Hepatitis, Cirrhosis)* indiquant que le concept *Hepatitis* est relié au concept *Cirrhosis* par la relation *is_complicated_by*. En analysant de plus près ces relations on remarque qu'elles correspondent aux qualificatifs du MeSH et que les concepts peuvent être mappés aux concepts du MeSH. Par exemple, la relation *is_complicated_by* correspond au qualificatif *complications* et la relation *is_treated_by* correspond au qualificatif *traitement*. La seule information qui est disponible dans notre base de connaissances est que les concepts *Cirrhose* et *Hépatite* sont tous deux subsumés par le concept *Maladies du Foie*. Les descriptions des concepts de l'arborescence Maladies pourront ainsi être complétées.

La base de connaissances peut être également exploitée dans d'autres applications que le CISMeF, notamment dans les catalogues de santé et Medline qui utilisent le MeSH *(Hersh et al., 1996) (Boyer et al., 1997) (Norman et al., 1998) (Deacon et al., 2001)*.

5.8 Vers une Ontologie ? le projet ATONANT

Il existe plusieurs méthodologies de construction d'ontologies que nous décrivons dans *(Soualmia, 2001a)*, notamment *(Noy & Hafner, 1997) (Uschold & Gruninger, 1996)(Fernanez-Lòpez, 1997)*. Une méthode consiste à construire des ressources ontologiques à partir de textes *(Assadi, 1998)(Maedche & Staab, 2000)(Aussenac-Gilles et al., 2000)*. Cette dernière a été notamment utilisée dans le projet MENELAS à partir de comptes rendus d'hospitalisation et dans *(Le Moigno et al., 2002)* pour la réanimation chirurgicale ou encore dans *(Lame, 2002)* pour une ontologie du droit. Dans le cadre du projet ATONANT (2004-2005)[43], une ontologie est une *spécification partagée d'une conceptualisation*. L'ontologie d'un domaine organise hiérarchiquement les concepts d'un domaine pour une application. Elle est construite en fonction d'une application finale dans laquelle elle sera utilisée. Le but final d'une ontologie est de construire un objet formel qui pourra être utilisé par un système informatique. L'objectif du projet est de fédérer au sein d'une plate-forme modulaire un ensemble de systèmes d'extraction et de fouille de textes associés à des fonctions d'exploration de ces données lexicales puis de modélisation et de structuration conceptuelle. Le projet s'appuie sur un certain nombre de travaux, outils et méthodes existants :

- TERMINAE, *(Biébow & Szulman, 2000)(Szulman et al., 2002)* un environnement de construction d'ontologie
- La méthode OntoSpec *(Kassel, 2002)* qui consiste à introduire une ontologie conceptuelle spécifiée dans une langue naturelle contrôlée et fortement structurée.

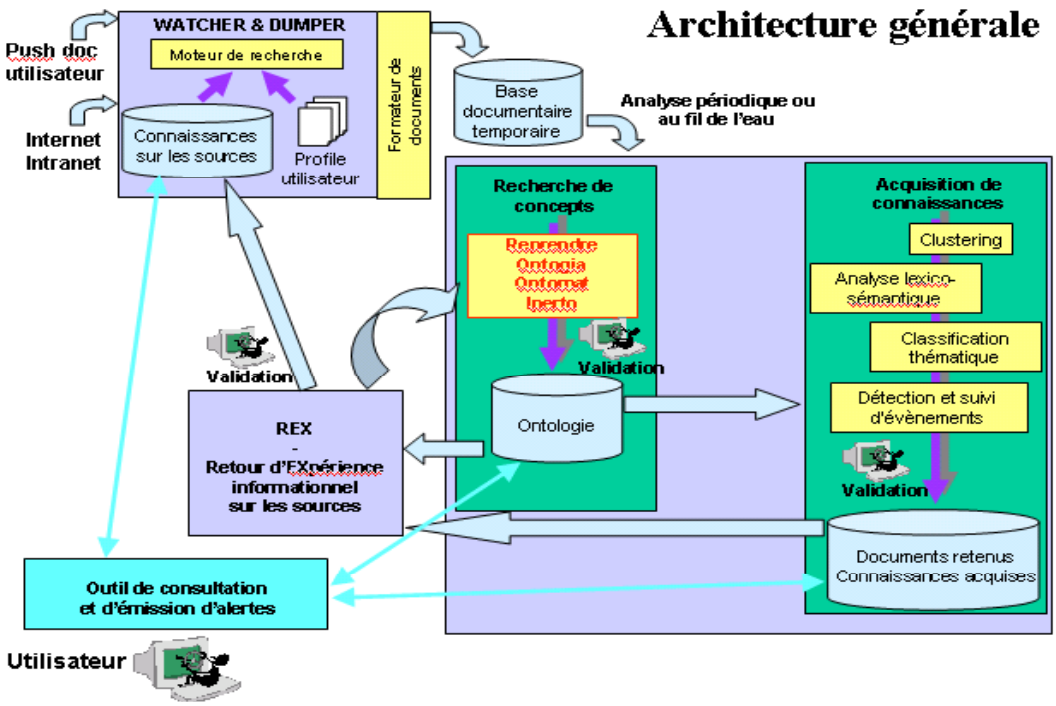


FIG.5.17. –Architecture générale de la plateforme ATONNANT

[43] Les partenaires de ce projet sont EADS, le LIPN, le LaRIA, le LIP6 et le CEA

5.8.1 Terminae

TERMINAE est une plateforme pour l'aide à la construction d'ontologies à partir de textes. L'outil permet d'exploiter des résultats d'outils de TAL pour élaborer au fur et à mesure des ressources terminologiques, allant de fiches terminologiques à une ontologie formelle.

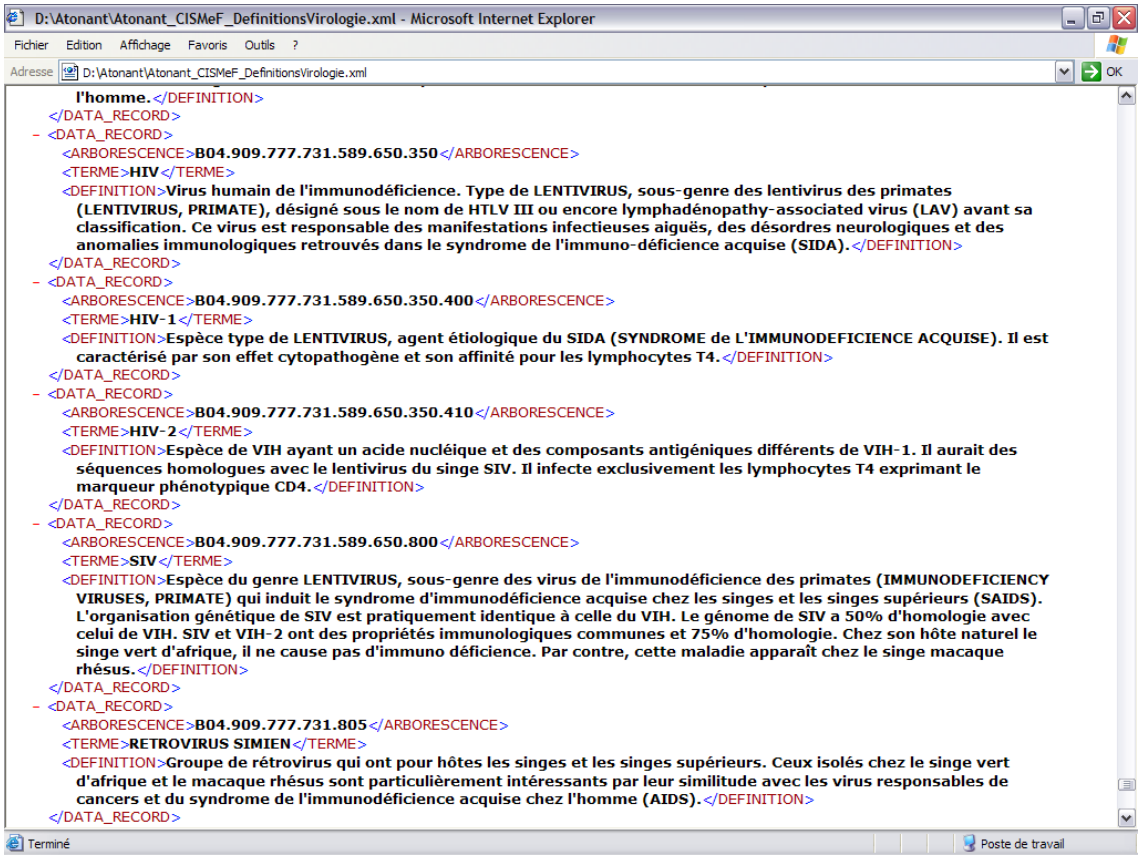
La méthode TERMINAE se réalise en quatre étapes principales :

- La constitution du corpus,
- L'étude linguistique consiste à identifier des termes et des relations lexicales en utilisant des outils TAL comme SYNTEX
- La normalisation sémantique conduit à définir des concepts et des relations sémantiques dans un langage formel
- La formalisation permet de préciser complètement et valider le modèle. Le langage de formalisation est de la famille $\mathcal{ALC}$ des logiques de description. Un module d'import d'ontologies en OWL est en cours de construction *(Szulman & Biébow, 2004)*.

5.8.2 Résultats dans le Cadre du Projet ATONANT

Un module d'aspiration et de normalisation des documents a été mis en place en amont de l'analyse de contenu afin de permettre une mise en forme des informations textuelles. Nous avons aspiré les textes intégraux relatifs à l'arborescence B04 (Virologie). Le corpus est de taille considérable. Nos choix se sont ensuite orientés vers les textes des définitions des concepts de l'arborescence virologie. En effet, B04 est la seule arborescence disposant de définitions en français. De plus les traductions ont été réalisées par la même personne, une virologue.

Nous avons donc généré un fichier XML à partir de la base de données en sélectionnant les attributs (arborescence+terme+définition). Ce fichier a été transformé par S.Szulman afin d'être chargeable dans Terminae.



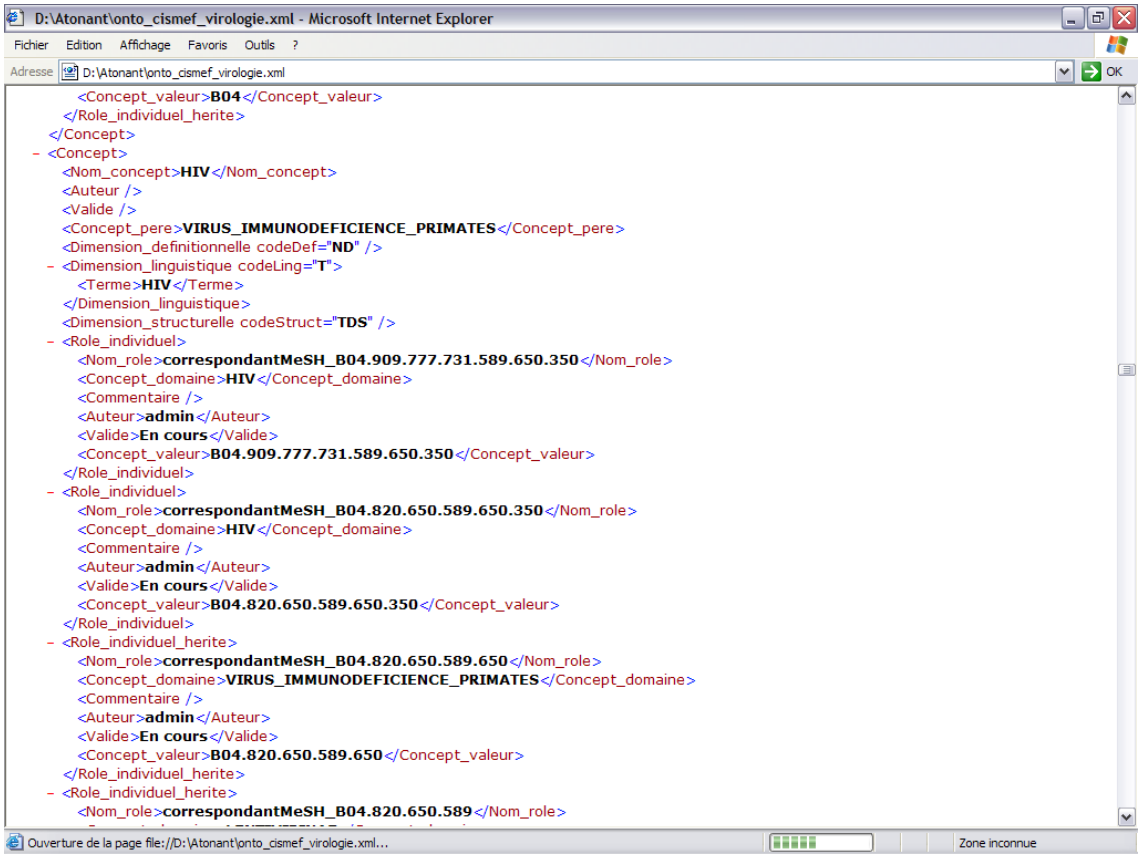


FIG.5.18. –Fichier XML généré pour l'arborescence B04 et Fichier pour Terminae

Le test de la plateforme va finalement être réalisé sur une dizaine de concepts et relations vu que l'intervention humaine va être importante pour valider l'organisation des concepts. Cette méthode suppose que l'expert connaît le domaine et qu'il peut organiser les concepts entre eux. Dans un domaine aussi vaste que la médecine, ou même d'une spécialité médicale, cela nous semble être un travail fastidieux. De plus, l'expérience montre que l'hypothèse selon laquelle l'expert d'un domaine serait le dépositaire d'un système conceptuel est restrictive. Les experts ne sont pas en mesure de verbaliser explicitement et complètement un ensemble de termes qui seraient la traduction verbale d'un système explicite de concepts. Nous avons donc réduit le nombre de concepts à 10. Le test sur une dizaine de concepts est très contraignant mais il devrait permettre de valider cette méthodologie de construction à partir des définitions de concepts.

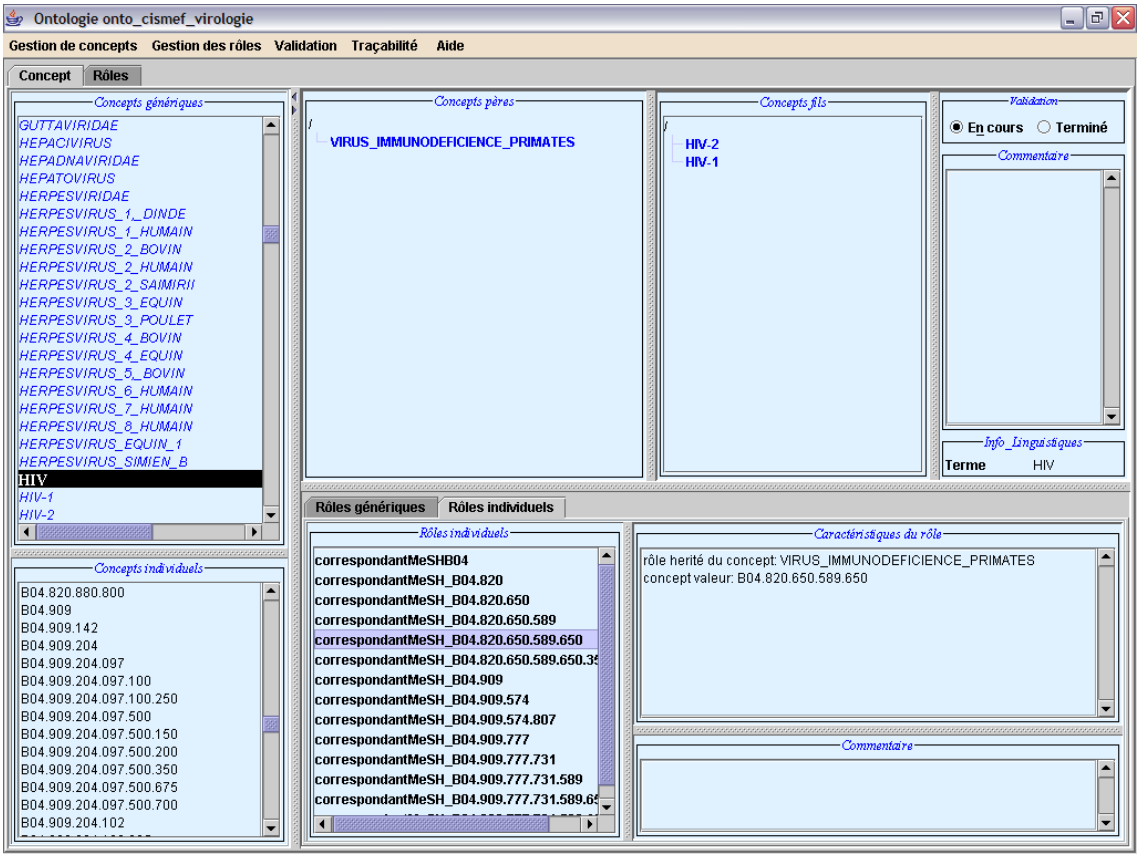


FIG.5.19. –Concepts de l'arborescence sous Terminae

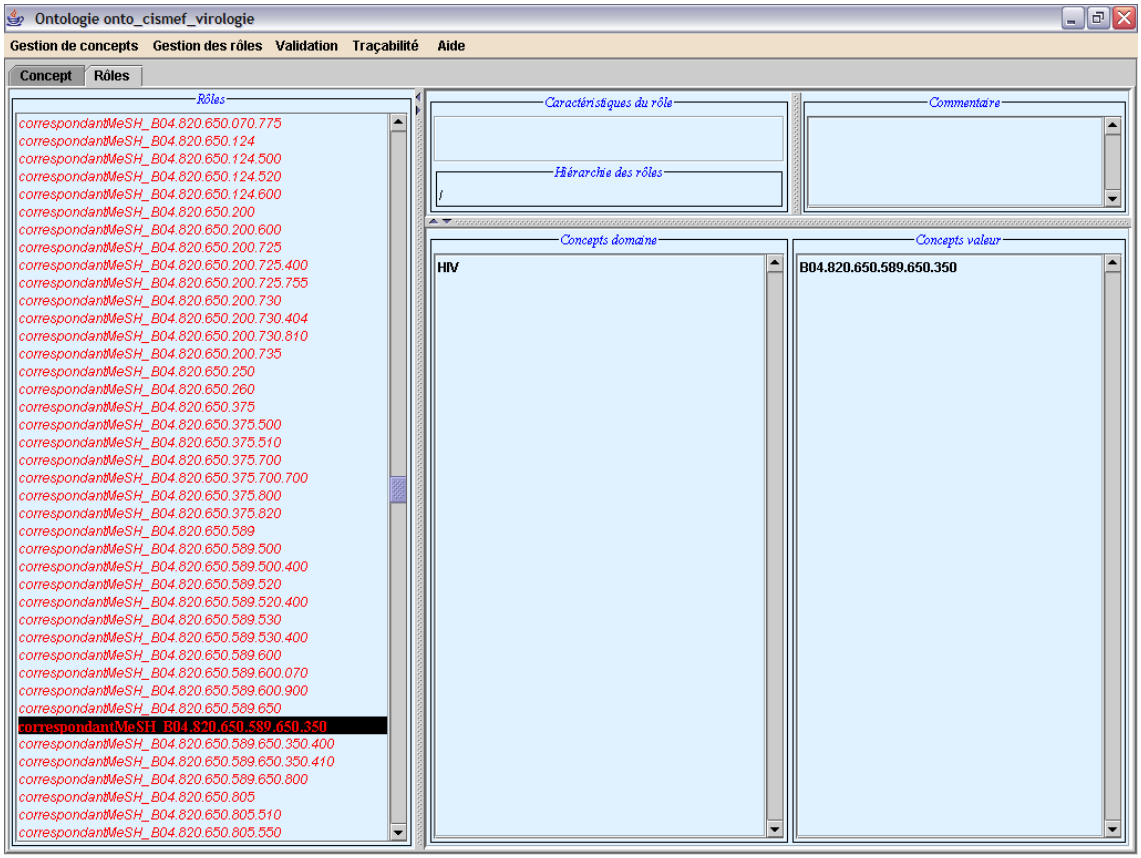


FIG.5.20. –Rôles générés

BIBLIOGRAPHIE

(*Aitken et al., 2004*) AITKEN JS., WEBBER BL., BARD JBL. (2004) Part-of relations in anatomy ontologies: a proposal for RDFS and OWL formalizations *Pacific Symposium on Bioinformatics* pp. 166-177.

(*Andreasen et al., 2004*) ANDREASEN T, JENSEN PA, J FISCHER NILSSON, PAGGIO P, SANDFORD PEDERSEN B, ERDMAN THOMSEN H. (2004) Content-based text querying with ontological descriptors, *Data & Knowledge Engineering 48*:199-219.

(*Assadi, 1998*) ASSADI H. (1998) Construction d'ontologies à partir de textes techniques- application aux systèmes documentaires. *Thèse Université Paris 6.*

(*Aussenac-Gilles & Condamines, 2004*) AUSSENAC-GILLES N., CONDAMINES A. (2004) Documents électroniques et constitution de ressources terminologiques ou ontologiques. *Information-Interaction-Intelligence, Vol. 4*, n°1, p.75-93.

(*Aussenac–Gilles et al., 2000*) AUSSENAC–GILLES N., BIEBOW B., SZULMAN S. (2000) Revisiting ontology design a methodology based on corpus analysis *EKAW 2000, Lecture Notes in AI # 1937* pp. 172-188

(*Baader et al., 2003*) BAADER F, CALVANESE D., MCGUINNESS D., et al (2003) The Description Logic Handbook: Theory, Implementation and Applications. *Cambridge University Press.*

(*Bachimont, 2000*) BACHIMONT B. (2000) Engagement sémantique et engagement ontologique : conception et réalisation d'ontologies en ingénierie des connaissances. *Ingénierie des connaissances*. pp. 305-323.

(*Baker, 2000*) BAKER T. (2000) A Grammar of Dublin Core. *Digital-Library Magazine, vol 6 n°10.*

(*Barry et al., 2001*) BARRY C., CORMIER C., KASSEL G., NOBECOURT J. (2001) Evaluation de langages opérationnels de représentation des connaissances, *IC' 2001* pp.309-327.

(*Benjamins & Fensel, 1998*) BENJAMINS V.R., FENSEL, D. (1998) The Ontological Engineering Initiative (KA)2. In *Frontiers in Artificial Intelligence and Applications*, Vol 46. p. 287-301.

(*Berners-Lee et al., 1998*) BERNERS-LEE T, FIELDING R, MASINTER. (1998) Uniform Resource Identifiers URI generic Syntax internet draft standard *RFC 2398*.

(*Berners-Lee et al., 2001*) BERNERS-LEE T., HEUDLER J., LASSILA O. (2001) The Semantic Web. *Scientific American, 284*(5):34-43.

(*Biebow & Szulman, 2000)* BIEBOW B., SZULMAN S. (2000) Terminae une approche terminologique pour la construction d'ontologies du domaine à partir de textes. *RFIA reconnaissance des formes et intelligence artificielle vol II* P 81- 90.

(*Bouaud et al., 1995*) BOUAUD J., BACHIMONT B., CHARLET J. AND ZWEIGENBAUM P. (1995) Methodological Principles for Structuring an « Ontology ». *Proceedings of IJCAI conference*.

(*Boyer et al., 1997*) BOYER C., BAUJARD O., BAUJARD V., et al. (1997) Health On the Net automated database of Health and medical information. *IJMI 47*(1-2):27-9.

(*Brachman & Schmolze, 1985*) BRACHMAN R. J, SCHMOLZE J. G. (1985) An Overview of the KL-ONE Representation System. In *Cognitive Science, Vol. 9*. p 171-216.

(*Brachman et al., 1999*) BRACHMAN RJ., BORGIDA A., MC GUINESS DL., PATEL-SCHNEIDER PF. (1999) Reducing Classic to Practice : Knowledge Representation Theory Meets Reality. *Artificial Intelligence 114*(1-2) pp 203-237

(*Brachman, 1977*) BRACHMAN, R. J. (1977) What's in Concept: Structural Foundations for Semantic Networks. In *International Journal of Man-Machine Studies, Vol . 9*. p 127-152.

(*Buckingham et al., 2000*) BUCKINGHAM SHUM, S., MOTTA, E., DOMINGUE, J. (2000) ScholOnto: an ontology-based digital library server for research documents and discourse. In *International. Journal on Digital Libraries*, Vol. 3, N°3. p 237-248.

(*Charlet, 2003*) CHARLET J. (2003) L'ingénierie des connaissances, développements, résultats, et perspectives pour la gestion des connaissances médicales. *HDR 2003*.

(*Chein & Mugnier, 1992*) CHEIN M., MUGNIER M-L. (1992) Conceptual Graphs: Fundamental Notions. In *Revue d'Intelligence Artificielle, 1992, Vol. 6-4*. p 365-406.

(*Chevallet, 1992*) CHEVALLET JP. (1992) Un modèle logique de recherche d'informations appliqué au formalisme des graphes conceptuels - Le prototype ELEN et son expérimentation sur un corpus de composants logiciels. *Thèse de doctorat : Université Joseph Fourier*.

(*Cornet & Abu-Hanna, 2002*) CORNET R., ABU-HANNA A. (2002) Usability of Expressive Description Logics – A Case Study in UMLS. *AMIA 2002*, 180-184.

(*Deacon et al., 2001*) DEACON P., SMITH JB., TOW S. (2001) Using metadata to create navigation paths in the HealthInsite Internet gateway. *Health Info Libr J. 18* (1) pp: 20-9.

(*Desmontils & Jacquin, 2002*) DESMONTILS E., JACQUIN C. (2002) Indexing a Web Site with a Terminology Oriented Ontology, In *The Emerging Semantic Web*, I.F. Cruz, S. Decker, J. Euzenat and D. L. McGuinness (Ed.), IOS Press, pp.181-197.

(*Domingue & Motta, 2000*) DOMINGUE J.B. , MOTTA E. (2000) Planet-Onto: From News Publishing to Integrated Knowledge Management Support. In *IEEE Intelligent Systems,* Vol. 15, N°3. pp.26-32.

(*Evett et al., 1993*) EVETT M.P., ANDERSEN W.A., HENDLER J.A. (1993) Massively Parallel Support for efficient Knowledge Representation. In Proceeding of the *13th International Joint Conference on Artificial Intelligence*, pp. 1325-1331.

(*Fensel et al., 1998*) FENSEL, D., DECKER, S., ERDMANN, M., STUDER, R. (1998) Ontobroker: The Very High Idea. In Proceedings of the *11th International Flairs Conference (FLAIRS-98)*.

(*Fensel et al., 2000*) FENSEL D., HORROCKS I., VAN HARMELEN F. (2000) OIL in a Nutshell. In *Proceedings of the 12th European Knowledge Acquisition Workshop, (EKAW 2000)*, pp 1-16.

(*Fernàndez-Lòpez et al., 1997*) FERNÀNDEZ-LÒPEZ M., GÒMEZ-PÉREZ, JURISTO M. (1997) Methontology : from Ontological Arts Towards Ontological Engineering. Workshop on Ontological Engineering, *AAAI Spring Symposium* p 33-40.

(*Gandon et al., 2002*) GANDON F., DIENG-KUNTZ R., CORBY O. & GIBOIN A. (2002) Web Sémantique et Approche Multi-Agents pour la Gestion d'une Mémoire Organisationnelle Distribuée. *Journées Ingénierie des Connaissances,* p.15-26.

(*Gandon, 2002*) GANDON F. (2002) Distributed artificial intelligence and knowledge management: ontologies and multi-agent systems for a corporate semantic web. *Phd Université de Nice*.

(*Golbeck et al., 2003*) GOLBECK J., FRAGOSO G., HARTEL F., et al. (2003) The National Cancer Institute's Thesaurus and Ontology. *Journal of Web Semantics*.

(*Gruber, 1993*) GRUBER TR. (1993) A Translation Approach to Portable Ontology Specifications. In *Knowledge Acquisition, Vol. 5*, N° 2. p199-220.

(*Guarino et al., 1999*) GUARINO N., MASOLO, C., VETERE, G. (1999) Ontoseek: content-Based Access to the Web. In *IEEE Intelligent Systems, Vol. 14*, N° 3. p 70-80.

(*Haarslev & Möller, 2001*) HAARSLEV V., MÖLLER R. (2001) Description of the RACER System and its Applications. In International *Workshop on Description Logics 2001 (DL2001)*.

(*Haarslev & Möller, 2003*) HAARSLEV V., MÖLLER R.(2003) Racer a core inference engine for the semantic web, *International Semantic Web Conference,* pp.27-36

(*Hacid et al., 2000*) HACID MS., SOUALMIA LF., TOUMANI F. (2000) Schema Extraction for Semi structured Data. *International Workshop on Description Logics*. pp 133-142.

(*Heflin et al., 1999*) HEFLIN, J., HENDLER, J., LUKE, S. (1999) Applying Ontology to the Web: A Case Study. *International Work-Conference on Artificial and Natural Neural Networks (IWANN'99)*. p 715-724.

(*Hendler & Mc Guiness,)* HENDLER J., MC GUINESS DL the DARPA Agent Markup Language: *IEEE Intelligent Systems 15*(6): 67-73

(*Hendrix, 1978*) HENDRIX, G. (1978) The Representation of Semantic Knowledge. In *Understanding Spoken Language. Edited by D.E. Walker*, p 121-226.

(*Hersh et al., 1996*) HERSH WR., BROWN KE., DONOHOE LC., et al.(1996) CliniWeb: managing clinical information on the World Wide Web. *JAMIA, 3*(4):273-80.

(*Horrocks & Tessaris, 2000*) HORROCKS I, TESSARIS S. (2000) A conjunctive query language for description logic ABoxes. *AAAI/IAAI* p 399-404

(*Horrocks & Rector, 1997*) HORROCKS I., RECTOR A. (1997) Experience Building a Large, Re-usable Medical Ontology using a Description Logic with Transitivity and Concept Inclusions. *Workshop on Ontological Engineering AAA Spring Symposium*.

(*Horrocks &Patel-Schneider, 2003*) HORROCKS, PATEL-SCHNEIDER (2003) Reducing OWL entailment to description logic satisfiability. *ISWC' Lecture Notes in Computer Science# 2870* pp.17-29

(*Horrocks et al., 2003*) HORROCKS I., PATEL-SCHNEIDER PF., VAN HARMELEN F. (2003) From SHIQ and RDF to OWL: The making of a web ontology language. *Journal of Web Semantics. 1*(1):7-26.

(*Horrocks, 1998*) HORROCKS I. (1998) Using expressive description logic : fact or fiction. *Principles of knowledge representation and reasoning. KR'98* pp.636-649

(*Hu et al., 2003*) HU B., DASMAHAPATRA S., SHABOT N. (2003) From Lexicon to mammographic Ontology : experiences and lessons. *International Workshop on Description Logics*.

(*Kashyap & Borgida, 2003*) KASHYAP V., BORGIDA A. (2003) Representing the UMLS Semantic Network using OWL. *International Semantic Web Conference*.

(*Kifer et al., 1995*) KIFER M., LAUSEN G., WU J. (1995) Logical Foundations of Object-Oriented and Frame Based Languages. In *Journal of the ACM,* Vol. 42, N°4. pp.741-843.

(*Lame, 2002*) LAME G. (2002) Construction d'ontologies à partir de textes : une ontologie du droit dédiée à la recherché d'information. *Thèse Doctorat Ecole des Mines Paris*.

(*Lassila & Swick, 1999*) LASSILA O., SWICK R. (1999) Resource Description Framework (RDF) Model and Syntax Specification. *W3C Candidate Recommandation*.

(*Laublet et al., 2002*) LAUBLET P., REYNAUD C., CHARLET J. (2002). Sur Quelques Aspects du Web Sémantique. *Actes des deuxièmes assises nationales du GdRI3*, p.59-78.

(*Le Moigno et al., 2002*) LE MOIGNO S., CHARLET J., BOURIGAULT D., DEGLOUET P., JAULENT MC. (2002) Terminology extraction from texts to build an ontology in surgical intensive care. Annual *symposium of the American Informatics Association AMIA* p.430-434.

(*Lindberg Dab et al., 1993*) LINDBERG DAB, HUMPHREYS BL, MCCRAY AT. (1993) The Unified Medical Language System. *Meth Inform Med, 32*(4): 281-91

(*Macgregor, 1988*) MACGREGOR R. M. (1988) A Deductive Pattern Matcher. In Proceedings of the *7th National Conference on Artificial Intelligence*, St. Paul, MN. p 403-408.

(*Maedche & Staab, 2000*) MAEDCHE A, STAAB S. (2000) Mining ontologies from texts. Proceedings of the *international conference EKAW 2000* .

(*Martin & Eklund, 2000*) MARTIN, P., EKLUND P. (2000) Knowledge Indexation and Retrieval and the Word Wide Web. In *IEEE Intelligent Systems, special issue "Knowledge Management and Knowledge Distribution over the Internet"*

(*Mayer et al., 2003*) MAYER MA., DARMONI SJ., FIENE M., KÖHLER C., & AL. (2003) MedCIRCLE - Modeling a Collaboration for Internet Rating, Certification, Labelling and Evaluation of Health Information on the Semantic World-Wide-Web. *Medical Informatics Europe* p.667-672.

(*Meghini et al., 1997)* MEGHINI C, SEBASTIANI F, STRACCIA U. (1997) Modelling the retrieval of structured documents containing texts and images. *Lecture Notes in Computer Science # 1324* pp 325-344

(*Michard, 2001*) MICHARD, A. (2001) Web mining : Des portails coopératifs au Web sémantique : les applications de RDF. In Tutorial du *XIXème Congrès Informatique des Organisations et Systèmes d'Information et de Décision (INFORSID).*

(*Miller, 1990*) MILLER GA. (1990) Nouns in WordNet: a lexical inheritance system. In *International Journal of Lexicography*, *Vol. 3*, N°4. p 245 - 264.

(*Minsky, 1975*) MINSKY, M. (1975) A Framework for Representing Knowledge. In The *Psychology of Computer Vision*, pp 211-281.

(*Motta et al., 2000*) MOTTA E., BUCKINGHAM-SHUM, S. AND DOMINGUE, J. (2000) Ontology-Driven Document Enrichment: Principles, Tools and Applications. In *International Journal of Human-Computer Studies*, Vol. 52. pp.1071-1109.

(*Motta, 1998*) MOTTA E. (1998) Reusable Components for Knowledge Models. *PhD Thesis : The Open University (UK).*

(*Nebel, 1990)* NEBEL B. (1990) Reasoning and revision in hybrid representation systems *Lecture Notes in Artificial Intelligence#422*.

(*Norman, 1998*) NORMAN F. (1998) Organising Medical Networks' information. *Med. Inf. 23*:43-51.

(*Noy & Hafner, 1997*) NOY NF, HAFNER CD. (1997) The State of the Art in Ontology Design: a Survey and Comparative Review. *AI Magazine 18*(3):53-74.

(*Noy et al., 2001*) NOY NF., SINTEK M., DECKER S., et al. (2001) Creating Semantic Web Contents with Protege-2000. *IEEE Intelligent Systems 16*(2):60-71.

(*Quillian, 1968*) QUILLIAN, M.R.. Semantic Memory. In *Semantic Information Processing*, Edited by M. Minsky. Cambridge: MIT Press. pp. 227-270.

(*Rodrigues et al., 1998*) RODRIGUES JM., TROMBERT-PAVIOT B., BAUD R., WAGNER J., MEUSINET-CARRIOT F. (1998) GALEN-In-Use : using Artificial Intelligence Terminology Tools to Improve the Linguistic Coherence of a National Coding System for Surgical Procedures. Cesnik et al. (eds). *MedInfo'1998.*

(*Roussey et al., 2001*) ROUSSEY C., CALABRETTO S., PINON J-M. (2001) A new Conceptual Graph Formalism Adapted for Multilingual Information Retrieval Purposes. In *Proceedings of the 12th International Conference on Database and Expert Systems Applications (DEXA'2001)*; pp.92-101.

(*Schank & Abelson, 1977*) SCHANK RC., ABELSON RP. (1977) Scripts, Plans, Goals and Understanding: an Inquiry into Human Knowledge Structures. *Hillsdale NJ : Erlbaum.*

(*Schmolze & Lipkis, 1983*) SCHMOLZE JG., LIPKIS T. (1983) Classification in the KL-ONE Knowledge Representation System. 8th *International Joint Conference on Artificial Intelligence*, pp.330-332.

(*Schreiber et al., 1997*) SCHREIBER ATH., VAN HEIJST G., WIELINGA BJ. (1997) Using explicit ontologies in KBS development, *International Journal of Human-Computer Studies*, 45, pp. 183-292.

(*Schulz & Hahn, 2001*) SCHULZ S. HAHN U. (2001) Medical Knowledge Re-engineering – converting major portions of the UMLS into a terminological knowledge base. *International Journal in Medical Informatics, 64*(2-3):207-221.

(*Sintek & Decker, 2001*) SINTEK M., DECKER S. (2001) TRIPLE- An RDF Query, Inference and Transformation Language. Proceedings of *Deductive Databases and Knowledge Management Workshop*.

(*Soualmia & Darmoni, 2003 a*) SOUALMIA LF., DARMONI SJ. (2003 a) Une Terminologie Orientée Ontologie pour la Recherche d'Information sur la Toile. *Journées Francophones de la Toile*. pp. 185-194.

(*Soualmia & Darmoni, 2003 b*) SOUALMIA LF., DARMONI SJ. (2003 b) Projection de Requêtes pour une Recherche d'Information Intelligente sur le Web; *Rencontres Jeunes Chercheurs en Intelligence Artificielle*.

(*Soualmia et al., 2003*) SOUALMIA LF., BARRY C., DARMONI SJ. (2003) "Knowledge-Based Query Expansion over a Medical Terminology Oriented Ontology"; *Lecture Notes in Artificial Intelligence# 2780*; pp 209-213.

(*Soualmia et al., 2004*) SOUALMIA LF., DAHAMNA B., DARMONI SJ. (2004) Représentation du thésaurus MeSH et de la terminologie CISMeF en OWL; *Journée Web Sémantique Médical 2004*.

(*Soualmia et al., 2004*) SOUALMIA LF., GOLBREICH C., DARMONI SJ. (2004) Representing the MeSH in OWL: Towards a Semi-Automatic Migration; KR-MED, *International Workshop on Formal Biomedical Knowledge Representation*, pp 72- 80.

(*Soualmia, 2001 a*) SOUALMIA LF. (2001 a) "Description Logics : a Modelling Support for Automatic Cataloguing in E-Commerce"; In Proceedings of the *8th Research Symposium on Emerging Electronic Markets, RSEEM'01*. pp 73-86.

(*Soualmia, 2001 b*) SOUALMIA LF. (2001 b) Les Logiques de Description et le Commerce Electronique. Actes de la *Conférence Internationale des Nouvelles Technologies de l'Information et de la Communication, NîmesTIC'2001*.

(*Sowa, 1984*) SOWA, J. (1984) Conceptual Structures: information processing in mind and machine. In *The System Programming Series*, Reading: Addison Wesley publishing Company.

(*Sowa, 2000*) SOWA JF. (2000) Ontology, Metadata and Semiotics. B.Ganter, G.W.Mineau (Eds), Conceptual Structures: Logical, Linguistic, and Computational Issues, *Lecture Notes in AI #1867*, pp.55-81.

(*Stuckenschmidt & Van Harmelen et al., 2002*) STUCKENSCHMIDT H., VAN HARMELEN F. (2002) Approximating terminological queries. Proceeding of *FQAS'2002,Flexible query answering systems*.

(*Szulman et al., 2002*) SZULMAN S., BIEBOW B., AUSSENAC GILLES N. (2002) Structuration de terminologies à l'aide d'outils TAL avec Terminae. *TAL* vol 43 n°1/pp.103-128.

(*Troncy, 2004*) TRONCY (2004) Formalisation des connaissances documentaires et des connaissances conceptuelles à l'aide d'ontologies : application à la description de documents audiovisuels. *Thèse de l'université de Grenoble*.

(*Uschold & Gruninger, 1996*) USCHOLD, M., GRUNINGER, M. (1996) Ontologies: Principles, Methods and Applications. In *Knowledge Engineering Review, Vol. 11*, N° 2. pp.93-155.

(*Woods, 1975*) [WOOD75] WOODS, W. A. (1975) What's in a Link: Foundation for Semantic Networks. In *Representation and Understanding: Studies in Cognitive Science*, Edited by D. Bobrow, A. Collins. New York: Academic Press, pp.35-82.

(*Wroe, 2003*) WROE C.J., STEVENS R., GOBLE C.A., ASHBURNER M. (2003) A Methodology To Migrate The Gene Ontology To A Description Logic Environment Using DAML+OIL. Proceedings of the *8th Pacific Symposium on Biocomputing (PSB)*.624-635

[RDFS] Resource Description Framework (RDF) Schema Specification.W3C Working Draft. http://www.w3.org/TR/WD-rdf-schema

[XML] eXtensible Markup Language (XML) 1.0. W3C Recommendation 15 june 1998. http://www.w3.org/TR/REC-XML

CONCLUSION GENERALE

Nous nous sommes intéressés dans le cadre de cette thèse à la recherche d'information sur le Web. Nous avons présenté les différents modèles de recherche d'information, les différentes mesures d'évaluation d'un système de recherche d'information ainsi que principaux problèmes relatifs à la recherche d'information. Nous travaillons dans le contexte 'appliqué' de la recherche de documents en santé. Nous avons proposé une recherche d'information fondée sur l'exploitation conjointe de différents types de connaissances.

Le premier type de connaissances est relatif à la morphologie, le second type à la fouille de données et enfin le troisième type de connaissances est relatif au raisonnement terminologique. Nous avons pour cela construit automatiquement les bases de connaissances en fonction des données disponibles dans le catalogue CISMeF, contexte de nos expérimentations.

La base de connaissances morphologiques a été construite à partir du vocabulaire de la terminologie CISMeF et d'un lexique général du français. Nous avons montré que cette méthode permettait de couvrir le vocabulaire à 92,77% avec une très bonne précision. L'utilisation de liens de synonymie, ou de hiérarchie, au sein même de la terminologie est une autre méthode de construction de lexique. Dans notre contexte ces liens sont exploités dans le processus de recherche d'information et ce lexique n'apporterait pas d'information supplémentaire.

Cette base de connaissances morphologiques a permis d'améliorer le processus de recherche d'information avec des résultats très encourageants. Ces connaissances ont été complétées par des traitements de phonémisation des requêtes des utilisateurs afin de les apparier au mieux avec notre vocabulaire, dans le cas de requêtes mal orthographiées.

La base de règles d'associations à été construite à l'aide d'un processus de fouille de données, fondé sur une analyse formelle de treillis de concepts. Nous avons montré comment il était possible d'extraire différents types de règles d'associations à partir des documents de la base en fonction de contextes divers, dépendant essentiellement des unités de connaissances à extraire. Nous avons pour cela adapté l'algorithme A-Close d'extraction de règles d'association à partir du treillis des itemsets fermés fréquents à nos données. Nous avons également introduit un nouveau paramètre afin de gérer la taille maximale des générateurs. Les règles d'association extraites sont plus précises que simples associations entre termes puisqu'elles peuvent être qualifiées et pondérées (avec la notion de majeur/mineur). L'évaluation d'un ensemble de règles d'association par un expert a montré que cette méthode permet de générer de nouvelles connaissances intéressantes, et a permis à l'expert de modéliser d'autres règles, dites règles expertes.

Leur exploitation dans un processus de recherche d'information interactif avec l'utilisateur permet de préciser leurs requêtes. Les propositions de termes à ajouter aux requêtes à partir des règles d'association ont été jugées *utiles* par 29,2% et *très utiles* par 62,5% des utilisateurs (Annexes). Ces règles peuvent également être utilisées dans un processus d'indexation automatique.

Enfin, nous avons proposé une méthode permettant de modéliser et de formaliser automatiquement les connaissances terminologiques du CISMeF à l'aide du langage OWL-DL, standard du Web Sémantique, mais également logique de description. Le but est de pouvoir exploiter des unités de connaissances formelles dans le processus de recherche d'information à l'aide de raisonnement et d'inférences, mais également de pouvoir vérifier la consistance de la terminologie, et de ce fait la consistance des requêtes des utilisateurs avec une explication quant à la cause d'une éventuelle inconsistance. Notre modélisation a permis de corriger les descriptions susceptibles de contenir des cycles en levant les ambiguïtés qui y étaient présentes, notamment par la distinction entre les notions.

Cette base de connaissances peut être complétée par les connaissances extraites par le processus de fouille de données en les représentant sous la forme de règles d'inférence. Cette formalisation peut être exploitée par d'autres portails utilisant le MeSH comme référence, conduisant ainsi vers un Web Sémantique médical.

Ces approches à base de connaissances se complètent dans le but d'une recherche d'information intelligente dans le contexte du Web Sémantique. En effet, dans la mesure où les connaissances morphologiques et les autres traitements linguistiques favorisent des traitements de base sur les requêtes elles-mêmes. Sans eux, les autres traitements à base de connaissances extraites ou d'inférences ne peuvent pas être réalisés. De même, l'extraction de connaissances permet de compléter la base de connaissances terminologiques à l'aide de règles d'inférence.

En termes de perspectives de ces travaux, les connaissances morphologiques pourront être complétées à l'aide du lexique UMLF dès qu'il sera exploitable. Ce lexique a été généré à partir d'une partie du corpus de CISMeF pour une spécialité donnée. On peut envisager de phonémiser également la base de connaissances morphologiques pour corriger les requêtes employant des dérivations et flexions n'étant pas orthographiées correctement. Toutes ces bases seront exploitables pour l'aide à l'indexation automatique.

Concernant la fouille de données, on peut utiliser d'autres indices statistiques pour la sélection de règles d'associations de qualité. En terme de recherche d'information interactive, l'affichage doit évoluer vers une navigation dans le contexte, au lieu d'une simple liste de concepts à sélectionner pour compléter les requêtes. On peut envisager de générer des règles d'association généralisées en exploitant les hiérarchies de concepts. L'extraction de règles d'associations floues entre les concepts en mesurant le degré d'appartenance d'un concept à un document, est encore une autre perspective.

Concernant la représentation des connaissances, outre l'attente des résultats du projet Atonant, on peut compléter les descriptions des concepts en exploitant le réseau sémantique de l'UMLS, ou encore à l'aide des règles d'association extraites. Celles-ci pourront être incluses dans une IBox ou bien représentées à l'aide du nouveau langage ORL (OWL Rule Language)[44] s'il devient un standard. En effet, les règles métier associent un concept à un autre peuvent être

[44] Horrocks I, Patel-Schneider PF, A proposal for an OWL Rules Language, *WWW Conference 2004*.

comparées à des règles de Horn (implémentées notamment dans le système PICSEL). Cependant, les règles d'association générées ne peuvent pas être assimilables à ces règles de Horn. En effet, l'antécédent et le conséquent sont composés de conjonctions de descriptions de concepts, d'où l'intérêt d'un nouveau langage pour exprimer des règles dans l'infrastructure du Web Sémantique. Dès lors que toutes ces conditions seront réunies, nous espérons que des inférences complexes, au-delà de l'utilisation de la simple relation de subsomption, pour être réalisables.

Liste des critères de qualité de l'information de santé sur l'Internet selon le Net Scoring.

1 Crédibilité (sur 99 points)	1.1 Source 1.1a Nom, logo et références de l'institution sur chaque document du site (critère essentiel) 1.1b Nom et titres de l'auteur sur chaque document du site (critère essentiel) 1.2. Révélation 1.2a Contexte : source de financement, indépendance de l'auteur (critère essentiel) 1.2b Conflit d'intérêt (critère important) 1.2c Influence, biais (critère important) 1.3 Mise à jour : actualisation des documents du site avec date de création, date de dernière mise à jour et éventuellement date de dernière révision (critère essentiel) 1.4 Pertinence / utilité (critère essentiel) 1.5 Existence d'un comité éditorial (critère essentiel) 1.5a Existence d'un administrateur de site ou maître-toile (critère important) 1.5b Existence d'un comité scientifique (critère important) 1.6. Cible du site Internet ; accès au site (libre, réservé, tarifé) (critère important) 1.7. Qualité de la langue (orthographe et grammaire) et/ou de la traduction (critère important) 1.8. Méta-données (critère essentiel)
2 Contenu (sur 87 points)	2.1 Exactitude (critère essentiel) 2.2 Hiérarchie d'évidence et indication du niveau de preuve (critère essentiel) 2.3 Citations des sources originales (critère essentiel) 2.4 Dénégation (critère important) 2.5 Organisation logique (navigabilité) (critère essentiel) 2.6 Facilité de déplacement dans le site 2.6a Qualité du moteur interne de recherche (critère important) 2.6b Index général (critère important) 2.6c Rubrique "quoi de neuf " (critère important) 2.6d Page d'aide (critère mineur) 2.6e Plan du site (critère mineur) 2.7 Exclusions et omissions notées (critère essentiel) 2.8 Rapidité de chargement du site et de ses différentes pages (critère important) 2.9 Affichage clair des catégories d'informations disponibles (informations factuelles, résumés, documents en texte intégral, répertoires, banque de données structurées) (critère important)
3 Hyper-liens (sur 45 points)	3.1 Sélection (critère essentiel) 3.2 Architecture (critère important) 3.3 Contenu (critère essentiel) 3.4 Liens arrière (back-links) (critère important) 3.5 Vérification régulière de l'opérationnalité des hyper-liens (critère important) 3.6 En cas de modification de structure d'un site, lien entre les anciens documents HTML et les nouveaux (critère important) 3.7 Distinction hyper-liens internes et externes (critère mineur)
4 Design (sur 21 points)	4.1 Design du site (critère essentiel) 4.2 Lisibilité du texte et des images fixes et animées (critère important)

	4.3 Qualité de l'impression (critère important)
5 Interactivité (sur 18 points)	5.1 Mécanisme pour la rétroaction, commentaires optionnels : courriel de l'auteur de chaque document du site (critère essentiel) 5.2 Forums, chat ("causette") (critère mineur) 5.3 Traçabilité : informations des utilisateurs de l'utilisation de tout dispositif permettant de récupérer automatiquement des informations (nominatives ou non) sur leur poste de travail (cookies,...) (critère important)
6 Aspects quantitatifs (sur 12 points)	6.1 Nombre de machines visitant le site et nombre de documents visualisés (critère important) 6.2 Nombre de citations de presse (critère mineur) 6.3 Nombre de productions scientifiques issues du site, avec indices bibliométriques (critère mineur)
7 Aspects déontologiques (sur 18 points)	7.1 Responsabilité du lecteur (critère essentiel) 7.2 Secret médical (critère essentiel) Le non-respect des règles déontologiques est un élément disqualifiant d'un site
8 Accessibilité (sur 12 points)	8.1 Présence dans les principaux répertoires et moteurs de recherche (critère important) 8.2 Adresse intuitive du site (critère important)
	Soit 312 points au maximum

Extrait d'arborescence du thesaurus MeSH (arborescence *cardiopathies* – C14-280).

cardiopathies
- *arrêt cardiaque*
  - *mort subite origine cardiaque*
- *cardiomégalie*
  - *cardiomyopathie congestive*
  - *hypertrophie ventricule gauche*
- *cardiomyopathie*
  - *cardiomyopathie congestive*
  - *cardiomyopathie éthylique*
  - *cardiomyopathie hypertrophique*
    - *cardiomyopathie hypertrophique familiale*
  - *cardiomyopathie restrictive*
  - *fibrose endomyocardique*
  - *Kearns et Sayre, syndrome*
  - *lésion reperfusion myocardique*
- *cardiomyopathie carcinoïde*
- *cardiomyopathies congénitales*
  - *anomalie vaisseau coronaire*
  - *atrésie tricuspidienne*
  - *canal artériel, persistance*
  - *coarctation aortique*
  - *cœur crisscross*
  - *cœur triatial*
  - *Communication intercavitaires cardiaques*
  - *complexe Eisenmenger*
  - *dextrocardie*
    - *Kartagener, syndrome*
  - *dysplasie ventriculaire droite arythmogène*
  - *Enstein, maladie*
  - *hypoplasie cœur gauche, syndrome*
  - *Marfan, syndrome*
  - *persistance tronc artériel commun*
  - *tétralogie Fallot*

Liste des qualificatifs hiérarchisés utilisés pour l'indexation des ressources

analyse (AN)
- *isolement & purification (IP)*
- *liquide céphalorachidien (CF)*
- *sang (BL)*
- *urine (UR)*

anatomie et histologie (AH)
- *cytologie (CY)*
- *anatomie pathologique (PA)*
- *ultrastructure (UL)*
- *embryologie (EM)*
- *malformations (AB)*
- *innervation (IR)*
- *vascularisation (BS)*

chirurgie (SU)
- *transplantation (TR)*

complications (CO)
- *secondaire (SC)*

composition chimique (CH)
- *agonistes (AG)*
- *analogues et dérivés (AA)*
- *antagonistes et inhibiteurs (AI)*
- *synthèse chimique (CS)*

cytologie (CY)
- *anatomie pathologique (PA)*
- *ultrastructure (UL)*

diagnostic (DI)
- *anatomie pathologique (PA)*
- *échographie (US)*
- *radiographie (RA)*
- *scintigraphie (RI)*

effets indésirables (AE)
- *intoxication (PO)*
- *toxicité (TO)*

embryologie (EM)
- *malformations (AB)*

épidémiologie (EP)
- *ethnologie (EH)*
- *mortalité (MO)*

étiologie (ET)
- *complications (CO)*
- *secondaire (SC)*
- *congénital (CN)*
- *embryologie (EM)*
- *génétique (GE)*
- *immunologie (IM)*
- *induit chimiquement (CI)*
- *microbiologie (MI)*
- *virologie (VI)*
- *parasitologie (PS)*
- *transmission (TM)*

métabolisme (ME)
- *biosynthèse (BI)*
- *déficit (DF)*
- *enzymologie (EN)*
- *liquide céphalorachidien (CF)*

- pharmacocinétique (PK)
- sang (BL)
- urine (UR)

microbiologie (MI)
- virologie (VI)

organisation et administration (OG)
- économie (EC)
- législation et jurisprudence (LJ)
- main d'œuvre (MA)
- normes (ST)
- ressources et distribution (SD)
- tendances (TD)
- utilisation (UT)

pharmacologie (PD)
- administration et posologie (AD)
- agonistes (AG)
- antagonistes et inhibiteurs (AI)
- contre-indications (CT)
- effets indésirables (AE)
  - intoxication (PO)
  - toxicité (TO)
- pharmacocinétique (PK)
- usage diagnostique (DU)

physiologie (PH)
- croissance et développement (GD)
- génétique (GE)
- immunologie (IM)
- métabolisme (ME)
  - biosynthèse (BI)
  - déficit (DF)
  - enzymologie (EM)
  - liquide céphalorachidien (CF)
  - pharmacocinétique (PK)
  - sang (BL)
  - urine (UR)
- physiopathologie (PP)
- sécrétion (SE)

statistiques et données numériques (SN)
- épidémiologie (EP)
  - ethnologie (EH)
  - mortalité (MO)
- ressources et distribution (SD)
- utilisation (UT)

thérapeutique (TH)
- diétothérapie (DH)
- chimiothérapie (DT)
- chirurgie (SU)
  - transplantation (TR)
- prévention et contrôle (PC)
- soins infirmiers (NU)
- radiothérapie (RT)
- rééducation et réadaptation (RH)

usage thérapeutique (TU)
- administration et posologie (AD)
- contre-indications (CT)
- effets indésirables (AE)
- intoxication (PO)

Liste hiérarchisée des types de ressources utilisés

[MH] mot clé MeSH ; [TP] type de publication Medline.

association (association)
- *association patients (association of patients)*
- *association professionnels santé (association of health professionals)*
  - *syndicat (labor unions)* [MH]

base de données (database) [TP]
- *banque d'images (image database)*
- *base de données bibliographiques (database, bibliographic)* [MH]

bibliothèque médicale (libraries, medical) [MH]
dessin architecture (architectural drawings) [TP]
éditeur (publisher)
enseignement et éducation (education) [MH]
- *formation (training)*
- *matériel enseignement (teaching material)* [MH]
  - *cours (lectures)* [TP]
    - *question d'internat (French pre-residency program examination)*
  - *lecture critique d'articles (critical appraisal)*
  - *manuel enseignement (textbooks)* [TP]
  - *matériel d'enseignement audio-visuel (audiovisual aids)* [MH]
    - *film éducatif (instruction)* [TP]
  - *problèmes et exercices (problems and exercises)* [TP]
    - *apprentissage du raisonnement clinique (clinical reasoning learning)*
    - *apprentissage par problème (problem-based learning)* [MH]
    - *cas clinique (case report)*
    - *corrigé d'annales (correct version of the examination)*
    - *lecture critique d'articles (critical appraisal)*
    - *questions réponses (examination questions)* [TP]
      - *questions à choix multiple (MCQ multiple choice quiz)*
      - *questions à réponses ouvertes et courtes (open and closed questions)*
  - *travaux dirigés (tutorials)*
  - *travaux pratiques(practicals)*
- *structure enseignement (teaching structure)*
  - *centre hospitalier universitaire* (*hospitals,teaching)* [MH]
  - *établissement enseignement médical ou apparenté (schools, health occupations)*
    - *école d'infirmiers* (*schools, nursing)* [MH]
    - *école dentaire* (*schools, dental)* [MH]
    - *école santé publique* (*schools, public health)* [MH]
    - *école vétérinaire* (*schools, veterinary)* [MH]
    - *faculté de médecine* (*schools, medicine)* [MH]
    - *faculté de pharmacie (schools, pharmacy)* [MH]

établissement santé (health facilities)
- *hôpital (hospital)* [MH]
  - *centre hospitalier universitaire* (*hospitals, teaching)* [MH]
  - *centre rééducation et réadaptation (rehabilitation centers)* [MH]
    - *centre traitement toxicomanie* (*substance abuse treatment centers*) [MH]
  - *hôpital spécialisé* (*hospitals, special)* [MH]
    - *hôpital enfant* (*hospitals, pediatric)* [MH]
    - *hôpital militaire* (*hospitals, military)* [MH]
    - *hôpital psychiatrique* (*hospitals, psychiatric)* [MH]
    - *maison convalescence (hospitals, convalescent)* [MH]
    - *maternité (hôpital)* (*hospitals, maternity)* [MH]
    - *service soins cardiologie* (*cardiac care facilities)* [MH]
  - *service hospitalier (hospital departments)* [MH]
    - *aumônerie hôpital* (*chaplaincy service, hospital)* [MH]
    - *département anesthésie hôpital* (*anesthesia department, hospital)* [MH]

département médecine nucléaire hôpital (*nuclear medicine department, hospital)* [MH]
pharmacie hôpital (*pharmacy service, hospital)* [MH]
service anatomopathologie hôpital (*pathology department, hospital)*
service cardiologie hôpital (*cardiology service, hospital)* [MH]
service chirurgie hôpital (surgery department, hospital) [MH]
service gynécologie et obstétrique hôpital (*obstetrics and gynecology department, hospital)* [MH]
service médical urgence (emergency medical services) [MH]
service oncologie hôpital (*oncology service, hospital)* [MH]
service psychiatrique hôpital (*psychiatric department, hospital)* [MH]
service radiologie hôpital (radiology department, hospital) [MH]
service rhumatologie hôpital (*rheumatology service, hospital)*
service urologie hôpital (*urology department, hospital)* [MH]
unité soins intensifs (intensive care units) [MH]
maison médicalisée personnes âgées (homes for the aged) [MH]
maison repos (nursing home) [MH]
services santé (health services) [MH]
centre anti-poison (poison control centers)[MH]
consultation médicale (office visits) [MH]
dispensaire hygiène mentale (community mental health services) [MH]
service diagnostic (diagnostic services) [MH]
service hygiène scolaire (school health services)[MH]
service médecine préventive (preventive health services) [MH]
service santé universitaire (student health services) [MH]
service soins domicile (home care services) [MH]
services de gériatrie (health services for the aged) [MH]
unité itinérante santé (mobile health units) [MH]
services soins ambulatoires (ambulatory care facilities) [MH]
centre public santé (community health centers) [MH]
centre public santé mentale (community mental health centers) [MH]
exposition (exhibits) [MH]
fondation (foundation) [MH]
forum et liste de diffusion (forum and mailing list)
forum et liste de diffusion patients (forum and mailing list for patients)
guide ressources (resource guides) [TP]
institution (establishment, institution, organization)
agence gouvernementale (governmental agency)
agence régionale (regional agency)
lettre d'information (*newsletter*)
liste de diffusion
logiciel (software)
matériel audio-visuel (audiovisual aids) [MH]
carte géographique (maps) [TP]
enregistrement vidéo (*video recording)* [MH]
film éducatif (instruction) [TP]
illustration médicale (medical illustration) [MH]
son (acoustique) (sound) [MH]
musées (museum) [MH]
ouvrage d'orientation (reference books) [MH]
atlas (atlases) [TP]
dictionnaire médical (dictionaries, medical) [MH]
encyclopédie (encyclopedias) [TP]
répertoire (directories) [MH]
annuaire (annual directory)
catalogue (catalogs) [TP]
patient
assistance par téléphone (hotlines) [MH]
association patients (association of patients)

forum et liste de diffusion patients (newsgroup and discussion list for patients)
information patient et grand public (popular works) [TP]
brochure information patient (patient education handout) [TP]
recommandation patients (patients guideline)
site personnel patient (patient personal Web site)
périodique (periodicals) [TP]
registre (registries) [MH]
réseaux coordonnés (community networks) [MH]
site personnel (personal Web site)
site personnel patient (patient personal Web site)
site personnel professionnel santé (health professional personal Web site)
société savante (scientific society)
structure recherche (research structure)
texte (text)
actes de congrès (congresses) [TP]
article de périodique (journal article) [TP]
bibliographie (bibliography) [TP]
biographie (biography) [TP]
biobibliographie (biobibliography) [MH]
cahier de laboratoire (laboratory manuals) [TP]
cours (lectures) [TP]
question d'internat (French pre-residency program examination)
dictionnaire médical (dictionaries, medical) [MH]
encyclopédie (encyclopedias) [TP]
étude comparative (comparative study) [MH]
étude évaluation (evaluation studies) [TP]
étude d'évaluation des pratiques professionnelles
étude d'évaluation économique
étude d'évaluation en santé publique
étude d'évaluation technologique
*guide (guide). V*oir aussi *recommandation.*
information patient et grand public (popular works)
brochure information patient (patient education handout
ligne directrice
manuel enseignement (textbooks) [TP]
méta-analyse (meta-analysis) [TP]
monographie (monograph) [TP]
problèmes et exercices (problems and exercises) [TP]
publication officielle (government publications) [TP]
recommandation (guidelines). Voir aussi guide.
recommandation patients (patients guideline)
recommandation pour la politique de santé (health policy guidelines)
recommandation professionnelle (practice guidelines)
conférence de consensus (consensus development conferences)
consensus formalisé d'experts (formal expert consensus)
recommandation pour la pratique clinique (clinical practice guidelines)
référence médicale (medical reference?)
recommandation de santé publique (public health guidelines)
rapport technique (technical report) [TP]
texte législatif (legislation) [TP]
thèse ou mémoire (dissertations, academic) [MH]

Code de Génération des contextes pour l'algorithme A-Close

```
-------------------------------------------------------------------------------------------
-- Processus : GenereItemSets.sql
-- Date : 09/05/2003
--
-- Description : Script PL/SQL de generation de fichier en entree de DataMiner.java
-------------------------------------------------------------------------------------------

-------------------------------------------------------------------------------------------
-- Pour le lancement sous SQLPlus :
-------------------------------------------------------------------------------------------
-- @C:\A\GenereItemSets.sql

Set serveroutput on size 1000000
spool C:\Lina\cismef\Output\GenereItemSets.txt

-------------------------------------------------------------------------------------------
-- BLOC DECLARATIF
-------------------------------------------------------------------------------------------
DECLARE
-------------------------------------------------------------------------------------------
-- Parcours des Numeros de Fiches
-------------------------------------------------------------------------------------------
   cursor C_FICHES is
   select distinct
       NUM_FICHE
   from
       TB_MOTCLEFS_FICHE
   where NUM_FICHE < 15000
   order by NUM_FICHE;

-------------------------------------------------------------------------------------------
-- Ou Parcours des Numeros de Fiches pour une specialite
-------------------------------------------------------------------------------------------
   cursor C_FICHES is
   select distinct
       NUM_FICHE
   from
       TB_MOTCLEFS_FICHE, TB_NMT
   where
         TB_NMT.NUM_META_PERE =13
   and TB_NMT.NUM_MOTCLEF_DES = TB_MOTCLEFS_FICHE.NUM_MOTCLEF
   and TB_MOTCLEFS_FICHE.MAJEUR = 'O'
   and NUM_FICHE < 15000
   order by NUM_FICHE;

-------------------------------------------------------------------------------------------
-- CONTEXTE 1 : Parcours des Motclefs attaches a la Fiche
-------------------------------------------------------------------------------------------
/* cursor C_ELEMENTS(P_FICHE varchar2) is
   select distinct
       TB_MOTCLEFS.MOTCLEF_SA ELEMENT
   from
       TB_MOTCLEFS_FICHE, TB_MOTCLEFS
   where
       TB_MOTCLEFS_FICHE.NUM_MOTCLEF = TB_MOTCLEFS.NUM_MOTCLEF
   and NUM_FICHE = P_FICHE
   order by 1;
*/
-------------------------------------------------------------------------------------------
-- CONTEXTE 2 : Parcours des Motclef OU Qualifs attaches a la Fiche
-------------------------------------------------------------------------------------------
  /* cursor C_ELEMENTS(P_FICHE varchar2) is
   select distinct
       TB_MOTCLEFS.MOTCLEF_SA ELEMENT
   from
       TB_MOTCLEFS_FICHE, TB_MOTCLEFS
   where
       TB_MOTCLEFS_FICHE.NUM_MOTCLEF = TB_MOTCLEFS.NUM_MOTCLEF
   and NUM_FICHE = P_FICHE
   UNION
   select distinct
```

```
       decode(TB_QUALIFICATIF.NUM_QUALIFICATIF,83,'',    TB_QUALIFICATIF.QUALIFICATIF_SA)
ELEMENT
   from
       TB_MOTCLEFS_FICHE, TB_QUALIFICATIF
   where
       TB_MOTCLEFS_FICHE.NUM_QUALIFICATIF = TB_QUALIFICATIF.NUM_QUALIFICATIF
   and NUM_FICHE = P_FICHE
   order by 1;
*/
-------------------------------------------------------------------------------------
-- CONTEXTE 3 : Parcours des couples {Motclef/Qualifs} attaches a la Fiche
-------------------------------------------------------------------------------------
/*   cursor C_ELEMENTS(P_FICHE varchar2) is
   select distinct
       TB_MOTCLEFS.MOTCLEF_SA  ||  decode(TB_QUALIFICATIF.NUM_QUALIFICATIF,83,'','/'  ||
TB_QUALIFICATIF.QUALIFICATIF_SA) ELEMENT
   from
       TB_MOTCLEFS_FICHE, TB_MOTCLEFS, TB_QUALIFICATIF
   where
       TB_MOTCLEFS_FICHE.NUM_MOTCLEF = TB_MOTCLEFS.NUM_MOTCLEF
   and TB_MOTCLEFS_FICHE.NUM_QUALIFICATIF = TB_QUALIFICATIF.NUM_QUALIFICATIF
   and NUM_FICHE = P_FICHE
   order by 1;
*/
-------------------------------------------------------------------------------------
-- CONTEXTE 4 : Parcours des Motclef OU Qualifs OU Type_Ress OU Metatermes attaches a la
Fiche
-------------------------------------------------------------------------------------
/*
   -- Motclefs
   cursor C_ELEMENTS(P_FICHE varchar2) is
   select distinct
       TB_MOTCLEFS.MOTCLEF_SA ELEMENT
   from
       TB_MOTCLEFS_FICHE, TB_MOTCLEFS
   where
       TB_MOTCLEFS_FICHE.NUM_MOTCLEF = TB_MOTCLEFS.NUM_MOTCLEF
   and NUM_FICHE = P_FICHE
   -- Qualificatifs
   UNION
   select distinct
       decode(TB_QUALIFICATIF.NUM_QUALIFICATIF,83,'',    TB_QUALIFICATIF.QUALIFICATIF_SA)
ELEMENT
   from
       TB_MOTCLEFS_FICHE, TB_QUALIFICATIF
   where
       TB_MOTCLEFS_FICHE.NUM_QUALIFICATIF = TB_QUALIFICATIF.NUM_QUALIFICATIF
   and NUM_FICHE = P_FICHE
   UNION
   -- Types de ressources
   select distinct
       TB_TYPE_RESS.TYPE_RESSOURCE_FRA_SA ELEMENT
   from
       TB_TYPE_RESS_FICHE, TB_TYPE_RESS
   where
       TB_TYPE_RESS_FICHE.NUM_TYPE_RESS = TB_TYPE_RESS.NUM_TYPE_RESS
   and NUM_FICHE = P_FICHE
   UNION
   -- Metatermes
   select distinct
       TB_META.META_SA ELEMENT
   from
       TB_NMT, TB_META, TB_MOTCLEFS_FICHE
   where
       TB_NMT.NUM_META_PERE = TB_META.NUM_META
   and TB_NMT.NUM_MOTCLEF_DES = TB_MOTCLEFS_FICHE.NUM_MOTCLEF
   and NUM_FICHE = P_FICHE
   order by 1;
*/

-------------------------------------------------------------------------------------
-- CONTEXTE 5 : Parcours des couples (Motclef/Qualifs) OU Type_Ress attaches a la Fiche
-------------------------------------------------------------------------------------
/*
```

```
   -- Motclefs
   cursor C_ELEMENTS(P_FICHE varchar2) is
   -- Couples (Motclefs/Qualifs)
   select distinct
       TB_MOTCLEFS.MOTCLEF_SA  ||  decode(TB_QUALIFICATIF.NUM_QUALIFICATIF,83,'','/'  ||
TB_QUALIFICATIF.QUALIFICATIF_SA) ELEMENT
   from
       TB_MOTCLEFS_FICHE, TB_MOTCLEFS, TB_QUALIFICATIF
   where
       TB_MOTCLEFS_FICHE.NUM_MOTCLEF = TB_MOTCLEFS.NUM_MOTCLEF
   and TB_MOTCLEFS_FICHE.NUM_QUALIFICATIF = TB_QUALIFICATIF.NUM_QUALIFICATIF
   and NUM_FICHE = P_FICHE
   UNION
   -- Types de ressources
   select distinct
       TB_TYPE_RESS.TYPE_RESSOURCE_FRA_SA ELEMENT
   from
       TB_TYPE_RESS_FICHE, TB_TYPE_RESS
   where
       TB_TYPE_RESS_FICHE.NUM_TYPE_RESS = TB_TYPE_RESS.NUM_TYPE_RESS
   and NUM_FICHE = P_FICHE
   order by 1;
*/
-------------------------------------------------------------------------------------------
-- CONTEXTE 6 : Parcours des {Motclef/Qualifs} OU Type_Ress OU Metatermes attaches a la
Fiche
-------------------------------------------------------------------------------------------
   -- Motclefs
   cursor C_ELEMENTS(P_FICHE varchar2) is
   select distinct
       TB_MOTCLEFS.MOTCLEF_SA  ||  decode(TB_QUALIFICATIF.NUM_QUALIFICATIF,83,'','/'  ||
TB_QUALIFICATIF.QUALIFICATIF_SA) ELEMENT
   from
       TB_MOTCLEFS_FICHE, TB_MOTCLEFS, TB_QUALIFICATIF
   where
       TB_MOTCLEFS_FICHE.NUM_MOTCLEF = TB_MOTCLEFS.NUM_MOTCLEF
   and TB_MOTCLEFS_FICHE.NUM_QUALIFICATIF = TB_QUALIFICATIF.NUM_QUALIFICATIF
   and NUM_FICHE = P_FICHE
  /* UNION
   -- Types de ressources
   select distinct
       TB_TYPE_RESS.TYPE_RESSOURCE_FRA_SA ELEMENT
   from
       TB_TYPE_RESS_FICHE, TB_TYPE_RESS
   where
       TB_TYPE_RESS_FICHE.NUM_TYPE_RESS = TB_TYPE_RESS.NUM_TYPE_RESS
   and NUM_FICHE = P_FICHE */
   UNION
   -- Metatermes
   select distinct
       TB_META.META_SA ELEMENT
   from
       TB_NMT, TB_META, TB_MOTCLEFS_FICHE
   where
       TB_NMT.NUM_META_PERE = TB_META.NUM_META
   and TB_NMT.NUM_MOTCLEF_DES = TB_MOTCLEFS_FICHE.NUM_MOTCLEF
   and NUM_FICHE = P_FICHE
   order by 1;
-------------------------------------------------------------------------------------------
-- VARIABLES
-------------------------------------------------------------------------------------------
   C_FICHES_REC C_FICHES%rowtype;
   C_ELEMENTS_REC C_ELEMENTS%rowtype;
   V_BUFF VARCHAR2(4000);
-------------------------------------------------------------------------------------------
-- BLOC EXECUTIF
-------------------------------------------------------------------------------------------
BEGIN
-------------------------------------------------------------------------------------------
-- Parcours des Fiches
-------------------------------------------------------------------------------------------
   for C_FICHES_REC in C_FICHES
   loop
```

```
   begin
      V_BUFF := C_FICHES_REC.NUM_FICHE;
-------------------------------------------------------------------------------------------
-- Ecriture des Elements de la Fiche
-------------------------------------------------------------------------------------------
      for C_ELEMENTS_REC in C_ELEMENTS(C_FICHES_REC.NUM_FICHE)
      loop
          V_BUFF := V_BUFF || '|' || C_ELEMENTS_REC.ELEMENT;
      end loop;

   --Dbms_output.Put_line(V_BUFF);
   P_DBMS_LONG_PUT(V_BUFF);
   end;
   end loop;
END;
/
-------------------------------------------------------------------------------------------
```

Classe de génération des règles d'association.

```
/**
 * Title :        Knowledge-based System for Query Expansion over CISMeF - DataMiner
 * Description :  This class extracts Exact and Approximative Associations Rules
 *                using the A_Close algorithm; from the CISMeF database and the
 *                database generated by Syntex.
 * Date:          MAY 2003
 */

//package cismef.knowqe;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.BufferedReader;
import java.io.PrintWriter;
import java.util.*;
import java.net.URL;
import java.lang.String;
/**
* @author <a href="mailto:Lina.Soualmia@chu-rouen.fr">Lina SOUALMIA</a>
 * @version 1.0
 * @see CismefMorphology.class
 * @see CismefOntology.class
 */
public class CismefDataMiner {
    private final static double MIN_SUPPORT = 10;
    private final static double MIN_CONFIDENCE = 0.7;
    private final static String CONTEXT_TYPE = "META_13_Text";
    private final static int MAX_GENERATORS = 20;
    private int generators_nb = 0;
    private SortedSet[] initialContext = null;
    private SortedSet[] context = null;
    private Hashtable correspondance = new Hashtable();
    private SortedSet items = new TreeSet();
    private Set[] generators = new SortedSet[MAX_GENERATORS + 1];
    private SortedSet[] [] close = null;

      public static void main(String[] args) {
         try {
               if (args.length != 1) {
               .........throw new IllegalArgumentException("Wrong parameters number");
               }
               else {
               System.err.println("****  "  +  args[0].substring(args[0].indexOf("/h")  +
1, args[0].length()) + " ****");
               ..........// convert file passed in argument into an array of sortedSet
               ......................CismefDataMiner dataMiner = new CismefDataMiner();
               dataMiner.initialContext                                                =
dataMiner.getInitialContext(dataMiner.readFileContent(args[0]));
               System.err.println(" *** BEGIN *** ");
               System.err.println(" --------- ");
               System.err.println(" INITIAL CONTEXT ");
               //dataMiner.showSets(dataMiner.initialContext);
               System.err.println(" --------- ");
               System.err.println(" CODED CONTEXT ");
               dataMiner.context = dataMiner.getCodedContext();
               //dataMiner.showSets(dataMiner.context);
```

```
                System.err.println(" --------- ");
                System.err.println(" GENERATORS ");
                for(int genLevel=1; genLevel <= MAX_GENERATORS; genLevel++){
                    System.err.println("   GEN LEVEL : " + genLevel);
                    dataMiner.addGenerators(genLevel);
                }
                //dataMiner.showSets(dataMiner.generators);
                System.err.println(" --------- ");
                System.err.println(" CLOSE ");
                dataMiner.close = dataMiner.getClose();
                System.err.println(" --------- ");
                System.err.println(" EXACTS RULES ");
                System.out.println(" **** EXACTS RULES **** ");
                dataMiner.showExactRules();
                System.err.println(" --------- ");
                System.err.println(" APPROX RULES ");
                System.out.println(" **** APPROX RULES **** ");
                dataMiner.showApproxRules();
                System.err.println(" --------- ");
                System.err.println(" *** END *** ");
                };
          }
          catch (Exception err) {
          System.err.println(err);
          System.err.println("Usage:             java             cismef.knowqe.DataMiner
<file:///C:/INPUT_FILE_LOCATION>");
          }
       }
       /**
        * Insert an input TextFile into a Vector
        * Returns Vector
        * @param  fileURL Path of the simple text file : file:///C:/INPUT_FILE_LOCATION
        * @return fileContent vector composed of TextFile lignes
        */
       public Vector readFileContent(String fileURL) {
          Vector fileContent = new Vector();
          InputStream in = null;
          String buf = new String();
          try {
                // File opening
                in = new URL(fileURL).openStream();
                BufferedReader flux = new BufferedReader(new InputStreamReader(in));
                // File to vector
                while((buf = flux.readLine())!= null) {
                if (buf.length() > 2) {
                fileContent.addElement(buf);
                }
                }
          }
          catch (Exception err) {
                System.err.println(err);
          }
          // Stream closing
          finally {
                try {
                in.close();
```

```
                }
                catch (Exception err) {
                }
          }
          return fileContent;
       }

       /**
        * Converts a Vector containing the simple TextFile into an annotations Vector
        * Returns Vector
        * @param  vText Vector containing text file strings
        * @return fileContent vector composed of annotations statements
        */
       private SortedSet[] getInitialContext(Vector vText) {
         SortedSet[] newContext = new SortedSet[vText.size() + 1];
         SortedSet itemset = new TreeSet();
         int numItemset = 1;
         String item = new String("");
          // Text to Annotation
          for(Enumeration enum = vText.elements(); enum.hasMoreElements();) {
                // One line parsing
                StringTokenizer               sTokens               =               new
StringTokenizer((String)enum.nextElement(),"\"|", false);
            // The fisrt token is the ressource number
                String firstToken = sTokens.nextToken().trim();
                // Predicate & Validation statements
                // Single/multiple object reading
                while (sTokens.hasMoreTokens()) {
                item = sTokens.nextToken().trim();
                if (item.length() > 0) {
                   itemset.add(item);
                }
                }
           newContext[numItemset ++] = new TreeSet(itemset);
           itemset.clear();
          }
       return newContext;
       }
       private SortedSet[] getCodedContext() {
         SortedSet initialItemSet = new TreeSet();
         for(int line=1; line< initialContext.length; line++){
             Iterator i = initialContext[line].iterator();
             while (i.hasNext()) {
                  initialItemSet.add(i.next());
             }
         }
         Integer itemNumber = new Integer(0);
         SortedSet itemSet = new TreeSet();
         Iterator j = initialItemSet.iterator();
         while (j.hasNext()) {
              itemNumber = increment(itemNumber);
              correspondance.put(j.next(), itemNumber.toString());
              itemSet.add(itemNumber.toString());
         }
         items = new TreeSet(itemSet);
         SortedSet[] newContext = new SortedSet[initialContext.length];
```

```
        for(int line=1; line< initialContext.length; line++){
            newContext[line] = new TreeSet();
            Iterator k = initialContext[line].iterator();
            while (k.hasNext()) {
                 newContext[line].add(correspondance.get(k.next()));
            }
        }
          return newContext;
    }
      private SortedSet[] [] getClose() {
        SortedSet[][] closeSets = new TreeSet[generators_nb + 1][3];
        int numGenerator = 0;
       for(int line=1; line< generators.length; line++){
            if (generators[line] != null) {
                Iterator i = generators[line].iterator();
                while (i.hasNext()) {
                     numGenerator ++;
                     closeSets[numGenerator][1] = new TreeSet(toSet(i.next()));
                     //System.err.println("- gen : " + closeSets[numGenerator][1]);
                     closeSets[numGenerator][2]                  =                  new
TreeSet(findClose(closeSets[numGenerator][1]));
                     //System.err.println("- ferme : " + closeSets[numGenerator][2]);
                     //System.err.println("- num gen : " + numGenerator);
                }
            }
        }
          return closeSets;
    }
    private Integer increment( Integer oldInt ) {
        return new Integer( oldInt.intValue() + 1 );
    }
      private void addGenerators(int level) {
        SortedSet generatorSet = new TreeSet();
        if (level == 1) {
            Iterator i = items.iterator();
            while (i.hasNext()) {
                 SortedSet subset = new TreeSet();
                 subset.add(i.next());
                 if (getSupport(subset) >= MIN_SUPPORT) {
                    generatorSet.addAll(subset);
                    generators_nb ++;
                 }
                 subset.clear();
              }
        }
       else  {
            Iterator i = generators[level - 1].iterator();
            while (i.hasNext()) {
                 SortedSet setA = toSet(i.next());
                 if (!setA.isEmpty()) {
                 //System.out.println("A:" + setA);
                 double supportA = getSupport(setA);
                //System.out.print(setA + " : ");
                //System.out.println(supportA);
                 Iterator j = generators[level - 1].iterator();
                 while (j.hasNext()) {
```

```
                    SortedSet setB = toSet(j.next());
                    if (!setB.isEmpty()) {
                    //System.out.println("  B:" + setB);
                    SortedSet newSet = new TreeSet();
                    newSet = getCandidat(setA, setB);
                    if (newSet.size() == level) {
                    //System.out.println("    CANDIDAT:" + newSet);
                    double support = getSupport(newSet);
                    if (support >= MIN_SUPPORT && support != supportA) {
                    generatorSet.add(newSet.toString());
                //System.out.print(newSet.toString() + " : ");
                //System.out.println(support);
                    generators_nb ++;
                    }
                    }
                    newSet.clear();
                    setB.clear();
                    }
                 }
                 setA.clear();
                 }
            }
        }
        generators[level] = new TreeSet(generatorSet);
      }
      private SortedSet toSet(Object obj) {
          String itemset = new String(obj.toString());
          String item = new String();
          SortedSet set = new TreeSet();
        itemset = itemset.replace('[',' ');
        itemset = itemset.replace(']',' ').trim();
        StringTokenizer sTokens = new StringTokenizer(itemset,",", false);
        while (sTokens.hasMoreTokens()) {
                item = sTokens.nextToken().trim();
                if (item.length() > 0) {
                   set.add(item);
                }
         }
        return set;
      }
      private SortedSet getCandidat(SortedSet set1, SortedSet set2) {
          SortedSet firstset = new TreeSet(set1);
          SortedSet secondset = new TreeSet(set2);
          Object last1 = firstset.last();
          Object last2 = secondset.last();
        firstset.remove(last1);
        if                    ((secondset.containsAll(firstset))                    &&
(last1.toString().compareTo(last2.toString()) < 0)) {
                secondset.add(last1);
            return secondset;
        }
        else return new TreeSet(Collections.EMPTY_SET);
      }
      private double getSupport(SortedSet subset) {
        double support = 0;
        for(int i=1; i< context.length ; i++){
```

```
          if (context[i].containsAll(subset)) { support++; }
      }
        return support;
    }
    private SortedSet findClose(SortedSet set) {
        SortedSet closeSet = new TreeSet();
        boolean firstOccurence = true;
      for(int line=1; line< context.length; line++){
         if (context[line].containsAll(set)) {
             if (firstOccurence) {
                closeSet = new TreeSet(context[line]);
                //System.out.println(" first occurence : " + closeSet);
                firstOccurence = false;
             }
             else {
                //System.out.print(" intersection : " + context[line]);
                //System.out.print(" ^ " + closeSet);
                closeSet = intersection(context[line], closeSet);
                //System.out.println(" = " + closeSet);
             }
         }
      }
        return closeSet;
  }
    private SortedSet intersection(SortedSet set1, SortedSet set2) {
        SortedSet newset = new TreeSet(set1);
      newset.retainAll(set2);
        return newset;
  }
    private SortedSet union(SortedSet set1, SortedSet set2) {
        SortedSet newset = new TreeSet(set1);
      newset.addAll(set2);
        return newset;
    }
    private SortedSet difference(SortedSet set1, SortedSet set2) {
        SortedSet newset = new TreeSet(set1);
      newset.removeAll(set2);
        return newset;
    }
    private void showSets(Set[] sets) {
     for(int line=1; line< sets.length; line++){
        if (sets[line] != null) {
           System.out.println(sets[line]);
        }
     }
    }
  private Object getCorrespondanceKey(Object value) {
        Object key = new Object();
      Enumeration e = correspondance.keys();
      boolean keyFound = false;
        while (e.hasMoreElements() && !keyFound) {
              key = (String)e.nextElement();
              if (correspondance.get(key).toString().equals(value.toString())) {
                 keyFound = true;
          }
```

```
         }
        return key;
    }
      private void showExactRules() {
        String antecedent = new String();
        String consequent = new String();
        double support = 0;
        int rulesNb = 0;
        for(int line=1; line< close.length; line++){
            support = getSupport(close[line][1]);
            antecedent = "";
            Iterator i = close[line][1].iterator();
            while (i.hasNext()) {
                 if (antecedent.length() > 0) antecedent = antecedent + "|";
                 antecedent      =      antecedent      +      getCorrespondanceKey(new
Integer(i.next().toString()));
            }
            consequent = "";
            Iterator j = difference(close[line][2], close[line][1]).iterator();
            while (j.hasNext()) {
                 if (consequent.length() > 0) consequent = consequent + "|";
                 consequent      =      consequent      +      getCorrespondanceKey(new
Integer(j.next().toString()));
            }
            if (consequent.length() > 0) {
                rulesNb ++;
                //System.out.println(rulesNb + " : " + antecedent + " -> " + consequent
+ " [" + support + "]");
               System.out.println("INSERT   INTO   TB_KNOWQE_REGLES_ASSOC   (CONTEXTE,
ANTECEDANTS,  CONSEQUENTS,  SUPPORT,  CONFIANCE)  VALUES  ('"  +  CONTEXT_TYPE  +  "',  '"  +
antecedent + "', '" + consequent + "'," + support + "," + 1.0 + ");");
            }
        }
      }
      private void showApproxRules() {
          SortedSet closeset = new TreeSet();
          SortedSet setA = new TreeSet();
          SortedSet setB = new TreeSet();
        String rule1 = new String();
        String rule2 = new String();
        String rule3 = new String();
        String antecedent = new String();
        String consequent = new String();
        double supportAuB = 0;
        double supportA = 0;
        double confidence = 0;
        int rulesNb = 0;
        for(int line=1; line< close.length; line++){
             setA = new TreeSet(close[line][1]);
             closeset = new TreeSet(close[line][2]);
            for(int cur=1; cur< close.length; cur++){
                setB = new TreeSet(close[cur][2]);
                if ( setB.containsAll(closeset) && !closeset.containsAll(setB) ) {
                   if ( !rule1.equals(setA + ":" + setB) &&
                        !rule2.equals(setA + ":" + setB) &&
                        !rule3.equals(setA + ":" + setB) ) {
                         rule3 = rule2;
```

```
                        rule2 = rule1;
                        rule1 = setA + ":" + setB;
                        //System.err.println(rule1);
                       supportAuB = getSupport(union(setB, setA));
                       if (supportAuB >= MIN_SUPPORT) {
                           supportA = getSupport(setA);
                           confidence = supportAuB / supportA;
                           if (confidence >= MIN_CONFIDENCE) {
                               antecedent = "";
                               Iterator i = setA.iterator();
                               while (i.hasNext()) {
                                    if   (antecedent.length()   >   0)   antecedent   =
antecedent + "|";
                                    antecedent = antecedent + getCorrespondanceKey(new
Integer(i.next().toString()));
                               }
                               consequent = "";
                               Iterator j = difference(setB, setA).iterator();
                               while (j.hasNext()) {
                                    if   (consequent.length()   >   0)   consequent   =
consequent + "|";
                                    consequent = consequent + getCorrespondanceKey(new
Integer(j.next().toString()));
                               }
                               if (consequent.length() > 0) {
                                   rulesNb ++;
                                   //System.out.println(rulesNb + " : " + antecedent +
" -> " + consequent + " [" + supportAuB + "]" + " [" + Math.round( 100.0 * confidence )
/ 100.0 + "]");
                                   System.out.println("INSERT                      INTO
TB_KNOWQE_REGLES_ASSOC (CONTEXTE, ANTECEDANTS, CONSEQUENTS, SUPPORT, CONFIANCE) VALUES
('" + CONTEXT_TYPE + "', '" + antecedent + "', '" + consequent + "'," + supportAuB + ","
+ Math.round( 100.0 * confidence ) / 100.0 + ");");
                               }
                           }
                       }
                   }
               }
           }
       }
   }
} // End Class
```

Règles expertes créées après la fouille de données.

MC_ANTECEDENT	QU_ANTECEDENT	MC_CONSEQUENT
abdomen	Radiographie	radiographie abdominale
abdomen	Radiographie	radiographie abdominale
abdomen	Traumatismes	traumatisme abdominal
abdomen	Traumatismes	traumatisme abdominal
accidents	prévention et contrôle	prévention accident
acide ascorbique	Déficit	acide ascorbique, carence
acide ascorbique	Déficit	acide ascorbique, carence
acide chlorhydrique gastrique	Analyse	dosage acidité gastrique
acide chlorhydrique gastrique	Analyse	dosage acidité gastrique
acide folique	Déficit	acide folique, carence
acide oxalique	analogues et dérivés	acides oxaliques
adenosine deaminase	Déficit	déficit immunitaire combiné sévère
adolescent	psychologie	psychologie adolescent
agaricales	intoxication	intoxication champignon
aldostérone	antagonistes et inhibiteurs	antialdostérones
aliment	intoxication	intoxication alimentaire
aliment	législation et jurisprudence	législation aliment
allergie et immunologie	méthodes	méthodes immunologiques
alpha 1-antitrypsine	déficit	alpha 1-antitrypsine, déficit
alpha-glucosidase	déficit	glycogénose type 2
alpha-mannosidase	déficit	alpha-mannosidose
androgènes	antagonistes et inhibiteurs	antiandrogènes
aorte	radiographie	aortographie
aorte, maladies	radiographie	aortographie
appareil cardiovasculaire	chirurgie	intervention chirurgie cardiovasculaire
appareil cardiovasculaire	malformations	malformations cardiovasculaires
appareil cardiovasculaire	physiologie	physiologie cardiovasculaire
appareil digestif	chirurgie	intervention chirurgie digestive
appareil digestif	malformations	malformations appareil digestif
appareil digestif	physiologie	physiologie appareil digestif
appareil locomoteur	croissance et développement	développement locomoteur
appareil locomoteur	croissance et développement	développement musculaire
appareil locomoteur	malformations	malformations appareil locomoteur
appareil locomoteur	physiologie	
appareil respiratoire	physiologie	physiologie respiratoire
appareil stomatognathique	malformations	
appareil urogénital	malformations	malformations urogénitales
appareil urogénital	radiographie	urographie
appendicite	chirurgie	appendicectomie
arachis hypogaea	effets indésirables	hypersensibilité arachide
arginase	déficit	hyperargininémie
argininosuccinate synthase	déficit	citrullinémie
arsenic	intoxication	intoxication arsenic
artères	radiographie	angiographie
artères	radiographie	angiographie
artères carotides	traumatismes	traumatisme artère carotide
artères carotides	traumatismes	traumatisme artère carotide
articulation cheville	traumatismes	traumatisme cheville

MC_ANTECEDENT	QU_ANTECEDENT	MC_CONSEQUENT
articulation cheville	traumatismes	traumatisme cheville
articulation genou	traumatismes	traumatisme genou
articulation hanche	traumatismes	traumatisme hanche
articulation hanche	traumatismes	traumatisme hanche
articulation poignet	traumatismes	traumatisme poignet
articulation poignet	traumatismes	traumatisme poignet
articulations	radiographie	arthrographie
articulations pied	traumatismes	traumatisme pied
arylsulfatases	déficit	leucodystrophie métachromatique
aspartate-semialdéhyde déhydrogénase	déficit	hydroxybutyrates
bactéries	physiologie	physiologie bactérienne
béryllium	intoxication	bérylliose
bêta-mannosidase	déficit	bêta-mannosidose
bêta-N-acétylhexosaminidase	déficit	Sandhoff, maladie
bêta-N-acétylhexosaminidase	déficit	Tay-Sachs, maladie
biotinidase	déficit	biotinidase, déficit
biotinidase	déficit	biotinidase, déficit
bouche	malformations	malformations bouche
bras	traumatismes	traumatisme membre supérieur
bronches	radiographie	bronchographie
bronches	radiographie	bronchographie
bronches, maladies	radiographie	bronchographie
cadmium	intoxication	intoxication cadmium
caecum, maladies	chirurgie	caecostomie
canaux biliaires	malformations	atrésie voies biliaires
canaux biliaires	radiographie	cholangiographie
carbamoyl-phosphate synthase (ammonia)	déficit	
carbamoyl-phosphate synthase (ammonia)	déficit	
carbamoyl-phosphate synthase (ammonia)	déficit	
caspase 1	analogues et dérivés	caspases
cataracte	chirurgie	extraction cataracte
cavité abdominale	traumatismes	traumatisme abdominal
cellule souche	transplantation	greffe de cellules souches
cellule souche hématopoïétique	transplantation	
cellules	physiologie	physiologie cellulaire
cellules	transplantation	transplantation cellulaire
cellules	transplantation	transplantation cellulaire
cheville	traumatismes	traumatisme cheville
cheville	traumatismes	traumatisme cheville
cholécalciférol	déficit	vitamine D, carence
cholinestérases	antagonistes et inhibiteurs	anticholinestérasiques
chromosomes	malformations	aberrations chromosomiques
ciguatoxines	intoxication	ciguatera
coeur	chirurgie	interventions chirurgie cardiaque
coeur	échographie	échocardiographie
coeur	malformations	cardiopathies congénitales
coeur	transplantation	transplantation cardiaque
coeur	transplantation	transplantation cardiaque
coeur	traumatismes	traumatisme coeur

MC_ANTECEDENT	QU_ANTECEDENT	MC_CONSEQUENT
coeur	vascularisation	vaisseaux coronaires
complexe IV de la chaîne respiratoire	déficit	cytochrome c oxydase, déficit
complexe IV de la chaîne respiratoire	déficit	cytochrome c oxydase, déficit
Convulsions	induit chimiquement	convulsivants
Cornée	transplantation	transplantation cornée
Cou	traumatismes	traumatisme cou
Cou	traumatismes	traumatisme cou
cysteine endopeptidases	antagonistes et inhibiteurs	inhibiteurs cystéine protéinase
cytochrome réductases	déficit	méthémoglobinémie
cytochrome-B(5) reductase	déficit	méthémoglobinémie
cytochrome-b(5) reductase	déficit	méthémoglobinémie
déformations main	congénital	déformations congénitales main
dent	malformations	malformation dentaire
dent	physiologie	physiologie dentaire
dent	traumatismes	traumatisme dentaire
dentisterie	économie	économie dentaire
dentisterie	enseignement et éducation	enseignement dentaire
dentisterie	éthique	éthique dentisterie
dentisterie	histoire	histoire art dentaire
dentisterie	instrumentation	instruments dentaires
dentisterie	législation et jurisprudence	législation dentisterie
dihydrolipoamide déhydrogénase	déficit	méthémoglobinémie
dipeptidyl carboxypeptidase I	antagonistes et inhibiteurs	
dipeptidyl peptidase I	déficit	Papillon-Lefèvre, maladie
doigt	traumatismes	traumatisme doigt
doigt	traumatismes	traumatisme doigt
dos	traumatismes	traumatisme dos
embryon	croissance et développement	développement embryonnaire et foetal
embryon	transplantation	transfert embryon
encéphale	analyse	biochimie encéphale
encéphale	analyse	biochimie encéphale
encéphale	composition chimique	biochimie encéphale
encéphale	composition chimique	biochimie encéphale
encéphale	échographie	échoencéphalographie
encéphale	traumatismes	traumatisme cérébral
endopeptidases	antagonistes et inhibiteurs	inhibiteurs protéinases
enzymes	antagonistes et inhibiteurs	anti-enzymes
épidémiologie	méthodes	méthode épidémiologique
épithélium	cytologie	cellule épithéliale
érythrocyte	malformations	érythrocyte anormal
érythrocyte	transplantation	transfusion érythrocytes
étudiant dentisterie	enseignement et éducation	enseignement dentaire
face	croissance et développement	développement maxillofacial
face	traumatismes	traumatisme face
facteur risque		
fibrinogène	déficit	afibrinogénémie
fluorures	intoxication	intoxication fluorure
foetoscopie	instrumentation	foetoscope
foetus	croissance et développement	développement embryonnaire et foetal

MC_ANTECEDENT	QU_ANTECEDENT	MC_CONSEQUENT
foetus	échographie	échographie prénatale
foetus	transplantation	transplantation tissu foetal
foie	transplantation	transplantation foie
froid	usage thérapeutique	cryothérapie
gaz	intoxication	intoxication gaz
genou	traumatismes	traumatisme genou
genou	traumatismes	traumatisme genou
glande salivaire	radiographie	sialographie
glande salivaire	radiographie	sialographie
glucose	sang	glycémie
glucose	urine	glycosurie
glucose 6-phosphate déhydrogénase	déficit	
glucose 6-phosphate déhydrogénase	déficit	
glucosylcéramidase	déficit	Gaucher, maladie
glucuronosyltransférase	déficit	Crigler-Najjar, syndrome
glycogen debranching enzyme	déficit	glycogénose type 3
glycogen phosphorylase, liver form	déficit	glycogénose type 6
glycogen phosphorylase, muscle form	déficit	glycogénose type 5
hanche	traumatismes	traumatisme hanche
hôpital	économie	économie hospitalière
hôpital	instrumentation	
hôpital	législation et jurisprudence	législation hôpital
hôpital	législation et jurisprudence	législation hôpital
hôpital	organisation et administration	administration hospitalière
hormones naturelles	antagonistes et inhibiteurs	antihormones
hormones thyroïdiennes	antagonistes et inhibiteurs	antithyroïdiens
hydroxyméthylglutaryl-CoA réductases	antagonistes et inhibiteurs	
hydroxyméthylglutaryl-CoA réductases	antagonistes et inhibiteurs	
hystéroscopie	instrumentation	hystéroscope
îlots Langerhans	transplantation	transplantation îlots Langerhans
immunoglobuline A	déficit	IgA, déficit
immunoglobuline G	déficit	IgG, déficit
insuline	antagonistes et inhibiteurs	antagoniste insuline
isosporidiose	chimiothérapie	coccidiostatiques
larynx	innervation	nerf laryngé
Lesch Nyhan, syndrome	A renseigner (par défaut)	hypoxanthine phosphoribosyltransférase
leucocyte	transplantation	transfusion leucocytes
lipoxygenase	antagonistes et inhibiteurs	inhibiteur lipoxygénase
lymphocyte	transplantation	transfusion lymphocytes
mâchoire	croissance et développement	développement maxillofacial
mâchoire	malformations	malformation mâchoire
magnésium	déficit	magnésium, carence
main	malformations	déformations congénitales main
main	traumatismes	traumatisme main
maladies transmissibles	prévention et contrôle	lutte contre maladie contagieuse
manganèse	intoxication	intoxication manganèse
manganèse	toxicité	intoxication manganèse
mannosidases	déficit	
maxillaire inférieur	traumatismes	traumatisme mandibulaire

MC_ANTECEDENT	QU_ANTECEDENT	MC_CONSEQUENT
médecine	économie	économie médicale
médecine	enseignement et éducation	enseignement médical
médecine	éthique	éthique médicale
médecine	histoire	histoire médecine
médecine	législation et jurisprudence	législation médicale
médecine	main d'oeuvre	personnel santé
médecine clinique	éthique	éthique clinique
médecine vétérinaire	enseignement et éducation	enseignement vétérinaire
médecine vétérinaire	législation et jurisprudence	législation vétérinaire
membre supérieur	malformations	
membre supérieur	traumatismes	
membres	malformations	anomalies congénitales membres
mercure	intoxication	hydrargyrisme
moelle épinière	traumatismes	traumatisme moelle épinière
moelle osseuse	cytologie	cellule moelle osseuse
moelle osseuse	transplantation	transplantation moelle osseuse
moelle osseuse	transplantation	transplantation moelle osseuse
monoxyde carbone	intoxication	intoxication oxyde de carbone
muscles	croissance et développement	développement musculaire
mycotoxine	intoxication	mycotoxicose
myocarde	chirurgie	interventions chirurgie cardiaque
myocarde	échographie	échocardiographie
myocarde	malformations	cardiopathies congénitales
myocarde	transplantation	transplantation cardiaque
myocarde	traumatismes	traumatisme coeur
myocarde	vascularisation	vaisseaux coronaires
nerf crânien	traumatismes	traumatisme nerf crânien
nerf facial	traumatismes	traumatisme nerf facial
nerf optique	traumatismes	traumatisme nerf optique
nez	chirurgie	rhinoplastie
oeil	chirurgie	
oeil	malformations	malformations oculaires
oeil	physiologie	physiologie oculaire
oeil	traumatismes	traumatisme oculaire
oreillette cardiaque	physiologie	fonction auriculaire
orthopédie	instrumentation	appareillage orthopédique
os	croissance et développement	croissance osseuse
os	transplantation	transplantation os
oxygène	déficit	anoxie
pancréas	transplantation	transplantation pancréas
peau	malformations	malformations cutanées
peau	physiologie	physiologie cutanée
peau	transplantation	transplantation peau
peau	transplantation	transplantation peau
peptide hydrolases	antagonistes et inhibiteurs	inhibiteurs protéinases
pharmacie (discipline)	économie	économie pharmaceutique
pharmacie (discipline)	enseignement et éducation	enseignement pharmacie
pharmacie (discipline)	éthique	éthique pharmacie
pharmacie (discipline)	législation et jurisprudence	législation pharmaceutique

MC_ANTECEDENT	QU_ANTECEDENT	MC_CONSEQUENT
phosphofructokinase-1, muscle type	déficit	glycogénose type 7
phosphofructokinase-1, muscle type	déficit	glycogénose type 7
pied	malformations	déformations congénitales pied
pied	traumatismes	traumatisme pied
plantes	intoxication	intoxication plante
plantes	physiologie	physiologie plantes
plantes		
plaquettes	transplantation	transfusion plaquettes
plomb	intoxication	intoxication plomb
poignet	traumatismes	traumatisme poignet
poignet	traumatismes	traumatisme poignet
potassium	déficit	potassium, carence
poumon	transplantation	transplantation poumon
préparations pharmaceutiques	antagonistes et inhibiteurs	antagonisme médicamenteux
préparations pharmaceutiques	intoxication	intoxication
préparations pharmaceutiques	législation et jurisprudence	
préparations pharmaceutiques	pharmacocinétique	pharmacocinétique
préparations pharmaceutiques	pharmacologie	pharmacologie
préparations pharmaceutiques	toxicité	toxicité pharmacochimique
préparations pharmaceutiques	usage thérapeutique	chimiothérapie
proaccélérine	déficit	facteur V, déficit
proconvertine	déficit	facteur VII, déficit
proconvertine	déficit	facteur VII, déficit
profession auxiliaire santé	main d'oeuvre	personnel santé auxiliaire
profession infirmier	économie	économie et soins infirmiers
profession infirmier	enseignement et éducation	enseignement infirmier
profession infirmier	éthique	éthique soins infirmiers
profession infirmier	histoire	histoire soins infirmiers
profession infirmier	législation et jurisprudence	législation soins infirmiers
prostaglandine E	analogues et dérivés	prostaglandine E synthétique
prostaglandine synthase	antagonistes et inhibiteurs	inhibiteur cyclooxygénase
prostaglandines	analogues et dérivés	prostaglandines synthétiques
prostaglandines	antagonistes et inhibiteurs	antagonistes prostaglandine
protéines	déficit	protéines, carence
protéines	liquide céphalorachidien	protéines liquide céphalorachidien
protéines	sang	protéines sang
protéines	urine	protéinurie
pyridoxine	déficit	vitamine B6, carence
pyruvate déhydrogénase, complexe	déficit	déficit en pyruvate déhydrogénase
pyruvate déhydrogénase, complexe	déficit	déficit en pyruvate déhydrogénase
rachis	traumatismes	traumatisme rachis
rein	transplantation	transplantation rein
sang	analyse	analyse de sang
sang	composition chimique	analyse de sang
sang	cytologie	cellule sanguine
sang	enzymologie	enzymes
sang	physiologie	physiologie sang
schizophrénie	psychologie	psychologie des schizophrènes
sein	échographie	échographie mammaire

MC_ANTECEDENT	QU_ANTECEDENT	MC_CONSEQUENT
sein	radiographie	mammographie
septum cardiaque	malformations	
sérine endopeptidases	antagonistes et inhibiteurs	inhibiteurs sérine protéinase
services santé	organisation et administration	administration services de soins
sports	traumatismes	traumatisme dû aux sports
système nerveux	chirurgie	interventions neurochirurgicales
système nerveux	malformations	malformations système nerveux
système nerveux	physiologie	physiologie système nerveux
système nerveux	traumatismes	traumatisme système nerveux
tendons	traumatismes	traumatisme tendon
tendons	traumatismes	traumatisme tendon
testosterone 5-alpha-reductase	déficit	pseudohermaphrodisme
Tête	traumatismes	traumatisme craniocérébral
thorax	radiographie	radiographie thoracique
thorax	traumatismes	traumatisme thorax
thorax	traumatismes	traumatisme thorax
thorax	traumatismes	traumatisme thorax
Tissu conjonctif	cytologie	cellule tissu conjonctif
transplantation	immunologie	immunologie transplantation
trompe Fallope	radiographie	hystérosalpingographie
trypsine	antagonistes et inhibiteurs	inhibiteurs trypsiques
utérus	radiographie	hystérosalpingographie
vaccin antibactérien		infections bactériennes
vaccin anticoquelucheux		coqueluche
vaccin anti-fièvre jaune		fièvre jaune
vaccin antigrippe		grippe
vaccin antihémophilus	prévention et contrôle	haemophilus influenzae
vaccin antihépatite B		hépatite B
vaccin antihépatite virale		hépatite virale humaine
vaccin antimorbilleux		rougeole
vaccin antiourlien		oreillons
vaccin antipoliomyélitique inactivé		poliomyélite
vaccin antirubéoleux		rubéole
vaccin anti-SIDA		SIDA
vaccin antityphoparatyphoïdique		fièvre typhoïde
vaccin antityphoparatyphoïdique		paratyphoïde, fièvre
vaccin antivaricelle		varicelle
vaccin antivariolique		variole
vaccin antiviral		maladies virales
vaccin BCG		tuberculose
vaccin diphtérie-tétanos-coqueluche		coqueluche
vaccin diphtérie-tétanos-coqueluche		diphtérie
vaccin diphtérie-tétanos-coqueluche		tétanos
vaccins anti-Alzheimer		Alzheimer, maladie
vaccins anti-charbonneux		charbon (maladie)
vaccins anticholériques		choléra
vaccins antiencéphalite japonaise		encéphalite japonaise
vaccins antihépatite A		hépatite B
vaccins anti-herpèsvirus		herpès

MC_ANTECEDENT	QU_ANTECEDENT	MC_CONSEQUENT
vaccins anti-maladie Lyme		Lyme, maladie
vaccins antiméningococciques		méningite méningococcique
vaccins antipneumococciques		streptococcus pneumoniae, infection
vaccins antipoliomyélitiques		poliomyélite
vaccins antirabiques		rage (maladie)
vaccins anti-rotavirus		rotavirus, infection
vaccins anti-salmonelle		salmonellose
vaccins anti-shigella		dysenterie bacillaire
vaisseaux coronaires	malformations	anomalie vaisseau coronaire
vaisseaux coronaires	radiographie	coronarographie
vaisseaux sanguins	chirurgie	intervention chirurgicale vasculaire
vaisseaux sanguins	radiographie	angiographie
vaisseaux sanguins	scintigraphie	angiographie isotopique
Veine porte	radiographie	splénoportographie
veines	radiographie	phlébographie
Venin	antagonistes et inhibiteurs	antitoxine venimeuse
ventricule cérébral	radiographie	ventriculographie
vésicule biliaire	radiographie	cholécystographie
Virus	physiologie	physiologie virus
vitamine A	déficit	rétinol, carence
vitamine B12	déficit	cyanocobalamine, carence
vitamine B6	déficit	vitamine B6, carence
vitamine D	déficit	vitamine D, carence
vitamine E	déficit	tocophérol, carence
vitamine K	déficit	vitamine K, carence
vitamines	déficit	avitaminoses
voie urinaire	physiologie	physiologie urinaire
voie urinaire	radiographie	urographie
voies biliaires	chirurgie	
1,4-alpha-glucan branching enzyme	déficit	glycogénose type 4
17-hydroxysteroid dehydrogenases	déficit	pseudohermaphrodisme
	antagonistes et inhibiteurs	inhibiteur croissance

Règles métier de A.Soualmia (médecin généraliste)

1. Complications (Hypertension artérielle, HTA) → accident vasculaire cérébral
2. Complications (Hypertension artérielle) → Insuffisance Rénale
3. Complications (Diabète) → Infarctus Myocarde
4. Complications (Diabète) → Neuropathies Diabétiques
5. Complications (Diabète) → Rétinopathies
6. Complications (Diabète) → Gangrène (gazeuse)
7. Complications (Ulcère gastrique) → Tumeur Estomac
8. Complications (Bronchite chronique) → insuffisance respiratoire
9. Complications (Insuffisance respiratoire) → cœur pulmonaire
10. Complications (Hépatite B) → Cirrhose
11. Complications (Alcoolisme) → Cirrhose
12. Complications (reflux gastrooesophagien, RGO) → œsophagite
13. Complications (œsophagite) →tumeur œsophage
14. Complications (Asthme) → insuffisance respiratoire
15. Complications (Valvulopathies) → endocardite bactérienne
16. Complications (angor instable) →Rhumatisme articulaire aigu (RAA)
17. Complications (vomissements) → déshydratation
18. Complications (Hyperthyroïdie) → troubles rythme cardiaque
19. Complications (hypothyroïdie) → constipation
20. Complications (hypertension artérielle ET grossesse) → mort fœtale… enfant
21. Complications (hypertension artérielle ET grossesse) → insuffisance rénale…mère
22. Complications (hypertension artérielle ET grossesse) → œdème pulmonaire…mère
23. Complications (pancréatite) → tumeur pancréas
24. Complications (ulcère duodénal) → perforation ulcère gastroduodénal
25. Complications (coeliaque, maladie) → anémie ferriptive
26. Complications (hématome rétroplacentaire) → mort fœtale ET rupture utérine

Classe de génération du fichier OWL à partir de la base de données

```
/**
 * Title :        CISMeFOWL-Generation...
* Date:           OCTOBER 2003
 */
//package cismef.knowqe;
import java.io.*;
import java.util.*;
import java.lang.String;
import oracle.jdbc.oci.*;
import oracle.jdbc.pool.*;
import oracle.jdbc.driver.*;
import java.sql.*;
/**
 * @author <a href="mailto:Badisse.Dahamna@chu-rouen.fr">Badisse DAHAMNA</a>
 * @version 1.0
 * @see CismefDataminer.class
 */
public class CismefOntology {
    /** {@value CHMN_FIC} */
    private final static String CHMN_FIC = "CismefOntology.owl";
    /** {@value ENTETE_OWL} */
    private final static String ENTETE_OWL = ""
        + "<?xml version=\"1.0\" encoding=\"UTF-8\"?> \n"
        + "<rdf:RDF xmlns:rdf=\"http://www.w3.org/1999/02/22-rdf-syntax-ns#\" \n"
        + "         xmlns:rdfs=\"http://www.w3.org/2000/01/rdf-schema#\" \n"
        + "             xmlns:owl=\"http://www.w3.org/2002/07/owl#\"> \n"
        + " \n"
        + "    <owl:Ontology rdf:about=\"\"> \n"
        + "        <rdfs:comment> \n"
        + "        CISMeF OWL Ontology \n"
        + "        Author: L.SOUALMIA (Lina.Soualmia@chu-rouen.fr) \n"
        + "        </rdfs:comment> \n"
        + "        <owl:imports rdf:resource=\"http://www.w3.org/2002/07/owl\" /> \n"
        + "    </owl:Ontology> \n"
        + " \n";
    /** {@value FIN_OWL} */
    private final static String FIN_OWL = "</rdf:RDF>";
     public static void main(String[] args) {
          try {
                // Arguments number control
                if (args.length != 0) {
                ..........throw new IllegalArgumentException("Wrong parameters number");
                }
                else {
                ..........// convert file passed in argument into an array of sortedSet
                ..............................CismefOntology CO = new CismefOntology();
                System.err.println(" *** DEBUT *** ");
                CO.lancement();
                System.err.println(" *** FIN *** ");
                };
          }
          catch (Exception err) {
                System.err.println(err);
                System.err.println("Usage: java CismefOntology");
```

```
         }
      }
      /**
       * lancement
       */
      public void lancement() {
          // Ouverture du fichier destination
        PrintWriter out = null;
        try {
         out = new PrintWriter(new FileOutputStream(CHMN_FIC));
        }
         catch (Exception err) { .............................System.err.println(err); }
        // Connexion a la base
        OracleOCIConnectionPool cpool = null;
        OracleOCIConnection connexionBase = null;
        try {
        cpool      =      new      OracleOCIConnectionPool      ("cismef",      "cismef",
"jdbc:oracle:oci8:@ORADCD1.WORLD", null);
        connexionBase = (OracleOCIConnection)cpool.getConnection();
        }
        catch (Exception err) { System.err.println(err); }
       // Variables pour requetes
        OraclePreparedStatement preStatement = null;
        OracleResultSet resultatRequete = null;
         String requete = new String();
        Object colonne1, colonne2;
        String ligne = new String();
        // Ecriture de l'entete du fichier
        out.println(ENTETE_OWL);
        // ------------------------------------------------------------------------------
        //                      MOTCLEFS
        // ------------------------------------------------------------------------------
         Vector motclefs = new Vector();
         String numMotclef = new String();
         String strClass = new String();
      try {
            /*requete    =    "    SELECT    DISTINCT    replace(replace(MOTCLEF_SA    ||
decode(TB_QUALIFICATIF.NUM_QUALIFICATIF,83,'',' ' || QUALIFICATIF_SA),' ','_'),',','_')
MOTCLEF_QUALIFIE, TB_MOTCLEFS_FICHE.NUM_MOTCLEF, TB_MOTCLEFS_FICHE.NUM_QUALIFICATIF ";
                  + " FROM TB_MOTCLEFS_FICHE, TB_MOTCLEFS, TB_QUALIFICATIF ";
                  + " WHERE TB_MOTCLEFS.NUM_MOTCLEF = TB_MOTCLEFS_FICHE.NUM_MOTCLEF ";
                  +       "       AND       TB_QUALIFICATIF.NUM_QUALIFICATIF       =
TB_MOTCLEFS_FICHE.NUM_QUALIFICATIF ";
                  //+     "     AND     TB_MOTCLEFS_FICHE.NUM_MOTCLEF     in     (select
TB_NMT.NUM_MOTCLEF_DES from TB_NMT where NUM_META_PERE = 6) ";
                  //+    "    AND    TB_MOTCLEFS_FICHE.NUM_QUALIFICATIF    in    (select
TB_NMTQ.NUM_QUALIF_DES from TB_NMTQ where NUM_META_PERE = 6) ";
                  + " ORDER BY 1 "; */
            requete       =       "       SELECT       DISTINCT       NUM_MOTCLEF,
replace(replace(TB_MOTCLEFS.MOTCLEF_SA,' ','_'),',','_') "
                  + " FROM TB_MOTCLEFS "
+ " WHERE ROWNUM < 10 "
                  + " ORDER BY 1 ";
           // On prepare la requete
           preStatement                                                              =
(OraclePreparedStatement)connexionBase.prepareStatement(requete);
           // Execution de la requête
```

```
            resultatRequete = (OracleResultSet)preStatement.executeQuery();
                while (resultatRequete.next())
                {
                Enregistrement              enrCourant              =              new
Enregistrement(resultatRequete.getObject(1),resultatRequete.getObject(2));
                 ....................................motclefs.addElement(enrCourant);
                }
                resultatRequete.close();
                preStatement.close();
          }
          catch(Exception err)
          {
                System.err.println (" ERREUR SQLException :" + err.getMessage());
                System.err.println ("        A la requete :" + requete);
          }
        OraclePreparedStatement preStatementMotclefPeres = null;
        OraclePreparedStatement preStatementMotclefProprietes = null;
        // On prepare les requetes sur 1 motclef en laissant un '?' comme parametre
          try {
              requete   =   "   SELECT   DISTINCT   replace(replace(TB_MC_PERE.MOTCLEF_SA,'
','_'),',','_') "
                    +   "   FROM   TB_MOTCLEFS   TB_MC_EN_COURS,   TB_NMOTS,   TB_MOTCLEFS
TB_MC_PERE "
                    + " WHERE TB_NMOTS.NUM_MOTCLEF_DES = TB_MC_EN_COURS.NUM_MOTCLEF "
                    + " AND TB_NMOTS.DIFF_NIVEAU = 1 "
                    + " AND TB_MC_PERE.NUM_MOTCLEF = TB_NMOTS.NUM_MOTCLEF_PERE "
                    + " AND TB_MC_EN_COURS.NUM_MOTCLEF = ? ";
            preStatementMotclefPeres                                                =
(OraclePreparedStatement)connexionBase.prepareStatement(requete);
              requete  =  "  SELECT  DISTINCT  NUM_MOTCLEF,  TB_MOTCLEFS.MOTCLEF  LIBELLE,
TB_MOTCLEFS.KEYWORD LIBELLE_ANG "
                    + " FROM TB_MOTCLEFS "
                    + " WHERE TB_MOTCLEFS.NUM_MOTCLEF = ? ";
            preStatementMotclefProprietes                                           =
(OraclePreparedStatement)connexionBase.prepareStatement(requete);
        }
          catch(Exception err)
          {
                System.err.println   ("   ERREUR   SQLException                    :"   +
err.getMessage());
                System.err.println (" A preparation de la requete :" + requete);
          }
        // ----------------------
        //  POUR CHAQUE MOTCLEF
        // ----------------------
        for (int i =0 ; i< motclefs.size() ; i++) {
              Enregistrement enrMotclef = (Enregistrement)(motclefs.elementAt(i));
            numMotclef = enrMotclef.Colonne1().toString();
            strClass = enrMotclef.Colonne2().toString();
            out.println("    <owl:Class  rdf:ID=\"" + enrMotclef.Colonne2().toString() +
"\"> ");
            // -------------------------
            //      PERES DU MOTCLEF
            // -------------------------
                try {
                Vector motclefPeres = new Vector();
                // On complete le statement
```

```
                preStatementMotclefPeres.setString(1, numMotclef);
                // Execution de la requête
                resultatRequete                                                    =
(OracleResultSet)preStatementMotclefPeres.executeQuery();
                while (resultatRequete.next())
                {
                      Enregistrement            enrCourant            =            new
Enregistrement(resultatRequete.getObject(1));
                ..........................      motclefPeres.addElement(enrCourant);
                }
                resultatRequete.close();
                // Ecriture de la class
                if (motclefPeres.size() == 0) {
                    out.println("      <rdfs:subClassOf> ");
                    out.println("                                                  <owl:Class
rdf:resource=\"#http://www.w3.org/2002/07/owl#Thing\" /> ");
                    out.println("      </rdfs:subClassOf> ");
                } else if (motclefPeres.size() == 1) {
                        Enregistrement                  enrCourant                  =
(Enregistrement)(motclefPeres.elementAt(0));
                    out.println("      <rdfs:subClassOf> ");
                    out.println("                          <owl:Class   rdf:about=\"#"   +
enrCourant.Colonne1().toString() + "\" /> ");
                    out.println("      </rdfs:subClassOf> ");
                } else {
                    out.println("      <rdfs:subClassOf> ");
                    out.println("      <intersectionOf rdf:parseType=\"Collection\"> ");
                  for (int j =0 ; j< motclefPeres.size() ; j++) {
                        Enregistrement                  enrCourant                  =
(Enregistrement)(motclefPeres.elementAt(j));
                        out.println("                    <owl:Class   rdf:about=\"#"   +
enrCourant.Colonne1().toString() + "\" /> ");
                    }
                    out.println("      </intersectionOf> ");
                    out.println("      </rdfs:subClassOf> ");
                }
          }
          catch(Exception err)
          {
                System.err.println (" ERREUR SQLException :" + err.getMessage());
                System.err.println ("        A la requete :" + requete);
          }
             out.println("   </owl:Class> ");
             /*
             // ------------------------------
             //      PROPRIETES DU MOTCLEF
             // ------------------------------
                try {
                // On complete le statement
                preStatementMotclefProprietes.setString(1, numMotclef);
                // Execution de la requête
                resultatRequete                                                    =
(OracleResultSet)preStatementMotclefProprietes.executeQuery();
                while (resultatRequete.next())
                {
                    out.println("     <owl:DatatypeProperty  rdf:ID=\""  +  "ID_CISMeF"  +
"\"> ");
```

```
                    out.println("                    <rdfs:domain   rdf:resource=\"#"   +
resultatRequete.getObject(1) + "\" /> ");
                    out.println("                                                        <rdfs:range
rdf:resource=\"http://www.w3.org/2001/XMLSchema#int\" /> ");
                    out.println("   </owl:DatatypeProperty> ");
                    out.println("              <owl:DatatypeProperty     rdf:ID=\""     +
"LIBELLE_ACCENTUE" + "\"> ");
                    out.println("                    <rdfs:domain   rdf:resource=\"#"   +
resultatRequete.getObject(2) + "\" /> ");
                    out.println("                                                        <rdfs:range
rdf:resource=\"http://www.w3.org/2001/XMLSchema#string\" /> ");
                    out.println("   </owl:DatatypeProperty> ");
                    out.println("   <owl:DatatypeProperty rdf:ID=\"" + "LIBELLE_ANGLAIS"
+ "\"> ");
                    out.println("                    <rdfs:domain   rdf:resource=\"#"   +
resultatRequete.getObject(3) + "\" /> ");
                    out.println("                                                        <rdfs:range
rdf:resource=\"http://www.w3.org/2001/XMLSchema#string\" /> ");
                    out.println("   </owl:DatatypeProperty> ");
                }
                resultatRequete.close();
          }
          catch(Exception err)
          {
                System.err.println (" ERREUR SQLException :" + err.getMessage());
                System.err.println ("        A la requete :" + requete);
          }
             */
          // Retour chariot
          out.println(" ");
        } // Fin For
        // ----------------------------------------------------------------------------
        //                      QUALIFICATIFS
        // ----------------------------------------------------------------------------
          Vector qualificatifs = new Vector();
          String numQualificatif = new String();
      try {
              requete   =   "   SELECT   TB_QUALIFICATIF.NUM_QUALIFICATIF,   'qu_'   ||
replace(replace(TB_QUALIFICATIF.QUALIFICATIF_SA,' ','_'),',','_') "
                    + " FROM TB_QUALIFICATIF "
//+ " WHERE ROWNUM < 10 "
                    + " ORDER BY 1 ";
            // On prepare la requete
            preStatement                                                            =
(OraclePreparedStatement)connexionBase.prepareStatement(requete);
            // Execution de la requête
            resultatRequete = (OracleResultSet)preStatement.executeQuery();
                while (resultatRequete.next())
                {
                Enregistrement              enrCourant              =              new
Enregistrement(resultatRequete.getObject(1),resultatRequete.getObject(2));
                ................................qualificatifs.addElement(enrCourant);
                }
                resultatRequete.close();
                preStatement.close();
          }
          catch(Exception err)
          {
```

```
                System.err.println (" ERREUR SQLException :" + err.getMessage());
                System.err.println ("        A la requete :" + requete);
          }
        OraclePreparedStatement preStatementQualificatifPeres = null;
        OraclePreparedStatement preStatementQualificatifProprietes = null;
        // On prepare les requetes sur 1 motclef en laissant un '?' comme parametre
          try {
              requete       =       "       SELECT       DISTINCT       'qu_'       ||
replace(replace(TB_QU_PERE.QUALIFICATIF_SA,' ','_'),',','_') "
                    +    "    FROM    TB_QUALIFICATIF    TB_QU_EN_COURS,    TB_NQUALIFS,
TB_QUALIFICATIF TB_QU_PERE "
                    +        "        WHERE        TB_NQUALIFS.NUM_QUALIF_DES        =
TB_QU_EN_COURS.NUM_QUALIFICATIF "
                    + " AND TB_NQUALIFS.NUM_QUALIF_PERE <> TB_NQUALIFS.NUM_QUALIF_DES "
                    + " AND TB_QU_PERE.NUM_QUALIFICATIF = TB_NQUALIFS.NUM_QUALIF_PERE "
                    + " AND TB_QU_EN_COURS.NUM_QUALIFICATIF = ? ";
            preStatementQualificatifPeres                                              =
(OraclePreparedStatement)connexionBase.prepareStatement(requete);
             requete = " SELECT DISTINCT NUM_QUALIFICATIF, TB_QUALIFICATIF.QUALIFICATIF
LIBELLE, TB_QUALIFICATIF.SUBHEADING LIBELLE_ANG "
                    + " FROM TB_QUALIFICATIF "
                    + " WHERE TB_QUALIFICATIF.NUM_QUALIFICATIF = ? ";
            preStatementQualificatifProprietes                                         =
(OraclePreparedStatement)connexionBase.prepareStatement(requete);
        }
          catch(Exception err)
          {
                System.err.println  ("  ERREUR  SQLException                    :"  +
err.getMessage());
                System.err.println (" A preparation de la requete :" + requete);
          }
        // ----------------------------
        //  POUR CHAQUE QUALIFICATIF
        // ----------------------------
        for (int i =0 ; i< qualificatifs.size() ; i++) {
              Enregistrement                      enrQualificatif                      =
(Enregistrement)(qualificatifs.elementAt(i));
            numQualificatif = enrQualificatif.Colonne1().toString();
            strClass = enrQualificatif.Colonne2().toString();
            out.println("                   <owl:ObjectProperty      rdf:ID=\""      +
enrQualificatif.Colonne2().toString() + "\"> ");
            // ------------------------------
            //      PERES DU QUALIFICATIF
            // ------------------------------
                try {
                Vector qualificatifPeres = new Vector();
                // On complete le statement
                preStatementQualificatifPeres.setString(1, numQualificatif);
                // Execution de la requête
                resultatRequete                                                        =
(OracleResultSet)preStatementQualificatifPeres.executeQuery();
                while (resultatRequete.next())
                {
                      Enregistrement            enrCourant            =            new
Enregistrement(resultatRequete.getObject(1));
                 ......................      qualificatifPeres.addElement(enrCourant);
                }
                resultatRequete.close();
```

```
                // Ecriture de la class
                if (qualificatifPeres.size() == 0) {
                } else if (qualificatifPeres.size() == 1) {
                        Enregistrement                    enrCourant                    =
(Enregistrement)(qualificatifPeres.elementAt(0));
                    out.println("      <rdfs:subPropertyOf> ");
                    out.println("                  <owl:ObjectProperty  rdf:about=\"#"  +
enrCourant.Colonne1().toString() + "\" /> ");
                    out.println("      </rdfs:subPropertyOf> ");
                } else {
                    out.println("      <rdfs:subPropertyOf> ");
                    out.println("      <intersectionOf rdf:parseType=\"Collection\"> ");
                  for (int j =0 ; j< qualificatifPeres.size() ; j++) {
                        Enregistrement                    enrCourant                    =
(Enregistrement)(qualificatifPeres.elementAt(j));
                        out.println("              <owl:ObjectProperty  rdf:about=\"#"  +
enrCourant.Colonne1().toString() + "\" /> ");
                    }
                    out.println("      </intersectionOf> ");
                    out.println("      </rdfs:subPropertyOf> ");
                }
          }
          catch(Exception err)
          {
                System.err.println (" ERREUR SQLException :" + err.getMessage());
                System.err.println ("        A la requete :" + requete);
          }
             out.println("   </owl:ObjectProperty> ");
          // Retour chariot
          out.println(" ");
        } // Fin For
        // ----------------------------------------------------------------------------
        //                      MOTCLEFS QUALIFIES
        // ----------------------------------------------------------------------------
          Vector motclefsQualifies = new Vector();
          String numMotclefAssocie = new String();
          String numQualificatifAssocie = new String();
          strClass = new String();
      try {
              requete    =    "    SELECT    DISTINCT    TB_MOTCLEFS_FICHE.NUM_QUALIFICATIF,
TB_MOTCLEFS_FICHE.NUM_MOTCLEF,  replace(replace(MOTCLEF_SA  ||  '_qu_'  ||  QUALIFICATIF_SA
,' ','_'),',','_') MOTCLEF_QUALIFIE "
                    + " FROM TB_MOTCLEFS_FICHE, TB_MOTCLEFS, TB_QUALIFICATIF "
                    + " WHERE TB_MOTCLEFS.NUM_MOTCLEF = TB_MOTCLEFS_FICHE.NUM_MOTCLEF "
                    + " AND TB_QUALIFICATIF.NUM_QUALIFICATIF <> 83 "
+ " AND ROWNUM < 100 "
                    +        "        AND        TB_QUALIFICATIF.NUM_QUALIFICATIF        =
TB_MOTCLEFS_FICHE.NUM_QUALIFICATIF "
                    + " ORDER BY 1 ";
              // On prepare la requete
              preStatement                                                              =
(OraclePreparedStatement)connexionBase.prepareStatement(requete);
              // Execution de la requête
              resultatRequete = (OracleResultSet)preStatement.executeQuery();
                  while (resultatRequete.next())
                  {
```

```
               Enregistrement              enrCourant              =              new
Enregistrement(resultatRequete.getObject(1),resultatRequete.getObject(2),resultatRequete
.getObject(3));
                ..........................motclefsQualifies.addElement(enrCourant);
               }
               resultatRequete.close();
               preStatement.close();
         }
         catch(Exception err)
         {
               System.err.println (" ERREUR SQLException :" + err.getMessage());
               System.err.println ("        A la requete :" + requete);
         }
        OraclePreparedStatement preStatementQualificatif = null;
        OraclePreparedStatement preStatementMotclef = null;
        // On prepare les requetes sur 1 motclef en laissant un '?' comme parametre
         try {
             requete  =  "  SELECT  DISTINCT  'qu_'  ||  replace(replace(QUALIFICATIF_SA,'
','_'),',','_'),    TB_QUALIFICATIF.QUALIFICATIF    LIBELLE,    TB_QUALIFICATIF.SUBHEADING
LIBELLE_ANG "
                  + " FROM TB_QUALIFICATIF "
                  + " WHERE TB_QUALIFICATIF.NUM_QUALIFICATIF = ? ";
            preStatementQualificatif                                                    =
(OraclePreparedStatement)connexionBase.prepareStatement(requete);
             requete = " SELECT DISTINCT replace(replace(MOTCLEF_SA,' ','_'),',','_'),
TB_MOTCLEFS.MOTCLEF LIBELLE, TB_MOTCLEFS.KEYWORD LIBELLE_ANG "
                  + " FROM TB_MOTCLEFS "
                  + " WHERE TB_MOTCLEFS.NUM_MOTCLEF = ? ";
            preStatementMotclef                                                         =
(OraclePreparedStatement)connexionBase.prepareStatement(requete);
        }
         catch(Exception err)
         {
               System.err.println  ("  ERREUR  SQLException                    :"  +
err.getMessage());
               System.err.println (" A preparation de la requete :" + requete);
         }
        // ----------------------
        //  POUR CHAQUE MOTCLEF
        // ----------------------
        for (int i =0 ; i< motclefsQualifies.size() ; i++) {
              Enregistrement                        enrMotclef                        =
(Enregistrement)(motclefsQualifies.elementAt(i));
            numQualificatifAssocie = enrMotclef.Colonne1().toString();
            numMotclefAssocie = enrMotclef.Colonne2().toString();
            strClass = enrMotclef.Colonne3().toString();
            out.println("    <owl:Class rdf:ID=\"" + enrMotclef.Colonne3().toString() +
"\"> ");
            out.println("      <rdfs:subClassOf> ");
            out.println("         <owl:Restriction> ");
            // -------------------------
            //      QUALIFICATIF
            // -------------------------
                try {
                Vector qualificatif = new Vector();
                // On complete le statement
                preStatementQualificatif.setString(1, numQualificatifAssocie);
                // Execution de la requête
```

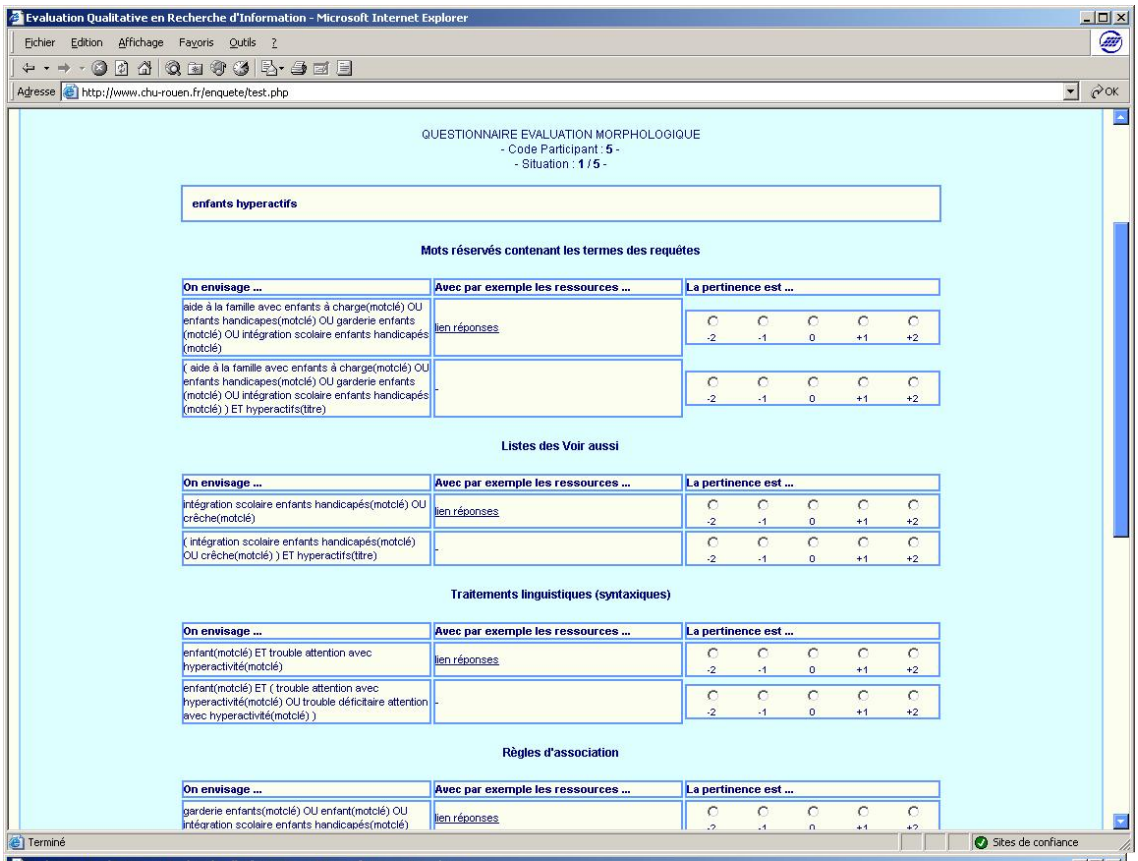
```
                resultatRequete                                                    =
(OracleResultSet)preStatementQualificatif.executeQuery();
                while (resultatRequete.next())
                {
                      Enregistrement            enrCourant            =            new
Enregistrement(resultatRequete.getObject(1));
                ..........................      qualificatif.addElement(enrCourant);
                }
                resultatRequete.close();
                // Ecriture de la class
                if (qualificatif.size() == 1) {
                       Enregistrement                  enrCourant                   =
(Enregistrement)(qualificatif.elementAt(0));
                    out.println("                  <owl:onProperty  rdf:resource=\"#"  +
enrCourant.Colonne1().toString() + "\" /> ");
                }
          }
          catch(Exception err)
          {
                System.err.println (" ERREUR SQLException :" + err.getMessage());
                System.err.println ("        A la requete :" + requete);
          }
             // -------------------------
             //      MOTCLEF
             // -------------------------
                try {
                Vector motclef = new Vector();
                // On complete le statement
                preStatementMotclef.setString(1, numMotclefAssocie);
                // Execution de la requête
                resultatRequete = (OracleResultSet)preStatementMotclef.executeQuery();
                while (resultatRequete.next())
                {
                      Enregistrement            enrCourant            =            new
Enregistrement(resultatRequete.getObject(1));
                ..............................      motclef.addElement(enrCourant);
                }
                resultatRequete.close();
                // Ecriture de la class
                if (motclef.size() == 1) {
                       Enregistrement                  enrCourant                   =
(Enregistrement)(motclef.elementAt(0));
                    out.println("               <owl:someValuesFrom rdf:resource=\"#" +
enrCourant.Colonne1().toString() + "\" /> ");
                }    ..............................................................
        }
          catch(Exception err)
          {
                System.err.println (" ERREUR SQLException :" + err.getMessage());
                System.err.println ("        A la requete :" + requete);
          }
             out.println("         </owl:Restriction> ");
             out.println("      </rdfs:subClassOf> ");
             out.println("   </owl:Class> ");
          out.println(" ");
         } // Fin For
         // Ecriture de la fin du fichier
```

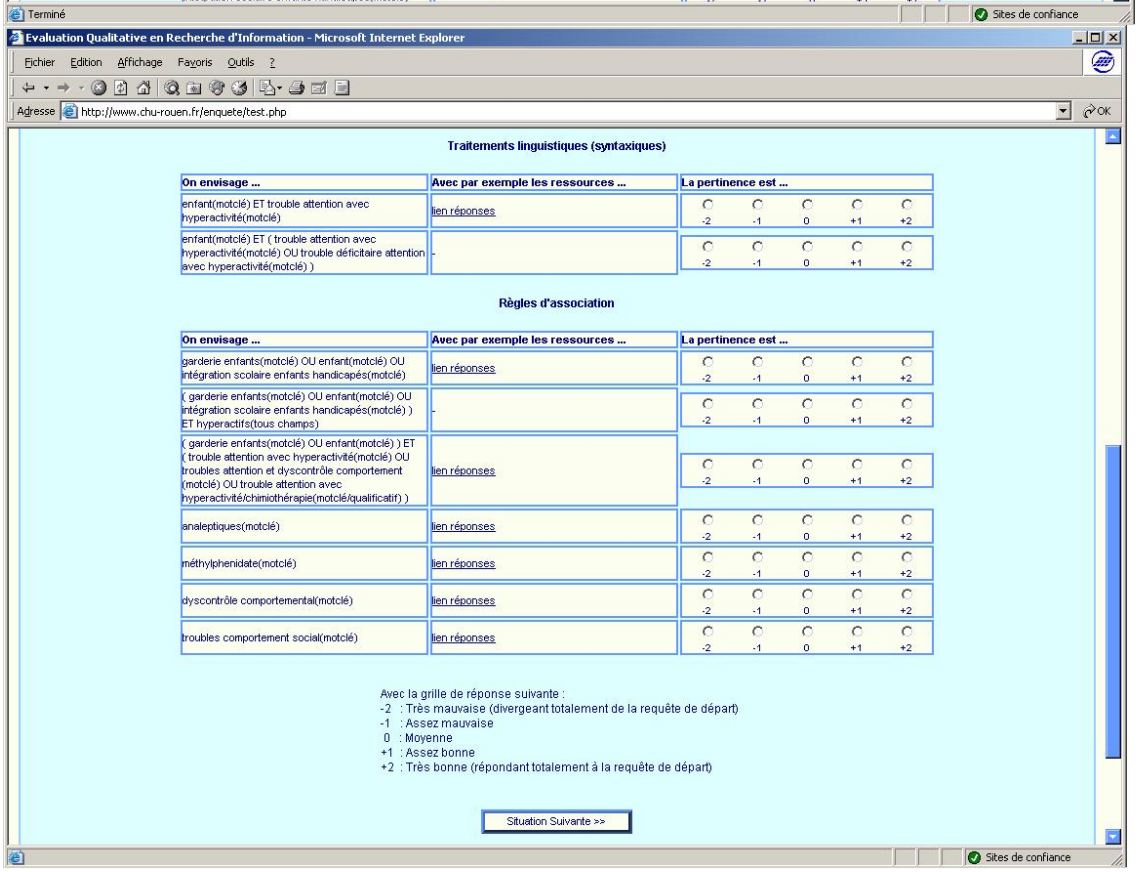
```
        out.println(FIN_OWL);
        // Fermetures
        try {
            preStatementMotclefPeres.close();
            preStatementMotclefProprietes.close();
            connexionBase.close();
             cpool.close();
               out.close();
         }
         catch (Exception err) {
             System.err.println (" ERREUR finally :" + err.getMessage());
         }
      }
    public class Enregistrement {
        Object colonne1;
        Object colonne2;
        Object colonne3;
        Object colonne4;
        public Enregistrement (Object colonne1) {
            this.colonne1 = colonne1;
        }
        public Enregistrement (Object colonne1, Object colonne2) {
            this.colonne1 = colonne1;
            this.colonne2 = colonne2;
        }
        public Enregistrement (Object colonne1, Object colonne2, Object colonne3) {
            this.colonne1 = colonne1;
            this.colonne2 = colonne2;
            this.colonne3 = colonne3;
        }
        public Enregistrement (Object colonne1, Object colonne2, Object colonne3, Object
colonne4) {
            this.colonne1 = colonne1;
            this.colonne2 = colonne2;
            this.colonne3 = colonne3;
            this.colonne4 = colonne4;
        }
        public Object Colonne1() {
            return colonne1;
        }
        public Object Colonne2() {
            return colonne2;
        }
        public Object Colonne3() {
            return colonne3;
        }
        public Object Colonne4() {
            return colonne4;
        }
    }
} // End Class
```

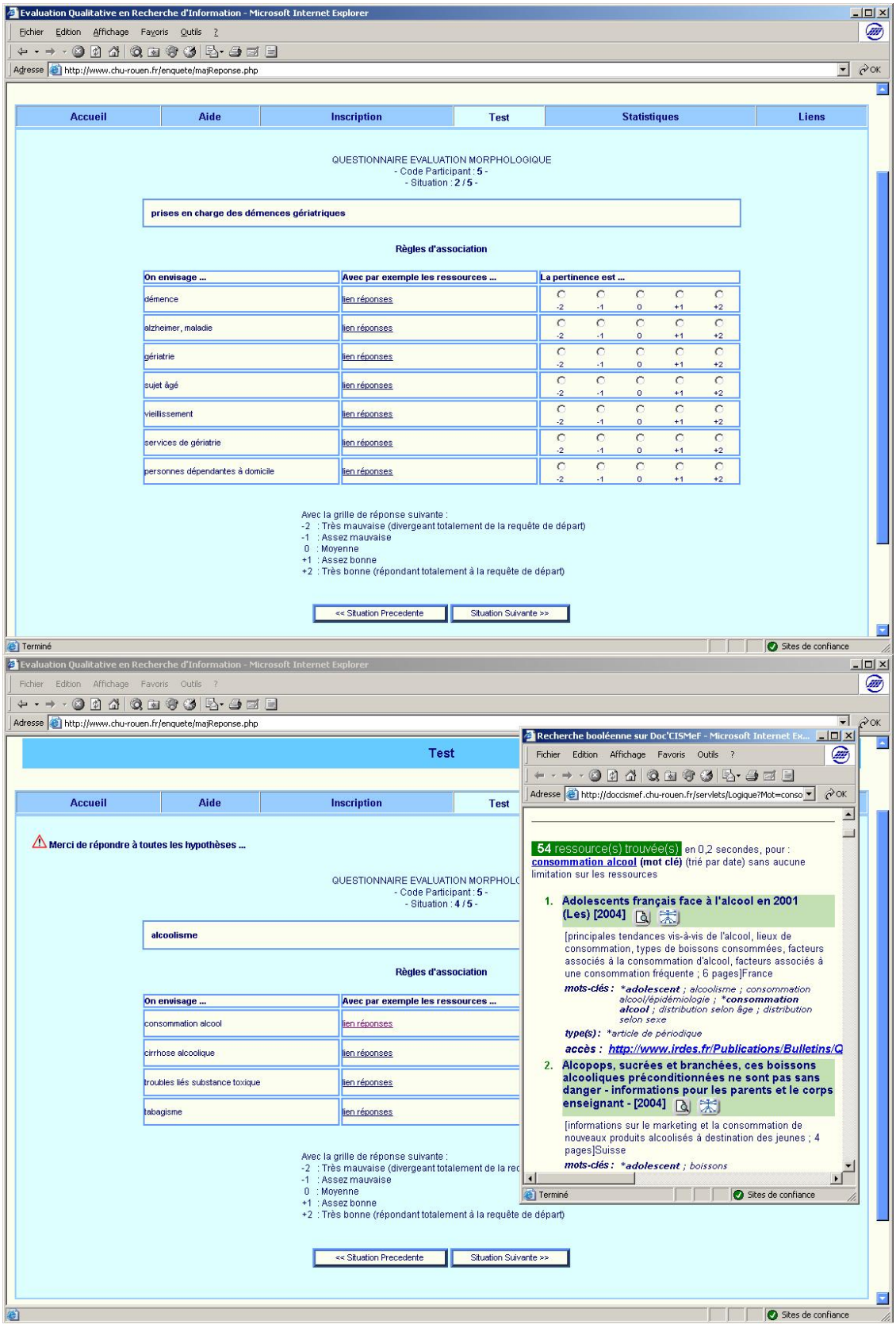
Extrait de Lexique (http://www.lexique.org)

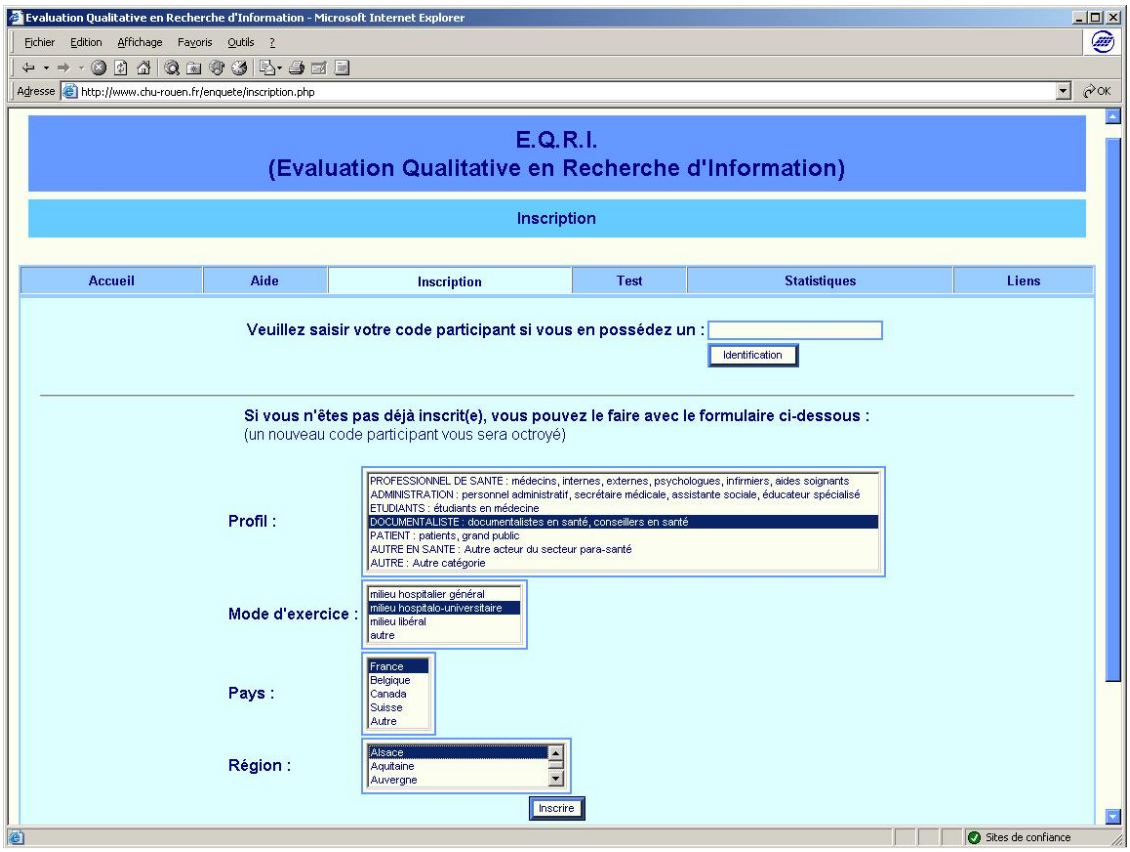
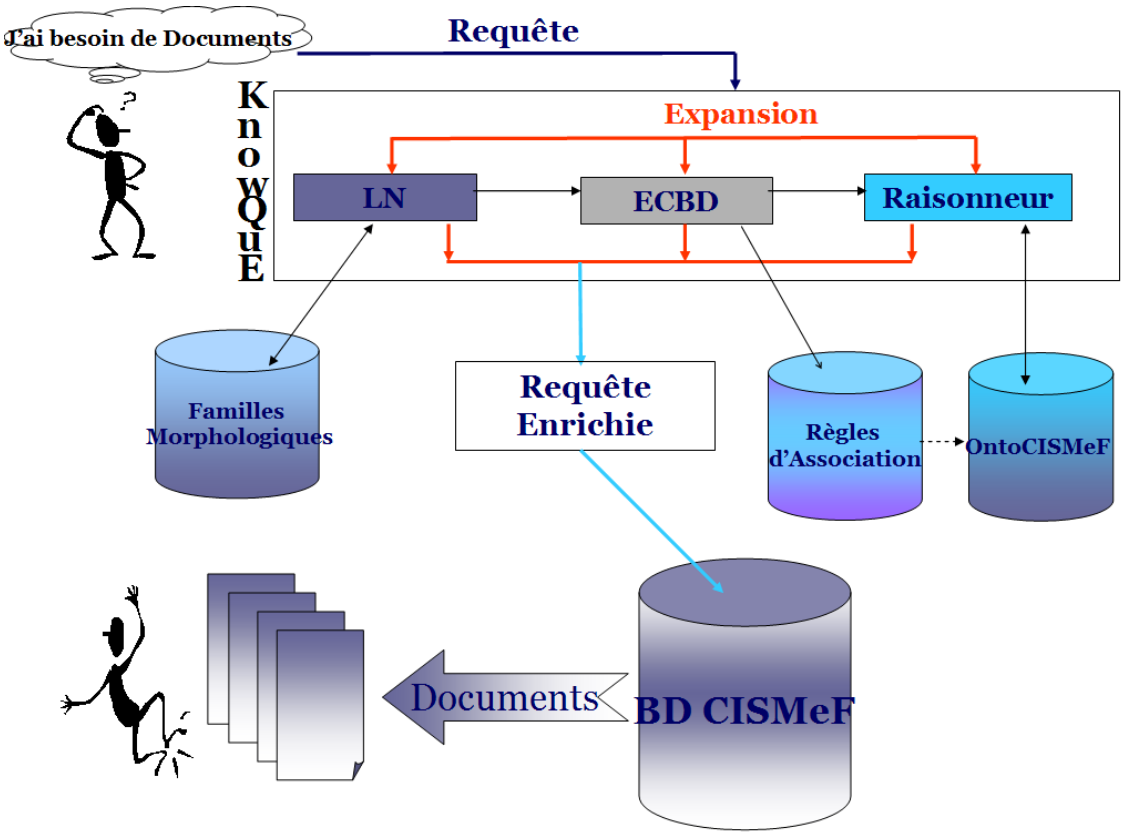
lem	graph	phon	cgram	genre
accident	accident;accidents	aksid@	NOM	m
accidenté	accidenté;accidentée;accidentées;accidentés	aksid@te	ADJ;VER:pper	f;m
accidentel	accidentel;accidentelle;accidentelles;accidentels	aksid@tEl	ADJ	f;m
accidentellement	accidentellement	aksid@tElm@	ADV	
accidenter	accidente;accidenté;accidentée;accidentées;accidentés	aksid@t;aksid@te	ADJ;VER:imp:pr;ind:pr;pper;sub:pr	f;m

Interfaces du serveur d'évaluation des projections intéractives.






Evaluation Qualitative en Recherche d'Information - Microsoft Internet Explorer
Fichier Edition Affichage Favoris Outils ?
Adresse http://www.chu-rouen.fr/enquete/majReponse.php
Accueil
Aide
Inscription
Test
Statistiques
Liens
QUESTIONNAIRE EVALUATION MORPHOLOGIQUE
- Code Participant : 5 -
- Situation : 2 / 5 -
prises en charge des démences gériatriques
Règles d'association
On envisage ...
Avec par exemple les ressources ...
La pertinence est ...
démence
alzheimer, maladie
gériatrie
sujet âgé
vieillissement
services de gériatrie
personnes dépendantes à domicile
lien réponses
-2 -1 0 +1 +2
Avec la grille de réponse suivante :
-2 : Très mauvaise (divergeant totalement de la requête de départ)
-1 : Assez mauvaise
0 : Moyenne
+1 : Assez bonne
+2 : Très bonne (répondant totalement à la requête de départ)
<< Situation Precedente
Situation Suivante >>
Terminé
Sites de confiance
Evaluation Qualitative en Recherche d'Information - Microsoft Internet Explorer
Test
Merci de répondre à toutes les hypothèses ...
QUESTIONNAIRE EVALUATION MORPHOLO
- Code Participant : 5 -
- Situation : 4 / 5 -
alcoolisme
Règles d'association
On envisage ...
Avec par exemple les ressources ...
consommation alcool
cirrhose alcoolique
troubles liés substance toxique
tabagisme
lien réponses
Recherche booléenne sur Doc'CISMeF - Microsoft Internet Ex...
http://doccismef.chu-rouen.fr/servlets/Logique?Mot=conso
54 ressource(s) trouvée(s) en 0,2 secondes, pour :
consommation alcool (mot clé) (trié par date) sans aucune limitation sur les ressources
1. Adolescents français face à l'alcool en 2001 (Les) [2004]
[principales tendances vis-à-vis de l'alcool, lieux de consommation, types de boissons consommées, facteurs associés à la consommation d'alcool, facteurs associés à une consommation fréquente ; 6 pages]France
mots-clés : *adolescent ; alcoolisme ; consommation alcool/épidémiologie ; *consommation alcool ; distribution selon âge ; distribution selon sexe
type(s) : *article de périodique
accès : http://www.irdes.fr/Publications/Bulletins/Q
2. Alcopops, sucrées et branchées, ces boissons alcooliques préconditionnées ne sont pas sans danger - informations pour les parents et le corps enseignant - [2004]
[informations sur le marketing et la consommation de nouveaux produits alcoolisés à destination des jeunes ; 4 pages]Suisse
mots-clés : *adolescent ; boissons


Evaluation Qualitative en Recherche d'Information - Microsoft Internet Explorer
Fichier Edition Affichage Favoris Outils ?
Adresse http://www.chu-rouen.fr/enquete/inscription.php
E.Q.R.I.
(Evaluation Qualitative en Recherche d'Information)
Inscription
Accueil
Aide
Inscription
Test
Statistiques
Liens
Veuillez saisir votre code participant si vous en possédez un :
Identification
Si vous n'êtes pas déjà inscrit(e), vous pouvez le faire avec le formulaire ci-dessous :
(un nouveau code participant vous sera octroyé)
Profil :
PROFESSIONNEL DE SANTE : médecins, internes, externes, psychologues, infirmiers, aides soignants
ADMINISTRATION : personnel administratif, secrétaire médicale, assistante sociale, éducateur spécialisé
ETUDIANTS : étudiants en médecine
DOCUMENTALISTE : documentalistes en santé, conseillers en santé
PATIENT : patients, grand public
AUTRE EN SANTE : Autre acteur du secteur para-santé
AUTRE : Autre catégorie
Mode d'exercice :
milieu hospitalier général
milieu hospitalo-universitaire
milieu libéral
autre
Pays :
France
Belgique
Canada
Suisse
Autre
Région :
Alsace
Aquitaine
Auvergne
Inscrire
Sites de confiance

Processus global d'expansion de requêtes