UNIVERSITÉ DU QUÉBEC À MONTRÉAL

UN SYSTÈME DATA MINING EN LIGNE POUR LA MAINTENANCE ONTOLOGIQUE D'UNE MÉMOIRE CORPORATIVE DM

THÈSE

PRÉSENTÉE

COMME EXIGENCE PARTIELLE

DU DOCTORAT EN INFORMATIQUE COGNITIVE

PAR

CHOUKRI DJELLALI

DÉCEMBRE 2014

REMERCIEMENTS

Je voudrais commencer par remercier tout particulièrement mes directeurs de recherche qui m'ont aidé à découvrir mes intérêts de recherche, pour leurs disponibilités très appréciées et pour avoir partagés avec moi leurs compétences personnelles et professionnelles afin de mener à bien mon projet de recherche.

J'exprime toute ma gratitude aux membres du jury.

J'exprime aussi ma gratitude à Belhard et Yassine qui ont bien voulu partager des renseignements bibliographiques, m'ont éclairé sur des approches nouvelles et m'ont fait des suggestions à la lumière de leurs expériences. Je ne peux passer sous silence le travail professionnel des secrétaires et tout particulièrement, la très grande disponibilité des directeurs de doctorat en informatique cognitive.

Je remercie tout particulièrement mes compagnons du Laboratoire LANCI (Laboratoire d'ANalyse Cognitive de l'Information) qui m'ont fourni une aide précieuse au quotidien pour discuter des mille et un petit problèmes rencontrés ainsi que pour leur support et d'innombrables bons moments. Je tiens aussi à dire un grand merci à: Rabi, Mohamed, Oxana, Wafa, Mathieu, Lise, Léo, Travers, Fathi, Fawzi, Radouane, Abdel Adime, Djeddi, Hossam, Kayhane Karim, Isabelle, Rahma, Eureka, Sanaa, etc.

Je remercie également le consultant Amine pour son aide précieuse sur le plan informatique. Ma gratitude va aussi à Fadila, Salim et Hanane pour leur aide appréciée et plus spécialement pour leur amitié qui a dépassé la simple relation de travail. Merci à Oxana, Annie, Wafa, Julie et Caroline pour leur patience, leur compréhension et leur amour apportés pendant toutes ces années à Montréal.

Je dédie ce mémoire à Rabi et à Mohamed ainsi qu'à toute ma famille pour m'avoir permis de poursuivre mes études en toute quiétude. Enfin, je me sens privilégié d'avoir été entouré de personnes dynamiques, brillantes et passionnées par la recherche tout au long de mon doctorat (Hafedh, Mounir, Abdel, Daniel, Aziz, Lorne, Hakim, etc). Elles m'ont inspiré le désir de me dépasser et d'en connaître toujours plus. Je souhaite enfin remercier ma famille à qui je dois énormément. Baba Lazhar, Mama, Soufiane, Moncef, Fawzi, Samir, Hatem, Ridha, Bachir et Ilhame. Bien à vous et bonne lecture.

DÉDICACE

À tous mes amis, dont je ne peux
citer tous les noms qui m'ont
encouragé jusqu'à la fin de mes études.

TABLE DES MATIERES

LISTE DES FIGURES

LISTE DES TABLEAUX

LISTE DES ACRONYMES

A-Box	Description du monde
ACI	Analyse en composantes indépendantes
ACP	Analyse en composantes principales
AFSM	Adaptive feature selection model
ALCQHIR +	Logique descriptive
API	Application Programming Interface
ART	La théorie de la résonance adaptative
ASP	Active Server Pages
AUML	Agent-based Unified Modelling Language
BDI agents	Belief Desire Intention BDI agents
CC	Le coefficient de corrélation
CHI	Le test $\chi 2$
CLARA	Clustering LARge Application
CLARANS	Clustering LARge Applications based on RANdomized Search
CLASA	Clustering LARge Application based on simulatewd Anealiting
COMMA	COrporate Memory Management through Agents
CRISP-DM	CRoss Industry Standard Process for Data Mining)
DTD	Déformation Temporelle Dynamique
DF	Document Frequency Tresholding
DIG	Description Logic Interface
DL	Description Logic
DM	Data Mining
DOM	Document Object Model
DTD	Déformation Temporelle Dynamique
DTW	Dynamic Time Warping
Doxygen	Document management system
EBD	Early Brain Damage
EBS	Early Brain Damage
Epoch	Itération
FATMAS	Agent-OrientedMethodology for Fault-Tolerant Multi-Agent Systems
FCA	Formal Concept Analysis
FLogic	Frame Logic
FTP	File Transfer Protocol
FTPS	FTP Secure
Fuzzy ART	Fuzzy Adaptative Resonance Theory
GEFS	Generic feature selection
GHI	Le Ghi carré
GI	Le gain informationnel
GML	Graph Modelling Language

GNG	Growing Neural Gas
Gold standard	Ontologie d'étallonage
GraphViz	Outil de visualisation
GRAPPA	GRAPh Package
GSBS	Generalized sequential backward selection
GSFS	Generalized sequential forward selection
GUI	Graphical User Interface
GXL	Graph eXchange Language
H	Hamming
HAC	Hierarchical Agglomeration Clustering
HC	Hill Climbing
HTTPS	Hypertext Transfer Protocol Secure
IA	Intelligence Artificielle
IIS	Internet Information Services
IM	Information mutuelle
INDUS	Intelligent Data Understanding System
IP	Internet Protocol
J	Jaccard
JAR	Java Archive
JNLP	Java Network Launch Protocol
JWS	Java Web Start
KNN	K Nearest Neighbor
KNIME	Konstanz Information Miner
LIMO	Application web (WAR) pour interagir avec un index
LOOM	Langage de représentation de la connaissance
LPO	La logique de prédicat du premier ordre
Luke	Lucene Index Toolbox
Lucene	API textuel Apache open source
MAS	Multi-Agent Systems
MaSE	Multi-agent Software Engineering
MDA	Multiple Discriminant Analysis
MOI	Mémoire d'organisation informatisée
OBD	Optimal Brain Damage
OBS	Optimal Brain Surgeon
OCD	Optimal Cell Damage
OCML	Operational Conceptual Modelling Language
OKBC	Open Knowledge Base Connectivity
OWL	Web Ontology Language
PAM	Partitionning Around Medoids
PTA	Plus l-take away r
RACER	Renamed ABox and Concept Expression Reasoner
RAM	Random Access Memory
RBC	Raisonnement à base de cas
RBF	Radial Basis Function
RDBMS	Relational DataBase Management System

RDFS	Resource Description Framework Schema
SABPO	Standards Based and Pattern Oriented
SAX	Simple Api for Xml
SBS	Sequential Backward selection
SFBS	Sequential Floating Backward Selection
SFFS	Sequential Floating Forward Selection
SFS	Sequential forward search
SFSM	Sequential floating search methods
SGBD	Système de gestion de bases de données
SGC	Système de gestion de la connaissance.
SIMC	Système d'information pour la mémoire corporative
SMA	Le système multi-agents
SMTP	Simple Mail Transfer Protocol
SODA	Societies in Open and Distributed Agent
SOM	Self Organizing Map
SOMM	Système d'organisation de la mémoire
SOTA	Self Organizing Tree Algorithm
SQL	Structured Query Language
SVD	Singular Value Decomposition
SW	Stop word
SWARM	Fourmi artificielle
T-Box	Terminologie
TCP	Transmission Control Protocol
TF-IDF	Term Frequency Inverse Document Frequency
T.L.N	Traitement du langage naturel
TS	La force du terme
URI	Uniform Resource Identifier
VSM	Vector Space Model
XGMML	eXtensible Graph Markup and Modeling Language
XML	eXtensible Markup Language
XOL	Ontology Exchange Language
3D	Trois dimensions

RÉSUMÉ

L'intégration de la connaissance dans la mémoire corporative (Ribière et Matta, 1998), (Dieng et al., 1998) fait face à l'hétérogénéité des données (Visser, Jones et al. 1997). L'utilisation de l'ontologie est une approche possible pour surmonter ce problème. Cependant, l'ontologie est une structure de donnée comme n'importe quelle structure informatique, elle est donc dynamique et évolue dans le temps à cause des conditions dynamiques résultant des changements du domaine conceptuel, les changements de conceptualisation, les changements de spécification, les changements descendants, etc (Yildiz, 2006).

Ces dernières années, plusieurs approches ont été proposées pour résoudre le problème de la maintenance des ontologies. Cependant, la précision et le rappel ne permettent pas de satisfaire les besoins des utilisateurs. De plus, ces approches ne prennent pas en compte toute l'information disponible pour prendre une décision réaliste.

Pour résoudre le problème de l'évolution de la connaissance dans les ontologies, nous proposons une approche hybride qui utilise l'apprentissage machine et un processus d'alignement qui contrôle les relations syntaxiques entre les entrées dans l'ontologie. De plus, des règles structurelles et des heuristiques sont appliquées pour améliorer le degré de similitude entre les entités ontologiques. Ce processus hybride crée des règles de correspondance qui définissent comment transformer les entrées dans l'ontologie en définissant tous les types d'associations possibles entre les entités ontologiques. L'approche d'enrichissement de l'ontologie exploite les techniques de la fouille de données, les techniques du traitement automatique du langage naturel et la recherche d'information pour améliorer la performance d'apprentissage durant la tâche d'enrichissement du domaine conceptuel.

L'évaluation des ontologies demeure un problème important et le choix d'une approche appropriée dépend des critères utilisés. Dans notre approche, nous adoptons la vérification de la cohérence décrite dans (Maziar Amirhosseini et al 2011) et (Abderrazak et al 2011).

MOTS CLÉS: Data Mining, traitement automatique du langage naturel, apprentissage machine, recherche d'information, intégration, ontologie, mémoire corporative, Web sémantique.

INTRODUCTION

Aujourd'hui, l'intégration de la connaissance est devenue un facteur de succès clef pour la gestion de la connaissance. La complexité et l'évolution des connaissances, les organismes virtuels, la documentation numérique, les réseaux sociaux et l'explosion de l'Internet exigent une gestion systématique des connaissances. Les solutions apportées sont souvent construites autour d'une mémoire corporative (CM de l'anglais: Corporate Memory ou OM: Organizational Memory) (Ribière et Matta, 1998), (Dieng et al., 1998). L'objectif principal d'une mémoire corporative est d'augmenter l'accessibilité à la connaissance corporative en fournissant un entreposage structuré. La mémoire corporative intègre des mécanismes pour faciliter l'accès, l'identification, l'acquisition, la diffusion et la réutilisation des ressources corporatives. Ainsi, pour les intégrer, nous devons résoudre le problème de l'hétérogénéité (Visser, Jones et al. 1997). L'ontologie définit bien la sémantique des données (Gruber, 1995), par conséquent, le repérage de la correspondance entre les connaissances corporatives est faisable. L'interopérabilité est considérée comme l'application principale de l'ontologie particulièrement dans les tâches d'intégration. Ainsi, l'utilisation de l'ontologie est une approche possible pour surmonter le problème de l'hétérogénéité. L'ontologie est vue comme un ensemble de définitions et de primitives de la représentation de la connaissance corporative. Elle peut expliciter la sémantique en utilisant un langage expressif tel que: RDF, RDFS, OWL, OCML, XOL, FLogic, LOOM, CLASSIQUE, OKBC, etc. L'ontologie est récemment devenue un centre d'intérêt dans plusieurs domaines de recherche car elle fournit une compréhension partagée d'un domaine d'intérêt grâce à une représentation calculable.

Il existe de nombreux outils pour éditer et créer les artefacts ontologiques dans plusieurs domaines, tels que: la gestion de la connaissance, la biologie, le clustering hiérarchique des documents, le repérage de l'information, la visualisation et la bioinformatique, etc. Dans la majorité de ces outils, l'ontologie a été traitée comme une structure statique, mais en réalité elle a une nature fortement dynamique. Les modifications de l'ontologie peuvent être initiées par les changements du domaine conceptuel ou les changements de conceptualisation ou les changements de spécification (Yildiz, 2006).

La maintenance d'une ontologie est une tâche difficile qui implique beaucoup d'efforts, en effet, c'est une tâche cruciale car les différents acteurs de la mémoire corporative peuvent avoir différentes significations au sujet des artéfacts ontologiques. Ainsi, ces dernières années le développement des ontologies a été marqué par la croissance des travaux traitant l'apprentissage. Ces travaux couvrent différents niveaux de la structure de l'ontologie. Nous trouvons les travaux sur la hiérarchie, l'interprétation, le voisinage sémantique, les signatures, l'analyse et l'extraction de données, la prise en compte d'autres

aspects de l'apprentissage comme, les instances, les règles, les axiomes, les heuristiques, les règles syntaxiques, les règles linguistiques, etc. Cependant, de nombreuses approches se sont concentrées sur des types limités de la connaissance dans l'ontologie et négligent les autres. En effet, peu d'entre elles traitent explicitement toute l'information disponible dans l'ontologie. Les approches graphiques traitent efficacement l'information structurelle mais elles ne sont pas fiables à cause de l'hétérogénéité. Les approches qui utilisent l'apprentissage machine négligent complètement l'information structurelle contenue dans l'ontologie. De ce fait, l'apprentissage axiomatique reste inexploré et elles ne sont pas appropriées aux problèmes avec une grande hétérogénéité. La précision et le rappel dans la majorité de ces approches ne satisfait les demandes des applications réelles et le choix d'une méthode d'évaluation appropriée dépend des critères utilisés pour chaque approche. En résumé, le goulot d'étranglement d'extraction des connaissances à partir de données s'est avéré être le principal obstacle pour le processus d'apprentissage des ontologies (Gomez-Perez et Manzano-Macho, 2003).

Notre objectif de recherche consiste à mettre en place un système Data Mining en ligne pour enrichir les ontologies en utilisant l'apprentissage machine, le traitement automatique du langage naturel et la recherche d'information. Ces techniques se basent sur les ressources textuelles disponibles dans la mémoire corporative.

L'apprentissage automatique ou l'apprentissage machine ou ML (de l'anglais: *Machine Learning*) (Duda, Hart et Stork, 2001) est un champ de recherche de l'intelligence artificielle, visant le développement des techniques d'apprentissage, c'est-à-dire, des techniques permettant d'acquérir automatiquement des connaissances par l'expérience en se basant sur l'analyse de données empiriques (exemple, forme, vecteur ou Pattern de l'anglais) provenant d'une base d'apprentissage. Les techniques d'apprentissage machine peuvent déduire les modèles cachées dans les données, évoluer suivant le changement de l'environnement, s'adapter à la dynamique du système et les distributions de données complexes.

Afin d'améliorer la performance d'apprentissage, il est nécessaire de supprimer le bruit dans le modèle de données. L'étape de prétraitement implique généralement la suppression des chiffres et les signes de ponctuation, le dictionnaire négatif et la troncature. Ces techniques réduisent la taille de la représentation et améliorent le temps d'exécution et la performance d'apprentissage.

Pour avoir un taux de reconnaissance élevé dans le processus d'apprentissage, il est nécessaire d'avoir un bon résumé du document textuel. Ainsi, la représentation des documents textuels joue un rôle important dans le processus de maintenance. Les documents doivent être transformés en une représentation interprétable par le processus de classification car les classificateurs ne peuvent pas les interpréter directement. Cette représentation est similaire à la forme utilisée dans le repérage de l'information(Frakes, A. et al. 2000). Chaque document est indexé par son contenu dans un modèle d'indexation et le contenu peut être pondéré par des algorithmes différents. Généralement, cette représentation génère un espace

hautement dimensionnel, là où le nombre de variables est beaucoup plus grand que le nombre de documents disponibles pour l'apprentissage, ceci pose un sérieux problème pour les approches de classification automatique. D'une part, les variables non pertinentes peuvent augmenter la taille de l'espace de recherche et le temps d'exécution nécessaire pour les algorithmes d'apprentissage. D'autre part, la plupart des approches basées sur l'apprentissage machine ont une mauvaise généralisation dû au grand nombre de variables.

La malédiction de la dimensionnalité (Rust, 1997) peut être corrigée par l'utilisation d'une technique pour la sélection des variables qui est un processus permettant de choisir les variables pertinentes parmi un grand ensemble de variables redondantes. Les variables sélectionnées améliorent la performance de la généralisation quand elles sont utilisées par le processus d'apprentissage (Jain et Zongker, 1997) et (Zhao *et al.*, 2002).

La classification automatique des documents est indispensable pour l'extraction des modèles cachés et elle est considérée comme le moteur Data Mining dans le système Data Mining de maintenance de l'ontologie. Elle permet d'acquérir les distributions des documents textuels et les relations de classification et de covariance des données dans la base d'apprentissage.

Tous les modèles cachés sont décrits par des mots clés représentatifs résultant de leurs contenus pour faciliter la gestion des documents (recherche & récupération, diffusion, compréhension & identification, etc.), et pour identifier les changements candidats dans le processus d'apprentissage.

Étant donné la tâche d'extraction des modèles cachés à partir des documents textuels disponibles dans la mémoire corporative, les règles de correspondance ou les similitudes entre les modèles cachés et les artéfacts ontologiques de base sont des propriétés qui améliorent considérablement la description et, de ce fait, enrichissent l'ontologie. Par conséquent, l'approche de l'enrichissement utilise également un processus d'alignement (Alexandru-Lucian et Iftene, 2010) basé sur la similitude pour achever l'interopérabilité entre les deux représentations.

L'importance d'une syntaxe et d'une sémantique bien définies sont deux conditions nécessaires pour assurer la complétude et la correction d'un système formel. En se basant sur une syntaxe et une sémantique claires nous pouvons implémenter le code de l'ontologie avec une représentation adéquate pour appuyer l'inférence. Ainsi, le modèle de représentation de l'ontologie devrait soutenir l'expressivité au maximum tout en maintenant la complétude pour assurer la décidabilité (Baader, 2003).

La cohérence de l'ontologie devrait être maintenue après l'application des changements. Cela permet de s'assurer que le raisonnement conceptuel est encore possible. Puisque l'extraction automatique demeure une question de recherche, nous considérons le processus d'apprentissage comme un processus semi-automatique.

En résumé, la tâche d'apprentissage de l'ontologie consiste à enrichir la terminologie par des artéfacts ontologiques extraits à partir d'un corpus et d'arranger ces termes taxonomiquement.

a) Contexte de la thèse

Tout au long de ce travail, notre préoccupation sera comme mentionné précédemment, de relier les différentes représentations pour piloter l'évolution de l'ontologie dans un espace dynamique. Ce sujet permet d'aborder différents domaines scientifiques, en particulier, le Data Mining, l'apprentissage machine, le traitement automatique du langage naturel ou (NLP de l'anglais: Natural Language Processing) et la recherche d'information ou le repérage de l'information (IR de l'anglais: Information Retrieval).

Le Data Mining est le processus permettant de découvrir les modèles cachées dans une base d'apprentissage en utilisant plusieurs techniques, en particulier, les réseaux connexionnistes, les algorithmes génétiques, les arbres de décision, les graphes probabilistes, les fourmis artificielles ou colonie de fourmis, l'optimisation combinatoire, etc (Berzal et Matín, 2002).

Dans un contexte d'entreposage des données, le Data Mining regroupe l'ensemble des techniques susceptibles d'extraire les modèles cachés à partir d'une grande quantité de données bruitées. C'est une discipline d'importance croissante car elle peut fournir un avantage concurrentiel significatif en exploitant le potentiel des entrepôts de données ou le contenu des mémoires corporatives.

Beaucoup de chercheurs considèrent le Data Mining comme le processus d'acquisition de connaissances (KA de l'anglais: Knowledge Acquisition). Néanmoins, les spécialistes en analyse multidimensionnelle en ligne (OLAP: On-Line Analytical Processing) (CHRYSAFIS, 2003) appréhendent le Data Mining comme une étape essentielle dans le processus d'acquisition de connaissances. Il faut noter que selon cette conception, le moteur Data Mining de notre approche est seulement une étape dans l'ensemble du processus de la maintenance de l'ontologie, c'est une étape très importante car elle découvre les modèles cachés. Toutefois, dans les systèmes d'information décisionnels ou SID (Decision Making Information System), le terme Data Mining est devenu un terme plus populaire que le terme Acquisition de Connaissances. De ce fait, au cours de cette thèse, nous choisissons d'utiliser le terme Data Mining.

Les techniques de Data Mining basées sur l'apprentissage machine imposent peu de restrictions et produisent des modèles faciles à comprendre, elles ont donc acquis une large popularité dans les applications Data Mining. Le but de l'apprentissage machine est de créer une entité de classification automatique qui peut apprendre en acquérant la connaissance pertinente, qui peut reconnaitre les formes en traitant les faits recueillis et finalement qui peut inférer de nouvelles connaissances ou prendre des décisions.

Comme une partie de l'I.A, l'apprentissage machine est l'un des champs principaux de recherche qui implique les concepts mathématiques, la logique et les probabilités pour formaliser l'apprentissage. Cette technique a été utilisée dans plusieurs champs de recherche, en particulier, le Data Mining, la reconnaissance des formes, l'optimisation combinatoire, l'analyse génétique, la robotique, le repérage d'information, la sécurité informatique, etc.

Le traitement automatique du langage naturel (parfois nommé ingénierie linguistique) est un ensemble de techniques artificielles visant à traiter de façon significative le texte d'un discours décrit dans le langage naturel (Pereira et Gross, 1994).

En outre, la recherche d'information (Frakes, A. et al. 2000) est la technique de repérage de l'information pertinente d'une collection de ressources d'information en utilisant les structures des données, les modèles d'indexation et les algorithmes de repérage, etc.

En résumé, notre vision de l'apprentissage de l'ontologie vise l'intégration d'une multitude de disciplines pour faciliter l'extraction des artéfacts ontologiques, en particulier, le Data Mining, l'apprentissage machine, le traitement automatique du langage naturel et la recherche d'information. Cette alternative de recherche nous semble prometteuse car elle étudie l'application des techniques de l'apprentissage machine au cas de maintenance des ontologies.

b) Plan de la thèse

Nous commencerons par un survol de la littérature concernant les notions relatives à l'ontologie et les défis auxquels les chercheurs ont eu à faire face dans le domaine de l'évolution de l'ontologie. Puis, nous effectuerons une analyse sur la maintenance de l'ontologie, ce qui nous amènera à présenter nos questions de recherche. Nous présenterons ensuite la méthodologie utilisée ainsi que le système Data Mining proposé.

Nous détaillerons l'intégration des connaissances dans la mémoire corporative et les mesures utilisées pour évaluer les performances d'intégration. Finalement, nous décrirons les résultats de la recherche qui reposent sur une analyse des données recueillies par l'intermédiaire de la chaîne Data Mining contenant les outils de prétraitement de données, la sélection des variables, le clustering, la sélection des modèles, l'alignement, la mise à jour de l'ontologie, le modèle calculable utilisé pour représenter l'ontologie enrichie, les techniques de raisonnement utilisées pour vérifier la cohérence des connaissances et l'outil de visualisation utilisé pour explorer et comprendre la structure ontologique. Nous tenterons également d'établir des conclusions sur la question de la maintenance des ontologies.

Le présent document est composé de huit chapitres. Dans le premier chapitre, nous présentons et nous discutons les différentes définitions d'une ontologie et les différentes formes qu'elle peut prendre. Nous y

présentons également les différentes techniques d'apprentissage machine utilisées pour enrichir les ontologies, les problématiques de l'évolution de l'ontologie, les objectifs et les hypothèses de recherche. Le deuxième chapitre présente la méthodologie suivie pour la maintenance de la connaissance dans l'ontologie. Il se concentre sur le prétraitement et le nettoyage du corpus textuel, la sélection des termes, la méthode et l'architecture du clustering et le processus d'enrichissement à partir d'un mécanisme d'alignement. Le troisième chapitre se concentre sur l'intégration de la connaissance dans la mémoire corporative; l'objectif de ce dernier est de passer en revue les bases conceptuelles de l'intégration et d'expliquer les processus d'intégration comprenant l'acquisition, l'indexation et la récupération de la connaissance. Ce chapitre constitue le volet cognitif de la thèse et nous y présentons le système d'intégration de la connaissance en utilisant l'ontologie corporative, l'analyse du texte, les propriétés particulières des éléments d'indexation, le modèle et la structure d'indexation, le repérage et les mesures utilisées pour évaluer la performance. Nous donnons également dans ce volet une brève description de l'ontologie utilisée dans le cadre de notre projet de recherche.

Afin de présenter les principaux concepts permettant d'étudier la représentation de la connaissance, nous allons nous référer dans ce chapitre introductif à un scénario certes limité, mais servant de cadre idéal à la majorité des recherches actuelles sur la représentation de la connaissance: celui de l'intégration de la connaissance. Il faudra bien sûr l'améliorer voir le modifier à l'occasion, mais il fournit un bon point de départ.

Le volet exploratoire montre l'évaluation du modèle proposé: Dans le quatrième chapitre, nous allons évaluer les différents modules implémentés dans la chaîne Data Mining, en particulier, le prétraitement, la sélection des variables, la réduction de l'espace représentation. Nous allons détailler dans ce chapitre la distribution et le niveau de bruit dans le corpus d'apprentissage, le processus de la sélection des variables, les stratégies de recherche, la manière de traiter la malédiction de la dimensionnalité et le critère d'évaluation choisi pour guider le processus de recherche dans la tâche de sélection des variables pertinentes.

Le cinquième chapitre porte sur l'analyse et le développement d'un modèle artificiel de réseau de neurones basé sur les principes de la théorie de la résonance adaptative floue pour les tâches de clustering, la configuration de l'architecture neuronal, l'initialisation typique, l'estimation l'erreur de généralisation, la sélection et l'évaluation du modèle connexionniste.

Dans le sixième chapitre, nous allons décrire un processus d'alignement avec agrégation et les mesures corrélatives utilisées pour calculer la corrélation entre les différents processus individuels d'alignement.

Le septième chapitre est consacré à l'étape de mise à jour pour enrichir le modèle conceptuel, le langage pour décrire le modèle calculable, l'étape de la vérification de la consistance terminologique utilisant un système de raisonnement basé sur des structures symboliques de faits et de règles, la méthode de

visualisation, l'outil pour adresser la documentation des définitions des artéfacts ontologiques, l'architecture logicielle, l'implémentation et le protocole utilisé pour déployer le système. Les cas d'utilisation et les diagrammes de séquence sont utilisés pour décrire le comportement de chaque module réalisé.

Nous concluons enfin par l'énumération de l'originalité et de la pertinence de notre approche, le bilan des travaux, la contribution de ces travaux dans d'autres domaines de recherche et les nombreuses perspectives qui en découlent. Les enjeux du paradigme de la gestion distribuée de la connaissance et la méta modélisation pour la maintenance des ontologies sont discutés et les perspectives méthodologiques sont présentées.

CHAPITRE I

PROBLEMATIQUE ET OBJECTIFS DE RECHERCHE

Résumé: Ce présent chapitre présente une brève introduction à l'ontologie afin de se familiariser avec ce concept. Puis, une présentation des avancées de l'intégration des connaissances corporatives basée sur les ontologies permet de se situer dans ce contexte spécifique. L'emphase est ensuite mise sur les changements et les différentes phases du processus d'évolution des ontologies. Ensuite, la troisième partie, la revue de littérature, recense les différentes recherches et les thèmes intervenant dans le projet. Plus précisément, trois axes d'investigation sont traités: l'apprentissage machine, le traitement automatique du langage naturel et enfin la recherche d'information ou le repérage de l'information. Notamment des approches d'évaluation de la performance sont présentées. Ceci permet de présenter dans la quatrième partie le contexte et la problématique de recherche, ainsi que les hypothèses de recherche, qui s'apparentent à une combinaison des cadres théoriques présentés préalablement.

Dans la suite de ce chapitre, nous donnons dans le paragraphe (1.1) une idée générale sur l'ontologie. Nous présentons et discutons les différentes définitions d'une ontologie et les différentes formes qu'elle peut prendre dans le paragraphe (1.2). Nous discutons l'utilisation des ontologies dans la tâche d'intégration de la connaissance corporative dans le paragraphe (1.3). Ensuite, nous présentons les techniques d'apprentissage machine utilisées pour enrichir les ontologies dans le paragraphe (1.4). Afin de traiter l'évolution des ontologies dans la mémoire corporative, nous allons montrer dans le paragraphe (1.5) que les modèles actuels qui traitent de l'évolution de l'ontologie ont plusieurs faiblesses. Nous verrons ensuite dans ce paragraphe les travaux effectués et les défis auxquels les chercheurs ont eu à faire face dans le domaine de l'évolution de l'ontologie. Le paragraphe (1.6) présente les objectifs et les hypothèses de recherche.

1.1 Définition d'une ontologie

Le partage de compréhension de la structure d'information est l'un des buts visés par le Web sémantique. Le Web sémantique est un modèle de référence qui permet de partager et réutiliser les ressources Web entre les partenaires d'interaction. Pour Tim Berners-Lee (Berners-Lee, 1999), le web sémantique est avant tout une question de contenu et son but se comprend comme:

> « *I have a dream for the Web [in which computers] become capable of analyzing all the data on the Web — the content, links, and transactions between people and computers. A "Semantic Web", which should make this possible, has yet to emerge, but when it does, the day-to-day mechanisms of trade, bureaucracy and our daily lives will be handled by machines talking to machines. The "intelligent agents" people have touted for ages will finally materialize*» (Berners-Lee,1999).

Selon cette perspective, toute discussion sur le Web sémantique nous amène à questionner l'analyse des ressources Web (contenu, lien et interopérabilité, ..). En fait, nous pourrions aborder le Web sémantique sous différents angles mais une chose semble faire l'unanimité, c'est que toute interprétation du contenu implique une description à l'aide d'annotations.
La définition suivante été adopté par la communauté du Web sémantique:

> « *The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation* ».
>
> Tim Berners-Lee, James Hendler, Ora Lassila, The Semantic Web, Scientific American, May 2001

Cette définition est la plus proche de la réalité puisqu'elle regroupe à la fois les processus de signification et de computationalité pour des fins d'interopérabilité.
Comme le montre la figure 1.1, le Web sémantique (plus techniquement appelé le Web de données ou Web 3.0) est décrit comme une plateforme multicouche, où l'ontologie se trouve au milieu de cette plateforme. (Berners-Lee, Hendler et Lassila., 2001). Selon (Stojanovic, Mentzas et Apostolou, 2006) et (Yildiz, 2006), l'ontologie se révèle être la meilleure solution pour la communication et le partage des connaissances. Elle fournit un niveau plus profond de la sémantique en offrant des métadonnées significatives et des engagements ontologiques pour le partage de la connaissance entre les partenaires d'interaction. Le but-sous jacent est de rendre explicite la sémantique des ressources Web au travers de métadonnées ou d'annotations. L'ontologie répond donc à un réel besoin en matière d'accès à la signification de l'information sur le Web.

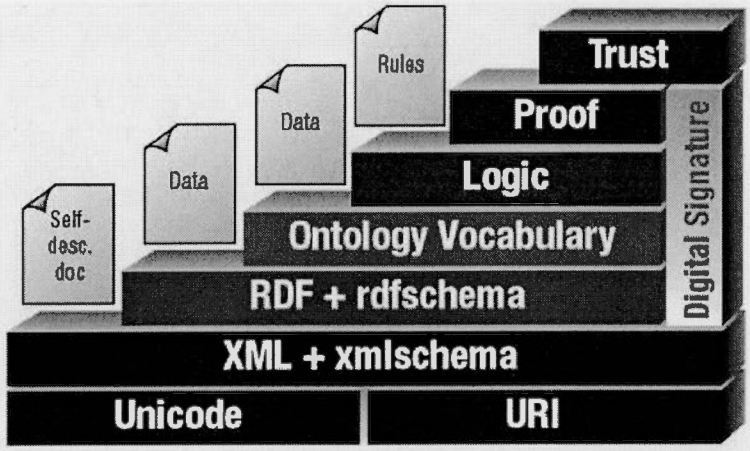


Figure 1.1 Architecture du Web sémantique (Berners-Lee, Hendler et Lassila., 2001)

La littérature contient de nombreuses définitions d'une ontologie. Toutefois, la plus connue et la plus citée est celle de (Gruber, 1995) *p.1*, qui est également la définition que nous adoptons dans notre thèse: *« Une ontologie est une spécification explicite d'une conceptualisation ».*

Le terme conceptualisation se rapporte à une représentation abstraite d'un domaine d'intérêt particulier (Gruber, 1995).

Les deux définitions initiales proposée par (Gruber, 1995) et (Borst, 1997) sont légèrement modifiées dans l'article (Jiehan et Dieng-Kuntz, 2004) p.40 comme suit:

« Une spécification formelle et explicite d'une conceptualisation partagée ».

Plusieurs concepts émergent à partir de cette définition:

- *Spécification*: décrire le domaine par des artéfacts ontologiques (concepts, instances, propriétés). La spécification permet d'identifier l'ensemble des termes à représenter, leurs caractéristiques et leurs granularités. Elle devrait être complète et concise.
- *Formelle*: des définitions rigoureuses grâce à un système formel par rapport à une certaine spécification.
- *Explicite*: les artefacts ontologiques explicitent le partage et la réutilisation des connaissances entre les partenaires d'interaction.
- *Conceptualisation*: se rapporte à un modèle abstrait d'un certain domaine d'intérêt.
- *Partagée*: la connaissance consensuelle entre les partenaires d'interaction (Jiehan et Dieng-Kuntz, 2004).

1.2 Typage ontologique

Les ontologies consistent en un ensemble de concepts et une interprétation d'un domaine d'intérêt particulier. Les caractéristiques d'une ontologie peuvent être regroupées comme suit:

1.2.1 Formalisation

Les ontologies peuvent être classées dans quatre groupes, en fonction de leur degré de formalité:

- Hautement informelle: représentée dans le langage naturel.
- Semi informelle: une représentation structurée en réduisant l'ambiguïté.
- Semi formelle: la représentation des concepts (instances) et les relations dans un système formel.
- Rigoureusement formelle: la définition des artéfacts ontologiques avec des interprétations formelles, des théories, des faits, des axiomes, des preuves de complétude, etc. Cette représentation peut aussi inclure des informations sur les propriétés d'une classe, les restrictions des valeurs, des contraintes logiques entre les concepts et les relations, des propriétés plus détaillées telles que les propriétés symétriques, les propriétés quantifiées, la description des instances, les descriptions par intersection/union/complémentaire/héritage, etc. (Uschold et Gruninger, 1996) p.6.

1.2.2 Généralisation

Selon (Guarino, 1998) p.6, les artéfacts ontologiques peuvent contenir des informations avec différents niveaux de détails. Ainsi il a distingué deux types d'ontologies:

- Les ontologies grossières (de l'anglais coarse shareable ontologies): les artéfacts ontologiques sont représentés dans un système formel avec une axiomatisation minimale pour réduire l'ambigüité.
- Les ontologies de référence: des ontologies détaillées spécifiant une sémantique formelle.

1.2.3 Conceptualisation

Comme montré à la figure 1.2, les ontologies peuvent être classées selon la conceptualisation (Guarino, 1998) p.7-8.

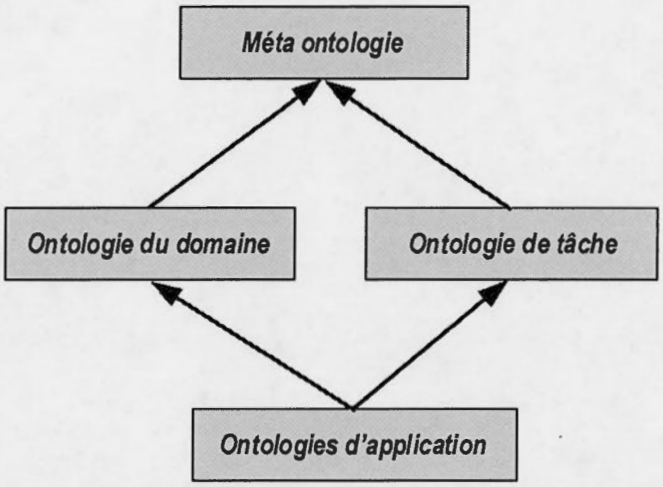


Figure 1.2 Typage ontologique (Guarino, 1998) p.7

- Les métas ontologies: des ontologies générales qui capturent les primitifs de représentation de la connaissance.
- Les ontologies générales ou de niveau supérieur: des ontologies spécifiant des connaissances abstraites indépendantes d'un détail dans le domaine du problème.
- Les ontologies du domaine: des ontologies spécifiques à un domaine particulier.
- Les ontologies d'application: relient les concepts du domaine avec les concepts procéduraux.

1.3 Intégration corporative basée sur l'ontologie

Les ontologies deviennent de plus en plus répandues dans la communauté d'entreposage des données. Elles ont maintenant un rôle spécifique dans l'IA, l'informatique linguistique, les théories des bases de données, l'ingénierie de la connaissance, la représentation de la connaissance, la conception des bases de données, l'intégration de l'information, l'analyse orientée objet, l'extraction et le repérage de l'information, l'organisation et l'intégration des entreprises, la conception des SMA, etc.

Les ontologies sont utilisées dans de nombreux types de systèmes d'intégration corporative: CoMMa (Corporate Memory Management through Agents) (Gandon *et al.*, 2002), MOMIS (Mediator EnvirOnment for Multiple Information Sources) (Beneventano *et al.*, 2003 ; Xing *et al.*, 2008), STATIS (Beneventano *et al.*, 2008), SAMOA (Moser *et al.*, 2009), etc.

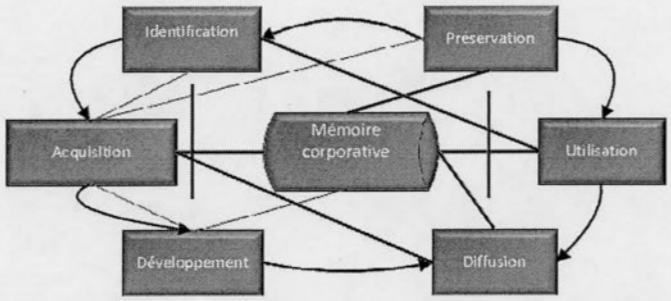


Figure 1.3 Gestion de la connaissance corporative (Abecker *et al.*, 1998), p.41

Dans ce contexte, la mémoire corporative définit l'information stockée dans l'historique de l'entreprise permettant de contrôler les activités de base de la gestion de la connaissance corporative (acquisition, indexation, repérage, diffusion, utilisation et préservation), qui peut être appliquée sur les décisions actuelles (figure 1.3) (Walsh et Ungson, 1991).p.61.

Selon (Guarino, 1998), leurs utilisations peuvent être bénéfiques au cours du développement et pendant l'exécution des systèmes d'information. Elles fournissent un jargon pour apprendre le lexique de la mémoire corporative (structures, processus, système d'information, artéfacts, ressources, etc.). Ainsi, l'ontologie est un outil conceptuel représentant la sémantique dans un système d'intégration.

Dans ce contexte, il existe trois approches qui utilisent les ontologies dans la tâche d'intégration de la connaissance corporative.

i. L'approche basée sur une ontologie unique: comme montré à la figure 1.4, cette approche utilise une ontologie globale pour spécifier la sémantique. Cette approche est susceptible aux changements qui peuvent affecter la conceptualisation du domaine représenté dans l'ontologie. L'approche CoMMa (Gandon *et al.*, 2002) utilise ce genre d'intégration. Son modèle inclut une base de connaissance terminologique hiérarchique avec des nœuds représentant les entités ontologiques.

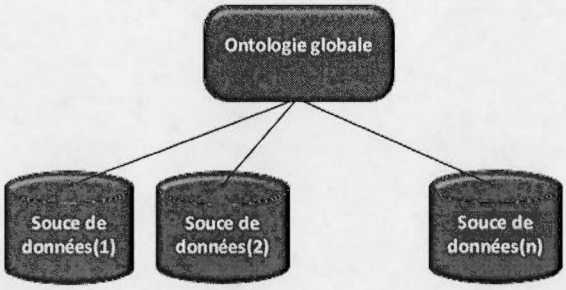


Figure 1.4 Intégration basée sur une ontologie unique (Wache *et al.*, 2001), p.2

ii. L'approche basée sur plusieurs ontologies: dans cette technique d'intégration, chaque source de la connaissance est décrite par sa propre ontologie. Un mécanisme d'alignement définissant la correspondance entre les ontologies est nécessaire pour comparer les différentes ontologies.

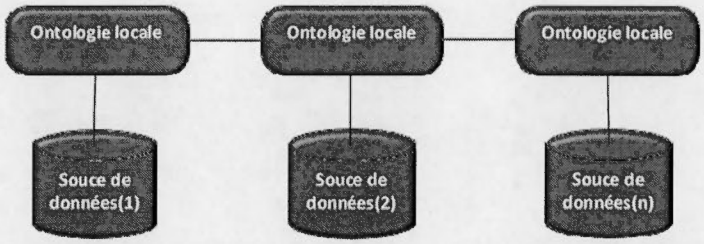


Figure 1.5 Intégration basée sur plusieurs ontologies (Wache *et al.*, 2001), p.2

iii. L'approche hybride: elle est développée pour pallier à certaines limitations des approches basées sur une ontologie unique et les approches basées sur plusieurs ontologies. La conception de l'architecture n'est pas dépendante des changements qui peuvent affecter la conceptualisation du domaine représentée dans l'ontologie. C'est parce que la sémantique de chaque source est décrite par sa propre ontologie. Le vocabulaire partagé est utilisé pour comparer les ontologies sources. C'est pourquoi cette approche ne fixe au préalable aucune forme d'alignement à la relation exprimant le vocabulaire partagé en fonction des ontologies locales. De plus, elle modélise les termes complexes en recourant aux techniques de l'intelligence artificielle telle que la logique combinatoire (Wache *et al.*, 2001).

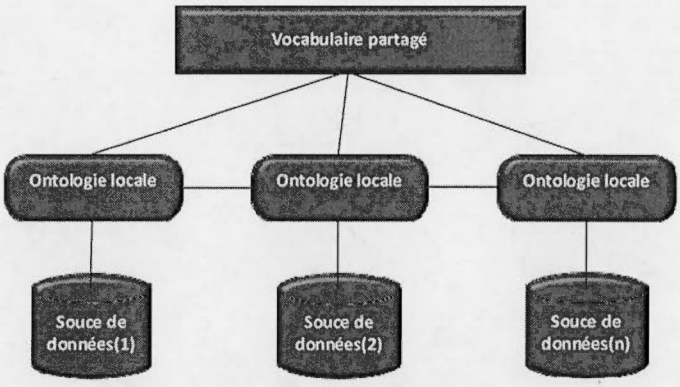


Figure 1.6 Intégration hybride (Wache *et al.*, 2001), p.2

Les systèmes d'intégration utilisent l'ontologie pour accéder à la signification de l'information. De ce fait, l'ontologie facilite l'intégration automatique des connaissances corporatives. Nous pouvons énumérer quelques avantages de l'utilisation de l'ontologie dans la mémoire corporative:

- Permettre la réutilisation de la connaissance corporative: l'ontologie forme une base solide pour résoudre le problème de l'hétérogénéité de données. De ce fait, elle facilite la réutilisation des connaissances corporatives.
- Rendre les axiomes du domaine explicites: la spécification déclarative de l'ontologie permet d'expliciter la sémantique cachée des axiomes déclarés.
- Analyser la connaissance du domaine: la spécification déclarative permet d'analyser la connaissance corporative en utilisant plusieurs mécanismes d'inférence, notamment, la consistance conceptuelle, la satisfiabilité, la subsomption et l'instanciation.
- La compréhension: les artéfacts ontologiques et la structure taxonomique permettent de comprendre la conceptualisation de la mémoire corporative.
- La communication: l'ontologie assure une communication non ambiguë entre les partenaires d'interaction.
- L'interopérabilité: l'ontologie assure la correspondance entre le modèle du domaine et les connaissances corporatives. De ce fait, elle facilite l'interopérabilité entre les différents schémas des représentations (Stojanovic, Mentzas et Apostolou, 2006) et (Yildiz, 2006).

Dans le prochain paragraphe, nous décrivons tout d'abord une présentation exhaustive de l'état de l'art sur les techniques d'apprentissage machine. Ensuite, la mise en contexte, les problématiques et les défis à chaque phase du processus d'extraction de connaissances sont discutés. Dans le dernier paragraphe, le processus d'apprentissage, les objectifs et les hypothèses de recherche sont présentés.

1.4 L'état de l'art

1.4.1 Processus d'évolution des ontologies

L'ontologie est une structure de donnée comme n'importe quelle structure informatique, elle est donc dynamique et évolue dans le temps à cause des conditions dynamiques résultant des changements du domaine conceptuel, les changements de conceptualisation, les changements de spécification (Yildiz, 2006). Nous adoptons ici la définition de l'évolution de l'ontologie suivante:

«Ontology Evolution is the timely adaptation of ontology to the arisen changes and the consistent propagation of these changes to dependent artifacts. » Source (Haase et Sure, 2004) p.6.

Comme montré à la figure (1.7), l'évolution de l'ontologie se compose de plusieurs phases qui représentent des activités dynamiques:

- La capture du changement: l'identification des changements en utilisant les méthodes ascendantes liées à l'extraction des connaissances à partir de l'ensemble de données ou les méthodes descendants représentant le besoin de l'utilisateur.
- La représentation: dans cette phase, le niveau approprié de granularité devrait être identifié pour représenter formellement et explicitement les changements.
- Les sémantiques de changement: cette étape devrait contrôler les effets possibles d'un changement et les problèmes qui pourraient provoquer l'inconsistance.
- La propagation des changements: tous les changements induits seront propagés aux parties concernées pour assurer la consistance et pour préserver les rapports concrets entre les entités de l'ontologie et l'ensemble de données (système d'information, agents, ontologies, processus, etc.). Cette étape devrait satisfaire les besoins des changements ascendants et descendants et en même temps la cohérence.
- L'implémentation: met-en-œuvre les changements identifiés.
- La validation: contrôle la complétude du domaine du problème et elle permet et d'annuler les changements inutiles qui provoque l'inconsistance ou la redondance (Klein, 2004) et (Yildiz, 2006).

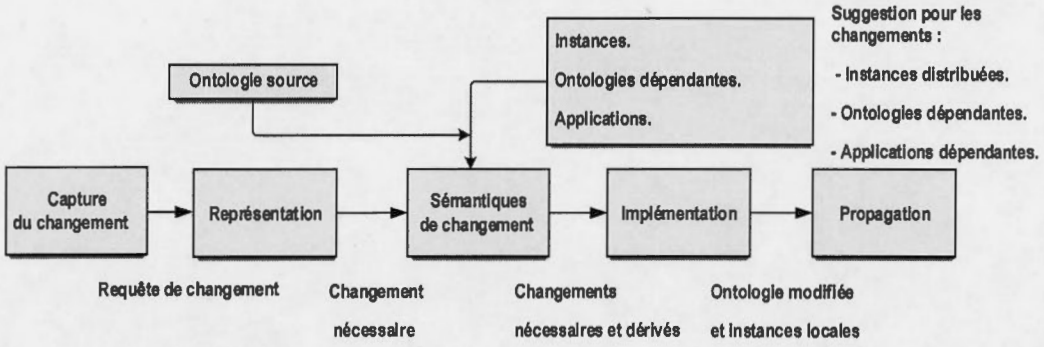


Figure 1.7 Processus d'évolution des ontologies (Stojanovic, Maedche et al. 2002) p.4

L'évolution de l'ontologie est le domaine qui étudie la manière de maintenir une ontologie conforme à un ensemble de conditions de consistance suite à un ensemble de changements. Il s'agit d'une technique permettant d'adapter les changements afin d'assurer la consistance de l'ontologie et la propagation des changements aux artéfacts ontologiques dépendants. Les changements peuvent être regroupés comme suit:

- Selon le type: comme montré à la figure (1.8), on peut distinguer trois types de changements. Premièrement, les changements conceptuels qui peuvent changer la conceptualisation et être exécutés sur les artéfacts de l'ontologie. Deuxièmement, les changements liés aux spécifications de la conceptualisation. Finalement, les changements dans la représentation de la conceptualisation.

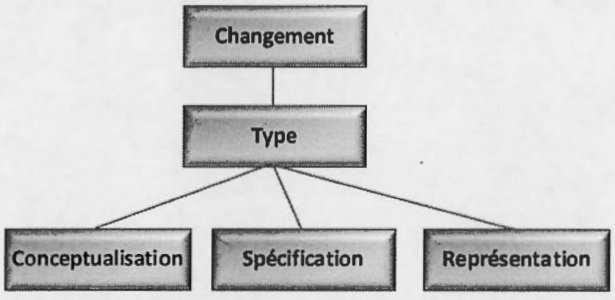


Figure 1.8 Catégorisation des changements selon de type

- Selon la granularité: nous pouvons distinguer les changements élémentaires et les changements complexes (figure 1.9) (Yildiz, 2006).

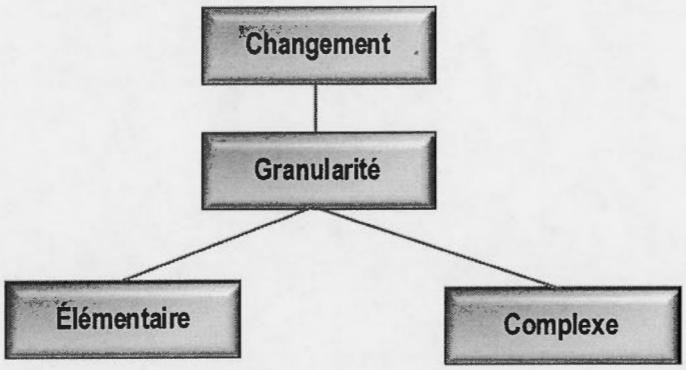


Figure 1.9 Catégorisation des changements selon la granularité

1.4.2 Apprentissage des ontologies

L'évolution de l'ontologie s'est avérée une tâche cruciale qui adresse la gestion des changements d'une façon systématique sans aucune perte de la connaissance existante en vérifiant et en maintenant la consistance. Dans ce contexte, plusieurs approches d'apprentissage ont été proposées pour assister l'utilisateur dans le processus de maintenance de l'ontologie. La majorité de ces approches est une combinaison des techniques du langage naturel, des techniques du Data Mining et de la recherche d'information. Parmi ces approches on retrouve:

(Faatz et Steinmetz, 2002) ont proposé une approche pour enrichir les ontologies en utilisant le texte recherché de l'internet. Le processus d'apprentissage est basé sur la comparaison entre la fréquence d'occurrence des termes dans un corpus et la structure de l'ontologie. Pour cela, cette approche propose trois étapes pour enrichir l'ontologie: le repérage des documents Web pour former le corpus, l'analyse du corpus pour identifier les cooccurrences des termes représentants les concepts candidats et l'évaluation de pertinence des concepts par un expert du domaine.

De leur côté, (Agirre et al., 2000) ont proposé une approche semi-automatique qui vise à enrichir l'ontologie en utilisant des requêtes terminologiques. Pour se faire, cette approche propose quatre étapes pour enrichir l'ontologie:

- Le repérage: la récupération des documents pertinents à partir des requêtes formées en utilisant les artéfacts ontologiques.
- La création des signatures des thèmes: l'analyse statistique de la fréquence des termes dans chaque document pour former les signatures.

- Le clustering descriptif: un processus de clustering est appliqué pour générer des collections suivant les significations des termes.
- L'évaluation: cette étape utilise le corpus d'étalonnage SemCor pour évaluer la qualité de l'ontologie enrichie.

Une autre approche intéressante pour enrichir l'ontologie a été *OntoLearn* (Missikoff, Navigli et Velardi, 2002) dans laquelle l'enrichissement est basé sur les techniques de traitement automatique du langage naturel et les techniques d'apprentissage machine. Cette approche utilise aussi le thesaurus Wordnet comme une source de base pour construire le noyau de l'ontologie. Le processus complet est composé de quatre étapes:

- *L'extraction de la terminologie*: les termes et les combinaisons des termes sont extraits à partir d'un corpus analysé en utilisant les techniques du traitement automatique du langage naturel.
- *L'interprétation sémantique*: cette étape permet d'identifier les relations sémantiques entre les concepts afin de créer des concepts définis en utilisant le voisinage sémantique WordNet , à savoir, synonymie, hypéronymie, méronymie, antonymie, etc.
- *La création de l'ontologie du domaine*: l'objectif de cette étape est d'intégrer la taxonomie obtenue avec le noyau de l'ontologie en utilisant Wordnet pour l'élagage et l'extension des concepts qui ne sont pas liés au vocabulaire du domaine.
- *L'évaluation:* la qualité de l'apprentissage est jugée par un expert du domaine.

(Maedche et Staab, 2001) de leur côté, ont proposé une classification de plusieurs approches d'apprentissage qui utilisent le texte structuré, le texte semi-structuré et le texte non structuré (Figure 1.10).

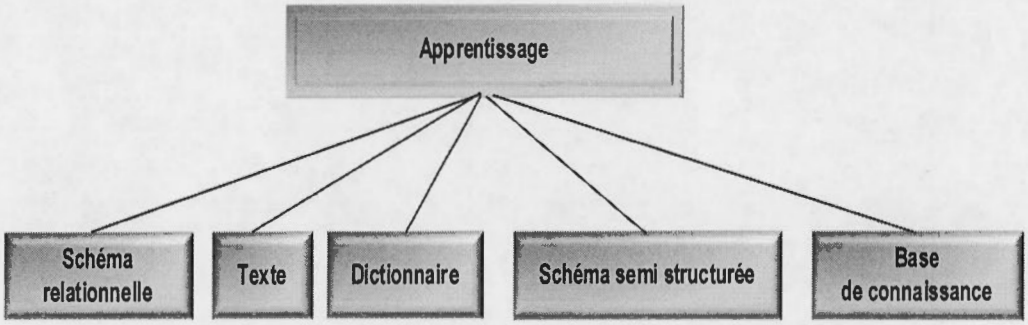


Figure 1.10 Classification des différentes approches pour l'apprentissage des ontologies

Les approches d'apprentissage des ontologies à partir du texte les plus connues se concentrent sur l'extraction des ontologies en appliquant les techniques de traitement automatique du langage naturel et les techniques d'apprentissage machine. Parmi ces approches on retrouve:

a) *L'élagage de l'ont*ologie: dans l'approche proposée par (Kietz, Maedche et Volz, 2000), le système d'apprentissage est composé de plusieurs composantes: la gestion du traitement textuel, le serveur de traitement textuel SMES, les algorithmes d'apprentissage et d'élagage et le système d'ingénierie des ontologies OntoEdit. Cette technique permet de créer le domaine de l'ontologie à partir de plusieurs sources hétérogènes en utilisant un processus de plusieurs étapes:
   - La création: un noyau générique de l'ontologie est utilisé pour représenter un domaine particulier.
   - L'encodage: le noyau générique est traduit dans un formalisme de représentation en utilisant le système d'ingénierie OntoEdit.
   - L'acquisition: un dictionnaire contenant 1500 termes est utilisé pour acquérir les concepts du domaine.
   - Le raffinement: les concepts qui ne sont pas liés au domaine de l'application sont supprimés en utilisant un noyau générique et un corpus d'apprentissage composé de 1000 documents intranet.

   Cette méthode peut construire rapidement l'ontologie mais la précision et le rappel ne sont pas bons.

b) *Le groupement conceptuel:* la méthode (Faure et Poibeau, 2000) exploite un processus hybride composé d'un système d'extraction INTEX avec un système d'apprentissage machine ASIUM. Ce dernier est basé sur une méthode de classification non supervisée pour le groupement conceptuel. Les concepts sont groupés selon la distance sémantique pour composer la hiérarchie conceptuelle. Mais cette approche ne prend pas en compte l'aspect contextuel pendant le calcul de la distance. Par conséquent, le processus de groupement conceptuel ne peut pas être contrôlé efficacement. De plus, avec cette méthode, seules les relations taxonomiques des concepts qui peuvent être générées.

c) L'analyse formelle de concepts (FCA: Formal Concept Analysis): dans l'approche (Cimiano, Hotho et Staab, 2005) utilisant l'analyse formelle de concepts, les concepts du domaine et leurs attributs peuvent être obtenus pour former le contexte formel et pour construire les treillis conceptuels. Après avoir ajouté les relations non taxonomiques, l'ontologie peut être formée. La hiérarchie est comparée avec deux hiérarchies conceptuelles issues d'une taxonomie manuelle pour deux

domaines: le tourisme et la finance. Cependant, le treillis conceptuel est une structure de donnée complexe et quand le contexte formel est grand, la construction des treillis de concepts est non triviale.

d) L'analyse conceptuelle relationnelle (RCA: Relational Concept Analysis): est une méthode basée sur le paradigme de l'analyse de données. Son but est d'inférer les relations entre les concepts formels basés sur les liens entre les objets formels. L'approche propose plusieurs étapes dans le processus d'apprentissage: l'analyse des textes pour l'extraction des termes, la génération de la famille des treillis (l'ontologie noyau) et le raffinement. L'étape d'évaluation utilise un corpus d'apprentissage composé d'un ensemble de résumés d'articles tirés d'un journal d'astronomie (Astronomy and Astrophysics, 2002-2004). Cependant, l'apprentissage dépend de la précision des outils d'extraction et la malédiction de la dimensionnalité de l'espace de représentation (Hacene *et al.*, 2008).

e) *Les règles d'association:* le mécanisme d'apprentissage proposé par (Maedche et Staab, 2000) est basé sur l'algorithme de découverte de règles d'association généralisées proposées par Srikant et Agrawal. Les règles d'association sont utilisées dans le processus d'apprentissage pour découvrir les relations conceptuelles dans une base d'apprentissage contenant 2234 documents HTML composés de 16 million de termes. L'apprentissage permet de découvrir les relations non taxonomiques entre les concepts en utilisant une hiérarchie conceptuelle comme connaissance de base. De ce fait, cette approche fournit seulement un soutien pour aider à générer l'ontologie.

f) *Extraction basée sur les patrons (de l'anglais: pattern extraction):* le processus OntoCase (Blomqvist, 2007) est basé sur la découverte des relations entre la séquence des mots dans le texte et le patron d'extraction. Le processus d'apprentissage est composé de plusieurs étapes: le repérage des cas pour la sélection des patrons appropriés en analysant le corpus, la réutilisation des patrons repérés, la construction de l'ontologie noyau, le raffinement de l'ontologie pour améliorer la qualité et la découverte de nouveaux patrons. Cependant, la conception des patrons nécessite l'intervention des experts en matière du domaine. De plus, la modification du patron bruite les données.

g) *L'apprentissage conceptuel*: est une approche où les nouveaux concepts acquis du corpus textuel permettent de modifier la taxonomie incrémentalement. Le processus d'apprentissage décrit dans l'approche (Hahn et Schulz, 2000) est composé de quatre étapes principales:

- L'extraction: les définitions des concepts sont générées automatiquement à partir de la source UMLS.

- La classification terminologique: le contrôle d'intégrité des hiérarchies taxonomiques est effectué par une classification terminologique.
- La cohérence: les points fixes, les cycles et les incohérences sont supprimés des déclarations.
- L'évolution: le raffinement de la base de connaissances en constante évolution est réalisé par un expert du domaine.

  Cependant, l'apprentissage conceptuel est considéré comme une partie du processus d'apprentissage de l'ontologie (Gomez-Perez et Manzano-Macho, 2003).

1.5 Mise en contexte et problématique

L'environnement de la mémoire corporative est en constante changement et l'ontologie devrait évoluer pour s'assurer qu'elle reflète l'ensemble de donnée décrivant le domaine d'intérêt. Comme précisé dans la section précédente, une ontologie est une structure dynamique, elle change pour prendre en compte l'évolution et la dynamique de l'environnement. Les changements peuvent être initiés par les changements conceptuels suite à l'évolution du domaine pour incorporer une fonctionnalité additionnelle ou les changements liés aux spécifications de la conceptualisation ou la représentation de la conceptualisation pour encoder l'ontologie.

La plupart des approches précédentes fournissent un soutien limité pour le processus d'apprentissage, en particulier, la phase d'extraction des connaissances. Dans ces approches, il n'existe pas de méthodes intégrées, ni d'outils qui combinent les différentes techniques d'apprentissage et les sources de connaissances hétérogènes avec les connaissances existantes pour accélérer le processus d'apprentissage. Dans les approches de l'analyse conceptuelle formelle, le treillis conceptuel est une structure de donnée compliquée et la construction des treillis de concepts n'est pas triviale. Les approches qui utilisent le groupement conceptuel négligent la structure conceptuelle contenue dans l'ontologie et il n'y a que les relations taxonomiques des concepts qui peuvent être générées. Les approches basées sur les règles d'association fournissent seulement un soutien limité pour aider à générer l'ontologie parce que l'apprentissage de l'ontologie utilise souvent des textes non-structurés. Le groupement conceptuel traite efficacement la hiérarchie conceptuelle mais elles ne sont pas fiables à cause de l'hétérogénéité sémantique. La précision et le rappel dans les approches d'élagage ne satisfait les demandes des applications réelles. Les approches basées sur les patrons nécessitent des experts pour la conception des patrons et les modifications bruitent les données. Dans les approches d'apprentissage conceptuel, l'apprentissage axiomatique reste inexploré et elles ne sont pas appropriées aux problèmes avec une grande hétérogénéité.

Dans la majorité des approches d'apprentissage précédentes, la représentation textuelle génère une dimensionnalité trop élevée de l'espace d'indexation. De ce fait, le nombre de variables excède l'échantillon de donnée. D'une part, la malédiction de la dimensionnalité ajoutera un bruit inutile dans les frontières de décision et affectera la généralisation de l'apprentissage machine. D'autre part, la malédiction de la dimensionnalité ou le fléau de la dimensionnalité augmente la taille de l'espace de recherche et le temps d'exécution. De plus, l'évaluation demeure un problème important dans la majorité des approches. Le choix d'une approche appropriée dépend des critères utilisés pour évaluer les ontologies.

En résumé, l'extraction des connaissances à partir de données s'est avérée être le principal défi pour l'apprentissage des ontologies. En conséquence, la conception et la réalisation d'un processus d'apprentissage ne sont pas évidentes; notamment l'extraction et la mise à jour peuvent soulever les problèmes suivants:

- Le but ultime de l'ontologie est de refléter la connaissance implicite d'un nombre de documents représentant le modèle du domaine. Cependant, le web, et particulièrement la mémoire corporative, sont connus par l'importance et l'évolution de leur connaissance. De ce fait, la dynamique l'environnement devrait être implicitement considérée, ses caractéristiques doivent être identifiées et représentées pour gouverner le processus d'apprentissage. En d'autres termes, le processus d'apprentissage devrait prendre en considération l'évolution des ressources textuelles plutôt que de traiter l'environnement en tant qu'entité statique. Dans cette optique, des algorithmes efficaces sont nécessaires pour extraire l'information pertinente dans un environnement de grande taille et dynamique.
- La malédiction de la dimensionnalité ou le fléau de la dimensionnalité associée à l'augmentation exponentielle de la dimension de l'espace d'indexation textuelle ajoutera un bruit inutile dans les frontières de décision et l'apprentissage obtient une performance significativement plus faible. Par conséquent, la sélection de l'information pertinente nécessite des transformations.
- Afin de coder et représenter l'ontologie, le processus de mise à jour devrait maintenir la cohérence de l'ontologie après l'ajout, la suppression et la modification des connaissances. D'une part, le modèle de la représentation devrait avoir un pouvoir expressif, c'est-à dire, les différentes variations expressives pour représenter la connaissance. D'autre part, l'inférence devrait contrôler la terminaison des preuves par réécriture et les cycles. Puisque l'inférence et l'expressivité sont deux conditions en opposition. Ainsi, le modèle de représentation devrait fournir des mécanismes efficaces pour se confronter au dilemme expressivité-efficacité.

Cette analyse justifie clairement le besoin d'une approche pour traiter les problèmes relatifs à l'apprentissage des ontologies et l'intervention humaine devrait constituer une option du processus d'apprentissage. Ainsi, notre objectif de recherche consiste à mettre en place un processus de maintenance des ontologies en utilisant l'apprentissage machine, le traitement automatique du langage naturel et la recherche d'information. Ces techniques se basent sur les documents textuels disponibles dans la mémoire corporative.

1.6 Objectif de la recherche

Comme précisé dans le paragraphe précédent, l'état de l'art des techniques existantes d'enrichissement des ontologies a permis de préciser la terminologie utilisée et d'explorer les différentes approches d'apprentissage machine. Par conséquent, notre vision de la maintenance des ontologies reflète une conception qui repose sur l'extraction de la connaissance. De ce fait, nous proposons dans ce projet une nouvelle approche semi-automatique pour enrichir l'ontologie en utilisant les techniques de traitement automatique du langage naturel, l'apprentissage machine et la recherche d'information. Le processus d'apprentissage commence par la capture des changements à partir des méthodes d'extraction de connaissances à partir de données. Ces dernières infèrent les changements en se basant sur les documents disponibles dans la mémoire corporative. Les changements résultants de ce genre de besoins sont les changements ascendants ou les changements pilotés par les données. Par conséquent, afin de permettre au système d'apprentissage de supporter l'extraction de la connaissance à partir des données, le choix des documents pour le prétraitement et le nettoyage est nécessaire. Le nettoyage supprime le bruit et les variables non pertinentes. L'ensemble des données est analysé pour identifier les modèles cachés, c'est-à-dire, ceux qui représentent les relations (caractéristiques, corrélations, distributions, contenus, etc.) entre les données en utilisant un engin de Data Mining. Ce dernier est la pièce maîtresse dans le processus d'extraction de la connaissance. Le modèle d'extraction est alors testé pour estimer son taux de reconnaissance et sa performance de généralisation. Tous les modèles cachés sont décrits par des étiquettes descriptives. Il devrait être possible de transformer le modèle caché pour aider les utilisateurs à atteindre leurs buts (identification, repérage, diffusion, etc.) et pour identifier les changements candidats. Par conséquent, il est nécessaire de stocker les rapports entre les entités de l'ontologie et les modèles cachés pour faciliter les révisions. Cette traçabilité permet de défaire et refaire les changements dans l'ontologie.

Les étapes dans le processus d'extraction sont exécutées itérativement jusqu'à ce que la connaissance pertinente soit extraite. Les modèles cachés sont utilisés pour alimenter le processus de mise à jour de

l'ontologie. Ils peuvent également être stockés également comme des nouvelles connaissances dans la base d'apprentissage ou dans la mémoire corporative. Les besoins explicites ou les changements descendants sont reflétés par les utilisateurs qui fournissent leurs avis a propos de la pertinence des artefacts de l'ontologie.

Dans la phase d'extraction, les modèles cachés sont modelés par l'apprentissage qui exploite les documents textuels disponibles dans la mémoire corporative. Ces derniers peuvent partiellement fournir le modèle caché dans la mémoire corporative. Cependant, le modèle caché reflète seulement une petite partie de la connaissance encodée dans l'ontologie. Une partie plus importante est fournie par les relations lexicographiques entre les étiquettes descriptives résultant de contenus des modèles cachés et les artéfacts ontologiques de base. Ainsi, les règles de correspondance entre les modèles cachés et les artéfacts de l'ontologie sont définies par un processus d'alignement. Ce dernier représente le mécanisme de base pour définir les règles de correspondance entre les deux représentations. Cependant, le mécanisme d'alignement ne peut pas décider d'accepter ou rejeter ou changer les artefacts ontologiques. Ainsi, l'intervention de l'expert dans le processus d'alignement est indispensable pour lier le modèle caché à un artefact ontologique et valider les changements candidats.

Les changements sont appliqués à l'ontologie dans un état cohérent. L'ontologie devrait rester consistante après l'exécution des changements pour préserver la sémantique des changements. La vérification de la consistance et la classification automatique des artéfacts ontologiques peuvent être exécutées en utilisant un outil de raisonnement automatique et un modèle de représentation de la connaissance.

Il est important de préciser que concevoir un processus d'apprentissage représente un véritable défi. En effet, le but principal est de représenter et d'adapter les changements afin d'assurer l'évolution et la consistance de l'ontologie: leur architecture doit être conçue de telle manière que les connaissances sont mises à jour et non contradictoires. A noter qu'un nombre important de sous-processus intervient dans le processus de maintenance, entre autres:

- Un mécanisme pour assurer que tous les changements appliqués sur l'ontologie gardent celle-ci dans un état cohérent. Il permet aussi de résoudre les changements sans ambiguïté, vérifier la consistance des concepts (instances) ou la satisfiabilité, les relations de subsomption (instanciation) implicites, contrôler les ré-récritures, les cycles, la complétude, la correction et d'en tirer des déductions.
- Un mécanisme pour traiter les changements transitifs (les changements requis à d'autres parties de l'ontologie) afin d'assurer la consistance et la propagation des changements aux artéfacts dépendants. Ce mécanisme représente la phase de propagation des changements qui devrait s'assurer que tous les changements induits seront propagés aux artéfacts concernés.

- L'ontologie reflète la connaissance implicite d'un nombre de documents représentant le modèle du domaine. Cependant, le volume de données interfère avec le processus d'apprentissage et ce volume est caractérisé par une taille dynamique. Cet impact est plus qu'important étant donné que, comme signalé précédemment, le principal défi est de définir les rapports entre les entités de l'ontologie et l'ensemble de données stockées dans la mémoire corporative. Ainsi, un mécanisme d'apprentissage incrémental est nécessaire, en particulier, pour l'évolution de l'ontologie où l'environnement est perçu en changement dynamique.
- Un mécanisme d'alignement exploitant les règles de correspondance ou les similitudes entre les artéfacts ontologiques et les modèles cachés résultants de l'extraction des connaissances afin de traiter les changements candidats. Les règles de correspondance sont définies sur les propriétés des artéfacts comparés.
- La dimensionnalité de l'espace de représentation textuelle excède souvent le nombre de documents disponibles pour l'apprentissage et cela représente un véritable défi pour l'apprentissage des ontologies. Ainsi, il est nécessaire de réduire la taille de l'espace de représentation avant d'appliquer l'algorithme d'apprentissage machine. De ce fait, le processus d'apprentissage devrait inclure un mécanisme pour conserver les caractéristiques qui englobent l'information pertinente. Ce mécanisme est complexe et nécessite la gestion d'une quantité importante de variables. D'une part, l'exploration de l'espace de représentation textuelle est critique puisque la sélection des variables est un problème d'optimisation combinatoire. D'autre part, un critère robuste de sélection est nécessaire pour mesurer les variables non pertinentes, corrélées, peu significatives, ou fortement corrompues par le bruit au regard de l'apprentissage machine.

- L'apprentissage de l'ontologie a recours aux documents textuels disponibles dans la mémoire corporative. En conséquence, il est nécessaire de stocker les rapports entre les entités de l'ontologie et les modèles cachés pour faciliter les révisions. Cette traçabilité permet de déterminer l'impact des nouveaux changements, défaire et refaire les changements dans l'ontologie. Elle permet aussi de retracer les modifications appliquées à l'ontologie pour préserver l'historique de l'évolution.
- Fournir une structure pour la hiérarchisation et la disposition des artéfacts ontologiques. De cette façon, nous pouvons distinguer clairement les relations hiérarchiques entre les artéfacts ontologiques (classe, subsomption, instanciation, relations, etc.). L'avantage de cette hiérarchisation est de pouvoir faire appel aux experts pour l'élagage et le raffinement.

- Un mécanisme pour identifier la stratégie d'apprentissage et pour évaluer la qualité de la généralisation.
- Une méthode pour représenter et évaluer la qualité de l'ontologie enrichie.

1.7 Hypothèses de recherche

L'apprentissage des ontologies est un processus piloté par les données. Il semble ainsi intéressant de se concentrer sur l'extraction des connaissances, processus fortement axé sur l'apprentissage machine. Ces principaux mécanismes concernent alors, le prétraitement, l'indexation la sélection des variables, la mise à jour, le modèle calculable en finissant par l'évaluation. Partant de ce constat, nous allons maintenant formuler les hypothèses qui seront utilisées dans notre cheminement. Ces hypothèses permettront de cerner le parcours et d'identifier les mécanismes nécessaires pour guider le processus d'apprentissage.

- « *La sélection des variables basée sur le modèle d'emballage de la décomposition en valeurs singulières tronquées ou TSVD (de l'anglais: Truncated Singular Value Decomposition) est plus efficace pour une suppression des termes sans perdre la performance d'apprentissage*»

  Le modèle de représentation des documents textuels génère une dimensionnalité très élevée (la malédiction de la dimensionnalité) même après les prétraitements et le nettoyage. En raison de cette dimensionnalité élevée, la plupart des variables sont redondantes et non pertinentes, ainsi la sélection des variables est utilisée pour réduire l'espace de représentation.

  Afin de sélectionner les variables pertinentes, un critère robuste de sélection est exigé. Deux approches ont été proposées pour résoudre le problème de la malédiction de la dimensionnalité de Bellman (MDB de l'anglais: Bellman's curse of dimensionality) (Rust 1997).

  La première approche est basée sur le modèle de filtrage qui se sert d'un filtre pour sélectionner les variables non pertinentes. La deuxième approche est l'approche basée sur le modèle d'emballage qui utilise un algorithme de Data Mining pour évaluer les variables sélectionnées (Jain et Zongker, 1997) et (Zhao *et al.*, 2002).

  La décomposition en valeurs singulières tronquées est un algorithme d'emballage (Jyh-Jong *et al.*, 2001). Il s'agit d'une méthode de détection des variables pertinentes en utilisant un apprentissage non supervisée. Elle transforme l'espace de représentation dans un espace réduit en utilisant la décomposition en valeurs singulières. Cette dernière permet de conserver les vecteurs singuliers qui englobent un maximum de variances en éliminant les axes qui correspondent à des variances faibles. Plus précisément, les vecteurs singuliers qui correspondent aux petites valeurs singulières seront éliminés.

Autrement dit, l'objectif de la décomposition en valeurs singulières tronquées est de trouver un espace réduit, issu d'une combinaison des termes, tel que la variance du nuage autour de cet axe soit maximale. Sur la base de cette idée, nous allons utiliser la décomposition en valeurs singulières tronquées pour la réduction de l'espace de représentation.

- *« L'enrichissement de l'ontologie peut-être réalisé au moyen d'un processus d'apprentissage non-supervisé »*

  L'apprentissage non-supervisé accumule la connaissance d'une façon incrémentale et est approprié pour les systèmes où l'environnement est perçu en changement dynamique tel que l'évolution de l'ontologie.

- *« La sélection des modèles basée sur la Théorie de la Résonance Adaptative est indispensable pour améliorer la performance d'extraction des modèles cachés»*

  La performance de la classification automatique est une question primordiale notamment pour la maintenance des ontologies. D'une part, dans la majorité des architectures connexionnistes de clustering, le nombre de clusters dans la couche de sortie est fourni par l'utilisateur. D'autre part, l'agrégation des précisions de plusieurs modèles indépendants permet de réduire la variance et donc d'améliorer la généralisation. Pour ces raisons, nous allons utiliser la sélection des modèles (Duda, Hart et Stork, 2001) basée sur La Théorie de la Résonance Adaptative ou ART « En anglais: Adaptive Resonance Theory » (Carpenter, Grossberg et Rosen, 1991).

- *«L'agrégation des alignements basés sur la similitude est une opération nécessaire pour associer les étiquettes aux artéfacts de l'ontologie»*

  L'alignement permet d'associer les étiquettes (substituer, insérer, supprimer) aux artéfacts de l'ontologie en exploitant la similitude entre les deux représentations. La mesure de similitude est basée sur la distance pour transformer l'occurrence de l'étiquette dans la chaine représentant l'artéfact ontologique. Autrement dit, le processus d'alignement applique des règles lexicographiques pour produire les similarités entre les étiquettes descriptives et les entités de l'ontologie. Il crée des règles d'alignement permettant de transformer les étiquettes dans l'ontologie. Cependant, le choix de la mesure de distance est étroitement lié à la détermination d'un alignement optimal. D'une part, la séquence des opérations d'alignement et le coût d'édition ne sont pas uniques en général. D'autre part, les algorithmes d'alignement varient selon le type de recherche et les méthodes utilisées pour réaliser la transformation optimale. Pour ces raisons, nous allons utiliser un mécanisme d'agrégation (Le Capitaine, 2009) des alignements pour chercher la transformation optimale entre les étiquettes et les artéfacts ontologiques.

CHAPITRE II

MÉTHODOLOGIE ET MODELE PROPOSÉ

Résumé: Ce chapitre présente la méthode de recherche suivie pour exécuter ce projet. Tout d'abord, le modèle conceptuel est proposé. Cette première partie concerne ainsi un survol du processus d'apprentissage et plus particulièrement le processus d'extraction des modèles cachés. La deuxième partie présente outre les outils de prétraitement, le modèle d'indexation ainsi que la stratégie de sélection des variables pertinentes. L'architecture de clustering est ensuite proposée. Elle représente le module d'extraction des connaissances, étape primordiale du processus de maintenance. Puis, les quatrième et cinquième parties se focalisent sur l'étiquetage et l'alignement. Enfin, l'étape de traitement de la consistance est décrite afin d'en assurer l'inférence et le raisonnement conceptuel.

Ce chapitre est organisé en cinq paragraphes. Le paragraphe (2.1) présente les différentes étapes du processus d'apprentissage. Dans le paragraphe (2.2) nous présentons les méthodes de prétraitement, le modèle d'indexation textuelle et la sélection des variables. Nous proposons dans ce paragraphe un algorithme d'emballage basé sur la décomposition en valeurs singulières tronquées pour traiter la malédiction de la dimensionnalité. Le paragraphe (2.3) présente le développement d'un modèle de clustering à travers un système hybride neuro-flou. Ce paragraphe vise à montrer les concepts liés à la théorie de la résonance adaptative, l'architecture et la configuration du réseau connexionniste. Le paragraphe (2.4) montre le processus d'alignement entre les modèles cachés et les étiquettes descriptives. Le traitement de la cohérence et l'inférence ontologique est présenté dans le paragraphe (2.5).

2.1 Processus d'apprentissage

Nous nous inspirons des méthodologies récentes sur l'évolution de l'ontologie et nous proposons dans le paragraphe (2.1) une approche semi-automatique qui utilise l'apprentissage machine, le traitement automatique du langage naturel et la recherche d'information pour découvrir les changements candidats qui ont été introduits récemment dans le corpus textuel.

La méthode commence avec un ensemble de documents concernant le domaine d'intérêt. En outre, l'ontologie initiale du domaine est nécessaire comme entrée. L'impact du bruit sur la performance de l'indexation textuelle est déterminé par le niveau et la distribution de bruit dans les données. Afin d'améliorer la performance d'indexation, Il est nécessaire de supprimer toutes les occurrences de bruit. Le dictionnaire négatif et la troncature sont les méthodes communément utilisées dans la recherche d'information pour supprimer le bruit. Les documents sont représentés en utilisant le modèle vectoriel, ce modèle traite le document comme un sac de mots. Une caractéristique importante de cette représentation est la dimensionnalité élevée de l'espace de variables qui induit un grand défi pour la performance d'apprentissage. Le clustering pourrait ne pas fonctionner efficacement avec un grand nombre de variables. Toutes les variables ne sont pas pertinentes pour l'algorithme de clustering textuel car elles peuvent diminuer la capacité de généralisation. Dans un tel cas, la sélection des variables mène souvent à une meilleure performance, elle permet de choisir les variables pertinentes à partir de l'ensemble original des variables, en procédant à l'élimination des variables redondantes. Cette étape traite la malédiction de la dimension de Bellman (Curse of dimensionality) et de ce fait, elle améliore le taux de reconnaissance dans le processus d'apprentissage. Les vecteurs liés aux variables sélectionnées sont les vecteurs qui englobent une quantité importante de la variance dans les données.

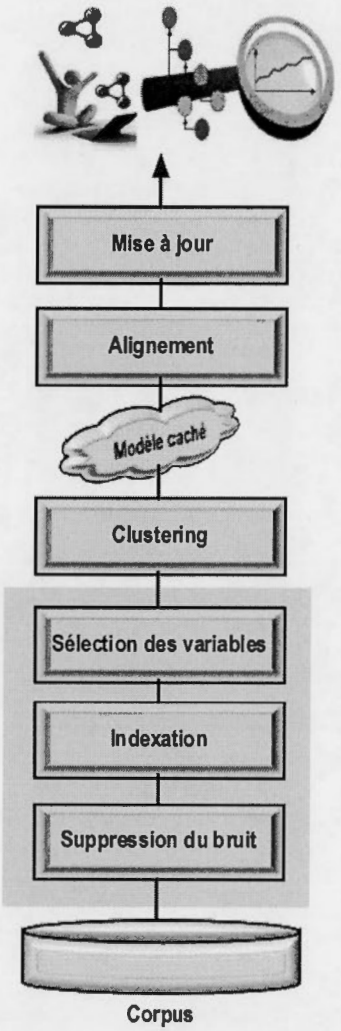


Figure 2.1 Processus d'apprentissage

Ceci peut être formulé par l'identification des indexes pertinents dans le modèle d'indexation. Les étapes de prétraitement, l'indexation et la sélection des variables sont les différentes formes de prétraitement des données, où les données sont préparées pour la fouille. Le clustering est utilisé pour découvrir les modèles cachés permettant ainsi de discerner les contenus dans des clusters qui ont une forte densité au sein de l'espace d'indexation. L'objectif de cette étape est d'organiser la collection de documents dans des groupes sur la base leurs distances. Une étiquette est attribuée à chaque sous-ensemble de documents pour en faciliter la gestion des documents et pour identifier les changements candidats.

Un processus d'alignement approximatif est utilisé pour repérer la correspondance entre les artefacts ontologiques et les étiquettes descriptives. Le processus concentre sur le problème d'alignement approximatif des chaînes permettant un nombre limité d'erreurs dans les correspondances. Il utilise des règles lexicographiques pour améliorer les degrés de similarité entre les entités ontologiques et les étiquettes descriptives. Ces règles sont fondées sur la distance entre les deux représentations. Les techniques de représentation de la connaissance sont utilisées pour décrire une conceptualisation explicite et formelle pour le modèle du domaine. La vérification de la consistance et la classification automatique des artéfacts ontologiques peuvent être exécutées en utilisant un outil de raisonnement automatique.

Le processus de la maintenance de l'ontologie est un système de Data Mining impliquant plusieurs étapes itératives. Dans les paragraphes suivants, nous allons détailler toutes les étapes décrites à la figure (2.1).

2.2 Prétraitement

2.2.1 Surpression du bruit

Le bruit peut induire en erreur le modèle d'indexation en définissant des corrélations inexistantes entre les documents. Avant de créer la représentation du document, il est nécessaire de supprimer toutes les occurrences de bruit. Cette étape de prétraitement implique généralement la suppression des chiffres et des signes de ponctuation, la conversion des mots en lettres minuscules, la suppression des mots fonctionnels et la troncature.

- La suppression de la ponctuation: il est nécessaire de supprimer toutes les occurrences de signes de ponctuation tels que *"?"*, *"&"*, *"("*, *")"*, *"©"*, *"/"*, *"@"*, *"0"*, *"1"*, *"2"*, etc.
- Le dictionnaire négatif: les mots fonctionnels tels que *"a"*, *"about"*, *"before"*, *"mostly"*, *"more"*, *"something"*, *"their"*, *"them"*, *"one"*, *"with"*, *"the"*, *"those"* ne sont pas importants pour la tâche d'apprentissage et leurs utilisations peuvent dégrader la capacité de généralisation.

- Troncature: les termes dans le corpus d'apprentissage ont de nombreuses variations morphologiques, par exemple, nous pouvons utiliser les mots « doing, do », « going, go » et « examples, example » dans le même document. Le remplacement des termes par les lexèmes pourrait améliorer la performance d'apprentissage. Cependant, il n'existe pas un algorithme communément utilisé et le choix dépend du langage utilisé et les paramètres d'analyse.

Le prochain paragraphe se focalise sur la représentation du document textuel, la pondération des poids et la normalisation de la représentation.

2.2.2 Indexation

La troisième étape dans le processus de prétraitement est la transformation des documents textuels en une représentation interprétable par le processus d'apprentissage.

Le modèle vectoriel (Salton, Wong et Yang, 1975) facilite l'application de divers algorithmes d'apprentissage machine et aide à calculer la similarité entre les documents. Dans cette représentation, un ensemble de termes sont choisis comme des mots clés pour indexer la base d'apprentissage. Chaque document est représenté comme un vecteur de dimension n.

$$\vec{d}_j = (w_{1j}, w_{2j}, \ldots\ldots, w_{ij}, \ldots w_{|T|j}) \quad (1 \le j \le n).$$

w_{ij} : représente le poids du terme t_i dans le document textuel d_j.

$|T|$: représente le nombre de différents termes.

$$\widehat{D} = \begin{pmatrix} tfidf_{1,1} & \cdots & tfidf_{1,n} \\ \vdots & \ddots & \vdots \\ tfidf_{m,1} & \cdots & tfidf_{m,n} \end{pmatrix}$$

Figure 2.2 La représentation vectorielle

Comme montré à la figure (2.2), chaque case dans la matrice [terme-document] représente les occurrences de chaque terme dans le document. Autrement dit, $D[i,j] = a_{ij}$ est la fréquence d'occurrence du terme i dans le document j.

Puisque chaque terme n'apparaît pas dans chaque document, la matrice D est une matrice creuse. Dans la pratique, une pondération peut être appliquée sur la matrice D afin d'améliorer le taux de repérage. Ainsi,

la fréquence d'occurrence d'un terme peut être transformée en utilisant un poids local $L(i,j)$ et un poids global $G(i)$, c'est-à-dire, $a_{ij} = L(i,j) \cdot G(i)$.

Le poids local peut prendre la forme suivante:

- La fréquence du terme: $L(i,j) = tf_{ij}$.
- Le logarithme: $L(i,j) = \log_2(tf_{ij} + 1)$.

tf_{ij}: est la fréquence du terme i dans le document j.

Le poids global peut prendre la forme suivante:

- Normalisation: $G(i) = \frac{1}{\sqrt{(\sum_j tf_{ij}^2)}}$.
- $GFIDF$: $G(i) = \frac{gf_i}{df_i}$.
- IDF: $G(i) = \log_2(\frac{nd}{df_i}) + 1$.
- Entropie: $G(i) = 1 - \sum_j \frac{p_{ij} \log_2(p_{ij})}{\log_2(n_d)}$, $p_{ij} = \frac{tf_{ij}}{gf_i}$.

$L(i,j)$: est le poids local du terme i dans le document j.

$G(i)$: est le poids global du terme i.

gf_i: est la fréquence globale du terme i dans le corpus d'apprentissage.

df_i: est le nombre de documents contenant le terme i.

n_d: le nombre de documents dans le corpus d'apprentissage.

Dans notre approche, les poids w_{ij} sont calculés en utilisant la méthode de la fréquence des termes et la fréquence inverse du document (Term Frequency Inverse Document Frequency):

$$tfidf(t_k, d_j) = \#(t_k, d_j) . \log \frac{|T_r|}{\#_{T_r}(t_k)}$$

Où:

$\#(t_k, d_j)$: la fréquence d'apparition du terme t_k dans le document d_j.

$\#_{T_r}(t_k)$: le nombre de documents qui contiennent t_k.

$|T_r|$: le nombre de documents dans l'ensemble T_r (la base d'apprentissage).

Pour compter les documents de différentes longueurs, chaque vecteur est normalisé à l'unité de longueur en utilisant la normalisation cosinus comme montré à la figure (2.3):

$$w_{kj} = \frac{tfidf(t_k, d_j)}{\sqrt{\sum_{s=1}^{|T|} (tfidf(t_s, d_j))^2}}$$

w_{kj} : représente le poids du terme t_k dans le document d_j .

	t_1	t_2	...	t_n
d_1	w_{11}	w_{12}	...	w_{1n}
d_2	w_{21}	w_{22}	...	w_{2n}
.	.	.	...	.
.	.	.	...	.
d_m	w_{n1}	w_{n2}	...	w_{mn}

Figure 2.3 Le modèle d'indexation textuelle

Cette représentation est moins discriminatoire si le terme apparaît dans plusieurs documents et considère la distribution du terme dans la corpus d'apprentissage.
Le prochain paragraphe s'intéresse plus particulièrement au paradigme de la sélection des variables qui permet de traiter la malédiction de la dimensionnalité de l'espace d'indexation.

2.2.3 Sélection des variables

La malédiction de la dimensionnalité est un défi décisif pour plusieurs problèmes liés au Data Mining. D'une part, le modèle ne peut pas délimiter les clusters par des frontières de décision complexes avec un espace de projection très faible. D'autre part, l'utilisation de nombreuses variables ajoutera un bruit inutile dans le modèle d'indexation [VSM] et affectera la généralisation. Dans notre approche, nous allons utiliser un modèle d'emballage (Jain et Zongker, 1997) et (Zhao *et al.*, 2002) basé sur la décomposition en valeurs singulières tronquées ou TSVD (Truncated Singular Value Decomposition) (Jyh-Jong *et al.*, 2001).

Au lieu d'utiliser le modèle d'espace vectoriel ou VSM, les termes d'indexation sont choisis sur la base de leurs scores distinctifs en fonction des valeurs singulières définies par le modèle d'emballage de la décomposition en valeurs singulières tronquées.

En termes mathématiques, une matrice d'indexation D [terme, document] est formée à partir du corpus textuel dans laquelle chaque entrée est un nombre réel représentant la fréquence des termes et la fréquence inverse du document d'un terme dans un document particulier.

Comme montré à la figure (2.4), la matrice $D\ [tfidf_{ij}]$ est factorisée en trois matrices singulières:

$$\hat{D} = U_r \Sigma_r V_r^T, UU^T = I_m \qquad et \quad VV^T = I_n \qquad (1)$$

$$\Sigma = diag[\sigma_1, \sigma_2, ..., \sigma_{\min(m,n)}], \sigma_i > 0 \quad (1 \prec i \leq r) \wedge \sigma_j = 0 \quad (j > r).$$

$$0 \prec i \leq m,\ 0 \prec j \leq n$$

I_m et I_n sont des matrices d'identité de tailles m et n (respectivement).

Telle que:

r : est le rang de la matrice.

U_r : est une matrice dense de taille $[m,r]$.

Σ_r : est une matrice diagonale de valeurs singulières positives triées par ordre décroissant.

V_r^T : est une matrice dense de taille $[r,n]$.

Les colonnes de la matrice U_r s'appellent « les vecteurs singuliers gauches ». Les colonnes de la matrice V_r^T s'appellent « les vecteurs singuliers droits ».

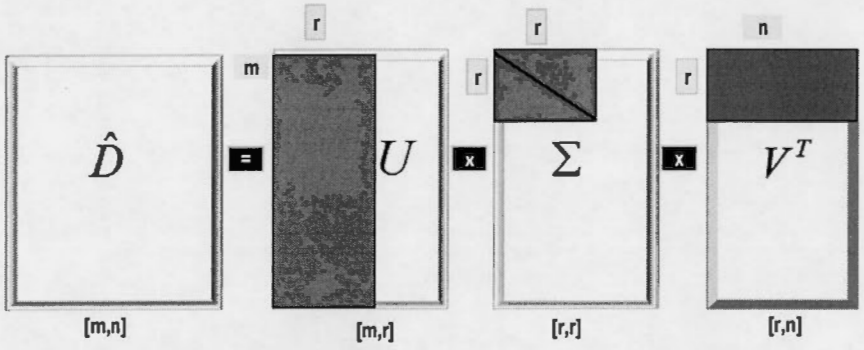


Figure 2.4 La décomposition en valeurs singulières

Les valeurs singulières représentent les directions qui expliquent le mieux la variance dans l'indexation d'un document particulier. Sur la base d'une combinaison de ces valeurs singulières, une nouvelle direction de l'espace peut être construite. Les variables avec les plus grandes valeurs singulières sont considérées pertinentes dans le modèle d'indexation et elles permettent d'expliquer une quantité importante de la variance dans les données. Tandis que les variables liées à des petites valeurs singulières sont considérés non pertinentes.

Le paragraphe qui suit présente le clustering dynamique, l'architecture connexionniste de la théorie de la résonance adaptative floue ainsi que les opérations de configuration.

2.3 Clustering

Le clustering est l'une des techniques puissantes de l'analyse de données qui utilise l'apprentissage non supervisé dans lequel les différents vecteurs-documents sont groupées sur la base leur distances. Cette technique a été utilisée avec succès dans plusieurs sous disciplines du Data Mining telles que la regression (Chung-Hsien *et al.*, 2010), les règles d'association, la reconnaissance des formes (Xin-xing *et al.*, 2010), apprentissage machine (Qu, 2009) et la recherche d'information (Frakes, A. et al. 2000) , etc.

Dans la majorité des algorithmes de clustering, le nombre de clusters requis au départ est fourni par l'utilisateur. Par contre, dans un environnement dynamique, cette connaissance est habituellement inconnue à l'avance. Il est souhaitable d'identifier automatiquement le nombre de clusters pour découvrir la structure intrinsèque dans l'ensemble de documents. Pour ces raisons, nous allons utiliser la théorie de la résonance adaptative ou ART (Adaptive Resonance Theory) pour organiser les documents dans des sous-ensembles thématiques suivant leurs sémantiques.

Les réseaux connexionnistes de la théorie de la résonance adaptative sont développés pour adresser le dilemme (stabilité - élasticité). Un réseau connexionniste possède le caractère d'élasticité s'il peut s'adapter indéfiniment aux entrées. Le dilemme (stabilité - élasticité) peut être décrit comme suit:

« How can a learning system be designed to remain plastic, or adaptive, in response to significant events and yet remain stable in response to irrelevant events?
How does the system know how to switch between its stable and its plastic modes to achieve stability without rigidity and plasticity without chaos?
In particular, how can it preserve its previously learned knowledge while continuing to learn new things?
And, what prevents the new learning to wash away the memories of prior learning? »

Source (Heins et Tauritz, 1995)

Les réseaux connexionnistes de la théorie de la résonance adaptative proposés par Grossberg permettent de résoudre ces deux notions antagonistes. En dehors de sa capacité à regrouper des formes, la théorie de la résonance adaptative a utilisée dans plusieurs systèmes en raison de l'utilisation réduite de ressources mémoire et la stabilité d'apprentissage. Dans un réseau ART, l'information se réverbère entre les couches du réseau. L'apprentissage est possible dans le réseau quand la résonance de l'activité neuronale se produit dans les cas suivants:

- La reconnaissance d'une forme déjà apprise.
- Le réseau se rend compte que la forme constitue une nouvelle information. Ainsi, le modèle passe à l'état de résonance pour la mémoriser.

Beaucoup d'architectures de la théorie de la résonance adaptative ont été développées pour améliorer les capacités du clustering telles que: la théorie de la résonance adaptative binaire ou ART1 (Binary Adaptive Resonance Theory), ART2, la carte de la théorie de la résonance adaptative ou ARTMAP, la théorie de la résonance adaptative floue ou Fuzzy ART et Fuzzy ARTMAP, etc.

L'architecture de la théorie de la résonance adaptative binaire ou ART1 est une sorte de réseau de neurones permettant de classer un vecteur d'entrée dans des catégories selon son degré de ressemblance avec les formes déjà apprises et si le vecteur d'entrée ne correspond à aucune forme stockée, une nouvelle catégorie est créée. De plus, aucune forme apprise n'est modifiée si elle ne correspond pas à la forme d'entrée courante. Ainsi, ce réseau peut résoudre le dilemme de la stabilité-élasticité qui a été mentionné précédemment. Néanmoins, ce réseau ne permet pas de traiter des données analogiques. ART2 quant à lui, a été conçu pour détecter les régularités dans un ensemble d'apprentissage analogique. Il utilise une architecture coûteuse du point de vue calcul, qui présente des difficultés pour le choix des paramètres. Pour surmonter ces difficultés, nous allons utiliser la théorie de la résonance adaptative floue.

Comme montré à la figure (2.5), l'architecture connexionniste de la théorie de la résonance adaptative floue se compose de deux couches: une couche d'entrée (F_1^a) appelée également la couche de comparaison et une couche de sortie (F2) appelée la couche de reconnaissance. La couche d'entrée contient n neurones, où n est la taille du vecteur document. Le nombre de neurones dans la couche de sortie est déterminé dynamiquement. Chaque neurone dans la couche de sortie a un vecteur prototype correspondant.

b_{ij} : les poids des connexions ascendants entre le neurone i de F_1^a et le neurone j de F_2^a.

t_{ij} : les poids des connexions descendantes entre le neurone j de F_2^a et le neurone i de F_1^a.

Gain1, Gain 2 et STM Reset: sont des fonctions de contrôle pour l'apprentissage et la reconnaissance.

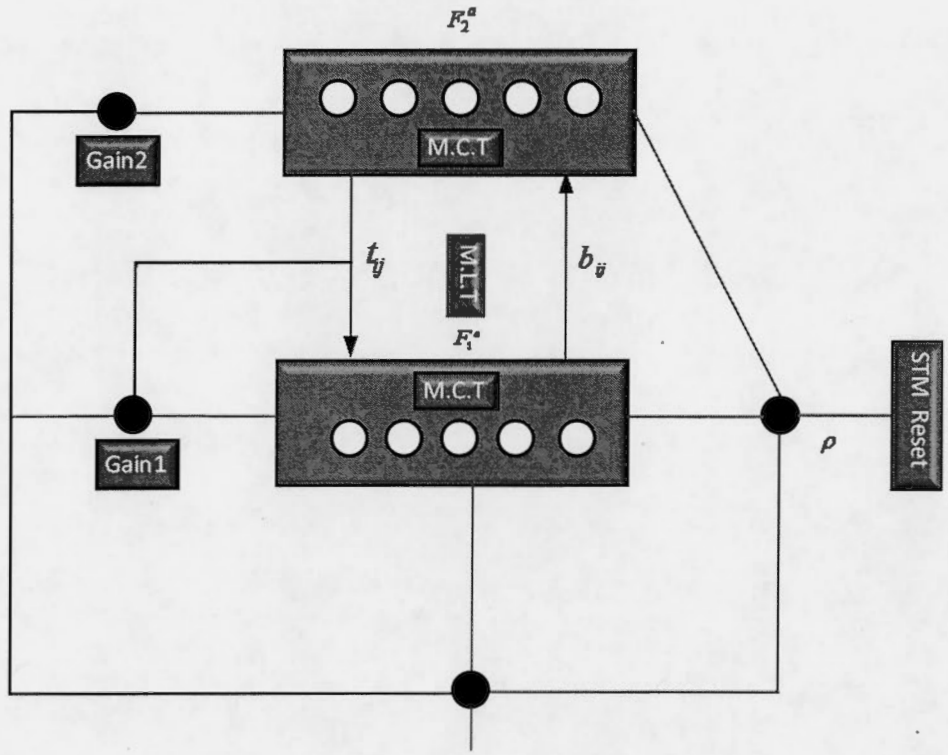


Figure 2.5 Architecture du réseau Fuzzy ART (Boukhadoum, 2010)

L'efficacité du processus de groupement est liée au paramètre de résonance du réseau connexionniste Fuzzy ART. Ce modèle est une généralisation de la théorie de la résonance adaptative binaire en utilisant la logique floue et il permet de concilier le compromis élasticité-stabilité avec son mécanisme dynamique.

La tâche de regroupement avec le réseau connexionniste exige un ensemble d'opérations de prétraitement avant de présenter les vecteurs-documents d'entrée au champ F_1^a. La première étape de prétraitement transforme chaque « vecteur-document » en un vecteur de sortie normalisé, c'est-à-dire chaque composante se trouve dans l'intervalle [0,1]. La normalisation euclidienne permet de convertir une forme d'entrée d_i de la façon suivante:

$$d_i \leftarrow \frac{d_i}{\|d_i\|},$$

C'est-à-dire,

$$tfidf_{ij} \leftarrow \frac{tfidf_{ij}}{\sqrt{\sum_{j=1}^{N} tfidf_{ij}^2}}; (1 \le j \le n), (1 \le i \le m), \quad \|d_i\| \succ 0$$

De ce fait, chaque vecteur d_i dans la base d'apprentissage prend la forme suivante:

$$\begin{cases} d_i(tfidf_{i1}, tfidf_{i2}, ..., tfidf_{in}) \\ (0 \le tfidf_{ij} \le 1, (1 \le i \le n), (1 \le j \le m) \end{cases}$$

Carpenter mentionne le problème de la compacité de clusters qui peut se produire avec Fuzzy ART durant l'apprentissage (Carpenter, Grossberg et Rosen, 1991). Ce phénomène est corrigé en utilisant le codage complémentaire des formes d'entrée. De ce fait, la deuxième étape de prétraitement produit un vecteur de sortie d_i, tel que:

$$\begin{cases} d_i = (tfidf_{ij}, tfidf_{ij}^{\ c}) = (tfidf_{i1}, ..., tfidf_{in}, tfidf_{i1}^c, ..., tfidf_{in}^c) \\ tfidf_{ij}^c = 1 - tfidf_{ij}, (1 \le i \le n), (1 \le j \le n) \end{cases}$$

Cette transformation est appelée le codage complémentaire (codage en complément de 1). Ce codage est effectué dans le champ préprocesseur désigné par F_1^a. Il offre plusieurs avantages, à savoir: la préservation complète de toute information stockée dans la longueur du vecteur, c'est-à-dire, la préservation de l'amplitude du vecteur et la génération de la redondance pour mieux distinguer les versions bruitées.
Comme montré à la figure (2.6), l'algorithme de la théorie de la résonance adaptative floue est un algorithme incrémental qui s'adapte indéfiniment aux nouvelles entrées. Dans le même temps, il ne permet pas aux nouvelles entrées de changer les poids de connexion b_{ij}, t_{ij} appris sauf si la forme d'entrée correspond à une forme apprise avec une certaine résonance ρ, c'est-à-dire que le système ne permet pas de changer les formes apprises à moins qu'elles ne soient suffisamment semblables[1]. La terminaison de l'algorithme est contrôlée par un seuil de terminaison, par exemple, le nombre maximal d'époques, à savoir, le nombre de fois où les données d'apprentissage sont présentées à plusieurs reprises au réseau connexionniste.

[1] Le dilemme (stabilité - élasticité).

Début

- ***Etape 1*** Initialisation
  - Les poids synaptiques: $t(0)_{ij} = 1; \quad 0 \prec b_{ij}(0) \prec \frac{L}{L-1+n}; L \succ 0.$
  - Paramètre de configuration:
    - Choisir les paramètres un seuil de résonance: $0 \prec \rho \leq 1, \alpha \succ 0, \beta \geq 0.$
  - Présenter le nouveau vecteur -document d_i.
  - Normalisation. $\begin{cases} d_i(tfidf_{i1}, tfidf_{i2}, ..., tfidf_{in}) \\ (0 \leq tfidf_{ij} \leq 1, (1 \leq i \leq m), (1 \leq j \leq n) \end{cases}$
  - Codage complémentaire $d_i = (tfid_{ij}, tfid_{ij}{}^c)$.
- ***Etape 2*** Trouver le vecteur prototype le plus proche dans l'ensemble des vecteurs prototypes

  Candidats: $J = \arg\max_j (y_j = \frac{\sum_{i=1}^{2n} \min(b_{ij}, tfidf_{ji})}{\alpha + \sum_{i=1}^{2n} b_{ij}}$
- ***Etape 3*** Vérifier si le vecteur prototype sélectionné est plus proche du vecteur d'entrée.
  - *Unification:* $x_i = \sum_{i=1}^{2n} \min(t_{Ji}, tfidf_{Ji}), \frac{\|\vec{x}\|}{\|\vec{d}_i\|} := \frac{\sum_{i=1}^{2n} x_i}{\sum_{i=1}^{2n} tfidf_{Ji}}$
  - *Comparaison:* $\frac{\|\vec{x}\|}{\|\vec{d}_i\|} \prec \rho$
- ***Etape 4*** Mise à jour du vecteur prototype approprié ou création d'une nouvelle catégorie.
  - Poids montants: $b_{Ji} = (1-\beta) b_{Ji} + \beta (b_{Ji}, tfidf_{ji})$
  - Poids descendants: $\beta \in [0,1]$ (β =1 pour l'apprentissage rapide).

Fin

Figure 2.6 Le pseudo code Fuzzy ART (Boukhadoum, 2010)

Le prochain paragraphe s'intéresse au processus d'alignement pour repérer la correspondance entre les artéfacts ontologiques et les modèles cachés.

2.4 Alignement

Pour mettre à jour l'ontologie, une des questions principales est l'alignement entre les modèles cachés et les artéfacts ontologiques de base. Une méthode automatique ou semi automatique devient nécessaire pour éviter l'alignement manuel qui est ardu et très coûteux.

Ces dernières années, plusieurs approches ont été proposées pour résoudre le problème d'alignement des ontologies. Parmi ces approches on retrouve: SKAT, S-Math, RFCA, Chimaera, CAIMAN, GLUE, IF-map, COMA, Rondo, etc. La liste complète peut être consultée dans les articles (Choi, Song et Han, 2006) et (Kalfoglou et Schorlemmer, 2003b). Cependant la précision et le rappel ne permettent pas de satisfaire les besoins des utilisateurs. De plus, ces approches ne prennent pas en compte toute l'information disponible dans l'ontologie. Elles sont souvent concentrées sur des types limités et négligent les autres. Par exemple, la méthode (Doan *et al.*, 2002) utilise les techniques d'apprentissage automatique basées sur les instances pour déterminer les concepts équivalents entre deux ontologies. Cette approche apporte une contribution acceptable mais elle néglige l'information structurelle contenue dans l'ontologie. De plus, les résultats dépendent de la taille et la représentativité des exemples utilisés pour apprendre le réseau Bayesien. L'approche décrite dans (Ehrig et Sure, 2004) utilise l'apprentissage machine pour apprendre comment calculer les similarités entre deux concepts. Cette approche présente plusieurs inconvénients: les minima locaux, le coût pour définir et valider l'architecture connexionniste, architecture stable qui ne prend pas en compte la dynamique de création des classes, les résultats dépendent de la taille et la représentativité des exemples utilisés pour l'apprentissage connexionniste. Les approches basées sur la logique (Kalfoglou et Schorlemmer, 2003a) permettent de vérifier l'inconsistance. Elles prennent en compte la structure taxonomique. Cependant, ces algorithmes ne sont pas appropriés aux problèmes avec hétérogénéité. Les approches basées sur les graphes et les arbres (Li, Hu et Hu, 2006), (Shihan et Jinzhao, 2010) montrent de bons résultats. Elles utilisent l'information structurelle efficacement mais elles rentrent dans les niveaux de détail sans considérer la structure ontologique générale. Ceci peut mener aux minimums locaux.

Dans ce projet nous proposons une nouvelle approche semi automatique pour aligner systématiquement toute l'information disponible dans l'ontologie avec les modèles cachés acquis par l'apprentissage. Dans cette approche, nous utilisons plusieurs étapes pour achever l'interopérabilité entre les deux représentations. Les étapes de la méthodologie proposée seront présentées en détails dans les paragraphes suivants.

- Étiquetage: nous mettons l'accent sur une méthode de clustering efficace permettant de fournir des étiquettes précises en plus de la découverte des clusters. Le processus d'étiquetage est composé de plusieurs étapes:

  Premièrement, les mots clés sont rangés en utilisant la formule de la fréquence inverse des documents:

$$tfidf(Label_k, C_j) = \#(Label_k, C_j).\log\frac{|C_r|}{\#_{T_r}(C_k)}$$

  $(Label_k, C_j)$: représente le nombre d'occurrences du mot clé $Label_k$ dans le cluster C_j.

  $\#_{T_r}(C_k)$: représente le nombre de clusters contenant le mot-clé $Label_k$.

  $|C_r|$: le nombre total des clusters.

  Cette approche donne moins de poids à un mot clé s'il est présent dans d'autres clusters. De plus, l'occurrence générale du mot clé représente le concept du domaine. Par conséquent, cette approche est très utile pour distinguer les différents clusters.

  Deuxièmement, dans l'étape d'étiquetage nous adoptons le voisinage cosinus (Berzal et Matín, 2002) et (Duda, Hart et Stork, 2001) avec une différence majeure concernant le modèle d'apprentissage « le vecteur de caractéristiques moyen avec un vote majoritaire ». La méthode du voisinage cosinus est une extrapolation du classifieur euclidien. Au lieu d'utiliser le vecteur caractéristique moyen comme le prototype d'un cluster, la méthode fait intervenir tous les vecteurs du voisinage selon un seuil donné par l'utilisateur. La distance cosinus entre chacun de ceux-ci et celui du prototype est calculée et l'étiquette assignée au cluster est alors celle du vote majoritaire.

  Formellement, étant donné un vecteur document D, la similarité cosinus est calculée en utilisant le produit scalaire et la norme.

$$\cos(d_i, \mu^k) = \frac{d_i \cdot \mu^k}{\|d_i\| \cdot \|\mu^k\|} = \frac{\sum_{j=1}^{2n} tfidf_{ij} \cdot \mu_j^k}{\sqrt{\sum_{j=1}^{2n} (tfidf_{ij})^2} \cdot \sqrt{\sum_{j=1}^{2n} (\mu_i^k)^2}}$$

  $\mu^k(\mu_1^k, \mu_2^k, ..., \mu_{2n}^k)$: le centroïde d'un cluster C_k.

  $d_i(d_{i1}, d_{i2}, ..., d_{i2n})$: le vecteur document.

  Le centroïde μ^k d'un cluster C_k est définie par l'équation suivante: $u^k = \frac{1}{m_k} \sum_{d_i \in C_k} d_i$

  m_k : le nombre de document dans le cluster C_k.

  La similarité cosinus indique la similitude intra-cluster et est utilisée pour normaliser la longueur du document lors de la comparaison. La fonction discriminante a la forme suivante:

$$Label_k = voteMajoritaire(Arg\max_i\{\cos(d_i, \mu^k)\}).$$

voteMajoritaire : la fonction du vote majoritaire.

L'étiquette finale est obtenue par un vote majoritaire des mots clés sélectionnés.

- Approximation: l'objectif général est de réaliser l'alignement des chaînes dans un modèle ontologique où les formes d'alignement ont un certain genre d'erreurs. Autrement dit, le but est de trouver la distance entre les étiquettes et les artefacts de l'ontologie permettant un nombre limité d'erreurs dans l'alignement. L'idée étant de rendre la distance minimale lorsqu'une des formes est susceptible d'être une variante incorrecte de l'autre. Ainsi, notre objectif est de réduire au minimum le coût total de la distance pour passer d'une chaine à l'autre.

  La mesure de similarité utilisant les techniques de distance est l'une des méthodes la plus utilisée pour évaluer la similarité entre deux chaînes. Les petites distances correspondent à de grandes similitudes et les grandes distances correspondent à des petites similitudes. Comme on peut le constater plusieurs mesures de similarité ont été proposées dans la littérature et la plupart d'entre elles s'appuient sur la fréquence d'occurrence et la structure hiérarchique des ontologies. Parmi ces similarités on retrouve: la mesure de Jaccard (Shibata, Kajikawa et Sakata, 2010), Hamming (Liu,2011), DICE (Jiannan, 2011), Dynamic Time Warning(Gang,2011), (Resnik,1995), (Hirst-St-Onge, 1998), (Leacock-Chodorow,1998), (Wu-Palmer,1994), etc. Une liste complète des fonctions de similarité est fournie dans les articles (Zargayouna et Salotti, 2004), (Ichise, 2009), (Alexandru-Lucian et Iftene, 2010) et (Jiannan, Guoliang et Jianhua, 2011) .

  Dans notre approche, nous utilisons plusieurs méthodes pour trouver les règles d'alignement entre les étiquettes descriptives et les artéfacts ontologiques en utilisant les techniques de la similarité basées sur les mesures de distance. Ainsi, pour créer des règles d'alignement, le processus d'extraction des similarités applique des techniques lexicographiques pour produire une matrice de similitudes reflétant des similarités entre les artéfacts ontologiques et les étiquettes. Autrement dit, il crée des règles d'alignement qui définissent comment transformer les étiquettes dans l'ontologie en définissant tous les types d'associations possibles entre les deux représentations.

2.5 Traitement de la consistance

La cohérence et la structure hiérarchique peuvent être utilisées pour vérifier l'inconsistance logique et l'héritage implicite.

Comme expliqué dans le paragraphe (1.1), le partage de la compréhension de la structure d'information est l'un des buts visés par le Web sémantique. La représentation et l'encodage de l'ontologie dépendent en grande partie de la disponibilité d'une sémantique bien définie et un raisonnement décidable. La logique descriptive semble être parfaitement adaptée à cette situation. Elle diffère de ses prédécesseurs, tels les graphes conceptuels, les graphes existentiels, les réseaux sémantiques et les cadres où elle est équipée par une sémantique formelle et des procédures de décision complètes et correctes. Outre cela, les principales tâches d'inférence, en particulier, l'instanciation et la subsomption dans la logique descriptive sont décidables. Cela nous permet de contrôler l'inférence et réduire le coût informatiques des ressources (Guohua *et al.*, 2007). Pour ces raisons, nous allons utiliser les outils d'inférence basés sur la logique descriptive pour vérifier la consistance de l'ontologie.

Conclusion

Dans ce chapitre, nous avons présenté une nouvelle méthode d'apprentissage des ontologies. La méthode proposée contient plusieurs phases.

Dans la première phase, les outils de prétraitement sont utilisés pour supprimer le bruit. Les documents textuels sont représentés en utilisant le modèle vectoriel. Avec cette représentation, chaque document est considéré comme un vecteur représenté à l'aide de la fréquence des termes et la fréquence inverse du document (Term Frequency-Inverse Document Frequency). Cependant, cette représentation génère un espace hautement dimensionnel, là où le nombre de variables est beaucoup plus grand que le nombre de documents disponibles pour l'apprentissage. La sélection des variables répond à un réel besoin en matière de réduction de la dimensionnalité de l'espace d'indexation.

Pour dériver une indexation optimale de l'espace de représentation vectoriel dans un espace réduit, le processus d'apprentissage utilise le modèle d'emballage basé sur décomposition en valeurs singulières tronquées.

La deuxième phase consiste à générer dynamiquement un ensemble de clusters en utilisant le clustering dynamique de la théorie de la résonance adaptative. Ce réseau connexionniste permet de rechercher la forme complexe des frontières de décision qui séparent les clusters. Ainsi, il peut induire effacement les changements dans le processus de maintenance. Les clusters sont considérés comme des marqueurs générés automatiquement pour alimenter le processus de mise à jour.

La troisième phase se focalise sur l'alignement approximatif pour trouver les règles de correspondance entre les modèles cachées et les artéfacts ontologiques de base. Les techniques de représentation de la connaissance sont utilisées pour décrire une conceptualisation explicite et formelle pour le modèle du domaine et pour vérifier la consistance de l'ontologie enrichie.

Dans le prochain chapitre, nous nous concentrerons sur l'intégration des connaissances et les mesures de performance utilisées pour évaluer l'indexation et le repérage.

CHAPITRE III

INTÉGRATION DE LA CONNAISSANCE

Résumé: Dans le contexte de ce volet cognitif, le centre d'intérêt est d'explorer l'architecture d'intégration des ressources corporatives en intégrant les domaines du traitement automatique du langage naturel et la recherche d'information. La première phase de ce système d'intégration, celle de description des ressources, s'intéresse aux descripteurs par lesquels nous avons indexé les ressources. La deuxième phase d'intégration, autrement dit, la phase d'analyse, concerne le prétraitement et le nettoyage. Ainsi, sont réalisées des étapes telles que le découpage des jetons, la ponctuation, le dictionnaire négatif et la troncature. Enfin les troisième et quatrième phases se focalisent sur la structure d'indexation et l'algorithme de repérage. Les modules sont présentés dans l'ordre où ils se situent habituellement dans le système à partir de l'étape de prétraitement jusqu'à la fin de la requête de repérage.

La suite du chapitre est organisée de la manière suivante. Dans le paragraphe (3.1), nous décrivons d'abord l'architecture d'intégration et nous discutons les différents modules implémentés, en particulier, l'acquisition, l'indexation et le repérage. Nous détaillons dans le paragraphe (3.2) les étapes principales d'intégration de la connaissance. Le paragraphe (3.3) montre les mesures de performance utilisées pour évaluer la performance du repérage de la connaissance corporative. Ce présent chapitre ne se préoccupe pas des procédures de nature purement pratiques telles que l'hébergement, la plateforme/les outils d'implémentation et la sécurité informatique, etc.

3.1 Architecture d'intégration

Dans notre travail, nous nous concentrons sur des ressources textuelles non structurées. Ainsi, notre objectif est de rechercher une méthode pour indexer et repérer le contenu des ressources corporatives.
Il existe une vaste panoplie d'applications textuelles (DataparkSearch, Apache Lucene, Zettair, Xapian, etc.) et nous avons décidé d'utiliser le produit Lucene[2] créé par la fondation Apache (Hatcher, Gospodnetic et McCandless, 2004). Il s'agit d'un moteur de recherche textuelle Open Source contenant une librairie de routines logicielles intégrées qui peuvent être utilisés pour mieux développer des mécanismes d'intégration. Il peut facilement ajouter des capacités de recherche et d'indexation à notre application textuelle et nous pouvons développer rapidement le système d'intégration de la connaissance corporative.
La figure (3.1) représente schématiquement le processus fonctionnel du système avec les cas d'utilisation. L'utilisateur envoie la chaîne de la requête à l'interface du système et demande le repérage des ressources corporatives. Le cas d'utilisation *(Analyse)* traite la chaîne de la requête et génère une expression de repérage contenant les unités fondamentales de recherche. La chaîne de la requête subit plusieurs opérations, à savoir, le découpage du flux, la suppression des chiffres et les signes de ponctuation, la conversion des termes en lettres minuscules, la suppression des mots fonctionnels, la troncature, etc. Ces opérations de prétraitements permettent de donner une forte indication sur le contenu de la requête.
Le module de repérage recherche dans le fichier d'index et soumet les documents au programme de triage. Ce dernier trie les documents suivant leurs pertinences et soumet les ressources triées à l'interface de recherche. L'interface du système présente les documents aux utilisateurs selon leurs scores calculés en fonction des formats préprogrammés. L'administrateur de la mémoire corporative gère un index dynamique qui prend en charge l'ajout et la récupération des documents de l'index. Il utilise les outils de gestion de la mémoire corporative pour inspecter/pondérer/déverrouiller/optimiser un index et également pour supprimer et restaurer des documents. Le cas d'utilisation *(Visualisation)* présente à l'utilisateur une pagination personnalisée de substituts de documents pour simplifier l'exploration.
Le cas d'utilisation *(verrouillage/déverrouillage)* contrôle l'accès à l'index pour empêcher la corruption durant les mises à jour.

[2] https://lucene.apache.org/

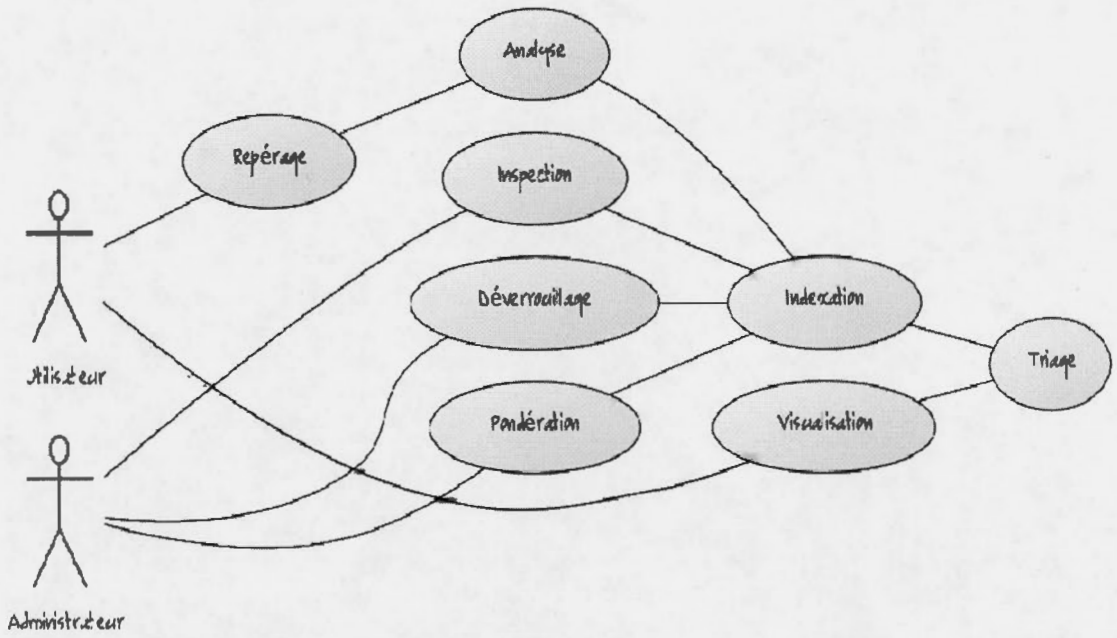


Figure 3.1 Le processus fonctionnel de la mémoire corporative

La figure (3.2) montre l'architecture de notre mémoire corporative composée de cinq modules: l'ontologie CRISP-DM-OWL, le module d'acquisition, le module d'indexation, le module de repérage et l'interface d'interaction.

Avant d'utiliser le module Data Mining fourni par l'application, l'utilisateur devra tout d'abord utiliser le module d'administration pour créer un index des documents contenus dans le corpus. Ceci est nécessaire pour mettre à jour le modèle d'indexation utilisé dans le module de repérage et dans le module d'apprentissage de l'ontologie. Au moment où l'indexation est créée le système devrait maintenir une version temporelle pour vérifier si le corpus a été modifié après l'indexation. Une fois cette étape complétée, l'utilisateur peut utiliser le module Data Mining pour enrichir l'ontologie. L'approche qui a été adoptée est la suivante: nous avons d'abord passé en revue chaque document dans le corpus puis nous avons enregistré les termes avec leurs poids dans un modèle d'indexation. Cette action sera répétée pour tous les documents ajoutés dans le corpus. Les indexes identifiés sont enregistrés dans une structure matricielle pour mettre à jour l'ontologie. Chaque ligne dans la matrice représente un vecteur d'apprentissage utilisé dans le processus d'apprentissage (l'étalonnage des modules Data Mining et le processus de maintenance de l'ontologie sont traités dans les chapitres IV et V).

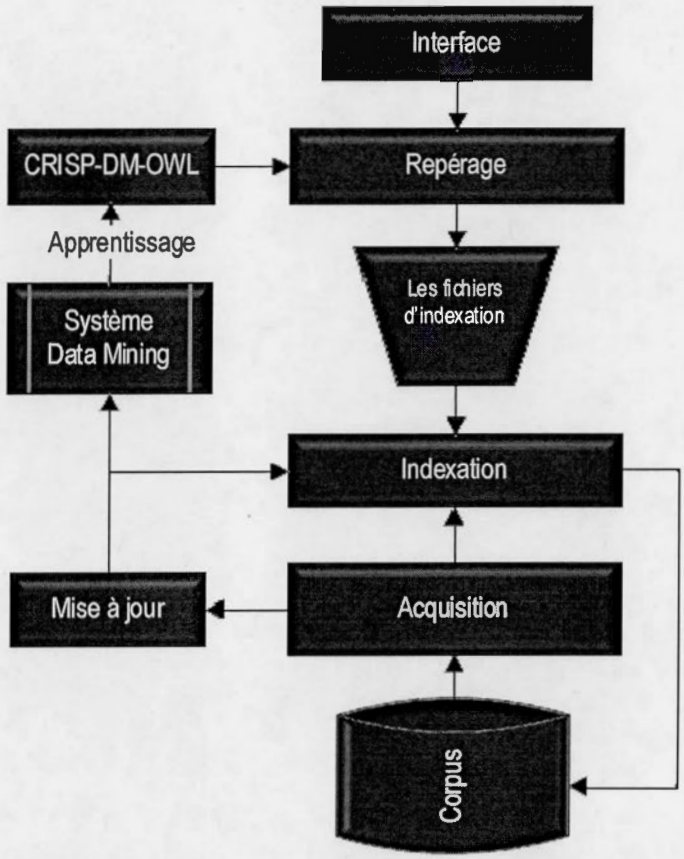


Figure 3.2 Architecture de la mémoire corporative (Djellali, 2013f)

L'ontologie CRISP-DM-OWL[3] utilisée dans le cadre de ce projet est intégrée dans un système hybride DM (Shen, 2007). Ce système est composé de sept éléments (figure 3.3):

- La base de cas DM: détient les informations sur les cas qui se présentent (les données de vérification de la qualité, les opérations de préparation des données, les paramètres du modèles, etc.). Chaque cas contient 53 variables, dont 15 sont indexées. Les variables permettent de représenter la description du problème, l'espace de solution et les sorties des activités. L'utilisation de la base de cas est consacrée aux objectifs suivant:
  - Éviter la dispersion de l'expertise en se concentrant sur la connaissance de tous les experts dans des cas spécifiques.
  - Permet d'avoir une base évolutive pour l'addition de nouveaux cas.

[3] http://www.elmanahel.ca/ontology/crisp-dm-owl.owl

- Permet d'utiliser le mécanisme de raisonnement à base de cas comme un mécanisme d'inférence.

- Une ontologie CRSIP-DM-OWL: décrivant les artefacts et les règles de base dans le système.
- Une interface d'assistance DM: le fonctionnement de l'assistant intelligent DM est engagé par une requête de l'utilisateur en spécifiant un problème DM.
- Un outil de raisonnement automatique à base de cas: le système de raisonnement à base de cas fournit un sous-ensemble de cas similaires aux cas spécifiés par l'utilisateur. Il offre un raisonnement de type inductif puisqu'il génère des comportements similaires à partir de situations similaires.

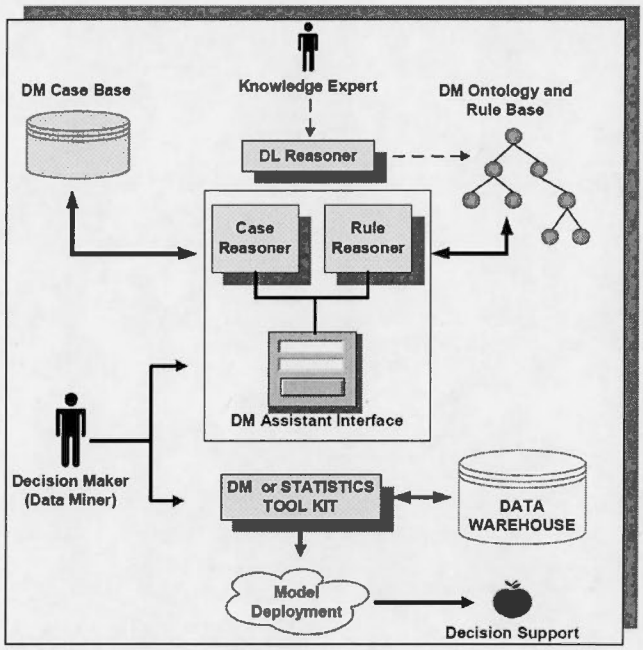


Figure 3.3 Le système hybride DM (Shen, 2007)

- Un modèle d'inférence basé sur les règles: pour contrôler la cohérence du système.
- Un modèle d'inférence basé sur des règles descriptives DL: le raisonnement basé sur la logique descriptive est strictement utilisé pour assurer la cohérence de l'ontologie. $O_{\text{CRISP-DM-OWL}}$.
- Un entrepôt de données: contient plusieurs bases de données fédérées composées de plusieurs tableaux, incluant les tableaux de faits et les tableaux de dimensions.

L'annexe (A) décrit brièvement l'ontologie CRISP-DM-OWL avec les artefacts impliqués dans chaque section. Pour simplifier la disposition des hiérarchies de concepts et pour permettre une meilleure

compréhension, nous montrons seulement quelques spécialisations. Le méta modèle OWL-DL de l'ontologie CRISP-DM-OWL est fourni dans l'annexe (B).

3.1.1 Cycle de vie de la mémoire corporative

La mémoire corporative peut être appréhendée à travers trois étapes principales représentées à la figure (3.4): acquisition, indexation et repérage, ces dernières peuvent être comparées aux phases de développement des systèmes à base de connaissances. L'étape d'acquisition est apparentée à la modélisation de la connaissance tandis que l'étape de l'indexation est assimilable à la représentation et la maintenance de la connaissance. Enfin, l'étape de repérage correspond à la validation des systèmes à base de connaissances.

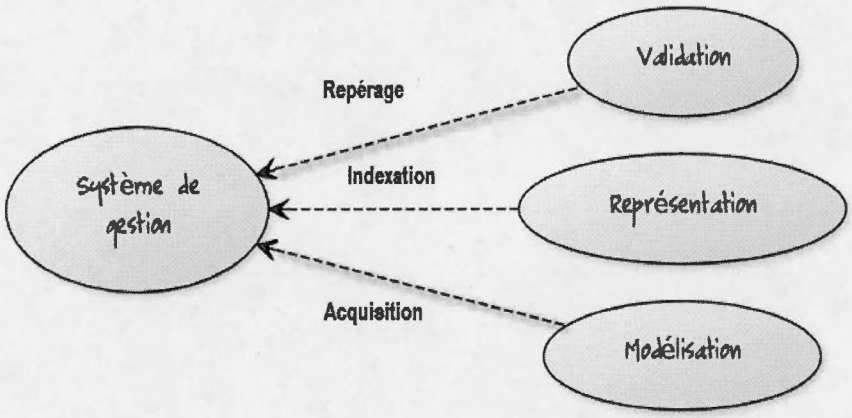


Figure 3.4 Cycle de vie du système de la mémoire corporative

a) Acquisition: comme montré à la figure (3.5), le cas d'utilisation *(Collecte)* est l'étape permettant de collecter les documents afin d'indexer les ressources corporatives.

La mémoire corporative peut avoir des changements en demandant à l'application de réindexer les nouvelles ressources et maintenir la cohérence ans la mémoire corporative. Le cas d'utilisation *(Mise à jour)* est principalement responsable de l'évolution des connaissances de manière à assurer la cohérence entre

l'index et les ressources corporatives (système d'information, ontologie, etc.). Ainsi, Ce cas d'utilisation devrait générer et maintenir l'index qui couvre l'ensemble des ressources de la mémoire corporative.

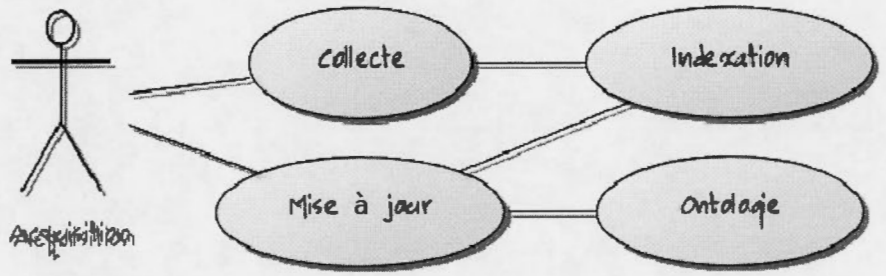


Figure 3.5 Le cas d'utilisation *«Acquisition»*

b) Indexation: comme montré à la figure (3.6), le cas d'utilisation *(Indexation)* prépare un modèle d'indexation pour la collection de documents textuels qui facilite le repérage des ressources corporatives.
L'objectif de cette étape est d'accroître la vitesse de recherche de la mémoire corporative, notamment pour les documents non structurés. Elle comprend la création et la fermeture des fichiers d'Index, la modification et l'ajout des documents à l'index. La création de l'index *(Ouvrir Index)* est le cas d'utilisation permettant d'ajouter le document dans les fichiers d'index utilisés pour extraire les connaissances pertinentes par le module de repérage et le système Data Mining. Dans le cas d'utilisation *(Analyse)*, le document est analysé pour produire un flux de jetons qui est ajouté à l'index dans une architecture segmentée. L'analyse permet également d'effectuer un certain nombre d'opérations optionnelles de prétraitement sur les jetons. Ces opérations diminuent le coût informatique des ressources et améliorent de manière significative le repérage.
Le cas d'utilisation *(Champs)* est utilisé pour indexer les champs des documents textuels. Il permet de spécifier de nombreuses options pour contrôler le repérage durant l'analyse de la requête. Chaque document contient un ensemble de champs interrogés pendant le repérage.
Le cas d'utilisation *(Pondération)* est une étape importante utilisée pour changer le facteur de pondération d'un document. Ainsi, le processus de repérage peut examiner la pertinence des documents récupérés.
L'indexation est soumise à la problématique d'évolution des ressources corporatives. L'administrateur peut mettre à jour l'index et réindexer les documents modifiés. Tout comme les documents, l'administrateur peut également re-construire/modifier/pondérer les champs des documents. Il utilise les outils d'administration système pour inspecter et modifier le contenu de l'indexation de plusieurs façons, à savoir, la suppression

sélective des documents, l'analyse des résultats de la recherche, l'optimisation d'indexation, la récupération et le filtrage des termes les plus fréquents, etc.

Dans le cas d'utilisation *(Optimisation)*, le processus d'indexation fusionne plusieurs segments pour générer un index optimisé qui consomme moins de descripteurs de fichiers et moins de ressources computationnelles durant le processus de repérage.

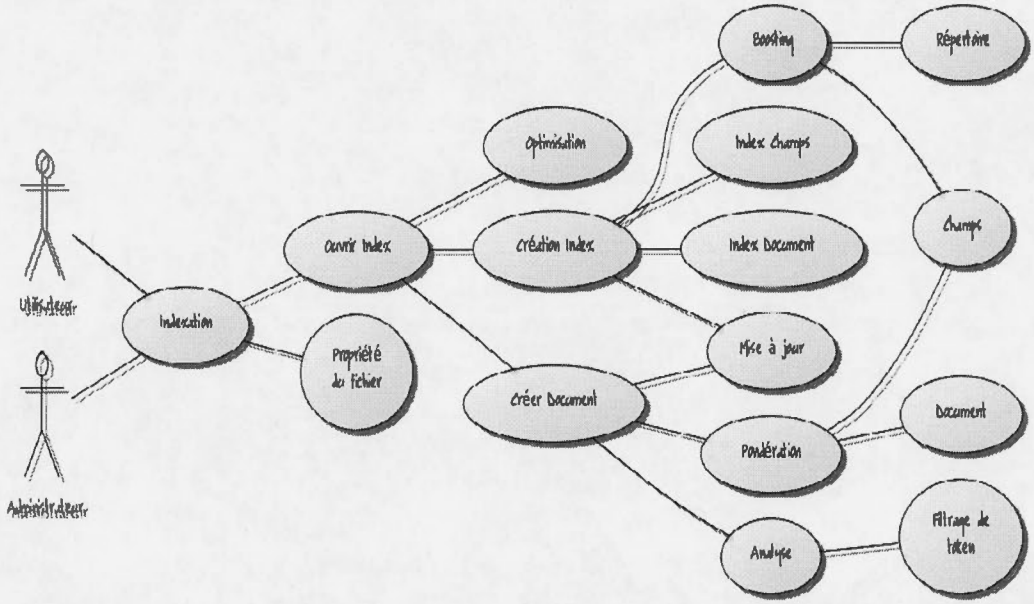


Figure 3.6 Le cas d'utilisation *«Indexation»*

c) Repérage: comme montré à la figure (3.7), le cas d'utilisation *(Repérage)* prend en charge plusieurs processus de recherche avancée telles que la recherche par terme spécifique, intervalle (numérique ou textuelle), préfixe ou générique, phrases, ou par correspondance floue des termes en utilisant des contraintes booléennes.

Lorsque l'utilisateur soumet une requête à la mémoire corporative, le système transmet d'abord la chaîne de la requête saisie à l'explorateur de requêtes. Ainsi, le cas d'utilisation *(Exploration)* contrôle plusieurs paramètres, en particulier, l'opérateur logique par défaut lorsque plusieurs termes sont utilisés, l'analyse de la date, la similitude minimale et la longueur de préfixe pour les requêtes floues, la résolution de la date, les requêtes génériques et diverse autres paramètres avancés. Pour repérer les mots clé pertinents, le cas d'utilisation *(Analyse)* traite les chaînes générées par plusieurs processus, en particulier, le découpage des jetons, la lemmatisation, la troncature, la suppression des mots fonctionnels, etc. Les différent processus impliqués dans l'analyse génèrent une expression composée de mots-clés valides. L'expression de la requête est transmise au module de repérage qui récupère les documents recherchés dans les fichiers

d'index selon l'expression de la requête et soumet les résultats au programme de triage. Les documents triés sont présentés aux utilisateurs par l'interface du système.

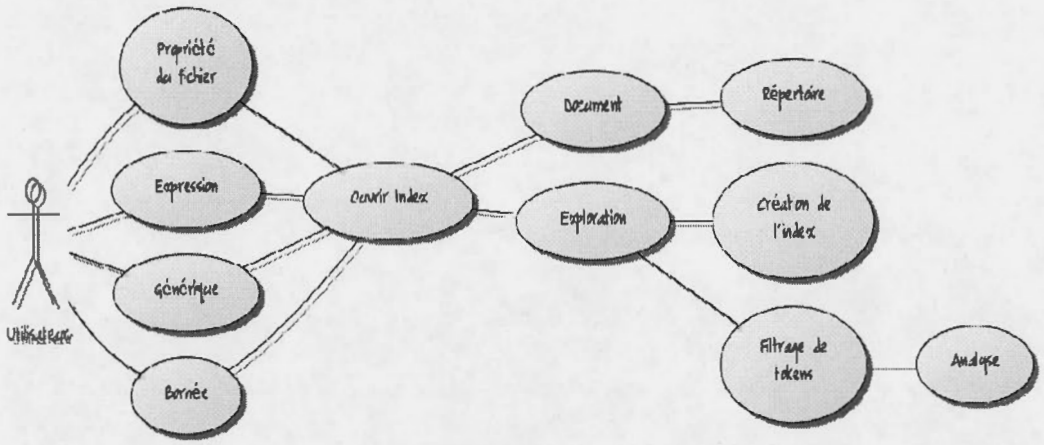


Figure 3.7 Le cas d'utilisation *«Repérage»*

Le prochain paragraphe explique non seulement comment utiliser Apache Lucene pour la construction d'un système d'intégration de la connaissance corporative, mais aussi des détails liés à l'ajustement des performances d'indexation.

3.2 Le processus d'intégration

Le processus d'intégration de la connaissance corporative implique cinq étapes principales:

- *La description des ressources*: le repérage dans la mémoire corporative est réalisé en précisant les mots-clés et/ou les champs de recherche. Les éléments par lesquels nous avons indexé les documents sont les mots qui se trouvent dans les documents. La structure du document associé est composée de cinq champs comme montré à la figure (3.8).

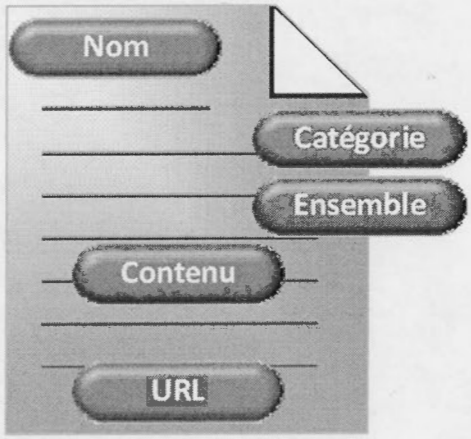


Figure 3.8 Représentation du document (Djellali, 2013g)

a) Nom: contient la collection de termes dans le nom local.
b) Catégorie: pour chaque ensemble de données, un index est construit et chaque document est étiqueté suivant sa catégorie.
c) Ensemble: décrit l'appartenance du document à l'ensemble de données (apprentissage ou test).
d) Contenu: le contenu de chaque document est représenté par un vecteur.
e) URL: le chemin d'accès au document dans le serveur local.

- *L'analyse du texte*: afin de mesurer la performance du système, nous avons utilisé des fichiers textuels non structurés à partir d'un corpus divisé en deux parties, une partie pour l'apprentissage et l'autre pour le test. Le corpus consiste en un ensemble de résumés des articles I.E.E.E. répartis dans plusieurs catégories. L'échantillon de données pré-classées contient 835 documents dans la base d'apprentissage et 283 documents dans l'ensemble de test. Le nombre de document dans chaque catégorie est fortement déséquilibré.

  Le système expérimental utilisé pour évaluer la capacité du système d'intégration a été exécuté sur un ordinateur avec un processeur Intel (R) Core™ 2, 6600 @ 2.40 GHz 2.39 GHz, un système d'exploitation Windows 7 Professionnel © 2009 64 bits et une mémoire (RAM) 4.00 Go. Le système est développé avec le langage Java sous l'environnement de développement intégré Eclipse JEE Juno 64 bits avec les serveurs MySQL et Apache Tomcat 7.0.29 et certaines bibliothèques de

fonctions telles que: JDK 7u3 + Java EE, Lucene 4.0-BETA, JAMA (de l'anglais: Java Matrix Package), etc.

Après le prétraitement des documents par le découpage des jetons, la ponctuation, le dictionnaire négatif et la troncature, il reste 59240 termes représentant le vocabulaire dans la base d'apprentissage et 19145 termes dans la base de test.

La figure (3.9) et la figure (3.10) montrent le déroulement de l'analyse à travers les documents dans l'ensemble d'apprentissage et l'ensemble de test. L'axe horizontal énumère les documents dans l'ensemble d'apprentissage (test) et l'axe vertical indique le vocabulaire accumulé pour chaque catégorie (Ponctuation), (Mot fonctionnel), (Lexème) et (Vocabulaire). (Zhou et Dieng-Kuntz, 2004).

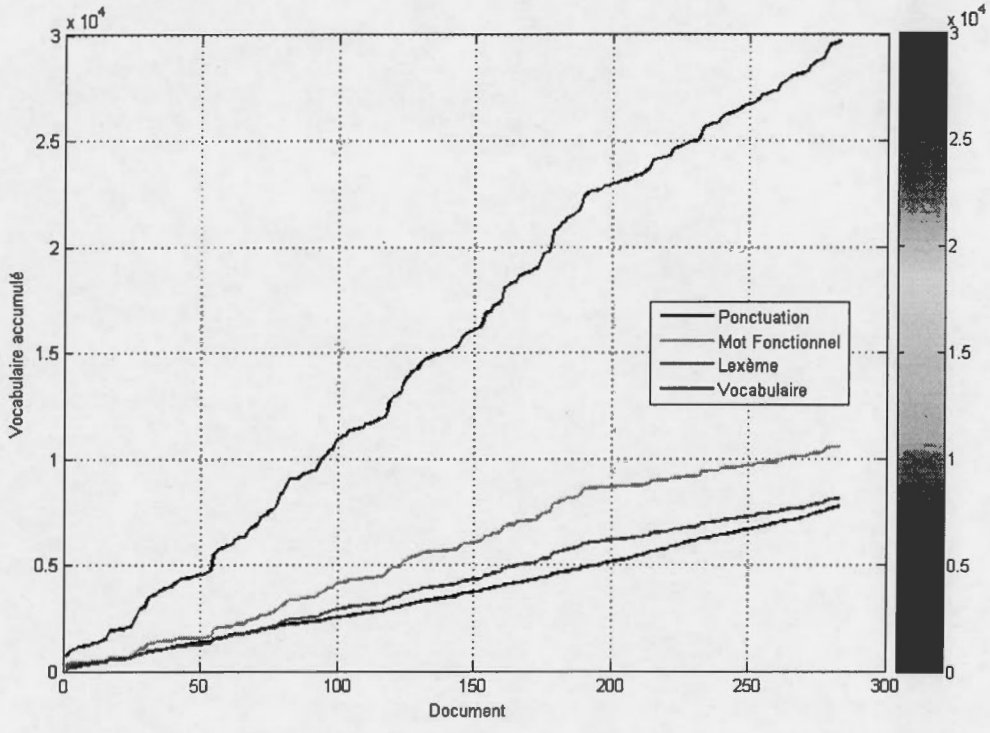


Figure 3.9 L'analyse de l'échantillon de test

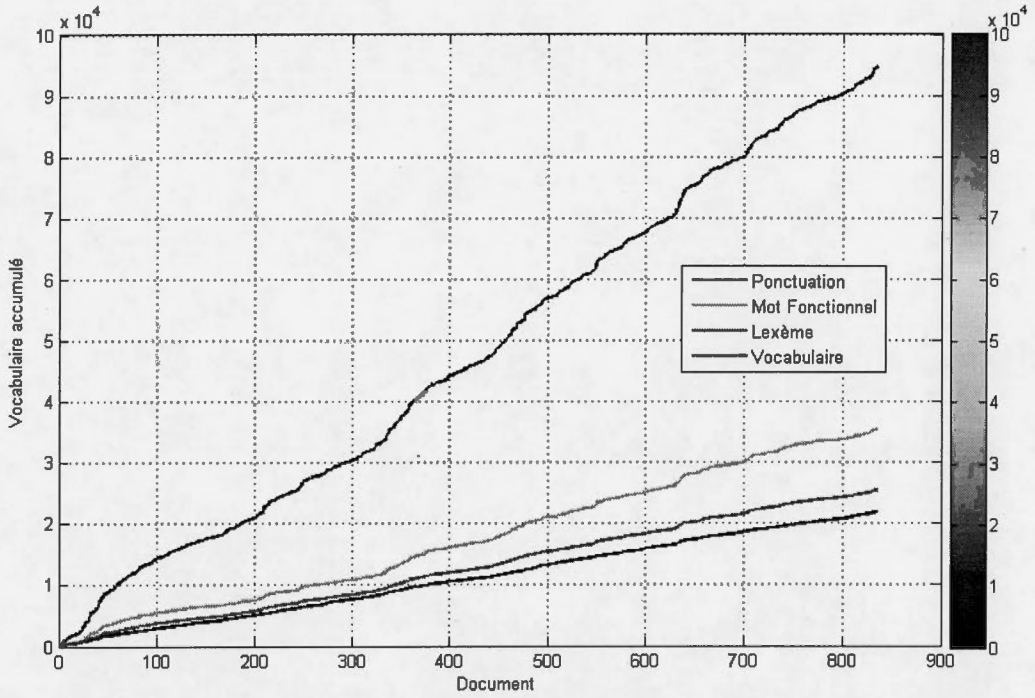


Figure 3.10 L'analyse de l'échantillon d'apprentissage

- *Structure d'indexation*: pour représenter le document textuel, nous avons utilisé le modèle vectoriel (VSM: Vector Space Model) (Salton, Wong et Yang, 1975). Dans cette représentation vectorielle, un ensemble de termes sont choisis comme des mots clés pour indexer la base d'apprentissage et la base de test. Les termes sont calculés en utilisant la méthode de la fréquence des termes et la fréquence inverse du document:

$$tfidf(t_k, d_j) = \#(t_k, d_j).\log\frac{|T_r|}{\#_{T_r}(t_k)}$$

Afin d'améliorer les performances d'indexation des ressources corporatives, nous avons analysé deux méthodes d'indexation: l'indexation composée et l'indexation multifichiers (Hatcher, Gospodnetic et McCandless, 2004). La figure (3.11) et la figure (3.12) montrent le traitement de l'indexation des documents dans l'ensemble d'apprentissage et l'ensemble de test. L'axe horizontal représente le document et l'axe vertical indique le temps d'indexation en millisecondes. Le temps d'indexation de la base de test égale à 138 millisecondes avec la méthode composée et 165 millisecondes en utilisant la méthode d'indexation multifichiers.

Pour indexer la base d'apprentissage, le temps d'exécution égale à 247 millisecondes durant l'indexation composée et 299 millisecondes durant la méthode d'indexation multifichiers.

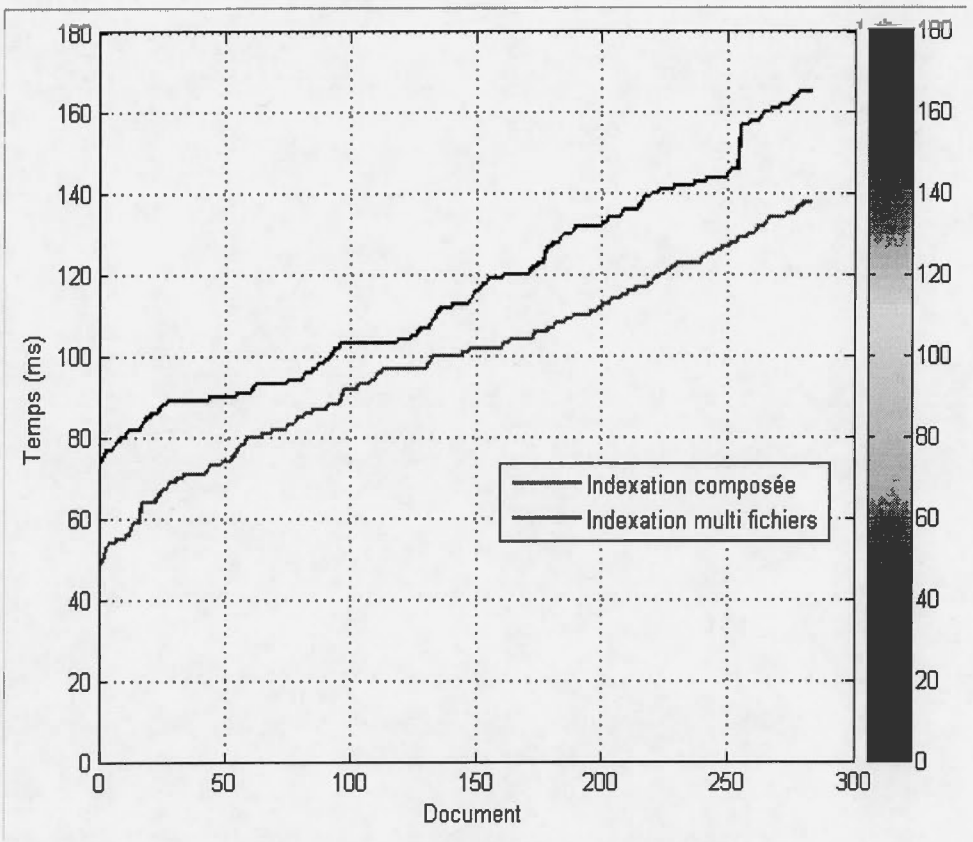


Figure 3.11 L'indexation de la base de test

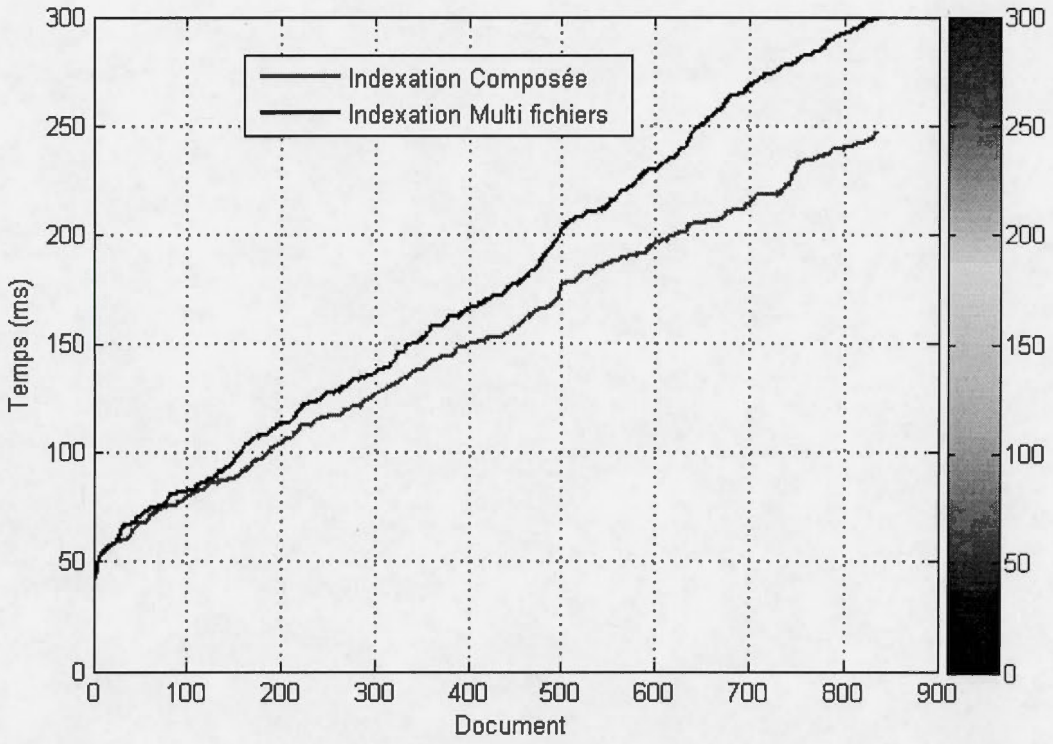


Figure 3.12 L'indexation de la base d'apprentissage

L'indexation multifichiers consomme plus de temps parce qu'elle stocke de nombreux fichiers séparés par segment. Comparée à l'indexation multifichiers, l'indexation composée réduit le nombre de fichiers d'index parce que la structure composée encapsule les fichiers individuels d'index dans un seul fichier composé, de ce fait elle donne des réponses rapides (ANNEXE C).

Les figures (3.13) et (3.14) montrent l'allocation de la mémoire (RAM) durant l'indexation composée et l'indexation multifichiers.

En raison de la création de nouveaux segments chaque fois que les documents sont ajoutés à l'index, il y aura un nombre variable de fichiers dans la mémoire avec les deux méthodes.

Pour indexer le corpus d'apprentissage, l'allocation de la mémoire libre durant indexation multi-fichier égale à 3447.53 kilo-octets et 7438.28 kilo-octets durant l'indexation composée. Pour l'ensemble de test, l'allocation de la mémoire libre égale à 1026.89 kilo-octets durant l'indexation multifichiers et 1274.89 kilo-octets durant l'indexation composée.

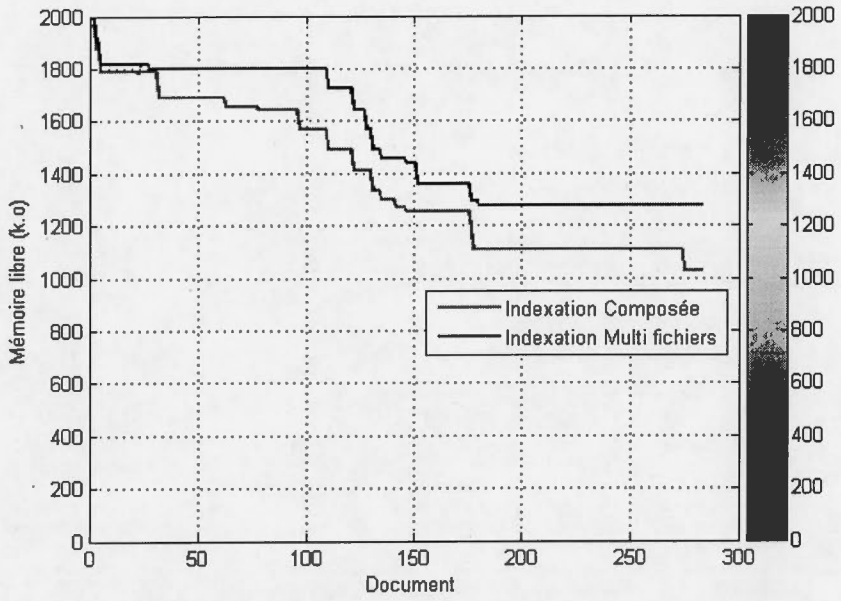


Figure 3.13 L'allocation de la mémoire de la base de test

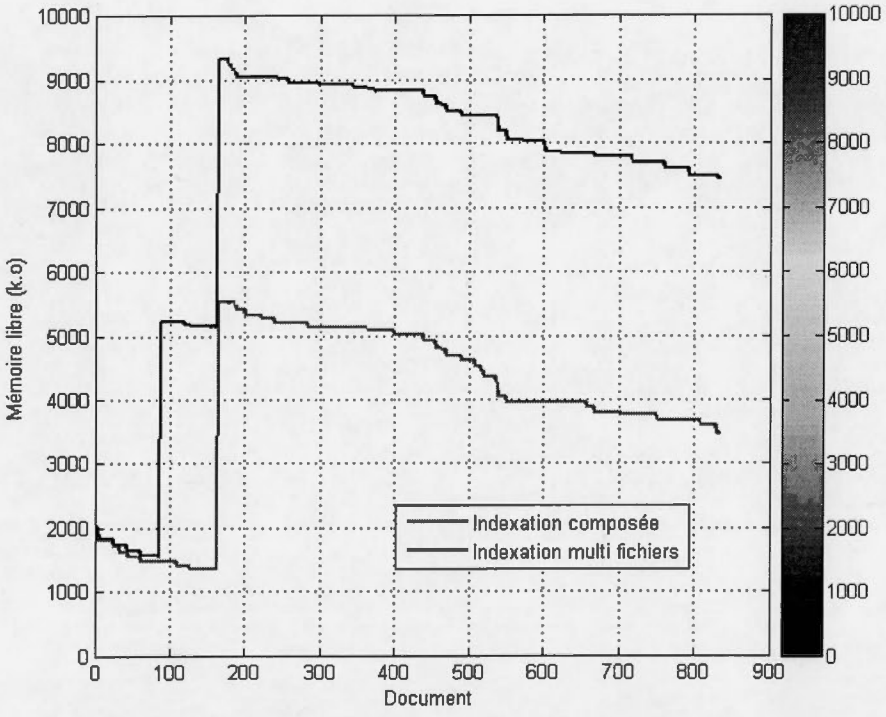


Figure 3.14 L'allocation de la mémoire de la base d'apprentissage

L'indexation composée réduit le nombre de fichiers d'index ouverts dans la mémoire parce qu'elle contient un seul fichier composé par segment. Cette structure optimisée consomme moins de descripteurs de fichiers et moins de ressources computationnelles durant le processus d'indexation. Cependant, chaque segment d'un index multifichiers se compose de plusieurs fichiers différents. De ce fait, l'indexation multi- fichiers n'améliore pas les performances et est coûteuse.

- L'enregistrement du texte dans un index: le système d'intégration permet de créer deux types d'index, le premier est créé dans la mémoire principale et l'autre est conservé sur le disque dur. Le module d'indexation fournit également deux protocoles d'accès, le premier offre un accès pour la récupération et le deuxième fournit des services pour le maintien de l'index.

  La structure de l'index est constituée de cinq classes de données:

  a) Le répertoire: toutes les données d'indexation sont enregistrées dans un répertoire.
  b) Le segment: un index comprend de nombreux segments indépendants qui peuvent être fusionnés.
  c) Le document: un document se compose d'une collection de champs.
  d) Le champ: comprend de nombreux types d'informations.
  e) Le terme: un terme est la plus petite pièce de l'index qui consiste en un nom de champ et une paire (texte-valeur) (ANNEXE D).

Le module d'indexation stocke l'entrée dans une structure composée représentée à la figure (3.15). Cette structure optimisée de l'index inversé permet une utilisation efficace de l'espace disque, une recherche par mot-clé et des réponses rapides. Chaque segment est un index autonome contenant un sous-ensemble de documents indexés.

Le fichier composé _X.cfs regroupe les fichiers d'index (_X.fdt, _X.fdx, _X.fnm, _X.frq, etc.) pour réduire le nombre de descripteurs de fichiers ouverts pendant l'accès et le repérage. De ce fait, les performances de recherche et d'indexation sont améliorées. Grâce à la fonction de hachage, chaque terme produit une seule valeur qui est stockée dans l'index.

Le module d'indexation sélectionne et fusionne les segments pour optimiser la structure de l'index.

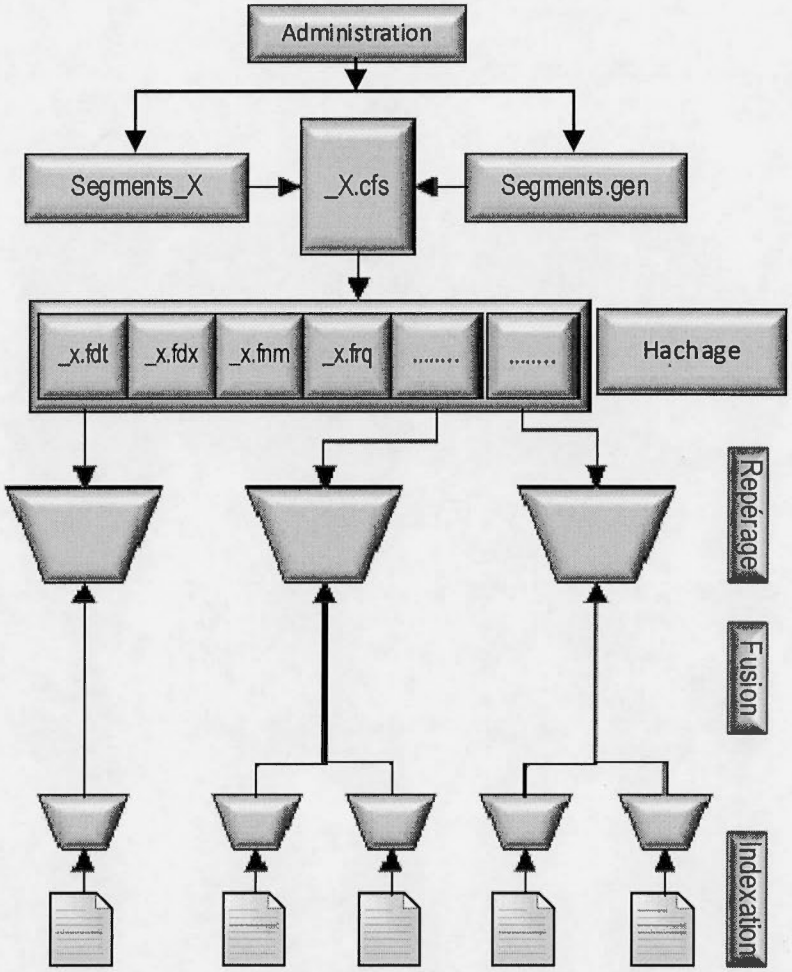


Figure 3.15 L'indexation composée

Dans l'indexation composée présentée au tableau 3.2, l'index de la base d'apprentissage (test) contient un seul segment « _0 » dont le nom est stocké dans le fichier « Segments_2 », ainsi le module d'indexation repère uniquement les fichiers avec le préfix _0. Le module d'indexation incrémente la génération (_2) du fichier des segments « Segments_2 » chaque mise à jour de l'index. Chaque index (d'apprentissage/test) contient un fichier unique de validation « *Segments.gen* » qui référence les segments courants pour déterminer la dernière mise à jour. Le module d'indexation consulte ce fichier pour déterminer les fichiers d'index liés aux instructions Ouverture/Lecture (Hatcher, Gospodnetic et McCandless, 2004) .

Tableau 3.1 Description de l'index

Base	Fichier	Contenu	Taille (k.o)	Accès	Rôle
Apprentissage	_0.cfs	_0.fdt, _0.fdx, ...	2741	A (L/E), U (L)	Indexation
	Segments.gen	copie	1	A (L/E), U (L)	Redondance et changements
	Segments_2	Les segments	1	A (L/E), U (L)	Repérage des fichiers d'index
Test	_0.cfs	_0.fdt, _0.fdx,...	1083	A (L/E), U (L)	Indexation
	Segments.gen	copie	1	A (L/E), U (L)	Redondance et changements
	Segments_2	Les segments	1	A (L/E), U (L)	Repérage des fichiers d'index

A: Administrateur, U: Utilisateur, L: Lecture, E: Écriture

L'administrateur de la mémoire corporative utilise les interfaces graphiques des outils système (Luke, LIMO, Hadoop, Zipf, etc.) pour inspecter tous les détails de l'index à partir des applications Desktop ou des applications en ligne.

Les différentes vues d'inspections générées par les outils système montrent les pièces majeures de l'index, y compris, la taille de l'index, le nombre de champs/documents/les termes, les fréquences, le repérage multicritères et les détails de la pondération. Les annexes (E,F,G) décrivent les outils système et les différentes vues d'inspections.

- Repérage: l'algorithme de récupération est basé sur le modèle de l'espace vectoriel (VSM). Le document et la requête sont représentés par deux vecteurs *tfidf* . De ce fait, la similitude entre le document et la requête est calculée par le cosinus de l'angle compris entre les deux vecteurs:

$$sim(d_i,q) = \frac{d_i \cdot q}{\|d_i\| \cdot \|q\|} = \frac{\sum_{j=1}^{N} tfidf_{ij} \cdot w_{qj}}{\sqrt{\sum_{j=1}^{N} (tfidf_{ij})^2} \cdot \sqrt{\sum_{j=1}^{N} (w_{qj})^2}} ; \ \forall d_i \in D$$

N : la taille du vecteur ; D : l'ensemble de document ; q : requête.

Un score est attribué à chaque document correspondant à une requête reflétant le degré de similitude en utilisant la formule suivante:

$$\sum_{t.q} \alpha(t,d) \cdot \beta(t,d,q)$$

$$\alpha(t,d) = (tf(t,d) \cdot idf(t)^2 \cdot boost(t.champs,d)$$

$$\beta(t,d,q) = LenghtNorm(t.champs,d).coord(q,d) \cdot queryNorm(q)$$

$$\forall t \in T, \forall d \in D, q \in Q$$

$tf(t,d)$: la fréquence du terme t dans le document d .

$idf(t)$: la fréquence du terme t dans l'index inversé.

$boost(t.champs,d)$: le facteur de pondération du champ (traité durant l'indexation).

$LengthNorm(t.champs,d)$: la valeur normalisée du champ calculée durant le processus d'indexation, c'est-à-dire, le nombre de termes dans le champ.

$coord(q,d)$: le facteur de coordination basé sur le nombre de termes de la requête présents dans le document.

$queryNorm(q)$: la valeur de normalisation pour une requête, c'est-à-dire, la somme des carrés des poids de chaque terme de la requête.

Les facteurs de pondération permettent de modifier la requête et les scores des champs. La pondération est fixée durant le mécanisme d'indexation par des facteurs explicites $boost(t.champs,d)$.

Le module de repérage prend en charge plusieurs types de recherche avancée telles que: la recherche bornée, la recherche générique, la recherche par expression et la recherche par terme.

Ces processus de recherche sont spécifiés dans plusieurs APIs, en particulier, Phrase Query, Fuzzy Query, Prefix-Query, Range Query, Filtered Query, Boolean Query, etc (Hatcher, Gospodnetic et McCandless, 2004).

3.3 Évaluation

Afin d'évaluer l'efficacité de la récupération de la mémoire corporative, nous avons utilisé les mesures de la précision, le rappel et l'indice équilibré F-mesure qui sont largement utilisés dans la recherche et le développement des systèmes de traitement du langage naturel, le repérage Web et l'apprentissage machine.

- La précision p est calculée en divisant le nombre de documents pertinents récupérés par le nombre total de documents retrouvés.

$$p = \frac{|\{\text{documents pertinents}\} \cap \{\text{documents retrouvés}\}|}{|\{\text{documents retrouvés}\}|}$$

$$= \frac{|\text{document correctement classé dans la catégorie x}|}{|\text{documents appartenant à la catégorie x}|} = \frac{tp}{tp+fp}$$

- Le rappel r est calculé en divisant le nombre de documents pertinents retrouvés par le nombre de documents pertinents disponibles dans la base de document.

$$r = \frac{|\{\text{documents pertinents}\} \cap \{\text{documents retrouvés}\}|}{|\{\text{documents pertinents}\}|}$$

$$= \frac{|\text{document correctement classé dans la catégorie x}|}{|\text{documents attribués à la catégorie x}|} = \frac{tp}{tp + fn}$$

tp : vrai positif ; fp : faux positif ; fn : faux négatif.

- L'indice équilibré F-mesure négocie le compromis de la précision par rapport au rappel, défini par la formule suivante: $\text{F - mesure} = \frac{2 \cdot p \cdot r}{p + r}$

Comme présenté dans le tableau 4.2, les expériences montrent que le système dispose d'une bonne performance de récupération, ce qui peut fournir un système d'intégration efficace pour les utilisateurs (Djellali, 2013f).

Tableau 3.2 L'efficacité du repérage

Précision	*Rappel*	*F-mesure*
0.56	0.91	0.69

Le système d'intégration offre plusieurs avantages, il s'agit notamment de la vitesse de recherche due à la structuration et l'optimisation d'indexation.

Conclusion

Nous avons conçu et mis en œuvre dans ce chapitre un système d'intégration qui se base sur l'analyse, l'indexation et le repérage. L'indexation et le repérage sont dérivés statistiquement par la cooccurrence des termes dans les documents et les requêtes. L'indexation consiste à attribuer les termes d'indexation aux ressources textuelles non structurées disponibles dans la mémoire corporative. Ces termes sont ensuite utilisés pour accéder aux ressources en utilisant une table indexée représentée par un fichier inversé. L'indexation composée a été utilisée pour réduire le nombre de fichiers d'index ouverts dans la mémoire. De ce fait, elle améliore les performances et elle n'est pas coûteuse.

Toutes les connaissances dans la mémoire corporative sont liées à une ontologie globale. Cette dernière décrit les artefacts et les règles de base pour améliorer le niveau d'intelligence du système. Elle agit comme une source de connaissances complémentaire dans le système. La communication entre les composantes de la mémoire corporative est basée sur les constructeurs de l'ontologie. Un des principaux avantages offerts par le système d'intégration est la construction d'analyseur de requêtes permettant de spécifier et de combiner des chaînes de requêtes complexes en toute simplicité. Cette fonctionnalité était un ingrédient clé dans le repérage de documents dans la mémoire corporative.

Dès que l'on s'intéresse à l'intégration des connaissances par les ontologies, il est pertinent d'examiner les inconvénients qui ont été observés dans la littérature, en particulier, l'évolution des connaissances. L'environnement dynamique de la mémoire corporative exige l'évolution et l'amélioration de la connaissance pour s'assurer qu'elle reflète le domaine d'intérêt. Il est nécessaire de mettre à jour périodiquement l'ontologie. Ceci est faisable parce que le système d'intégration soutient des mécanismes pour retrouver et détecter les changements des ressources corporatives.

Le processus de l'évolution exige la définition d'un cycle de vie qui s'étend de l'extraction des connaissances à la maintenance, aussi bien que des techniques qui pilotent le processus de mise à jour, le traitement du changement et la consistance. Dans le prochain chapitre, nous examinerons plus en détail l'étalonnage des modules de Data Mining pour la sélection des variables pertinentes utilisées dans le processus d'apprentissage.

CHAPITRE IV

PRÉTRAITEMENT ET SÉLECTION DES VARIABLES

Résumé: Les variables pertinentes ne peuvent être trouvées facilement. Par conséquent, un problème évident pour n'importe quelle approche méthodologique de sélection des variables est de fournir une stratégie de recherche et un critère d'évaluation des performances.

Ce chapitre présente, dans un premier temps, l'étape de prétraitement permettant d'identifier la distribution et le niveau de bruit. Dans une deuxième partie de ce chapitre, nous présentons le processus typique de la sélection des variables avec un panorama des stratégies de recherche. Nous présentons ensuite un état de l'art des quelques travaux qui émergent dans le domaine de sélection des variables. Ceci nous permet de développer ainsi les trois phases du processus de sélection des variables que nous avons défini autour d'un modèle d'emballage, à savoir, la génération des variables, l'évaluation et le critère d'arrêt.

Dans la suite de ce chapitre, nous présentons dans le paragraphe (4.1) la distribution et le niveau de bruit dans le corpus d'apprentissage. Nous discutons dans le paragraphe (4.2) une vue d'ensemble sur le processus de la sélection des variables, les stratégies de recherche et les critères d'évaluation. Dans le paragraphe (4.3), nous décrivons les modèles et les différents algorithmes de la sélection des variables. Nous discutons dans le paragraphe (4.4) de la manière de traiter la malédiction de la dimensionnalité et le critère d'évaluation choisi pour guider le processus de recherche dans la tâche de sélection des variables pertinentes utilisées dans le processus d'apprentissage.

4.1 Prétraitement

Les données bruitées peuvent confondre un algorithme d'apprentissage en brouillant les frontières de décision. L'impact du bruit sur la performance de la classification est déterminé par deux facteurs: la distribution et le niveau de bruit dans les données.

4.1.1 La distribution du bruit

Le filtrage des mots fonctionnels et la troncature sont deux méthodes de prétraitement communément utilisées pour supprimer le bruit.

a) *Le filtrage des mots fonctionnels:* nous avons utilisé la liste Glasgow[4] (Zaman, Matsakis et Brown, 2011) comme une liste de mots fonctionnels dans nos expériences. Cette liste est sans doute la liste des mots fonctionnels les plus utilisés, elle en couvre 351 mots fonctionnels. Le sac de mots (de l'anglais: bag of words) est déterminé avec une approche de génération semi-automatique en classant tous les termes par leurs occurrences, en sélectionnant les termes qui ont des fréquences plus élevées que le seuil prédéterminé, puis la suppression automatique de certains noms communs tels que *"abstract", "section", "introduction", "conclusion","perspective", "work", "future",* etc. Chaque terme est haché dans une table contenant la liste des mots fonctionnels. Si l'emplacement résultant est vide, le terme n'est pas un mot fonctionnel. Les comparaisons doivent être faites pour déterminer si la valeur hachée correspond vraiment à une entrée dans la table de hachage.

b) La troncature: la plupart des approches de la troncature sont basées sur des règles morphologiques de la langue cible. La suppression du suffixe est contrôlée par des restrictions quantitatives ou qualitatives. Certaines règles d'orthographe ad hoc peuvent être appliquées pour améliorer la précision. Les approches basées sur les corpus reflètent l'aspect syntaxique plutôt que l'utilisation des règles grammaticales. Les approches hybrides (Bhamidipati et Pal, 2007) permettent de supprimer les suffixes dérivés, habituellement utilisés pour générer une nouvelle partie du terme à partir d'un lexème donné.

Il existe une difficulté inhérente à la réalisation d'un algorithme de troncature. La plupart des algorithmes génèrent un ensemble de lexèmes avec un rappel élevé et une précision faible. Parmi plusieurs implémentations de l'algorithme de Porter (Issac et Jap, 2009), nous avons choisi la version qui a été publiée par (Gonzalo Parra, 2006). Cette version a l'avantage d'une séparation claire entre les règles de substitution et les procédures qui testent les conditions attachées à un lexème particulier. Elle est basée sur une série de mesures permettant de supprimer les suffixes par des règles de substitution. Ces règles ne

[4] http://ir.dcs.gla.ac.uk/resources/linguistic_utils/stop_words

s'appliquent que lorsque certaines conditions sont satisfaites. Les conditions permettent de quantifier le nombre de séquences voyelle-consonne qui sont présentes dans un lexème donné.

4.1.2 Le niveau du bruit

Nos données ont été préparées comme décrit dans la section (3.2). Ainsi, après l'exécution des outils de prétraitement, nous avons obtenu la distribution statistique comme illustré dans la Figure (4.1).

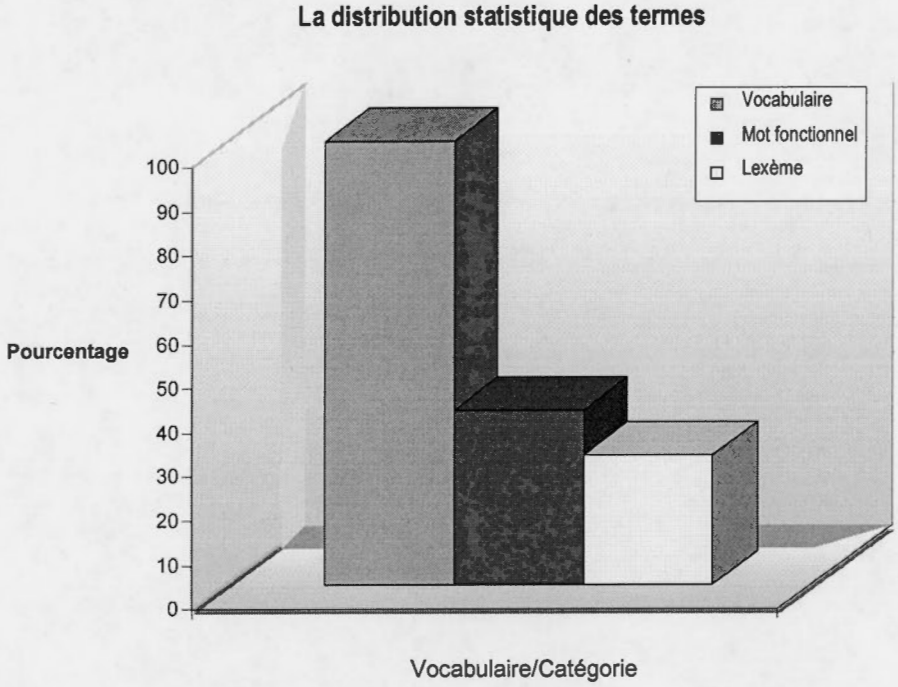


Figure 4.1 La plage des classes représentant le vocabulaire

Le graphique en histogramme empilé (100 %) illustre les changements affectant le corpus d'apprentissage après l'exécution des outils de prétraitement. Il fournit des renseignements sur la distribution statistique des termes en comparant la proportion dans laquelle chaque valeur contribue à un total dans chaque catégorie. Il illustre le rapport du nombre des termes représentant le vocabulaire (le mot fonctionnel, lexème) sur le nombre total de termes dans le corpus d'apprentissage. Les valeurs des séries sont affichées sous la forme d'un pourcentage de chaque catégorie *(Vocabulaire), (Mot fonctionnel) et (Lexème).*

Les documents type IEEE 830 (ANNEXE J-K) illustrent les comportements des outils de prétraitement implémentés dans le système.

Le paragraphe qui suit présente le processus généralisé de la sélection des variables, les critères d'évaluation ainsi que les stratégies de recherche.

4.2 Sélection des variables

Il y a eu une reprise d'intérêt pour les méthodes de sélection des variables où le but est de récupérer les variables pertinentes à partir de l'ensemble d'apprentissage. De ce besoin est apparue la sélection des variables et elle a eu une répercussion importante aussi bien dans le monde industriel que dans la communauté scientifique. Elle est étudiée dans plusieurs champs de recherche, en particulier, la reconnaissance des formes, l'apprentissage machine, le repérage d'information et le Data Mining.

La sélection des variables est définie de la façon suivante:

« The main idea of feature selection is to choose a subset of input variables by eliminating features with little or no predictive information » (T.Deepa, 2012) .

Comme montré à la figure (4.2), la sélection des variables cherche à projeter l'espace des variables dans un espace réduit issu d'une combinaison des variables originales, tel que les variables sélectionnées englobent un maximum de variances du nuage de points autour de cet espace. La sélection des variables permet d''éliminer les variables redondantes ou moins informatives, de sorte que le sous-ensemble choisi ne contienne que les variables pertinentes. Ces dernières sont celles qui ont la plus grande influence sur la position du vecteur document dans l'espace

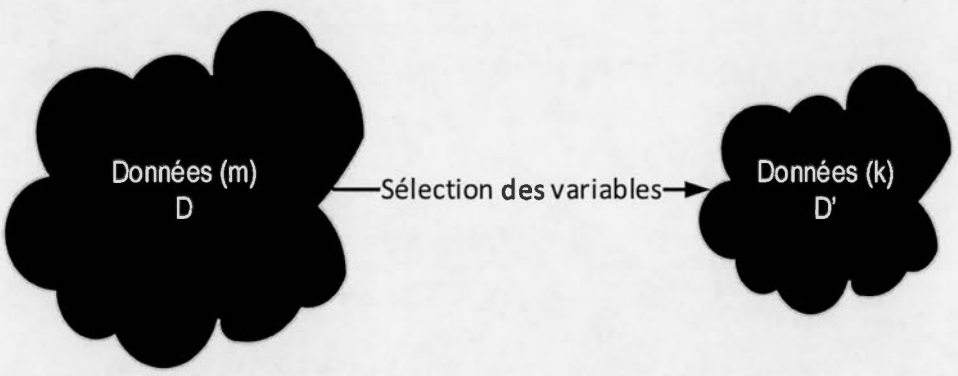


Figure 4.2 La projection de l'espace de représentation

Formellement, la sélection des variables est définie comme un problème d'optimisation combinatoire permettant de trouver un sous ensemble D' à partir de l'ensemble des variables originales D, tels que:

$$\begin{cases} |D'| = k \\ \wedge \\ J(|D'|) = \max_{Z \subseteq D, |Z| = k} J(Z) \\ k \prec m \end{cases} \quad (1)$$

D : ensemble de variables , $|D| = m$

J : critère monotone de sélection

k :taille de l'espace réduit.

Les k variables sélectionnées peuvent être considérés en tant que combinaisons des m variables originales et expliquent une proportion maximale de la variance.

En général, le processus de la sélection des variables se compose de quatre phases de base (montrées à la figure (4.3)), que sont la génération d'un sous-ensemble de variables, l'évaluation, le critère d'arrêt et la validation des résultats.

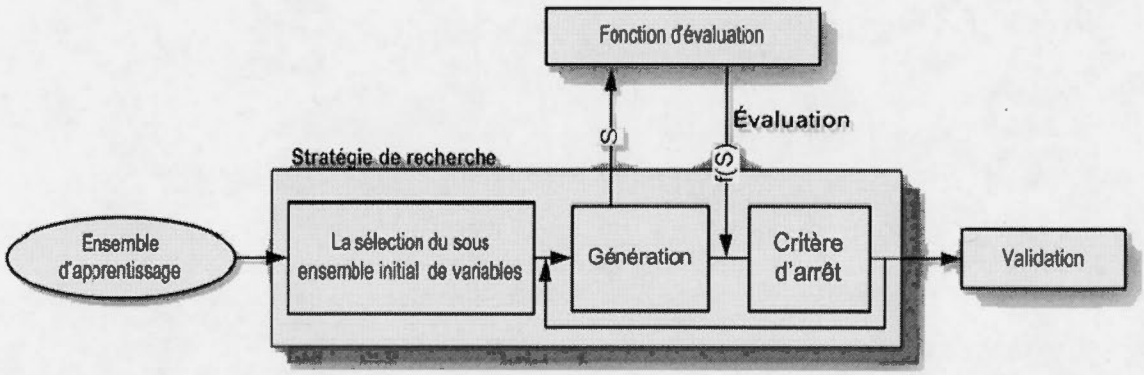


Figure 4.3 Le processus généralisé de la sélection des variables (Huan et Lei, 2005)

Plusieurs critères de sélection ont été proposés dans la littérature pour guider le processus de recherche et mesurer la pertinence des sous-ensembles de variables. Parmi ces critères on retrouve:

- Critère indépendant: s'intéresse aux corrélations entre les variables au lieu d'utiliser des algorithmes Data Mining pour la sélection des variables. Parmi les mesures qui ont été développées dans cette catégorie on retrouve:
    - Mesures de consistance: des coefficients basés sur le biais et la performance d'apprentissage. On trouve plusieurs mesures dans cette catégorie: taux d'incohérence, divergence, Pawlak rough set, paire incohérence, entropie conditionnelle, incertitude symétrique complémentaire, etc. (Shin, Fernandes et al. 2011), (Huan et Lei, 2005).
    - Mesures de dépendances: indices corrélatifs pour la distribution des données. Les mesures les plus connues dans cette catégorie sont: entropie basée sur l'information mutuelle, corrélation basée sur la distance Mahalanobis ou CMD, Jaccard (Chiang et al. 2001), indice Bravais-Pearson, etc. (Zighed, Abdesselam et al. 2013).
    - Mesures de distance: indices qualitatifs de divergence des classes (catégories). Parmi les mesures qui ont été développées dans cette catégorie on retrouve: indice Russell/Rao (Zhang et al. 2003), coefficient d'alignement, coefficient Dice, etc. (Cha, 2007).
    - Mesures d'information: représentent le gain informationnel d'une variable sélectionnée. On retrouve dans cette catégorie les mesures suivantes: indice IT-Sim, Gain Informationnel, Information Mutuelle ou MI (Tapia et Perez, 2013), critère d'indépendance Hilbert Schmidt ou HSIC, etc. (Bo *et al.*, 2009), (Huan et Lei, 2005)
- Critère dépendant: mesure la pertinence d'une variable selon la performance d'apprentissage. On retrouve dans cette catégorie les critères suivants: séparabilité des classes (Lei, Chunhua et Hartley, 2011), cohésion, la divergence de Jensen-Shannon (Duda, Hart et Stork, 2001), la marge maximale, etc. (Bin *et al.*, 2009), (Huan et Lei, 2005).

La génération est un processus de recherche qui produit des variables candidates pour l'évaluation basée sur une stratégie de recherche. Ces stratégies peuvent être classées dans trois catégories qui sont récapitulées à la figure (4.4).

Premièrement, la recherche complète est une (page 56 ligne 2 ?) stratégie de recherche exhaustive permettant de trouver le résultat optimal selon le critère d'évaluation utilisé. Afin de diminuer l'ordre de l'espace de recherche $o(2^n)$, un nombre modéré de sous ensembles sont évalués comme dans les approches de la branche bornée et la recherche bornée (Zongker et Jain, 1996). Secondement, la recherche séquentielle est une stratégie qui ajoute et/ou supprime les variables, une à la fois, dans le

processus de sélection. Les algorithmes les plus connus dans cette catégorie sont: la sélection séquentielle en avant (SFS), la sélection séquentielle descendante (SBS), la sélection séquentielle généralisée ascendante (GSFS), la sélection séquentielle généralisée descendante (GSBS, PTA(l,r)): plus l met_à_coté r, la sélection séquentielle flottante ascendante (SFFS), la sélection avec la recherche Max-Min, HC (Hill Climbing), etc. (Somol *et al.*, 1999) et (Sun, Babbs et Delp, 2005).

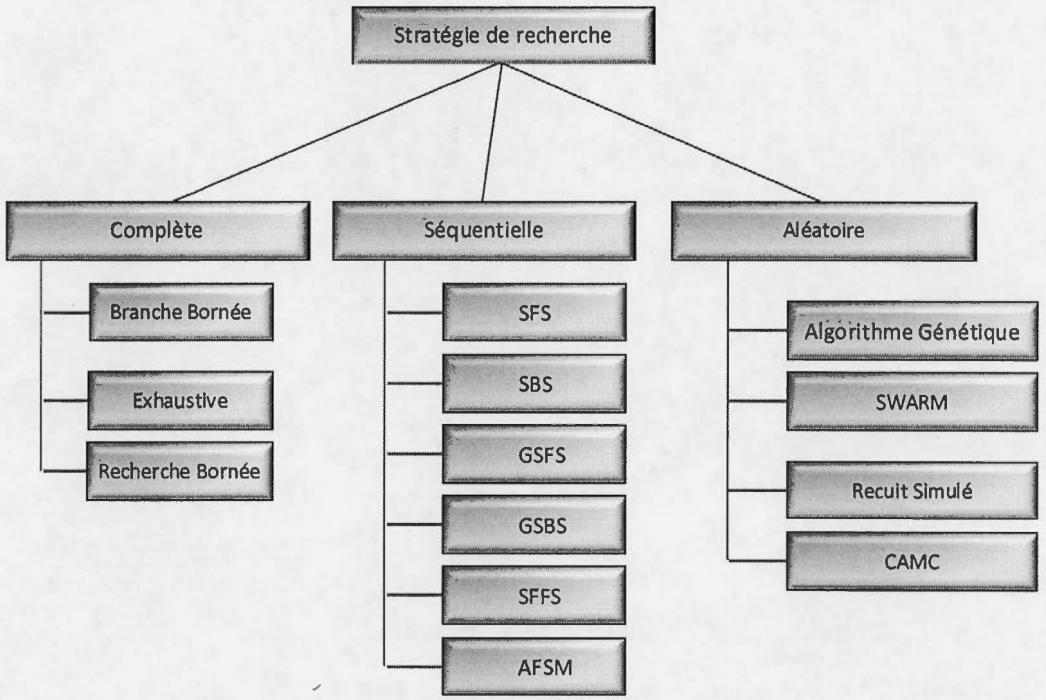


Figure 4.4 Les stratégies de recherche

Troisièmement, la recherche aléatoire qui injecte l'aspect aléatoire dans les approches séquentielles précédentes pour s'échapper des optimums locaux dans l'espace de recherche. Le processus de génération et d'évaluation des sous-ensembles est répété jusqu'à la satisfaction d'un critère d'arrêt donné. Parmi les techniques qui ont été développées dans cette catégorie on retrouve: la sélection génétique GEFS (Zongker et Jain, 1996), l'algorithme LVF(Las Vigas Algorithm), l'algorithme incrémental Las Vegas ou LVI, RELIEF, GAS-SEFS, la sélection des variables basé sur le recuit simulé ou SAFS (Simulated Annealing Feature Selection) (Huan et Lei, 2005).

Dans le prochain paragraphe, nous passons brièvement en revue les principaux algorithmes et les modèles analytiques du paradigme sélection des variables.

4.3 Les modèles de la sélection des variables

Les algorithmes de sélection des variables conçus avec différents critères d'évaluations peuvent être classés dans deux catégories. La différence entre elles dépend de la façon dont l'algorithme de sélection est intégré dans l'algorithme Data Mining. Le modèle de filtrage (Figure 4.5) se fonde sur la corrélation des données sans utiliser un algorithme Data Mining. Le processus de sélection est répété jusqu'à la satisfaction d'un critère d'arrêt prédéfini. Le principal avantage de ce modèle est sa faible complexité de calcul.

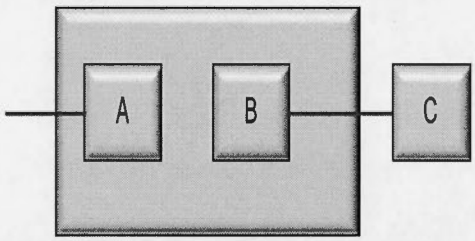


Figure 4.5 Le modèle de filtrage (Zhao *et al.*, 2002)

Tel que:

A: Générateur des sous-ensembles de variables.

B: Évaluateur des sous-ensembles de variables.

C: Suite de la chaîne Data Mining.

D: Algorithme Data Mining.

Le modèle d'emballage (Figure 4.6) utilise un algorithme Data Mining au lieu d'une mesure de dépendance pour la sélection des variables. Ce modèle ne diminue pas le coût computationnel mais il améliore de manière significative la précision prédictive des modèles appris à partir de l'ensemble d'apprentissage (Jain et Zongker, 1997) et (Zhao *et al.*, 2002).

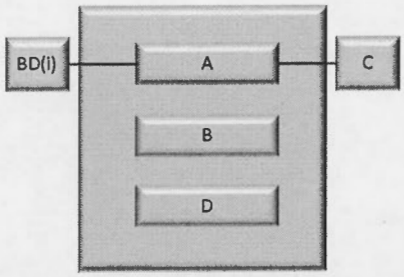


Figure 4.6 Le modèle d'emballage (Zhao *et al.*, 2002)

Plusieurs modèles analytiques ont été utilisés dans la sélection des variables ; parmi ceux-ci on retrouve:

- Les approches basées sur un critère de performance: sont des techniques indépendantes des algorithmes Data Mining et servent comme des filtres pour sélectionner les variables non pertinentes. On retrouve dans cette catégorie les technique suivantes: le seuil de la fréquence (DF) (Document Frequency Tresholding), le test $\chi 2$ (CHI), l'information mutuelle (IM), le gain informationnel (GI), le Ghi carré (GHI), le coefficient de corrélation (CC), le coefficient GSS et la force du terme TS, etc. La liste complète est fournie dans l'article (Sebastiani, 2002).
- Les approches statistiques multi-variables: conservent les variables qui englobent un maximum de variances en éliminant les axes qui correspondent à des variances faibles. Plus formellement, les axes des vecteurs qui correspondent aux petites variances seront éliminés de l'espace d'indexation. On retrouve dans cette catégorie les techniques suivantes: l'approche d'analyse en composantes principales ou ACP, l'analyse en composantes indépendantes ou ACI (Xingfu et Xiangmin, 2011), ACP non linéaire, hyperplan discriminant de Fisher, maximum de vraisemblance ou ML, analyse discriminante multiple ou MDA, l'analyse factorielle de vraisemblance maximale, etc. (Duda, Hart et Stork, 2001), (Stuhler *et al.*, 2011).
- Les approches connexionnistes: sont généralement optimisées par des méthodes d'apprentissage. Les approches les plus populaires dans cette catégorie sont: MDA (Multiple Discriminant Analysis), OBD (Optimal Brain Damage), OBS (Optimal Brain Surgeon), EBD (Early Brain Damage), EBS (Early Brain Damage), OCD (Optimal Cell Damage), etc. Un inventaire des diverses techniques des approches connexionnistes est donné par Bennani (Bennani, 2001).
- Les modèles noyau: on trouve plusieurs approches dans cette catégorie: la machine à vecteurs importés ou IVM, la sélection de vecteur de variables ou FVS (Braun, Weidner et Hinz, 2011), l'analyse en composantes principales noyaux ou KPCA (kernel principal component analysis), etc. (Zhanli *et al.*, 2005),(Djellali, 2013b).

Dans le prochain paragraphe, nous proposons une méthode pour traiter la malédiction de la dimensionnalité en utilisant le critère de la variance du nuage de points.

4.4. Traitement de la malédiction de la dimensionnalité

L'importance de la sélection des variables tout au long du processus d'apprentissage est due au fait que les variables sont souvent corrompues par le bruit et la redondance plutôt que des variables pertinentes.
En effet, les variables non pertinentes peuvent confondre un algorithme d'apprentissage causant un sur-apprentissage pour accommoder le bruit. Afin de surmonter ce problème, nous avons appliqué la décomposition en valeurs singulières tronquées ou TSVD (Truncated Singular Value Decomposition). La transformation est calculée en utilisant la décomposition en valeurs singulières (SVD) de la matrice « terme-document ».
La technique TSVD calcule une approximation de rang inférieur qui tire profit de la structure implicite des corrélations des termes comme montré à la figure ci-dessous:

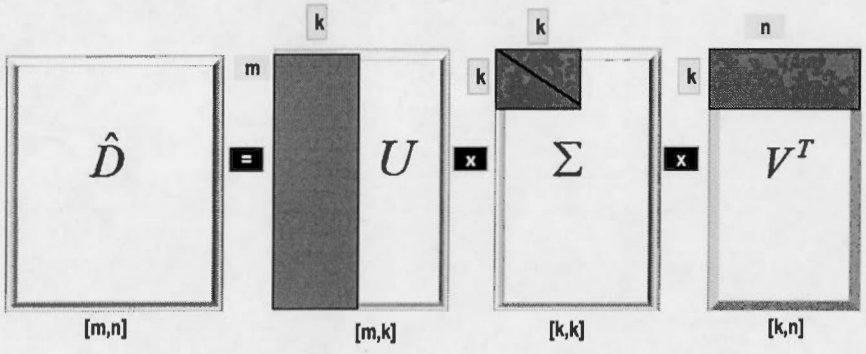


Figure 4.7 La décomposition en valeurs singulières tronquées

$$\hat{D} = U_k \Sigma_k V_k^T$$

$$\Sigma_k = diag[\sigma_1, \sigma_2, \ldots\ldots\ldots \sigma_k]$$

Tel que:

$U_k U_k^T = I_m \wedge V_k^T V_k = I_n$;

$I_m(I_n)$: la matrice identité de taille $m(n)$.

$\hat{D}$: est une matrice réduite « terme-document».

U_k : est une matrice dense de taille $[m,k]$ pour les premières k colonnes de U.

Σ_k : est une matrice positive diagonale de taille $[k,k]$ qui contient les k premières plus grandes valeurs singulières.

V_k^T : est une matrice dense de taille $[k,n]$ contenant les k premières lignes de la matrice V^T.

k : un entier, $k \prec \min(m,n)$.

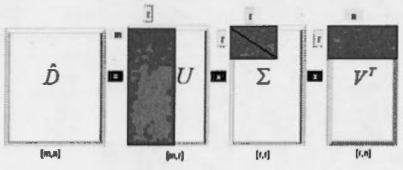
Sur la base des principes expliqués ci-dessus, le processus de base de la décomposition en valeurs singulières tronquées est résumé comme suit:

Début

- ***Etape 1*** Indexation: D [terme-document], $D[tfidf_{ij}]$, $d_{ij} = tfidf_{ij}$, $0 \le i \le m, 0 \le j \le n$.
- ***Etape 2*** Décomposition en valeurs singulières ou DVS.

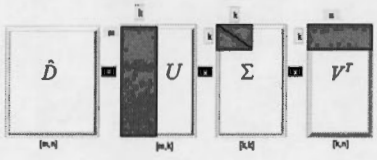
$$D \to \hat{D}$$

$$\hat{D} = U_r \Sigma_r V_r^T$$



$$(r \le \min(m,n); r = rang(\hat{D}) = |\{\sigma_i|; \sigma_i \succ 0)\,.$$

- ***Etape 3*** Approximation ou DVST: $\hat{D}_k = U_k \Sigma_k V_k^T = \sum_{i=1}^{k} u_i \sigma_i v_i^T$



- ***Etape 4*** Génération de l'espace: Requête $D_q U_k$; document $\Sigma_k V_k^T$.
- ***Etape 5*** Repérage $sim(\vec{q},\vec{d}) = (\vec{q}U_k)(\Sigma_k V_k^T),\ \vec{d} \in D; \vec{q} \in Q$
- ***Etape 6*** Triage (sim_i)

Fin

Figure 4.8 Le pseudo code de la décomposition en valeurs singulières tronquées

Les étapes (1, 2 et 3) représentent les actions d'indexation et les étapes (4, 5 et 6) représentent les actions répétitives pour la réduction de la dimensionnalité et le repérage des documents (plus de détails sont fournis dans les articles (Yong, Bin et Long-bin,2008 et Mokris et Skovajsova, 2008).

La figure (4.9) montre la décomposition en valeurs singulières de la matrice terme-document D. Les valeurs singulières (appelées aussi multiplicateurs canoniques) sont des réels positifs ou égales à zéro. Par convention elles sont triées par ordre décroissant le long de la diagonale de la matrice des valeurs singulières. Elles indiquent l'importance des vecteurs singuliers dans l'espace d^{1100}. Les 1100 directions de l'espace représentent le mieux les corrélations entre les termes dans la matrice d'approximation. Nous pouvons obtenir plusieurs approximations de la matrice terme-document $\hat{D}$ en diminuant le rang de la matrice de r à k, tel que: $0 \prec k \leq r; r = rang(\hat{D}) = |\{\sigma_i\}| = 1100;\ \sigma_i \succ 0$. De cette façon, nous pouvons supprimer l'effet de la variation d'utilisation d'un terme k dans la matrice terme-document.

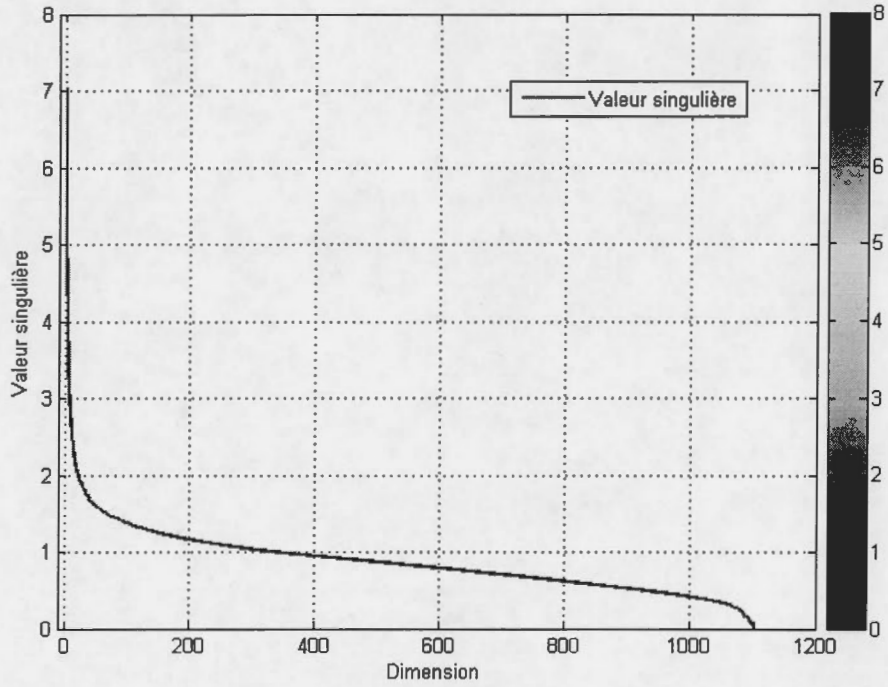


Figure 4.9 La décomposition en valeurs singulières

Le nombre de variables nécessaires pour expliquer un pourcentage prédéterminé de la variance peut varier considérablement en fonction de la collection de documents, les paramètres de l'analyse et le pourcentage souhaité. Le compromis doit être obtenu entre le nombre de variables pertinentes et la quantité de la variance souhaitée.

Pour contrôler ce compromis, nous avons utilisé l'algorithme du rapport d'énergie (Wei, Chang et al. 2001) comme critère pour chercher le nombre de variables pertinentes. Le principe de cette méthode est de trouver un axe issu d'une combinaison des vecteurs singuliers, tel que la variance du nuage de points autour de cet axe soit maximale.

La $i^{ème}$ valeur singulière est proportionnelle à la variance supplémentaire i décrite par l'équation suivante:

$$\mathrm{var}_i = \frac{\sigma_i^2}{\sum_{j=1}^{j=r} \sigma_j^2} \; ; \; r = rang(\hat{D}) = \left|\{\sigma_1, \sigma_2, ..., \sigma_r\}\right| ; \sigma_i \succ 0$$

r : le rang de la matrice (r =1100).

$\sigma_i \succ 0$: la valeur singulière i.

var_i : la variance supplémentaire i.

Chaque valeur singulière témoigne de son importance pour expliquer la variance. Plus précisément, le carré de chaque valeur singulière est proportionnel à la variance expliquée par chaque vecteur singulier. En conséquence, le rapport indique la variation captée par la $i^{ème}$ valeur singulière.

La figure (4.10) montre la quantité de la variance expliquée par les 1100 valeurs singulières. La relation entre la valeur singulière et la variance est inversement proportionnelle (plus la valeur singulière est grande, plus la variance est petite) et les taux de variation ne sont pas constants.

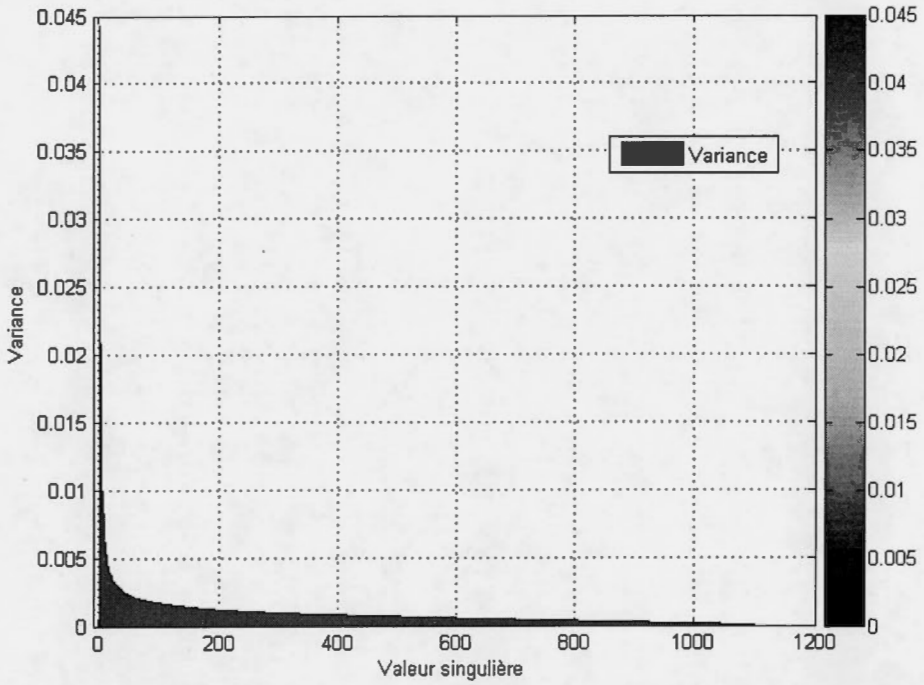


Figure 4.10 La variance expliquée Vs Valeur singulière, fourchette [1,1100]

Le problème est de trouver le meilleur espace affine de dimension (k) dans le sens où il maximise le cumul de la variance. Autrement dit, notre objectif est de trouver un espace réduit qui assure un cumul maximal de la variance souhaitée. Ainsi, le modèle d'emballage TSVD pour choisir le sous-ensemble des variables pertinentes $S_1 = \Phi$, étant donnée l'ensemble de valeurs singulières $S = \{diag(\Sigma_k)\} = \{\sigma_1,..,\sigma_r\}$ est défini par la formule suivante:

$$\Psi^{k+1} = ArgMax_{s_1 \subset S}\left(\frac{\sum_{i=1}^{i=k}\sigma_i^2}{\sum_{j=1}^{j=r}\sigma_j^2}\right);\ r = rang(\hat{D}) = \left|\{\sigma_1,\sigma_2,...,\sigma_r\}\right|;\sigma_i \succ 0$$

Le cumul de la variance est donné par l'équation suivante:

$$C_k = ArgMax_k\left(\sum_{i=1}^{k} \mathrm{var}_i\right) = ArgMax_k\left(\frac{\sum_{i=1}^{i=k}\sigma_i^2}{\sum_{j=1}^{j=r}\sigma_j^2}\right)$$

Ainsi, la sélection des variables est un problème d'optimisation combinatoire qui peut être formulé comme suit:

$$\begin{cases} S^o = ArgMax_{s_1 \subset S}\left(\dfrac{\sum_{i=1}^{i=k}\sigma_i^2}{\sum_{j=1}^{j=r}\sigma_j^2}\right) \\ |S| \le r; \qquad 0 \prec k \le r \end{cases}$$

L'optimalité d'un sous-ensemble de variables S est mesurée par un critère d'évaluation monotone, c.à.d. l'ensemble augmenté S^{l+1} est déterminé de la façon suivante: $S^{l+1} = S^l + x, C(S^l) \le C(S^{l+1})$.

Où,

$|\ |$: la cardinalité d'un ensemble candidat.

r : le rang de la matrice.

C : le cumul de la variance.

x : la variable selectionneé.

La figure (4.11) montre le cumul de la variance expliqué en utilisant toutes les valeurs singulières. Plus de 91.13% de la variance dans les données ont été expliqués par les 721 premières valeurs singulières et une petite explication supplémentaire de la variance est découverte dans la fourchette [722,1100]. Ainsi, la ligne horizontale maximise le cumul de la variance du nuage de points projetés dans un espace réduit et la ligne verticale détermine la valeur singulière correspondante σ_{721} .

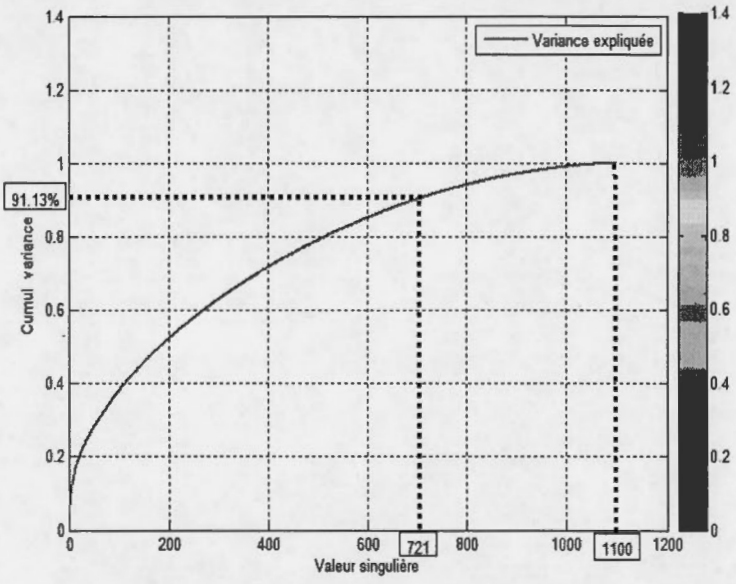


Figure 4.11 Le cumul de la variance expliquée

Les premières 721 valeurs singulières sont beaucoup plus grandes que les dernières valeurs singulières ($\sigma_1 \geq \sigma_2 .. \geq \sigma_{721} > \sigma_{722} .. \sigma_r > 0$) (Figure 4.9) et l'effet cumulatif de la variance des dernières 379 valeurs singulières ne dépasse pas la contribution des premières valeurs singulières (figure 4.12 et figure 4.13). De plus, les variables liées à de petites valeurs singulières sont pratiquement non-pertinentes et n'influencent pas les mesures de similitude entre les documents, c'est-à-dire que leur inclusion réduirait la performance de sélection ($\cos(d_i, d_j) = \cos(d_i^T U_k \sum_k^{-1}, d_j^T U_k \sum_k^{-1})$) (Djellali, 2013d). En conséquence, nous avons généré un espace de projection réduit en gardant seulement les 721 premières valeurs singulières dans la matrice Σ_{1100}.

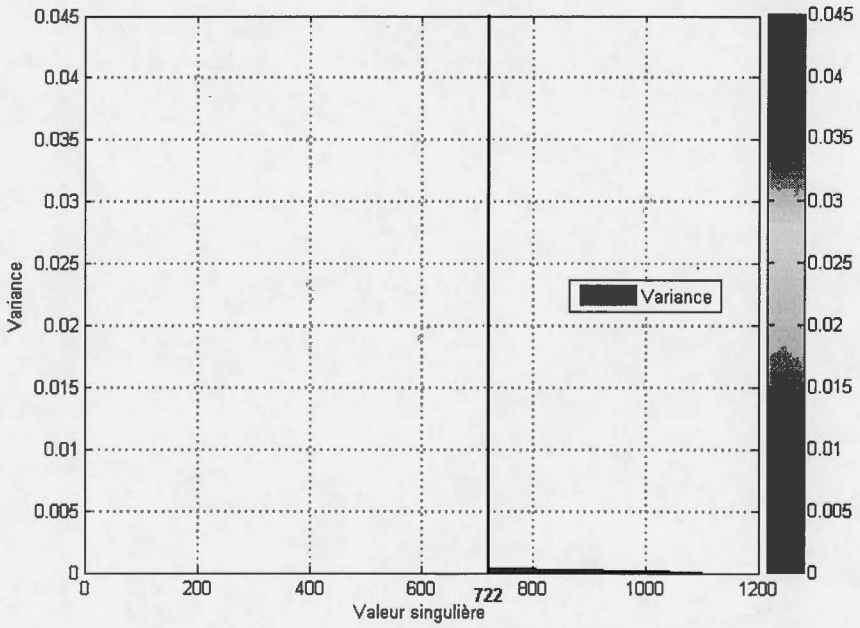


Figure 4.12 Variance VS Valeur singulière, fourchette [722,1100]

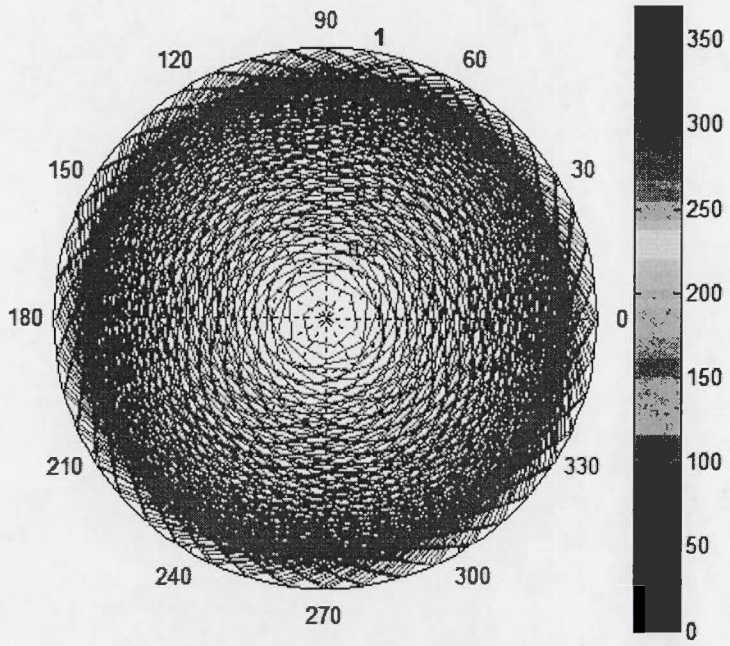


Figure 4.13 Les cordonnées polaires de la variance, fourchette [1,1100]

Les variables d'indexation ont été remplacées par les 721 variables pertinentes en utilisant le modèle d'emballage de la décomposition en valeurs singulières tronquées. Les variables choisies sont des combinaisons des variables originales avec les coefficients de pondération appliqués à partir des valeurs singulières. Les vecteurs singuliers maintenus sont ceux qui ont la plus grande influence sur la position du vecteur document dans l'espace d'approximation (d^{1100}) (Djellali, 2013b) et (Djellali, 2013h).

En résumé, nous avons vu que la sélection des variables implique un processus d'optimisation combinatoire basé sur un critère d'évaluation monotone et utilisant un processus d'apprentissage non supervisé. Pour sa part, la génération été décrite comme une fonction objective qui vise à maximiser le cumul de la variance par l'accumulation des variances supplémentaires proportionnelles aux valeurs singulières.

Conclusion

Nous avons présenté dans ce chapitre un modèle d'emballage pour la sélection des variables pertinentes. Nous avons utilisé la décomposition en valeurs singulières tronquées comme un premier niveau dans le processus d'apprentissage pour deux avantages: d'abord la réduction du coût informatique et en second lieu la réduction de la dimensionnalité tout en conservant la généralité. Il s'agit d'une technique qui utilise une recherche exploratoire portant sur les vecteurs singuliers expliquant au mieux la variance des nuages de données. L'élimination des petites valeurs singulières réduit l'espace d'indexation vectorielle en un espace réduit. La mesure de similitude cosinus calculée en utilisant le produit vectoriel des vecteurs singuliers est plus fiable que celle basée sur les vecteurs originaux. Les variables sélectionnées peuvent être considérées en tant que combinaison des variables originales qui expliquent une proportion maximale de la variance. De plus, le modèle d'emballage de la décomposition en valeurs singulières tronquées considère le biais de l'algorithme de la décomposition en valeurs singulières et l'effet du sous-ensemble des variables choisies. Ainsi, elle peut supprimer efficacement les variables redondantes ou les variables inutiles. En sélectionnant uniquement les premières valeurs singulières, nous avons construit un modèle d'indexation simplifié et précis. Par conséquent, l'apprentissage machine peut s'exécuter sur des données contenant seulement les variables pertinentes pour le clustering. Ce dernier est utilisé pour l'extraction des modèles cachés représentant les changements dans le processus de mise à jour.

Le prochain chapitre porte sur l'analyse et le développement d'un modèle artificiel de réseau de neurones basé sur les principes de la théorie de la résonance adaptative floue pour les tâches de clustering. Par conséquent, les paragraphes qui vont être abordées visent à montrer les concepts liés à l'apprentissage machine, la littérature associée aux modèles inspirés de théorie de la résonance adaptative et l'architecture connexionniste.

CHAPITRE V

CLUSTERING

Résumé: Dans le chapitre précédent, nous avons présenté et discuté un modèle d'emballage conçu pour se confronter au dilemme Exploration/ Exploitation ou la malédiction de la dimensionnalité de Bellman. Le but de ce chapitre n'est pas de donner une présentation exhaustive du clustering, mais plutôt d'en donner certaines clefs pour la configuration de l'architecture neuronale, l'estimation de la précision et la sélection de l'architecture connexionniste. Pour ce faire, nous ferons une introduction générale du clustering. Nous présenterons un panorama des travaux sur les architectures de clustering et les fonctions du coût. Ceci nous permet de définir les nouveaux défis relevant de la configuration des architectures et l'ordre de présentation. L'emphase est ensuite mise sur l'initialisation des paramètres qui permettent de configurer l'architecture neuronale floue pour réduire le temps de calcul et pour chercher une convergence rapide vers le voisinage de la solution. Pour terminer, nous présentons la méthode de sélection permettant de choisir le meilleur modèle de clustering et estimer le taux de reconnaissance basée sur les statistiques de performance, l'architecture connexionniste et l'échantillon de données.

La suite du chapitre est organisée de la manière suivante. Dans le paragraphe (5.1) nous donnons une brève présentation de différentes techniques de clustering. Nous discutons dans le paragraphe (5.2), de l'initialisation des poids synaptiques et de la configuration de l'architecture connexionniste. Nous l'aborderons au travers de l'initialisation typique des poids et les paramètres de configuration. Dans le paragraphe (5.3) nous montrons comment le modèle opère sur le contenu de la base d'apprentissage, en analysant la sélection des modèles, qui non seulement trace la convergence vers le voisinage de la solution mais estime l'erreur de généralisation à partir de l'agrégation des modèles entrainés. Enfin, la sélection de l'architecture connexionniste est présenté dans le paragraphe (5.4).

5.1 Clustering

Le clustering est une technique puissante permettant d'extraire l'information prédictive cachée dans la base d'apprentissage. Il s'agit d'une méthode non supervisée qui aide à se concentrer sur l'information pertinente dans le corpus d'apprentissage. De ce fait, elle prévoit les tendances et l'évolution des connaissances. Elle permet ainsi, de prendre des décisions prédictives basées sur les exemples d'apprentissage.

Le clustering est définie comme suit:

« A cluster is a collection of data objects that are similar to one another within the same cluster and are dissimilar to the objects in other clusters. The process of grouping a set of physical or abstract objects into classes of similar objects is called clustering » (Berzal et Matín, 2002).

Formellement, l'algorithme de clustering cherche à regrouper les documents dans des clusters. $(C_i)_{i=1}^{s}$, Tel que:

$$\bigcup_{i=1}^{s} C_j = (d_1, d_2, \ldots, d_n) \quad \wedge \quad C_i \cap C_j = \phi \text{ si } i \neq j.$$

Si $d_i \in C_j$ alors $d_i^T \omega_j \succ d_i^T \omega_i \ (i = 1,2,\ldots s \wedge i \neq j)$

$$\omega_j = \frac{\overline{\omega}_j}{\left\|\overline{\omega}_j\right\|} \ (j = 1,2,\ldots s \wedge i \neq j)$$

$$\overline{\omega}_j = \frac{1}{f_j} \sum_{d_i \in C_j} d_i \quad \wedge \quad f_j = \left\|C_j\right\| \ (1 \leq j \leq s)$$

Où d_i est i^{ème} document dans le corpus d'apprentissage $D[d_1, d_2, \ldots, d_n]$.

$\overline{\omega}_j$: le vecteur centroïde.

f_j : le nombre de document dans un cluster C_j.

De nombreuses techniques de clustering ont été proposées dans la littérature, parmi ces techniques on retrouve:

- Les techniques de groupement hiérarchique: permettent de créer un arbre hiérarchique de clusters par un raffinement itératif suivant une certaine fonction objective. Elles commencent par un groupement séparé et dans chaque itération, les groupements les plus semblables sont fusionnés dans un nouveau groupement. Le raffinement termine par la création d'un groupement global contenant les exemples d'apprentissage. Parmi les modèles de groupement hiérarchique, on retrouve: l'algorithme de l'arbre auto-organisatrice SOTA (Self Organizing Tree Algorithm), MSOT

(version améliorée de SOTA), le groupement hiérarchique d'agglomération ou HAC(Hierarchical Agglomeration Clustering), SLINK, CLINK, WLA, ULA, COBWEB, etc. (Berkhin, 2006).

- Les techniques de groupement partiel: commencent par un nombre fixe de clusters améliorés itérativement selon une fonction objective. Parmi les modèles proposés dans cette catégorie on retrouve: KNN, c-means, CLARANS (Clustering LARge Applications based on RANdomized Search), fuzzy, c-means, fuzzy k-medoid, ssAHC, genetic k medoid, ssFCM, PAM (Partitionning Around Medoids), CLARA (Clustering LARge Application), CLASA (Clustering LARge Application based on simulated Anealiting), ssAHC, OC-SVM, ss-FCM, SVC,.etc.))(Rui et Wunsch, 2005).
- Les modèles connexionnistes: des réseaux de neurones modélisant le clustering par l'apprentissage des poids synaptiques. Parmi les réseaux connexionnistes populaires dans cette catégorie, on retrouve: la carte auto-organisatrice ou SOM (Self Organizing Map), la quantification vectorielle par apprentissage ou LVQ (Learning Vector Quantization), SOFM, SPLL, la théorie de la résonance adaptative binaire ou ART1, ART2 (amélioration d'ART1), FUZZY ART (version améliorée d'ART en utilisant la logique floue), GNG (Growing Neural Gas), etc.
- Les modèles probabilistes: des méthodes d'ordre supérieur qui supposent que les clusters sont générés selon une distribution de probabilités. Les techniques les plus populaires dans cette catégorie sont: analyse factorielle des correspondances multiples, mélange de gaussiennes, espérance-maximisation ou EM (maximisation expectation), analyse sémantique latente probabiliste ou pLSI, densité mélange de gaussiennes non négative, factorisation matricielle non-négative ou NMF (Baowen, Jianjiang et Gangshi, 2003), décomposition de densité mélange de gaussiennes or GMDD (Weiwei et al., 2012), Allocation latente de Dirichlet ou LDA (Latent Dirichlet Allocation) (Heping, Jie et Shuwu, 2011), etc.
- Les modèles graphiques: des techniques qui se basent sur la maximisation ou la minimisation d'une certaine fonction objective définie sur les propriétés des graphes. On retrouve dans cette catégorie les modèles suivant: CHAMELEON, CLICK, CAST, normalized cuts, Inc-Cluster, AMOEBA, Graphe projectif latente ou LPG (Latent projective graph), etc.
- Les modèles combinatoires: parallèlement à la dynamique scientifique exposée ci-dessus, les nouvelles approches combinatoires injectent l'aspect aléatoire pour accélérer l'apprentissage. Les modèles les plus populaires dans cette catégorie sont: genetic-TS, Genetically guided algorithm or GGA (Hall, Ozyurt et Bezdek, 1999), genetic k-medoid, simulated annealing Fuzzy c-means or SA-FCM,etc.).

De nombreuses mesures ont été proposées pour évaluer la qualité du clustering. La plupart d'entre elles supposent la notion de l'homogénéité des données, la séparation des clusters et la pertinence pour une requête particulière. Ces mesures peuvent être classées dans deux catégories:

- Mesures internes: sont largement utilisées dans le repérage d'information et la reconnaissance des formes (Lu et al., 2007), (Frakes, A. et Baeza-Yates, 2000). Les mesure les plus populaires dans cette catégorie sont : Davies–Bouldin, index Dunn, coefficient de silhouette, homogénéité, séparation, modularité, etc. (Chen, Jaradat et al. 2002).
- Mesures externes: sont les plus connues et les plus utilisés pour évaluer la qualité du clustering. On retrouve dans cette catégorie les mesures suivantes : F-measure, index Jaccard, Fowlkes–Mallows, Mutual Information, matrice de confusion, etc (Berkhin, 2006) et (Duda, Hart et Stork, 2001).

Une étude plus détaillée sur les métriques et les différentes architectures peut être consultée dans les articles (Rui et Wunsch, 2005), (Berkhin, 2006) et (Duda, Hart et Stork, 2001).
Dans le prochain paragraphe, nous présenterons dans un premier temps les fonctions de coût, puis les problématiques de clustering et enfin l'initialisation typique des poids synaptiques et les paramètres de configuration de l'architecture connexionniste choisie.

5.2 La configuration de l'architecture neuronale

La majorité des méthodes de groupement utilisent une certaine fonction de coût afin de refléter la distribution des données. Parmi celles-ci, citons en quelques-unes: la somme des carrés des erreurs, la moyenne du groupe, séparation, SSB (group sum of squares), coefficient de silhouette, matrice de proximité, pureté, les densités (maximum de vraisemblance ou ML, maximisation des attentes ou EM) (Rui et Wunsch, 2005) et (Duda, Hart et Stork, 2001), etc. La tache du groupement devrait chercher l'ensemble de clusters qui maximise ou minimise cette fonction. Cette approche présente certains inconvénients quand l'ensemble d'apprentissage contient des distributions variables. En outre, il existe généralement plusieurs façons pour regrouper l'ensemble d'apprentissage dans un ensemble de clusters. C'est une des principales causes du fait que les méthodes traditionnelles sont susceptibles de générer des résultats insatisfaisants sur l'ensemble de données. De plus, la majorité des approches précédentes ne permettent pas de rechercher la forme complexe des frontières de décision qui séparent les clusters, car les algorithmes utilisés dépendent fortement des formes utilisés et de la mesure de similitude. Il n'y a aucune garantie pour la convergence vers l'optimum global et le résultat n'est pas indépendant des groupes initiaux (Djellali, 2013a), (Djellali, 2014k), (Kim, Lee et al.2000).

Pour surmonter ces problèmes, nous avons utilisé le clustering dynamique de la théorie de la résonance adaptative floue. Il s'agit d'une architecture connexionniste composée d'une couche d'entrée (F_1^a) et une couche de sortie (F2). Les couches F_1^a et F_2^a enregistrent les informations concernant les formes d'entrée durant le processus d'apprentissage. Elles sont en interaction par les poids ascendants b_{ij} et les poids descendants t_{ij} .

Le tableau (5.1) montre l'initialisation typique de l'architecture connexionniste de la théorie de la résonance adaptative floue. La dynamique du réseau est régie par deux sous-systèmes, à savoir, un sous-système d'attention et un sous-système d'orientation qui sont en interaction par les processus ascendants b_{ij} et les processus descendants t_{ij} . Les poids ascendants sont initialisés par des valeurs faibles et les poids descendants sont initialisés à la valeur 1.

Tableau 5.1 Initialisation typique (Boukhadoum, 2010)

Paramètre	Valeurs admissibles	Valeur type
b_{ij}	$0 \prec b_{ij}(0) \prec \frac{L}{L-1+n}$	0.0001
t_{ij}	$t(0)_{ij} = 1$	1

La configuration de notre architecture prend en considération à la fois les poids ascendants et les poids descendants de sorte que nous pouvons introduire le mécanisme d'unification. Ce mécanisme joue un rôle important pour la sélection du neurone vainqueur dans le champ F_2^a .

Le tableau (5.2) montre la configuration des paramètres de l'architecture connexionniste de la théorie de la résonance adaptative floue. Le paramètre de résonance ρ contrôle le nombre de neurones dans la couche de sortie. Quand le seuil de résonance augmente, le nombre de catégorie dans la couche F_2^a augmente aussi. Si ρ =1, le réseau connexionniste génère une nouvelle classe pour chaque vecteur d'entrée dans la base d'apprentissage. Le paramètre α « le paramètre de choix » prend ces valeurs dans l'intervalle $[0,\infty[$. La valeur typique de α est de 0.001. Le paramètre L « le paramètre du choix non engagé » prend ces valeurs dans l'intervalle $[1,\infty[$. Le pas d'apprentissage β est indépendant du temps, il se place dans la plage [0,1].

Tableau 5.2 Configuration de l'architecture Fuzzy ART (Boukhadoum, 2010)

Paramètre	Valeurs admissibles	Valeur type
L	L>1	1
ρ	0< ρ <=1	0.9
α	$[0,\infty[$	0.001
β	[0,1]	0.9

L'initialisation typique et la configuration de l'architecture connexionniste ont été utilisées pour réduire le temps de calcul et pour chercher une convergence rapide vers le voisinage de la solution.

Dans le prochain paragraphe, nous rappelons brièvement quelques principes généraux des méthodes de sélection des modèles puis la méthode de la validation croisée choisie et enfin l'estimation de l'erreur de généralisation.

5.3 Estimation de la précision

Dans notre travail, nous nous concentrons principalement sur les problèmes de la classification dynamique de la théorie de la résonance adaptative. Étant donné l'ensemble de données disponibles et un algorithme connexionniste Fuzzy ART, nous sommes confrontés à deux tâches: chercher le modèle le plus précis à partir d'une base d'apprentissage et estimer le taux de reconnaissance à partir d'une base de test. Ce problème est connu comme le problème de la sélection de modèles. Le but de cette technique est de sélectionner le modèle le plus précis. Dans cette méthode, la variance peut être partiellement réduite en minimisant le biais entre la précision réelle et la précision estimée[5].

Il existe plusieurs approches permettant de former un ensemble de modèles alternatifs, telles que la variation de la complexité d'un modèle donné (Boosting, AdaBoost, WeightBoost, SMOTEBoost, RUSBoost, Jackknife, LogitBoost, Gradient boosting, etc.) (Berzal et Matín, 2002) et l'exécution de différents modèles sur le même ensemble de données (Bagging, Random forest, v-fold validation, non-dependent cross-validation, leave-one-out cross-validation, Monte Carlo cross validation, k-fold validation, etc.) (Duda, Hart et Stork, 2001).

Notre objectif est de choisir le meilleur modèle de clustering basé sur les statistiques de performance, l'architecture connexionniste et le nombre d'exemple dans l'ensemble de données.

Comme précisé précédemment, nous avons contrôlé explicitement la complexité du modèle sélectionné en utilisant l'initialisation typique et la configuration des paramètres (ANNEXE L). Ainsi, le problème de la

[5] Le compromis biais-variance.

sélection du modèle est réduit à la définition suivante: en utilisant le processus de la sélection du modèle avec un ensemble prédéfini de paramètres typique, nous devrons générer le modèle le plus performant à partir de l'échantillon disponible.

Nous avons appliqué la méthode de la validation croisée avec deux blocs aussi appelée méthode de rotation binaire de l'estimation (En anglais: Cross Validation with two blocks) pour générer à la fois un modèle de clustering et une estimation de son taux de reconnaissance à partir l'échantillon disponible. Dans cette méthode, l'échantillon d'apprentissage est divisé en: un ensemble d'apprentissage pour induire le modèle et un ensemble de test pour évaluer le taux de reconnaissance.

Typiquement, 30% des données sont retenues pour tester le modèle (aucune justification théorique pour ce pourcentage). Par conséquent, toutes les expériences se concentreront sur la performance de la validation croisée double (plutôt que 10 ou 20 fois) comme une fonction déterminant le rapport d'apprentissage/test. La figure (5.1) résume les étapes répétitives de la validation croisée, kBlocs.

Algorithme La Validation croisée K bloc

Début /*** Subdiviser le corpus en K bloc ***/

$Corpus = \{(D_1, T_1), (D_2, T_2), ..., (D_k, T_k)\}$.

$Corpus = D_i \cup T_i$; $(D_i \cap T_i) = \Phi$

/*** L'induction sélectionnée du modèle ***/

Pour i=1 **jusqu'à** k

Tant que $E_p \leq \theta$ **faire**

Construire le modèle $FuzzyART_i$ à partir de l'ensemble D_i

Poids montants: $b_{Ji} = (1-\beta) b_{Ji} + \beta(b_{Ji}, I_i)$

Poids descendants: $t_{iJ} = x_i$

($\beta \in [0,1]$ (β =1 pour l'apprentissage rapide).

FTQ

Calculer le taux de reconnaissance $Taux_i$ sur l'ensemble de test T_i

Le vecteur prototype: $J_{t_i} = \arg\max_j (y_j = \dfrac{\sum_{i=1}^{2n} \min(b_{ji}, x_i)}{\alpha + \sum_{i=1}^{2n} b_{ji}})$; $t_i \in T_i$

/*** L'estimation de la précision ***/

Calculer la moyenne des taux de reconnaissance:

$$Taux_E = \frac{1}{k} \sum_{i=1}^{k} Taux_i$$

Fin

Figure 5.1 La validation croisée K bloc

$Taux_i$: Le taux de reconnaissance mesuré du modèle Fuzzy ART$_i$.

$Taux_E$: Le taux de reconnaissance estimé.

Le tableau (5.3) illustre en détails les statistiques de distribution des termes dans le corpus d'apprentissage en utilisant la validation croisée deux blocs.

Tableau 5.3 Les statistiques de distribution des termes avec la validation croisée 2 blocs

Itération	Bloc	$\overline{L}$	$\overline{L}_{SW}$	$\overline{L}_{stem}$	σ_L	σ_{SW}	σ_{stem}
K=1	Apprentissage	186.033	73.201	56.056	61.207	61.295	50.757
	Test	175.071	82.847	65.035	59.041	67.285	49.232
K=2	Apprentissage	179.046	72.117	57.007	60.111	59.213	51.508
	Test	181.215	75.129	58.281	62.067	60.493	52.019

Où, $\overline{L}$: La longueur moyenne du document.

$\overline{L}_{SW}$: La longueur moyenne des mots fonctionnels dans le document.

$\overline{L}_{stem}$: La longueur moyenne des lexèmes dans le document.

σ_L : L'écart type de la longueur du document.

σ_{SW} : L'écart type des mots fonctionnels dans le document.

σ_{stem} : L'écart type des lexèmes dans le document.

Nous avons appliqué l'algorithme de la moyenne quadratique minimale comme un critère d'arrêt pour l'apprentissage. Le critère équivaut à la somme des carrés des erreurs entre les sorties réelles et les sorties désirées. Il s'agit d'une méthode qui suppose que la courbe de meilleur ajustement est la courbe qui a la somme minimale des carrés des écarts entre les sorties réelles et les sorties désirées. Le critère de la moyenne quadratique minimale est donné par la formule suivante:

$$E_p = \sum_{j=1}^{n_L} (e_{1j}^{|L|})^2$$

Où, le signal d'erreur non linéaire est défini par la formule suivante:

$$e_{1j}^{[L]} = d_j^{|L|} - y_j^{|L|}$$

$d_j^{|L|}$, $y_j^{|L|}$ sont respectivement la sortie réelle et la sortie désirée.

Les précisions de la validation croisée double et les erreurs sont rapportées dans le tableau (5.4). L'estimation de l'erreur de généralisation obtenue à partir de la validation croisée double n'est pas fondée sur un modèle sélectionné, mais la moyenne des erreurs de tous les réseaux de neurones entraînés. Le taux de reconnaissance du premier modèle est égal à 97% après 26 itérations. Le deuxième modèle

apprend après 46 itérations avec un taux de reconnaissance égal à 95%. Le taux de la reconnaissance estimé dans la méthode de la validation croisée double est donné par la moyenne des taux de reconnaissances mesurés, c'est-à-dire 96%.

Tableau 5.4 Le taux de reconnaissance

BLOC	Epoch	Taux	Erreur
K=1	26	97%	0.0106
K=2	46	95%	0.0101

Les figures (5.2) et (5.3) montrent la phase d'apprentissage du réseau connexionniste. La sélection des modèles est effectuée de façon indépendante dans chaque expérience par l'intermédiaire de la minimisation de la fonction objective de la moyenne quadratique minimale. Dans chaque itération (epoch), toutes les entrées sont présentées au réseau connexionniste et chaque sortie est calculée individuellement. L'objectif est d'ajuster la matrice de poids b_{ij} et t_{ij} pour minimiser l'erreur. Ceci mène à un problème d'optimisation non linéaire des erreurs carrées. L'axe horizontal représente le nombre d'itérations et l'axe vertical indique l'erreur quadratique après la présentation des formes.

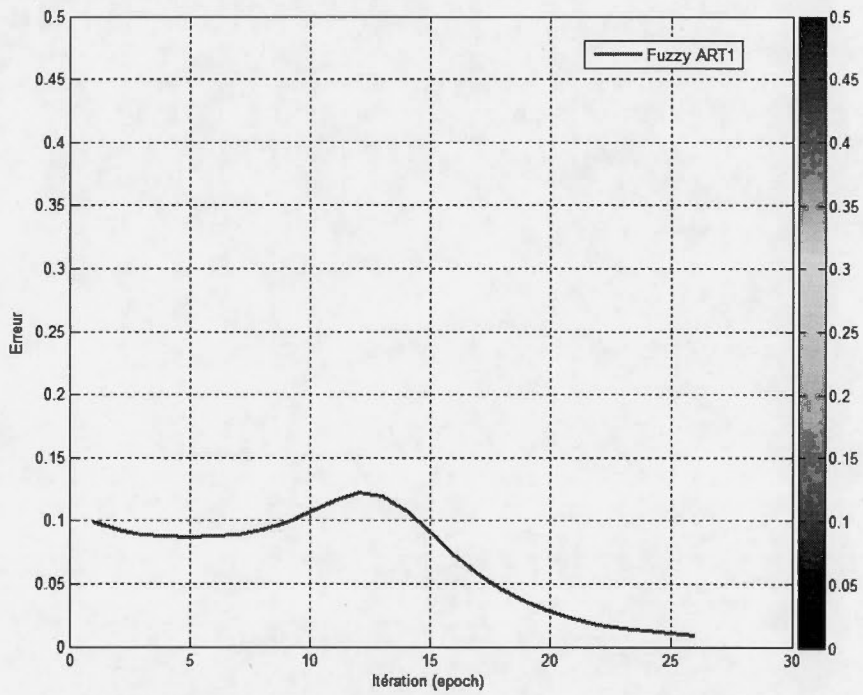


Figure 5.2 Erreur quadratique vs. Nombre d'itérations (modèle 1)

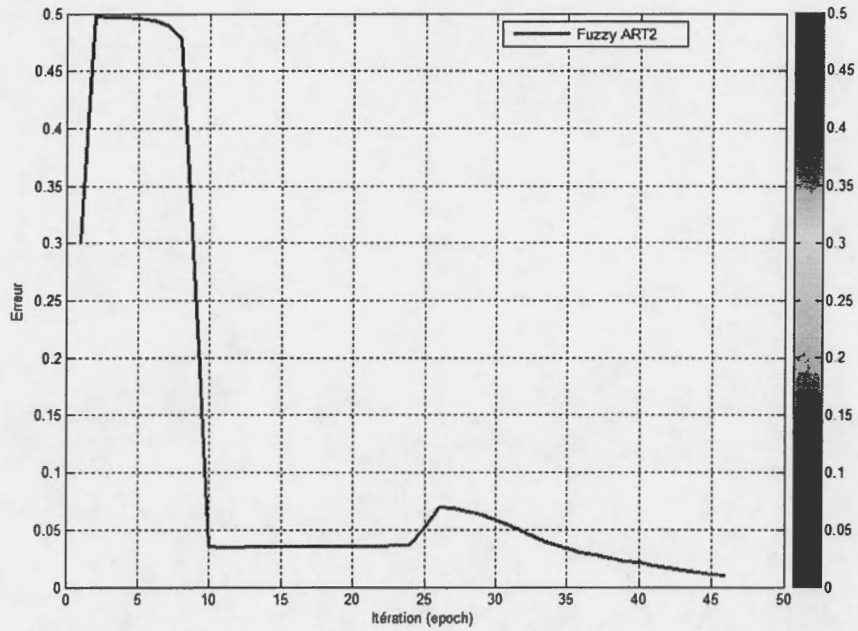


Figure 5.3 Erreur quadratique vs. Nombre d'itérations (modèle 2)

Lorsque le réseau connexionniste a appris à modéliser son apprentissage par la modification des poids synaptiques, le comportement souhaité du réseau est le suivant: On présente un vecteur-document à couche d'entrée F_1^a, celle-ci propage vers la sortie les valeurs d'activation correspondantes en utilisant la règle de propagation de la théorie de la résonance adaptative floue. La reconnaissance du vecteur de sortie généré par l'intermédiaire de la couche de sortie F_2^a devrait alors correspondre à la sortie désirée, telle qu'apprise lors de la phase d'apprentissage.

Nous verrons dans le prochain paragraphe, les techniques utilisées pour sélectionner la meilleure architecture connexionniste entrainée.

5.4 Sélection du modèle

Dans cette section, nous passons en revue les mesures d'évaluation utilisées pour sélectionner le modèle de clustering. Nos données ont été préparées comme décrit dans la section (5.3). Chaque cluster obtenu peut être considéré comme le résultat d'une requête, tandis que l'ensemble de documents pré-classés peut être considéré comme l'ensemble désiré des documents pour cette requête.

a) Le rappel et la précision: le rappel $r(i, j)$ est la fraction des documents récupérés qui sont pertinents à une requête, c'est-à- dire,

$$r(i,j) = \frac{|\{\text{documents pertinents}\} \cap \{\text{documents récupérés}\}|}{|\{\text{documents pertinents}\}|} = \frac{n_{ij}}{n_i},\ 1 \le i \le k.$$

Où, n_{ij} : est le nombre de documents de la catégorie i qui sont présents dans un cluster j.

n_i : est le nombre de documents de la catégorie i.

k : est le nombre de clusters retournés par le regroupement textuel ($1 \le i \le k$).

Il est trivial d'obtenir un rappel 100% en retournant tous les documents en réponse à une requête particulière. Par conséquent, nous devrons mesurer le nombre de documents non pertinents en calculant la précision $p(i,j)$ comme suit:

$$p(i,j) = \frac{|\{\text{documents pertinents}\} \cap \{\text{documents récupérés}\}|}{|\{\text{documents récupérés}\}|} = \frac{n_{ij}}{n_j},\ 1 \le i \le k.$$

n_j : est le nombre de documents dans un cluster j.

$p(i,j)$ mesure la fraction des documents récupérés dans un cluster qui sont pertinents pour le besoin de l'utilisateur. Elle décrit le nombre de documents valides versus le nombre de documents recherchés (valide et non valide).

F-mesure est la moyenne pondérée de la précision et le rappel, qui fournit un moyen raisonnable pour évaluer le compromis précision-rappel. La valeur F-mesure d'un cluster i et une classe j est alors donnée par l'équation suivante:

$$F(i,j) = \frac{2}{\frac{1}{p(i,j)} + \frac{1}{r(i,j)}} = \frac{2 \cdot p(i,j) \cdot r(i,j)}{p(i,j) + r(i,j)}$$

La valeur F-mesure pour le résultat du groupement entier est définie de la façon suivante:

$$F_\mu = \sum_i \frac{n_i}{n} \max_j F(i,j)$$

En général, plus la valeur F-mesure est grande plus le résultat de groupement est bon. Pour évaluer la performance maximale, nous avons exécuté le premier modèle connexionniste sur notre corpus et nous avons obtenu une efficacité maximale égale à 88%. Nous avons ensuite, évalué le deuxième modèle sur le même corpus de données et nous avons constaté une augmentation de 5% de la performance de clustering. La performance maximale du deuxième modèle égale à 93% (figure 5.4).

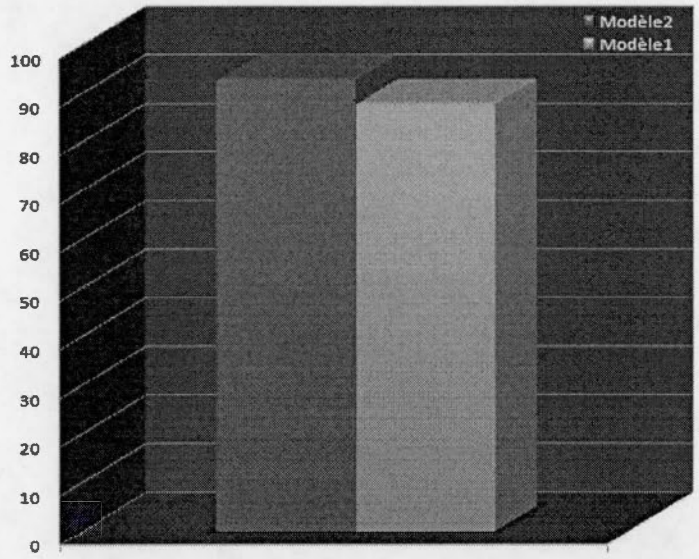


Figure 5.4 La performance du clustering

b) La dispersion: nous avons utilisé la fonction de dispersion comme une fonction objective pour mesurer la qualité du clustering. En d'autres termes, nous calculons l'erreur de chaque vecteur-document, c'est-à-dire, sa distance euclidienne au plus proche centroïde, puis nous calculons la somme totale de l'erreur quadratique. La dispersion du clustering est définie comme suit:

$$E_{C_l} = \sum_{l=1}^{k} \sum_{D \in C_l} d_E(c_l, D); \quad d_E(c_l, D) = \sqrt{\sum_{j=1}^{n} \left| c_l^j - d_j \right|^2}$$

Où, c_l : le centroïde d'un cluster C_l.

d_E :t la distance euclidienne entre deux objets dans l'espace euclidien.

Le centroïde qui minimise la somme de l'erreur quadratique d'un cluster C_l est défini par l'équation suivante:

$$c_l = \frac{1}{m_l} \sum_{D \in C_l} D$$

m_l : le nombre de vecteurs documents dans le cluster C_l.

Tableau 5.5 La dispersion du clustering

Bloc	Modèle	E_{C_l}
K=1	Fuzzy ART$_1$	621.475
	Fuzzy ART$_2$	528.714
K=2	Fuzzy ART$_1$	718.145
	Fuzzy ART$_2$	514.727

Comme mentionné précédemment, étant donné la fonction d'objectif de dispersion, le regroupement peut être traité comme un problème d'optimisation. Une manière de résoudre ce problème est d'énumérer toutes les façons possibles de diviser les vecteurs d'apprentissage en clusters, puis choisir l'ensemble des clusters qui répondent le mieux à la fonction objectif de dispersion, celle qui réduit au minimum le total E_{C_i} (trouver un minimum global).

Les résultats donnés dans le tableau (5.5) montrent que le modèle Fuzzy ART_2 est une solution de clustering représentant le minimum global de la fonction de dispersion, tandis que Fuzzy ART_1 montre un clustering sous-optimal qui est seulement un minimum local. Autrement dit, étant donné deux ensembles de clusters qui sont produits par les deux itérations, nous avons sélectionné le modèle connexionniste Fuzzy ART_2 avec la plus petite erreur quadratique puisque les centroïdes de ce regroupement sont une meilleure représentation des points dans leur groupe.

c) La cohésion: pour illustrer le fait que l'architecture connexionniste de théorie de la résonance adaptative floue n'est pas limitée aux données dans l'espace euclidien, nous considérons la matrice [document-terme] et la mesure de similarité cosinus. Le document est représenté par un vecteur [$tfidf$] comme décrit précédemment. Notre objectif est de maximiser la similarité des documents au centre de gravité du cluster, cette quantité appelée la cohésion du cluster. La quantité analogique au total E_{C_i} est la cohésion totale C_{C_i}, donnée par l'équation suivante:

$$C_{C_i} = \sum_{i=1}^{k} \sum_{D \in C_i} \cos(c_i, D), \quad \cos(c_i, D) = \frac{c_i \cdot D}{\|c_i\| \cdot \|D\|} = \frac{\sum_{i=1}^{n} c_i^k \cdot d_i}{\sqrt{\sum_{i=1}^{n} (c_i^k)^2} \cdot \sqrt{\sum_{i=1}^{n} (d_i)^2}}$$

Tableau 5.6 La cohésion totale du clustering

Bloc	Modèle	C_{C_i}
K=1	Fuzzy ART_1	819.774
	Fuzzy ART_2	1087.001
K=2	Fuzzy ART_1	908.170
	Fuzzy ART_2	915.420

Comme montré dans le tableau (5.6), le modèle Fuzzy ART_2 maximise la qualité objective de la cohésion totale du clustering.

d) L'information mutuelle: dans la théorie de l'information, l'information mutuelle mesure la dépendance entre deux distributions de probabilité (Duda, Hart et Stork, 2001), (Haifeng, Keshu et Tao, 2004). Cette méthode a été appliquée avec succès à une gamme variée de problèmes d'estimation et d'optimisation, y compris l'allocation du tampon mémoire, les files d'attente, la détection du signal, la séquence génétique, l'alignement, l'optimisation combinatoire, etc. La forme générale de l'information mutuelle est la suivante:

$$E_j = \sum_i p_{ij} \log(p_{ij})$$

Pour chaque cluster j, nous calculons p_{ij} la probabilité d'appartenance d'un document d'un cluster j à la catégorie i. E_j mesure l'erreur entre les sorties du réseau connexionniste (cluster j - catégorie i). Ces sorties sont considérées comme des hypothèses indépendantes. Elle est maximale quand les distributions sont complètement corrélées.

La figure (5.5) montre l'information mutuelle des deux modèles, l'axe horizontal représente le nombre d'itérations et l'axe vertical indique les valeurs de l'information mutuelle.

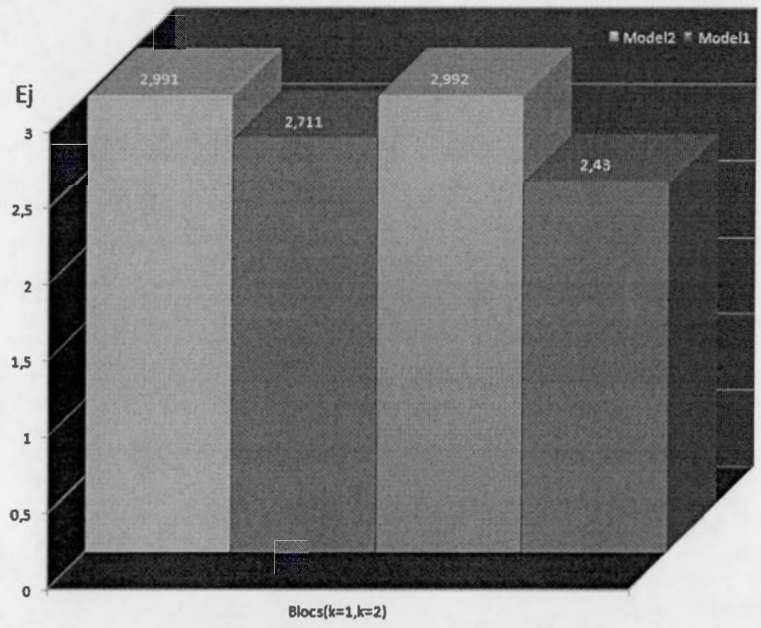


Figure 5.5 L'entropie de 2 itérations

Pour avoir un modèle de clustering efficace, une idée interprétable en termes de l'analyse de données est donc de chercher parmi les deux modèles connexionnistes celui qui possède la plus grande corrélation avec la répartition en classes/catégories. Autrement dit, nous cherchons celui qui a la meilleure information mutuelle avec la distribution des points d'apprentissage sur les classes/catégories. Comme illustré dans la figure (5.5), le modèle Fuzzy ART_2 a la meilleure information mutuelle.

De ce qui précède, il ressort que le modèle Fuzzy ART_2 améliore considérablement la qualité du clustering. Par conséquent, nous avons retenu ce modèle pour l'acquisition des changements candidats dans le processus d'apprentissage de l'ontologie.

La méthode que nous venons de décrire pour l'estimation de la précision et la sélection des modèles présente deux principaux avantages par rapport aux méthodes classiques: elle est automatique (estimation de la précision) et elle est optimale (sélection des modèles). La performance et les résultats des différents réseaux entraînés à partir de la validation croisée double ont été considérés dans l'estimation prédictive du modèle de clustering sélectionné pour la mise a jour de l'ontologie (Djellali, 2012).

Conclusion

Pour assurer une meilleure généralité de l'apprentissage, nous avons utilisé le modèle connexionniste de la théorie de la résonance adaptative floue, la sélection des modèles, l'initialisation typique, la configuration des paramètres et l'apprentissage en ligne. Le choix de ce modèle est dicté par le fait qu'il représente d'une part le clustering dynamique, et d'autre part, il ne dépend pas de l'ordre de présentation (plasticité - élasticité). L'initialisation typique et la configuration des paramètres de l'architecture connexionniste réduisent le temps de calcul et améliorent la vitesse de convergence finale pour atteindre le voisinage de la solution. De plus, l'apprentissage de la théorie de la résonance adaptative donne des fonctions de sortie avec une bonne capacité de généralisation. Il a les avantages suivants sur les autres techniques d'apprentissage machine:

- Fuzzy ART permet d'éviter les minimas locaux.
- Ne dépend pas de l'ordre de présentation en ligne (plasticité - élasticité).
- N'est pas affecté par le sur-apprentissage et est résistant au bruit.
- L'interaction entre le sous-système d'orientation et celui d'attention permet au réseau d'ajuster automatiquement sa taille en fonction de la complexité de la tâche de clustering.
- Fuzzy ART peut être utilisé comme un système de quantisation vectorielle.
- Permet la modélisation des relations complexes.

La modèle de clustering de la théorie de la résonance adaptative a été testé en utilisant une seule boucle de la validation croisée double, où l'échantillon des données est utilisé pour déterminer l'architecture connexionniste et pour estimer le taux de reconnaissance. L'estimation de l'erreur a été itérée sur deux échantillons d'apprentissage puis les estimations moyennes du biais et de la variance sont calculées sur deux échantillons de test. C'est une estimation de meilleure qualité que l'estimation séparée des taux de reconnaissance. La méthode proposée a également fait preuve d'une grande stabilité de reconnaissance, tout en assurant l'efficacité elle permet d'optimiser le compromis biais-variance sur l'architecture connexionniste sélectionnée. De ce fait elle permet d'automatiser le processus d'acquisition de la connaissance pour identifier les changements candidats dans le processus d'apprentissage.

La tâche d'apprentissage de l'ontologie consiste à enrichir le vocabulaire du domaine par des artéfacts ontologiques extraits à partir d'un corpus et d'arranger ces termes taxonomiquement. Dans le prochain chapitre, nous présenterons la manière dont les clusters raffinés peuvent être utilisés pour enrichir l'ontologie. Nous examinerons plus en détails le processus d'alignement utilisé pour achever l'interopérabilité entre les étiquettes descriptives et les artefacts de l'ontologie.

CHAPITRE VI

ALIGNEMENT

Résumé: Dans le chapitre précédent, nous avons présenté et discuté l'initialisation des poids synaptiques, la configuration de l'architecture connexionniste et de la sélection des modèles. Le présent chapitre porte sur la stratégie que nous adoptons pour chercher l'alignement optimal entre les étiquettes et artéfacts de l'ontologie. Cette stratégie consiste essentiellement en l'utilisation des processus individuels d'alignement pour achever l'interopérabilité entre les deux représentations. Dans une deuxième partie de ce chapitre, nous étudions un mécanisme d'agrégation pour élaborer un compromis des diverses décisions, ce qui nous permettra de former un alignement faible et robuste. Nous traitons particulièrement du cas d'agrégation pondérée. L'objectif de l'utilisation d'un opérateur d'agrégation pondérée est de faciliter le processus de combinaison des décisions des alignements individuels de manière à en obtenir une seule.

Dans la suite de ce chapitre, nous présentons dans le paragraphe (6.1) une approche d'alignement avec agrégation qui utilise plusieurs processus individuels d'alignement. Dans le paragraphe (6.2), nous allons nous intéresser plus spécifiquement aux processus individuels d'alignement qui ont été proposés dans le cadre de notre modèle conceptuel. Nous allons passer succinctement les avantages et les inconvénients de chaque processus. Le paragraphe (6.3) présente l'impact de l'agrégation sur le processus d'alignement en fonction de la pondération et de la structure des agrégats. Suivront enfin la corrélation (paragraphes 6.4) et l'évaluation en paragraphes (6.5).

6.1 Processus d'alignement

L'extraction des règles d'alignement s'avère difficile car l'ontologie est une structure complexe, ainsi, l'établissement des correspondances n'est pas facile. En analysant les artéfacts de l'ontologie et les étiquettes associées, nous pouvons repérer les correspondances entre les entités qui ne sont pas immédiatement évidentes. Pour repérer la correspondance entre les artéfacts ontologique et les étiquettes descriptives, nous proposons une nouvelle approche semi-automatique d'alignement en utilisant plusieurs étapes (figure (6.1)). Premièrement, dans l'étape d'extraction de la similarité, nous utilisons plusieurs processus individuels d'alignement. Chaque processus est une opération de recherche semi-automatisée permettant de trouver une forme représentant l'étiquette descriptives dans un modèle ontologique OWL. Les règles sont basées sur la distance entre l'information contenue dans l'ontologie et les étiquettes descriptives. Une fois obtenue la similarité de chaque alignement individuel, il est nécessaire dans une deuxième étape d'obtenir un modèle qui permettra de faire une décision globale. En effet, nous proposons, en se fondant sur les mesures de similarité individuelles précédemment calculées, un processus d'agrégation pour pondérer les décisions des alignements individuels.

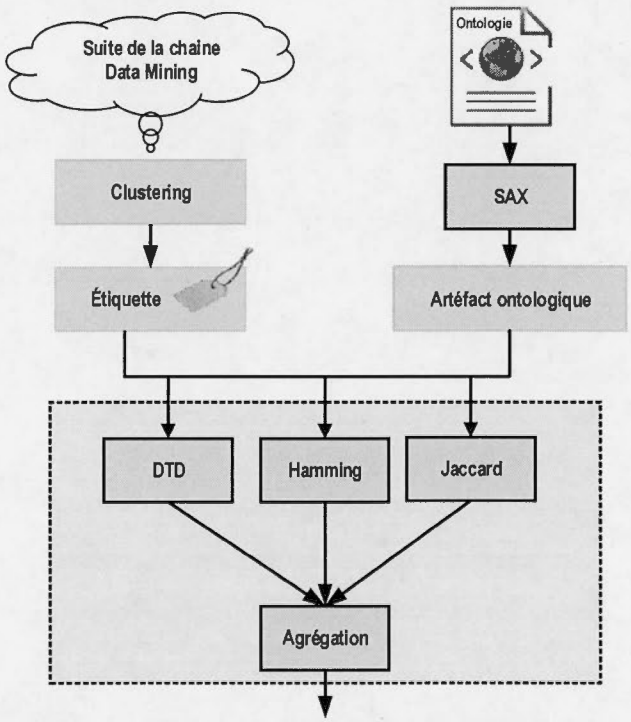


Figure 6.1 Processus d'alignement

L'interface de programmation SAX [6] (Simple Api for Xml)(Su Cheng et Krishna Rao, 2007) est utilisée pour extraire et analyser la syntaxe du document OWL en utilisant les mécanismes de notification (en anglais: callback). Il s'agit d'une interface qui s'adapte au volume de données par une analyse incrémentale. C'est pourquoi ce mode d'analyse utilise moins de coût informatique, au sens de l'utilisation optimale des ressources mémoire et le temps d'exécution nécessaire pour l'extraction des constructeurs syntaxiques. L'analyseur syntaxique intégré dans SAX permet de parcourir le document OWL et d'en extraire la structure hiérarchique des artéfacts ontologiques.

Dans le prochain paragraphe, nous allons présenter les résultats des expérimentations menées sur l'ontologie CRISP-DM-OWL décrit précédemment. Ces expérimentations ont été effectuées sur les étiquettes descriptives générées par le modèle connexionniste de la théorie de la résonance adaptative. Les tests effectués ont pour but de mesurer l'impact de la distance sur le processus d'alignement en fonction de la séquence des opérations et le coût d'édition.

6.2 Alignements individuels

Nous avons utilisé la méthode de la Déformation Temporelle Dynamique, DTD (DTW: Dynamic Time Warping) (Gang *et al.*, 2011) pour calculer la similarité entre les artéfacts ontologiques et les étiquettes descriptives. La similarité entre chaque concept dans l'ontologie et les étiquettes générées par le clustering descriptif est donnée par l'équation suivante:

$$Cof_{sim_D}(Label_k, C) = 1 - \frac{\lambda_D(Label_k, C)}{\max(|Label_k|, |C|)},\ C \in \Gamma^m, Label_k \in \Gamma^n_{OWL}.$$

La distance de la Déformation Temporelle Dynamique $\lambda_D(Label_k, C)$ est définie comme le coût global minimal pour passer de la chaîne C à la chaîne $Label_k$, c'est-à-dire:

$$\forall c_1, c_2 \in \Gamma,\ \alpha(c_1) = \beta(c_2) = 1, \delta(c_1, c_2) = 1 \text{ quand } c_1 \neq c_2 \text{ et } \delta(c_1, c_1) = 0.$$

Pour toute chaîne $C \in \Gamma^*$ on note sa longueur par $|C|$.

Γ : est un ensemble d'alphabets dont les éléments sont utilisés pour construire les chaînes des étiquettes descriptives.

[6] http://www.saxproject.org/

Γ_{OWL} : l'ensemble d'alphabets des chaînes représentant les artefacts de l'ontologie CRISP-DM-OWL. π_i, π_s, π_r sont des ensembles d'entiers. La fonction $\alpha : \Gamma \to \pi_i$ représente la fonction de coût d'insertion, c'est-à-dire que $\alpha(c_1)$ est le coût d'insertion de l'élément $c_1 \in \Gamma$ dans une chaîne donnée.
De même, nous définissons la fonction de coût de suppression comme $\beta : \Gamma \to \pi_s$, de sorte que $\beta(c_1)$ est le coût de suppression d'un élément $c_1 \in \Gamma$ d'une chaîne donnée.
Ensuite, nous définissons la fonction de coût de substitution $\delta : \Gamma \times \Gamma \to \pi_r$, de sorte que, $\forall c_1, c_2 \in \Gamma$, $\delta(c_1, c_2)$ est le coût de remplacement de l'élément c_1 par l'élément c_2 dans une chaîne donnée.
Finalement, nous utilisons $\lambda_D(i, j)$ pour dénoter la distance de la Déformation Temporelle Dynamique entre deux sous- chaînes $c_1^1 c_1^2 \ldots\ldots c_1^i$ et $Label_k^1 Label_k^2 \ldots\ldots Label_k^j$ (Zhiwei, Hui et McClean, 2012).

Nous adoptons les deux scénarios suivants:

$$\lambda_D(0,0) = 0.$$

$$\lambda_D(i,0) = \sum_{k=1}^{i} \alpha(c_1^k).$$

et

$$\lambda_D(0,j) = \sum_{k=1}^{j} \beta(Label_k^j),$$

$$1 \le i \le m, \quad 1 \le j \le n,$$

$$Label_k \in \Gamma, C \in \Gamma_{OWL}.$$

Alors la distance de la Déformation Temporelle Dynamique (DTD) $\lambda_D(m,n)$ est définie par la relation de récurrence suivante:

$$\lambda_D(i,j) = \min \begin{bmatrix} \lambda_D(i-1,j) + \beta(Label_k^j) \\ \lambda_D(i,j-1) + \alpha(c_1^i) \\ \lambda_D(i-1,j-1) + \delta(c_1^i, Label_k^j) \end{bmatrix}$$

Le calcul de la distance de la Déformation Temporelle Dynamique n'est pas optimisé parce que plusieurs éléments λ_D^{ij} $(\lambda_D(i,j))$, $(2 \le i \le p, \quad 2 \le j \le q)$ sont calculés plusieurs fois. Pour calculer $\lambda_D^{p,q}$, tous les termes $\lambda_D^{i,j}$ $(1 \le i \le p, \quad 1 \le j \le q)$ doivent être calculés. Les termes limites $\lambda_D^{i,0}$, $\lambda_D^{0,j}$ $(1 \le i \le p, \quad 1 \le j \le q)$ sont d'abord initialisés et tous les termes sont séquentiellement calculés (ligne par ligne ou colonne par colonne). Par conséquent, la complexité algorithmique est $o(|C| \times |Label_k|)$ tandis que l'espace computationnel nécessaire est seulement $o(\min(|C|, |Label_k|))$. L'algorithme de programmation dynamique nécessite l'initialisation d'une matrice de taille $[(m+1),(n+1)]$ (Xiao-Li, Cheng-Kui et Zheng-Ou, 2006),(Gang *et al.*, 2011).

La figure (6.2) montre les étapes itératives de l'algorithme de la Déformation Temporelle Dynamique.

Pseudo code $(\lambda_D(Label_k^1 Label_k^2 \ldots\ldots Label_k^m), (c_1^1 c_1^2 \ldots\ldots c_1^n))$

Début /** Initialisation **/

  Pour i=1 jusqu'à m **faire**

    $\lambda_D^{i,0} = i$.

  Fp

  Pour j=1 jusqu'à n **faire**

    $\lambda_D^{0,j} = j$.

  Fp

  /** Le calcul itératif de $\lambda_D^{p,q}$ **/

  Pour i=1 jusqu'à m **faire**

    Pour j=1 jusqu'à n **faire**

      $\lambda_D^{i,j} = \min(\lambda_D^{i-1,j} + 1, \lambda_D^{i,j-1} + 1, \lambda_D^{i-1,j-1} + \Psi(c_1^i, Label_k^j))$.

    Fp

  Fp

    imprimer ($\lambda_D^{m-1,n-1}$).

Fin

Figure 6.2 Le pseudo code de l'algorithme standard de la distance DTD

À titre d'exemple, le fragment de l'ontologie[7] à aligner est illustré dans la figure (6.3). Le code OWL-DL correspondant est fourni dans l'annexe (13) et les étiquettes choisies sont le résultat de la classification automatique descriptive.

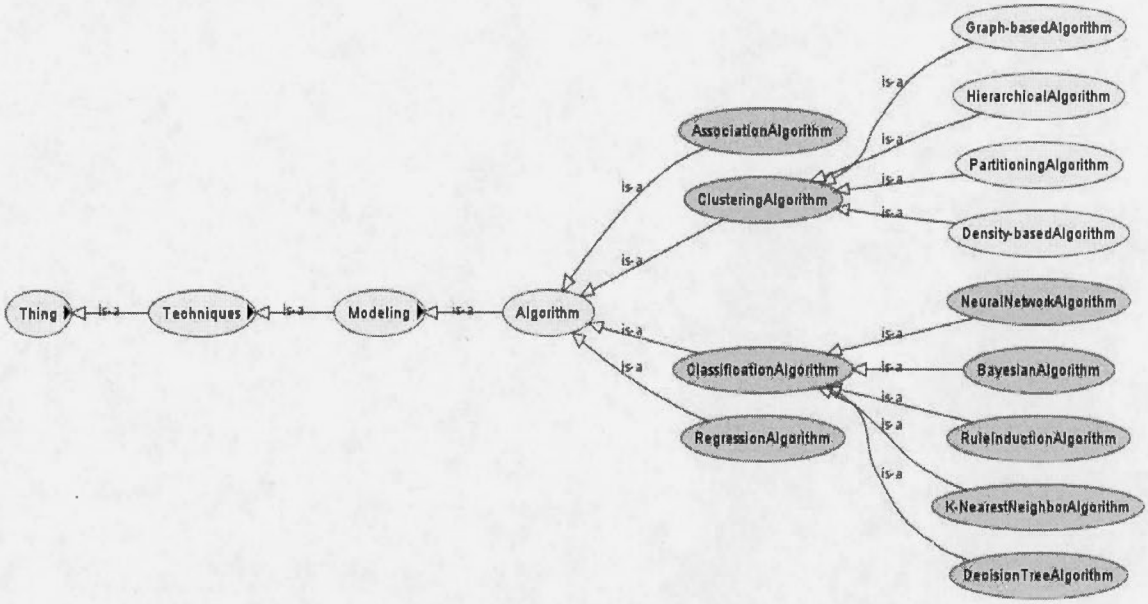


Figure 6.3 Les techniques de modélisation

Le tableau (6.1) montre la matrice produite lorsque la distance de la Déformation Temporelle Dynamique est calculée entre les deux chaînes « clustering » et « ClusteringAlgorithm ».

Dans cet exemple d'illustration nous pouvons voir que les dix premiers caractères s'alignent exactement, alors la distance de la Déformation Temporelle Dynamique jusqu'à ce point est zéro. D'autre part, lorsque le processus arrive au 11ème caractère, un décalage se produit correspondant à une primitive d'édition (insertion), donc la distance entre les deux chaînes est incrémentée. Cette procédure se poursuit jusqu'à ce que tous les caractères soient examinés, résultant en une distance totale de neuf (9) opérations entre les deux chaînes (le nombre minimal des primitives d'édition pour passer de la chaîne « clustering » à la chaîne « ClusteringAlgorithm » égale à 9).

Les séquences d'opérations effectuées pour passer de la chaîne « clustering » à la chaîne «ClusteringAlgorithm » peuvent être facilement récupérées de la matrice en parcourant le chemin $\lambda_D^{|\text{clustering}|-1,|\text{ClusteringAlgorithm}|-1} \rightarrow \lambda_D^{0,0}$, c'est-à-dire, la séquence des opérations qui correspondent à la formule de mise à jour. Les entrées en gras indiquent le chemin $\lambda_D^{0,0} \rightarrow \lambda_D^{|\text{clustering}|-1,|\text{ClusteringAlgorithm}|-1}$ (plusieurs chemins peuvent exister).

[7] Pour des raisons de lisibilité, seule la relation de hiérarchie *is-a* a été représentée.

Tableau 6.1 La distance DTD entre les deux chaînes « clustering » et « ClusteringAlgorithm »

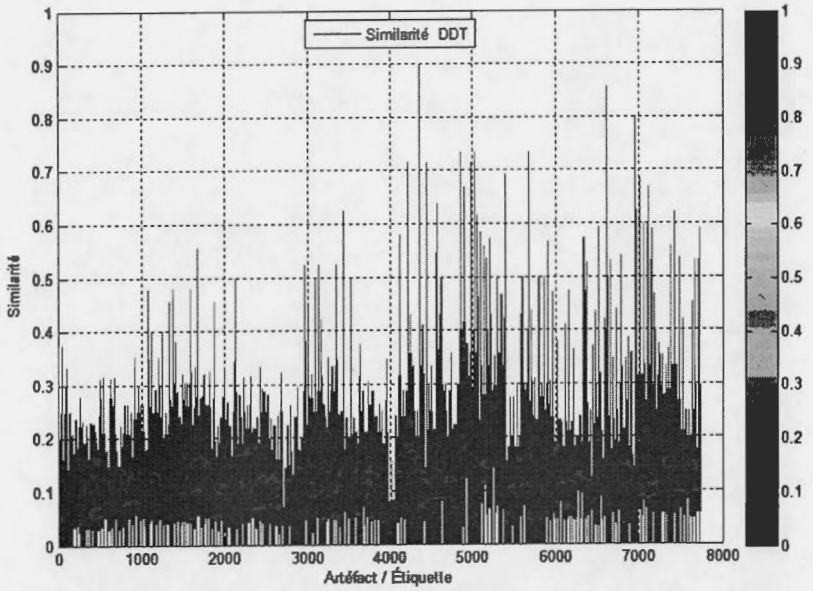
		c	l	u	s	t	e	r	i	n	g
	0	1	2	3	4	5	6	7	8	9	10
C	1	0	1	2	3	4	5	6	7	8	9
l	2	1	0	1	2	3	4	5	6	7	8
u	3	2	1	0	1	2	3	4	5	6	7
s	4	3	2	1	0	1	2	3	4	5	6
t	5	4	3	2	1	0	1	2	3	4	5
e	6	5	4	3	2	1	0	1	2	3	4
r	7	6	5	4	3	2	1	0	1	2	3
i	8	7	6	5	4	3	2	1	0	1	2
n	9	8	7	6	5	4	3	2	1	0	1
g	10	9	8	7	6	5	4	3	2	1	0
A	11	10	9	8	7	6	5	4	3	2	1
l	12	11	10	9	8	7	6	5	4	3	2
g	13	12	11	10	9	8	7	6	5	4	3
o	14	13	12	11	10	9	8	7	6	5	4
r	15	14	13	12	11	10	9	8	7	6	5
i	16	15	14	13	12	11	10	9	8	7	6
t	17	16	15	14	13	12	11	10	9	8	7
h	18	17	16	15	14	13	12	11	10	9	8
m	19	18	17	16	15	14	13	12	11	10	9

Le tableau (6.2) montre la similarité calculée entre les étiquettes descriptives et les artéfacts du fragment de l'ontologie décrit à la figure (6.3). La similarité de la Déformation Temporelle Dynamique est utilisée dans le processus individuel d'alignement.

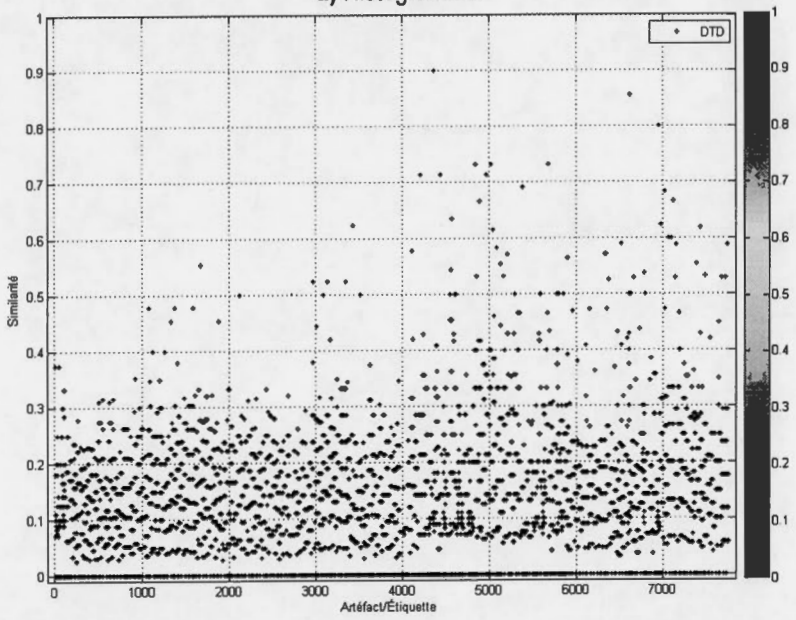
Tableau 6.2 L'alignement avec la similarité de la Déformation Temporelle Dynamique

	Association Algorithm	Clustering Algorithm	Classification Algorithm	Regression Algorithm	Ruleinduction Algorithm
Classification	0.300	0.316	✓0.609	0.211	0.182
regression	0.150	0.158	0.130	✓0.526	0.227
association	✓0.550	0.211	0.391	0.211	0.182
clustering	0.150	✓0.526	0.261	0.158	0.227
neural	0.100	0.211	0.087	0.211	0.182
sampling	0.200	0.158	0.174	0.158	0.136
selection	0.250	0.211	0.261	0.263	0.318
boosting	0.250	0.263	0.217	0.211	0.182
bagging	0.150	0.158	0.174	0.211	0.136
Induction	×0.250	0.158	0.217	0.158	0.248

La figure (6.4) montre l'utilisation de la programmation dynamique pour calculer les règles d'alignement. L'axe horizontal représente l'alignement de la Déformation Temporelle Dynamique et l'axe vertical indique les valeurs de similitude calculées entre les étiquettes générées par l'algorithme de clustering descriptif et l'ensemble des artéfacts dans l'ontologie CRISP-DM-OWL.



a) Histogramme



b) Dispersion

Figure 6.4 La similarité de la Déformation Temporelle Dynamique entre les concepts et les étiquettes

Dans le deuxième processus individuel d'alignement, nous avons utilisé la distance de Hamming pour calculer la similarité entre les étiquettes descriptives et les concepts ontologiques (Liu, Ke et Torng, 2011).

La distance de Hamming $\lambda_H(Label_k, C)$ est le nombre de positions où les deux chaînes sont différentes:

$$\lambda_H((c^1c^2.....c^p),(Label_k^1 Label_k^2.....Label_k^p)) = \sum_{i=1}^{p} \Psi(c^i, Label_k^i).$$

$$\forall c^i, Label_k^i \in \Gamma \times \Gamma_{OWL}.$$

La distance entre deux symboles c^i et $Label_k^i$ est définie de la façon suivante:

$$\Psi(c^i, Label_k^i) = \begin{cases} 0 & si \quad c^i = Label_k^i \\ 1 & Sinon \end{cases}, \ \forall c^i \in \Gamma, Label_k^i \in \Gamma_{OWL}.$$

Le coefficient de similitude de Hamming est donné par l'équation suivante:

$$Cof_{sim_H}(Label_k, C) = 1 - \frac{\lambda_H(Label_k, C)}{\max(|Label_k|, |C|)}$$

Si $|C| \neq |Label_k|$, alors la chaîne la plus courte peut être remplie par des espaces. Par exemple, la distance de Hamming des chaînes « clustering » et « ClusteringAlgorithm » n'est pas définie, puisque la chaîne « clustering» contient dix caractères, tandis que la chaîne « ClusteringAlgorithm » en contient dix-neuf. Cependant, si neuf espaces vides sont ajoutés à la chaîne « clustering », les deux chaînes ont la même longueur.

Le tableau (6.3) illustre l'algorithme de Hamming appliqué pour chercher la forme « clustering » dans le texte « ClusteringAlgorithm ». Les distances des symboles Ψ et la distance de Hamming λ_H sont alors calculées comme suit:

Tableau 6.3 La distance Hamming entre « clustering » et « ClusteringAlgorithm »

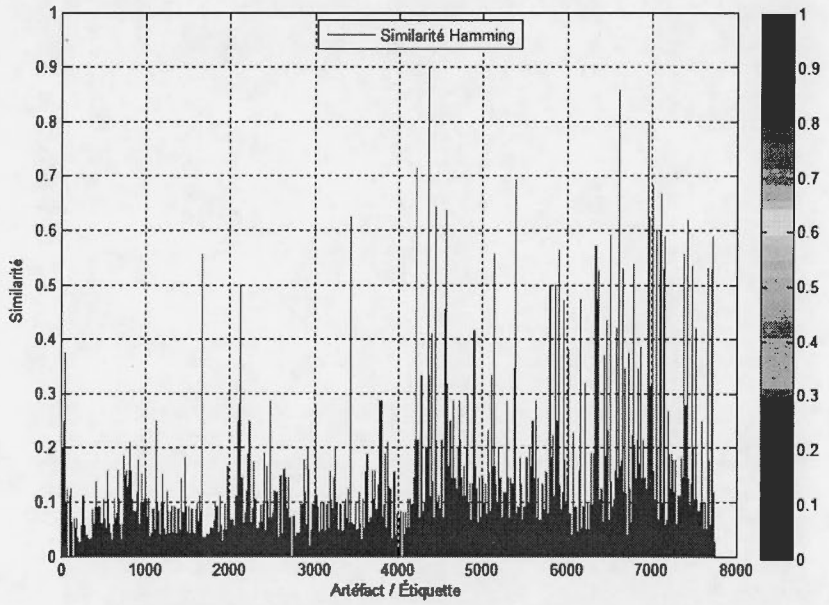
$Label_k^i$	c	l	u	s	t	e	r	i	n	g	_	_	_	_	_	_	_	_	_
c^i	C	l	u	s	t	e	r	i	n	g	A	l	g	o	r	i	t	h	m
Ψ	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
λ_H	0	0	0	0	0	0	0	0	0	0	1	2	3	4	5	6	7	8	9

Le tableau (6.4) montre la similarité calculée entre les étiquettes descriptives et les artéfacts ontologiques représentés à la figure (6.3) en utilisant l'alignement de Hamming.

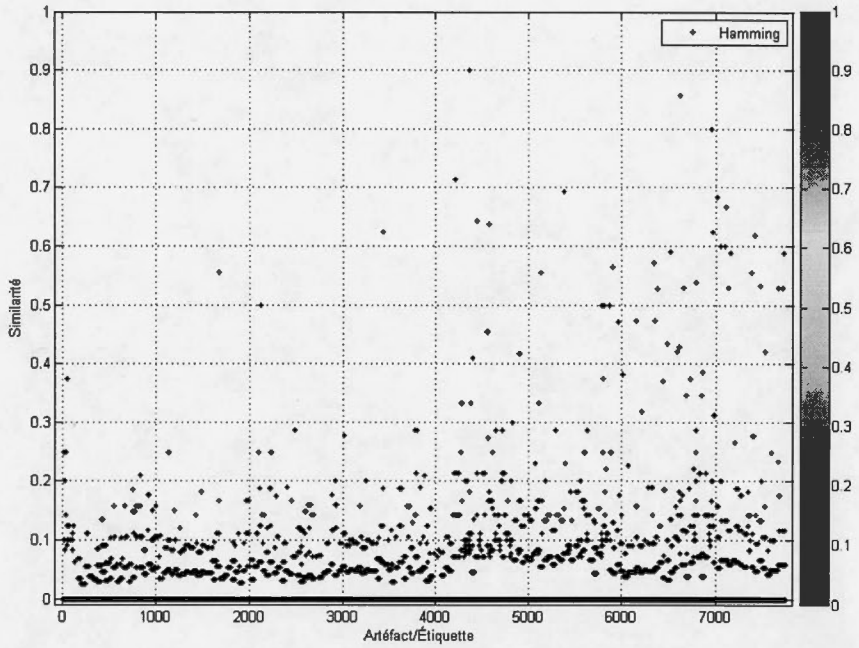
Tableau 6.4 L'alignement avec la similarité de Hamming

	Association Algorithm	Clustering Algorithm	Classification Algorithm	Regression Algorithm	Ruleinduction Algorithm
Classification	0.050	0.211	✓ 0.609	0.053	0.045
regression	0.000	0.053	0.043	✓ 0.526	0.045
association	✓ 0.550	0.000	0.043	0.000	0.000
clustering	0.000	✓ 0.526	0.174	0.053	0.000
neural	0.000	0.053	0.000	0.105	0.000
sampling	0.050	0.000	0.043	0.000	0.000
selection	0.050	0.053	0.000	0.053	0.091
boosting	0.050	0.105	0.087	0.000	0.000
bagging	0.000	0.000	0.000	0.053	0.091
Induction	0.050	✓ 0.053	0.000	0.000	✓ 0.000

La figure (6.5) montre les valeurs de similitude calculées entre les étiquettes générées par l'algorithme de clustering descriptif et l'ensemble des artéfacts dans l'ontologie CRISP-DM-OWL. L'axe horizontal représente l'alignement Hamming et l'axe vertical indique les valeurs de similitude.



a) Histogramme



b) Dispersion

Figure 6.5 La similarité de Hamming entre les concepts et les étiquettes descriptives

La troisième méthode utilise l'indice de Jaccard, également connu comme le coefficient de similitude de Jaccard défini comme la taille de l'intersection divisée par la taille de l'union des échantillons c^i et $Label_k^i$ (Shibata, Kajikawa et Sakata, 2010).

$$Cof_{sim_J}(C, Label_k) = J_\delta(Label_k, C) = \frac{|Label_k \cap C|}{|Label_k \cup C|} = \frac{|\{c^i\} \cap \{Label_k^i\}|}{|\{c^i\} \cup \{Label_k^i\}|}$$

$$\lambda_J(Label_k, C) = 1 - \frac{|Label_k \cap C|}{|Label_k \cup C|}$$

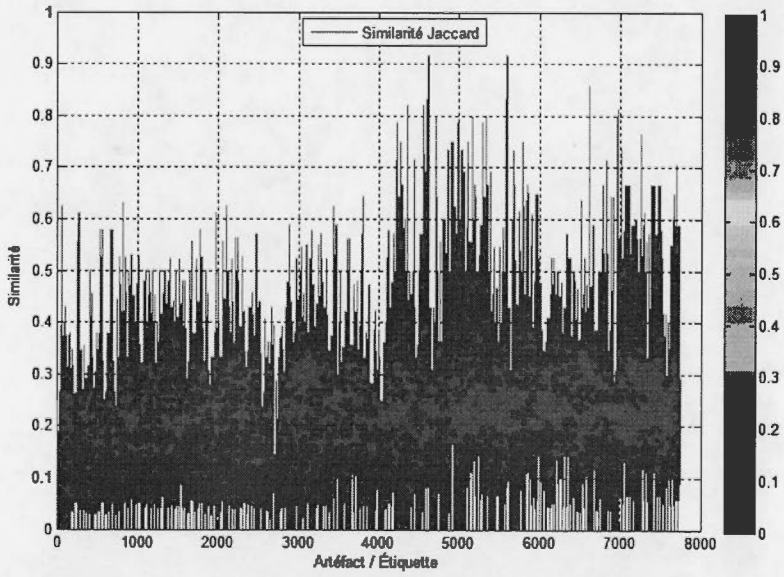
La complexité temporelle de l'algorithme ci-dessus est égale à $o(|C| \times |Label_k|)$.

Le tableau (6.5) montre la similarité de Jaccard calculée entre les étiquettes descriptives et les artefacts ontologiques représentés à la figure (6.3).

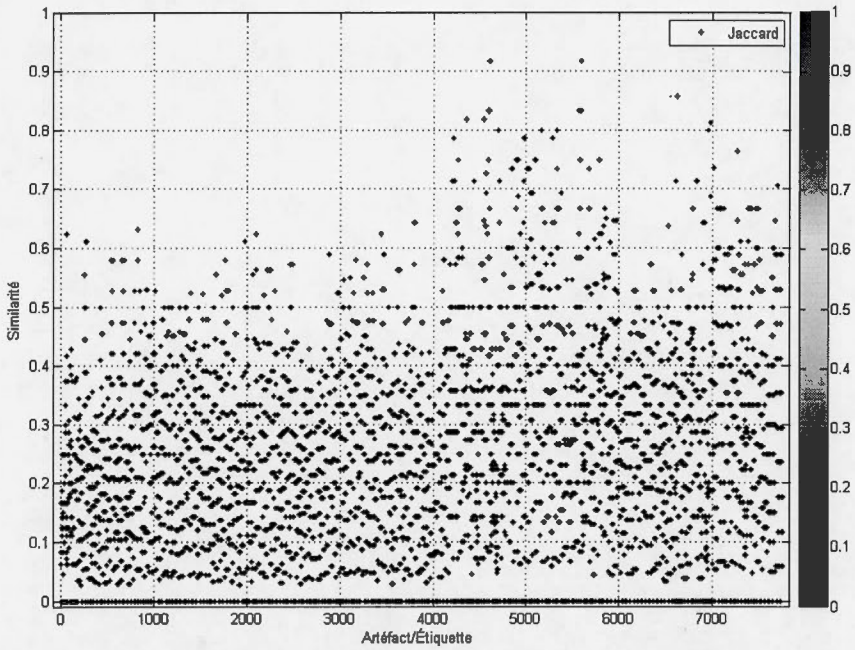
Tableau 6.5 Les résultats de l'alignement avec la similarité de Jaccard

	Association Algorithm	Clustering Algorithm	Classification Algorithm	Regression Algorithm	Ruleinduction Algorithm
Classification	0.650	✓0.684	0.609	0.579	0.500
regression	0.400	✓0.526	0.348	0.526	0.364
association	0.550	✓0.579	0.478	0.526	0.409
clustering	0.400	✓0.526	0.348	0.421	0.409
neural	0.200	0.316	0.174	0.263	0.273
sampling	0.350	0.368	0.304	0.368	0.273
selection	0.350	0.474	0.304	0.421	0.364
boosting	0.350	0.368	0.304	0.368	0.273
bagging	0.300	0.316	0.261	0.316	0.273
Induction	0.300	0.351	0.261	0.263	✓0.364

La figure (6.6) montre l'utilisation de l'indice de chevauchement Jaccard pour calculer les règles d'alignement. L'axe horizontal représente l'alignement Jaccard et l'axe vertical indique les valeurs de similitude calculées entre les étiquettes générées par l'algorithme de clustering descriptif et l'ensemble des artéfacts dans l'ontologie CRISP-DM-OWL.



a) Histogramme



b) Dispersion

Figure 6.6 La similarité de Jaccard entre les concepts et les étiquettes descriptives

À partir des tableaux (6.2), (6.4) et (6.5), nous pouvons voir clairement que les trois méthodes ont repéré les concepts « RegressionAlgorithm » et « ClusteringAlgorithm» qui correspondent aux étiquettes «regression» et « clustering ». Dans les trois résultats des étiquettes « association » et « classification », il est claire que la méthode de la Déformation Temporelle Dynamique et la méthode de Hamming ont sélectionné les deux concepts correspondants « AssociationAlgorithm » et « classificationAlgorithm » (respectivement), mais la méthode Jaccard ne trouve pas le bon concept parce que la similarité de Jaccard est variante selon les transformations et elle n'est pas suffisante pour la détection de l'appariement des contenus dupliqués dans les deux représentations. Elle est communément appliquée sur les données binaires où la fonction similitude calcule des valeurs binaires. Globalement, la mesure de distance calculée montre que la fonction de distance de Jaccard souffre de quelques inconvénients tels les transformations et les bruits. Pendant le calcul, nous avons trouvé que la distance de Jaccard est rigidement basée sur les points de chevauchement entre deux vecteurs. Par conséquent, s'il existe des occurrences de bruit qui n'appartiennent pas à la chaine de caractères, le nombre total de mutation exigé augmente aussi. Toutefois, ces limitations peuvent être surmontées par l'ajout de tâches de prétraitement. D'après les trois résultats de la chaîne « induction », la situation est inversée. En effet, la méthode de Hamming et la Déformation Temporelle Dynamique ne parviennent pas à trouver le concept correspondant dans l'ontologie, mais la méthode Jaccard a récupéré le bon concept «RuleinductionAlgorithm» parce que la distance de Hamming entre deux chaînes dépend

fortement de la position des caractères en commun. Similaire à la distance de Hamming, la distance de la Déformation Temporelle Dynamique nécessite un nombre élevé de primitives d'édition pour détecter l'alignement dans lequel les séquences d'opérations peuvent passer d'une chaine à l'autre. Les inconvénients liés à la distance de Hamming et à la distance de la Déformation Temporelle Dynamique sont surmontés par la distance de Jaccard qui considère n'importe quelle paire des sous-chaînes dans la procédure d'alignement.
Par conséquent, le choix d'une mesure de distance appropriée pour répondre aux besoins d'applications est une tâche cruciale et une attention particulière devrait être accordée à la sélection d'une mesure appropriée pour chaque alignement individuel.

À noter aussi qu'il existe quelques lacunes dans les mesures de distance utilisées dans les processus individuels d'alignement:

- Les fonctions de distance nécessitent des tâches approfondies de prétraitement telles que la suppression du bruit.
- Les fonctions de distance sont sensibles aux transformations de la forme (translation, rotation, transposition, mise à l'échelle, etc.).
- La séquence des opérations d'alignement et le coût d'édition ne sont pas uniques en général.
- Les algorithmes varient selon le type de recherche et les méthodes utilisées pour réaliser la transformation optimale.
- Le choix de la mesure de distance est étroitement lié à la détermination d'un alignement optimal.
- L'alignement dépend fortement du type d'erreur considérée et du coût computationnel d'exécution (Djellali, 2013h) et (Djellali, 2013e) .

Afin de remédier à ces problème et élaborer un compromis des diverses décisions, nous allons discuter dans le prochain paragraphe un processus d'alignement avec agrégation.

6.3 Agrégation

Une manière naturelle d'arbitrer entre les processus individuels d'alignement consiste à prendre une agrégation des règles d'alignement, de telle sorte que le résultat agrégé prenne en compte toutes les valeurs de similarité individuelles. Il s'agit d'un processus qui combine les décisions des alignements individuels de manière à en obtenir une seule. Cette technique est étudiée dans plusieurs champs de recherche, en particulier, la logique floue, la planification, le Data Mining, l'entreposage des données, la prise de décision, l'apprentissage machine, etc.

Il existe une vaste panoplie d'opérateurs d'agrégation utilisés dans la littérature, parmi ceux-ci, on retrouve: les opérateurs conjonctifs (norme triangulaire, OU flou, Dombi, Frank, etc.), disjonctifs (conorme triangulaire, Hamacher, etc.), de compromis (Min-Max ordonné pondéré, opérateur OWA (Ordered Weighted Averaging), GOWA, OWA généralisé induit ou GIOWA, AWFO, MEOWA, AWFO, intégrales floues, moyenne quasi-arithmétique, moyenne ordonnée, etc.), de compensation (l'opérateur γ , combinaison exponentielle, somme symétrique, etc.), de renforcement (le triple, noble, t-normes et t-conormes, identité commutative monotone ou MICA (Monotonic Identity Commutative Aggregator), etc.), pondérés (quasi-arithmétique pondérée, la moyenne arithmétique, moyenne pondérée, minimum et maximum pondérés, etc.). Cette étage d'agrégation, à elle seule, fait l'objet de plusieurs améliorations dans les articles de (Bahi, Guyeux et Makhoul, 2010), (Dandach, Carli et Bullo, 2010) et la thèse de (Le Capitaine, 2009).

Afin d'élaborer un compromis des diverses décisions, nous avons utilisé le critère de la moyenne pondérée (ou un vote) pour agréger les similarités calculées. Ainsi, l'agrégation des valeurs de similitude est donnée par l'équation suivante:

$$Cof_{sim_A}(Label_k, C) = \sum_i w_i Cof_{sim_i} \text{, où les } w_i \in [0,1] \wedge \sum_i w_i = 1.$$

La fonction de similarité Cof_{sim_A} permet d'agréger les indices de similarités individuels Cof_{sim_i} et dépend de l'importance de la pondération w_i . Les facteurs de pondérations w_i vont d'abord modifier le degré d'influence des indices de similarités individuelles Cof_{sim_i} sur la procédure d'agrégation et ainsi commanderont le niveau global d'arbitrage Cof_{sim_A} . Par souci de simplicité, nous avons supposé que l'utilisateur ne peut pas contrôler explicitement la complexité d'agrégation. En effet, nous avons quantifié les similarités calculées par poids unitaire à chacun des alignements individuels. Il est cependant possible de compléter le processus d'agrégation en considérant également des facteurs différents, la procédure étant analogue.

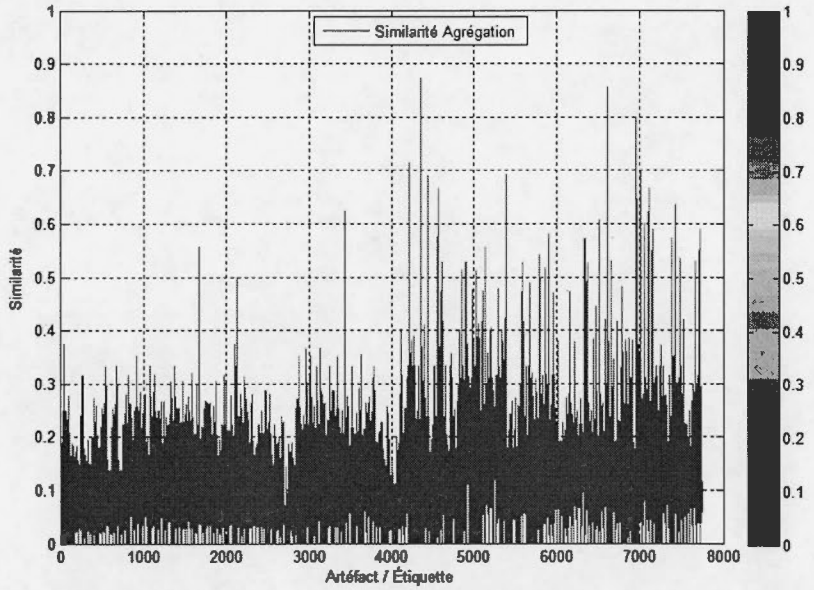
Le tableau (6.6) montre la moyenne pondérée des similarités calculées entre les étiquettes descriptives et les artefacts ontologiques représentés à la figure (6.3).

Tableau 6.6 Les résultats de l'alignement avec la méthode d'agrégation

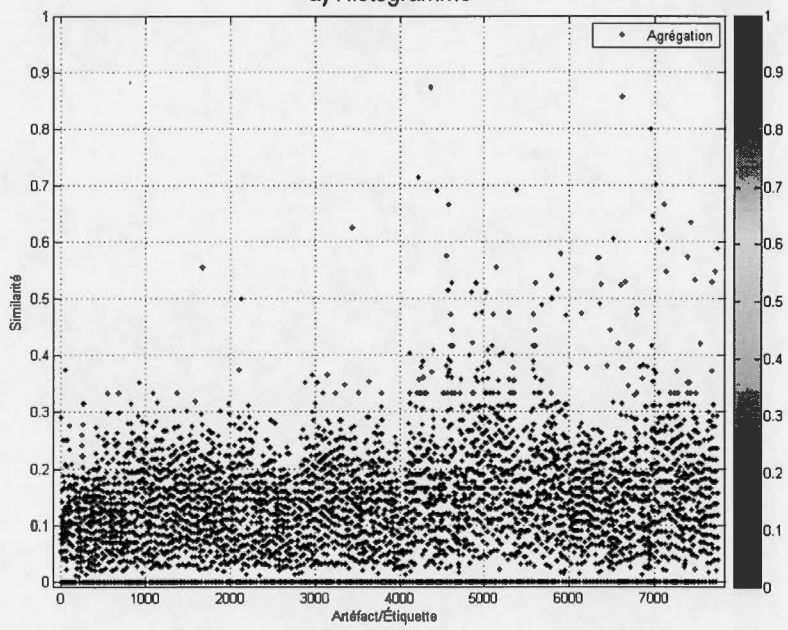
	Association Algorithm	Clustering Algorithm	Classification Algorithm	Regression Algorithm	Ruleinduction Algorithm
Classification	0.333	0.404	0.609	0.281	0.242
regression	0.183	0.246	0.174	0.526	0.212
association	0.550	0.263	0.304	0.246	0.197
clustering	0.183	0.526	0.261	0.211	0.212
neural	0.100	0.193	0.087	0.193	0.152
sampling	0.200	0.175	0.174	0.175	0.136
selection	0.217	0.246	0.188	0.246	0.258
boosting	0.217	0.245	0.203	0.193	0.152
bagging	0.150	0.158	0.145	0.193	0.167
Induction	0.200	0.193	0.159	0.140	0.204

Si nous utilisons le processus d'alignement avec agrégation (tableau (6.6)), nous pouvons conclure que les étiquettes «association», « classification », « regression », « clustering » et « induction » correspondent aux concepts «AssociationAlgorithm»,«ClassificationAlgorithm», «RegressionAlgorithm»,« ClusteringAlgorithm » et « RuleinductionAlgorithm » dans l'ontologie.

La figure (6.7) montre l'utilisation de l'agrégation pour calculer les règles d'alignement. L'axe horizontal représente l'alignement d'agrégation et l'axe vertical indique les valeurs de similitude calculées entre les étiquettes générées par l'algorithme de clustering descriptif et l'ensemble des artéfacts dans l'ontologie CRISP-DM-OWL.



a) Histogramme



b) Dispersion

Figure 6.7 La similarité d'agrégation entre les concepts et les étiquettes descriptives

Afin de conclure ce paragraphe, on peut dire que chaque alignement individuel est évidemment moins performant mais l'agrégation conduit finalement à un alignement optimal. L'agrégation des valeurs de similitude repose sur le concept de moyenne pondérée avec des facteurs unitaires. De ce fait, le processus

d'agrégation que l'on a proposé ne dépend pas donc du poids accordé à chaque processus individuels d'alignement.

Les paragraphes qui suivent présentent la dispersion des valeurs de similitude, la corrélation entre les processus individuels d'alignement ainsi que leurs évaluations.

6.4 Corrélation

Nous avons utilisé les graphiques de dispersion et la distribution polaire pour identifier les dépendances entre les variables de la Déformation Temporelle Dynamique, Hamming, Jaccard représentées sur chacun des axes. Chaque point représente une valeur de similitude et constitue une partie du nuage de points.

La figure (6.8) présente un diagramme de dispersion utilisant les coordonnées cartésiennes pour afficher les valeurs de similarité. Il s'agit d'une représentation qui utilise les formules trigonométriques pour exprimer la relation entre les trois variables. La matrice de similitude est affichée comme une collection d'objets [X,Y]. L'abscisse X sur l'axe horizontal montre une valeur de similarité et l'ordonnée Y sur l'axe vertical montre une autre. L'histogramme représente la répartition des valeurs de similitude.

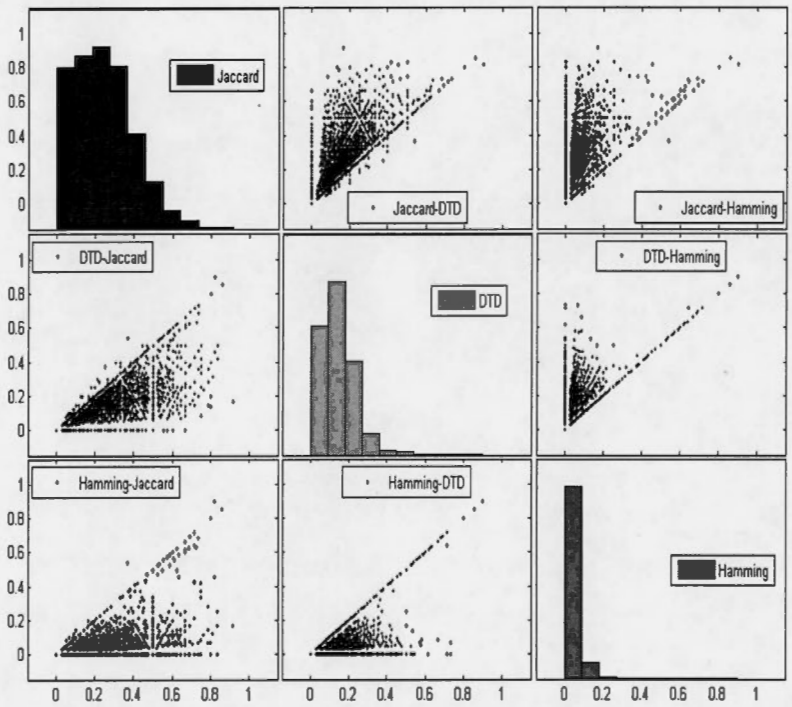


Figure 6.8 Diagramme de dispersion des valeurs de similitude

La figure (6.9) montre la dispersion des valeurs de similarité avec les coordonnées polaires. Cette représentation est particulièrement utile parce que la relation entre les valeurs de similitude est facilement exprimée en termes d'angle et de distance. Dans ce système à deux dimensions, chaque point est déterminé par les coordonnées polaires c'est-à-dire, les coordonnées radiales et angulaires.

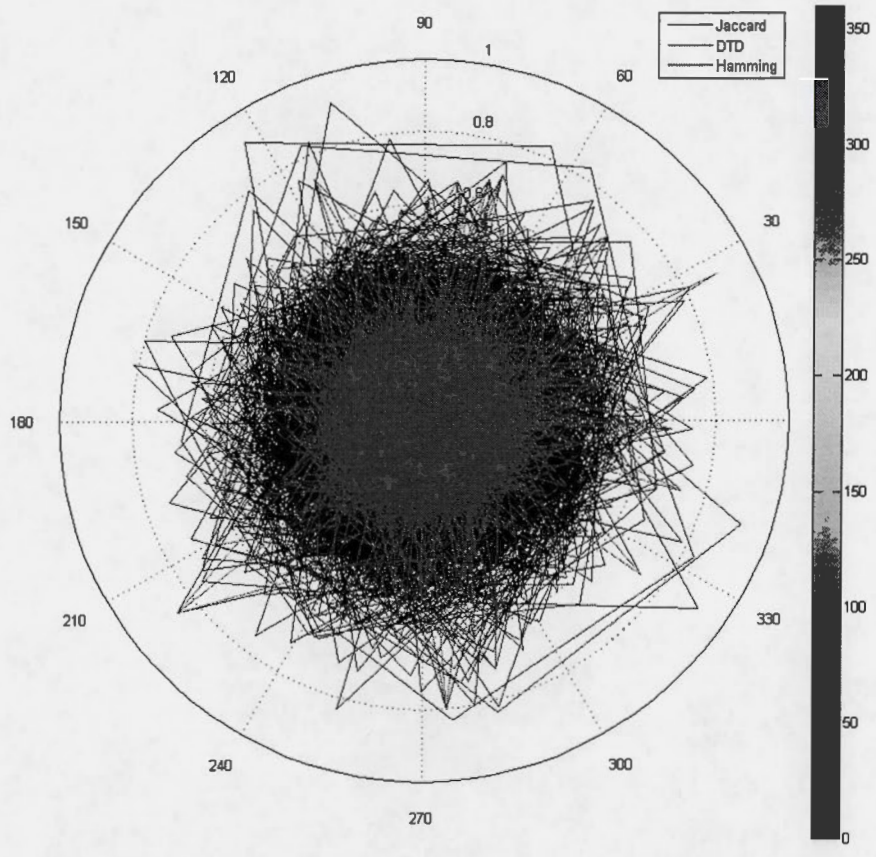


Figure 6.9 La distribution des cordonnées polaires des valeurs de similitude

Nous avons utilisé l'outil de corrélation de Pearson (Chartier, Meunier et Djellali, 2010) pour rechercher l'indice numérique qui décrit le degré corrélatif entre les trois variables DTD, Hamming et Jaccard. Cette mesure sert à caractériser une relation linéaire positive ou négative. Le degré corrélatif est égal à (1) dans le cas d'une parfaite corrélation et (-1) dans le cas d'une parfaite anti-corrélation. Une valeur comprise entre (-1) et (1) indique le degré de dépendance entre les variables et cette valeur se rapproche de zéro pour les variables non corrélées. L'indice numérique de Pearson est donné par l'équation suivante:

$$r_P = \frac{\sum_{i=1}^{N}(s_i - \bar{s}) \cdot (s_i^{'} - \bar{s}^{'})}{\sqrt{\sum_{i=1}^{N}(s_i - \bar{s})^2} \cdot \sqrt{\sum_{i=1}^{N}(s_i^{'} - \bar{s}^{'})^2 \cdot}}$$

N: est la taille de l'échantillon.

s_i, $s_i^{'}$: les valeurs de similitude calculées.

$\bar{s}$, $\bar{s}^{'}$: la valeur de similitude moyenne.

Dans le tableau (6.7), nous illustrons les types de liaisons qui peuvent exister entre les trois variables.

Tableau 6.7 Les mesures corrélatives entres les variables

Mesure	DTD-Jaccard	DTD- Hamming	Jaccard-Hamming
Corrélation	0.7373	0.4840	0.3766
Covariance	0.0109	0.0028	0.0037

Selon les statistiques, il est évident que la variable de la Déformation Temporelle Dynamique est liée avec l'ensemble des variables. La variable de la Déformation Temporelle Dynamique est corrélée positivement avec la variable Jaccard. La corrélation avec Jaccard semble plus élevée que la corrélation avec Hamming. Le degré corrélatif entre Jaccard et Hamming est faible.

La comparaison entre les résultats des graphiques de dispersion (figure 6.8) et les cordonnées polaires (figure 6.9) et les résultats montrés dans le tableau (6.7) confirme qu'il Il y a une forte liaison entre la similarité de la Déformation Temporelle Dynamique et Jaccard.

6.5 Évaluation

La qualité de l'alignement dépend fortement de la complexité de la distance et pour mesurer cette qualité, nous utilisons la métrique standard utilisée dans le repérage d'information (Kiu et Lee, 2006).

- Le rappel: décrit le nombre de règles d'alignement valides par rapport au nombre total de règles d'alignement existantes.

$$r = \frac{|règles\ d'\ alignement\ valides|}{|règles\ d'\ alignement\ possibles|}$$

- La précision: décrit le nombre de règles d'alignement valides versus le nombre de règles d'alignement recherchées (valide et non valide).

$$p = \frac{|règles\ d'alignement\ valides|}{|règles\ d'alignement\ trouvées|}$$

- F-mesure : l'indice équilibré F-mesure négocie le compromis précision-rappel, est défini par la formule suivante:

$$\text{F - mesure} = \frac{2 \cdot p \cdot r}{p + r}$$

D'après ces résultats montrés dans le tableau (6.8), nous pouvons conclure que la méthode d'agrégation améliore considérablement la performance du processus d'alignement parce qu'elle ignore les processus individuels d'alignement générant des taux d'erreurs extrêmes. Elle combine les résultats estimés par les processus individuels d'alignements en utilisant un estimateur robuste de précision, c'est-à-dire, l'agrégation (Djellali, Meunier et Delisle, 2012) .

Tableau 6.8 La performance des alignements

Alignement	Précision %	Rappel %	F-mesure %
DTD	91.75	80.01	85.47
Jaccard	71.13	95.74	81.62
Hamming	82.81	77.26	79.93
Agrégation	94.07	87.78	90.81

Conclusion

Nous avons utilisé des processus individuels d'alignement pour achever l'interopérabilité entre les étiquettes descriptives et les artéfacts ontologique. La méthode de la Déformation Temporelle Dynamique est très efficace parce que sa version généralisée est assez puissante pour plusieurs applications mais elle est très coûteuse. La méthode de Jaccard exécute une recherche pour les points de chevauchement entre deux chaînes et nécessite un temps de calcul minimal. Dans ce sens, la distance de Jaccard est sensible aux transformations de la forme mais elle est meilleure dans la reconnaissance des formes binaires. L'effort computationnel nécessaire pour calculer la distance de Hamming dépend linéairement de la taille des chaînes et la performance de calcul se dégrade rapidement si la taille augmente. Les méthodes approximatives et les heuristiques sont des approches à suivre pour surmonter ces problèmes. Ces approches utilisent des hypothèses fondées sur l'ensemble de données pour accélérer le calcul. Cela permet de converger rapidement au voisinage de la solution mais elles peuvent conduire à un alignement sous-optimal.

À partir de notre étude nous pouvons conclure que chaque méthode a des inconvénients et des avantages. Une seule méthode ne peut pas compléter le processus d'alignement parce qu'elle n'exploite qu'une partie des informations dans l'ontologie.
Le processus utilisant l'agrégation que nous avons proposé dans ce chapitre, la moyenne pondérée, permet d'élaborer un compromis des diverses décisions, ce qui nous permettra de former un alignement faible et robuste.

Dans le chapitre suivant, nous présenterons l'outil pour implémenter et mettre à jour l'ontologie, le modèle calculable utilisé pour représenter l'ontologie enrichie, les techniques de raisonnement utilisées pour vérifier la cohérence et l'outil de visualisation utilisé pour explorer et comprendre la structure ontologique.

CHAPITRE VII

MISE À JOUR

Résumé: L'objectif principal de ce chapitre est de présenter l'étape de mise à jour supportant le processus d'évolution et ainsi la représentation explicite de l'ontologie dans un langage formel.

Étant donné la composante d'évaluation de la performance dans ce projet de recherche, il est essentiel de présenter certains concepts théoriques sur ce thème. Tout d'abord, la revue de littérature traite une vue d'ensemble des erreurs qui peuvent se produire pendant la construction des taxonomies et donne un bref aperçu des approches qui ont été présentées dans la littérature pour l'évaluation des ontologies. Une rapide mise en contexte sur notre approche d'évaluation est alors proposée. L'axe d'investigation s'intéresse à la consistance, la correction et la complétude, et plus particulièrement les caractéristiques principales de la hiérarchie conceptuelle aussi bien que l'inférence ontologique. L'évaluation par un outil de raisonnement automatique est ensuite présentée. Il s'agit d'un système d'inférence permettant de vérifier le raisonnement terminologique et la description du monde et, entre autres, d'évaluer la cohérence, la subsomption, l'instanciation, les cycles et les points fixes. Pour terminer, nous présenterons notre architecture logicielle et le protocole de déploiement.

La suite du chapitre est organisée de la manière suivante. Le paragraphe (7.1) décrit l'étape de mise à jour qui utilise le Plug-in OWL Protégé comme un modèle de référence pour enrichir l'ontologie et pour décrire le modèle calculable. Dans le paragraphe (7.2), nous passons brièvement en revue les principales approches d'évaluation et nous présentons l'étape de vérification de la consistance. Cette étape utilise un outil de raisonnement en tant que système expert distribué basé sur des structures symboliques de faits et de règles. L'évaluation s'articule sur la méthode des tableaux sémantiques définis dans un système formel basée sur la logique descriptive $\mathcal{SHOIN}$. Nous décrivons dans le paragraphe (7.3) un outil de visualisation permettant d'explorer les associations et les tendances des connaissances ontologiques. Puisque le succès d'une ontologie dépend fortement des définitions consensuelles, nous proposons dans le paragraphe (7.4), un outil pour adresser la documentation des définitions des artéfacts ontologiques. Le paragraphe (7.5) présente l'architecture logicielle, l'implémentation et le protocole utilisé pour déployer le système.

7.1 La mise à jour

Le processus de mise à jour permet d'établir un modèle calculable dans un langage formel. C'est une représentation explicite de la conceptualisation acquise dans l'étape d'extraction en utilisant un langage formel. L'étape mise à jour implique en particulier de:

1. Choisir un langage de représentation pour encoder l'ontologie: OWL-DL est le code calculable utilisé pour représenter l'ontologie enrichie d'une façon structurée et formelle. Ce langage a une sémantique qui peut être décrite via une traduction dans une expression de la logique descriptive $\mathcal{SHOIN}$, c'est-à-dire, un sous-ensemble de la Logique de Prédicat de premier Ordre ou LPO (FOL de l'anglais: First Order Logic). Il utilise une syntaxe convenable basée sur RDFS et fournit un raisonnement décidable. Ce langage est conçu pour traiter le contenu plutôt que la présentation de l'information. L'ontologie enrichie OWL-DL est représentée à l'aide d'un document référé par le biais d'un identifiant uniforme de ressource. Le document fait référence à des annotations non logiques qui contiennent la description de l'auteur et d'autres informations non logiques associées à la réutilisation. Il décrit l'ontologie enrichie à travers un ensemble d'axiomes qui affirment des faits, les relations de subsomption entre les concepts, les assertions, les restrictions des propriétés, des références, etc.
2. Décrire les spécifications de l'ontologie suivant l'éditeur choisi: la maintenance de l'ontologie est réalisée en utilisant le plug-in OWL[8] (extension de Protégé). Ce plug-in est utilisé pour modifier l'ontologie, accéder aux outils de raisonnement basés sur la logique de description et pour acquérir la structure taxonomique. Il étend le modèle Protégé avec des fonctionnalités pour représenter la spécification OWL-DL[9] et est prend en charge trois types de raisonnement DL: la vérification de la cohérence, la subsomption et l'instanciation.

   Le processus de mise à jour offre une vue orientée objet pour l'ontologie et se sert des techniques orientées objet pour représenter les entités de l'ontologie. Il décrit la structure de l'ontologie en termes de classes et des propriétés similaire à l'approche orientée objet. Ceci donne aux développeurs une interface orienté objet convenable pour la maintenance des ontologies. L'ontologie enrichie est représentée comme un gabarit configurable et tous les objets sont des instances de ce gabarit.

Dans le prochain paragraphe, nous allons nous intéresser plus spécifiquement aux approches été présentées dans la littérature pour l'évaluation des ontologies. Nous allons passer succinctement les

[8] http://protege.stanford.edu/
[9] http://www.w3.org/TR/owl-features/

avantages et les inconvénients de chaque processus. Ensuite, nous proposons une méthode d'évaluation de la cohérence de l'ontologie enrichie à l'aide d'un système formel.

7.2 Évaluation

Ces dernières années, l'évaluation des ontologies a émergé en ayant un impact spectaculaire sur le Web sémantique et le repérage d'information. C'est une méthode en plein essor qui concerne à la fois plusieurs aspects: la consistance, la complétude, la redondance, l'interopérabilité, la sémantique, la structure hiérarchique, etc.

(Gómez-Pérez, 1996), p.2 définit l'évaluation comme :*« la méthode qui permet de contrôler la bonne réalisation du contenu de l'ontologie en assurant que les définitions introduites dans l'ontologie implémentent correctement les besoins de l'ontologie et les questions de compétence ».*

L'évaluation n'a pas eu uniquement un impact sur les caractéristiques principales de l'ontologie, mais également sur les aspects d'inférence ontologique et les questions de compétence. Dans ce contexte, (Gruber, 1995) (p.2.3) a proposé des critères de conception afin de supporter l'interopérabilité dans systèmes d'intégration basées sur les ontologies. Ces critères peuvent être regroupés comme suit:

- La clarté: décrire la signification appropriée des artéfacts ontologiques.
- La cohérence: se rapporte au fait qu'elle devrait garantir la consistance entre la connaissance inférée et les axiomes définis dans l'ontologie.
- L'extensibilité: fait référence à une structure dynamique, c'est-à-dire, l'évolution et l'amélioration de l'ontologie pour s'assurer qu'elle reflète le modèle de données.
- Le biais d'encodage minimal: reflète la notion de la représentation intermédiaire.
- L'engagement ontologique minimal: se rapporte au fait qu'elle devrait garantir une expressivité suffisante pour soutenir le partage des connaissances.
- L'abstraction: une description indépendante de l'implémentation.

Comme précisé précédemment, l'évaluation des ontologies concerne plusieurs aspects. Un des aspects spécifiques concerne les erreurs taxonomiques. Dans ce contexte, (Gómez-Pérez, 1999) a donné une classification complète des erreurs qui peuvent se produire pendant la construction des taxonomies. Ces erreurs sont classées suivant les critères d'inconsistance, d'incomplétude et de redondance.

Le tableau (7.1) montre une vue d'ensemble des erreurs taxonomiques selon le critère.

Tableau 7.1 Les erreurs taxonomiques (Gómez-Pérez, 1999)

Inconsistance	**Les erreurs de regroupement**	Partition des sous classes avec des instances (des classes) communes.
		Partition exhaustive des sous classes avec des instances (classes) communes.
		Partition exhaustive des sous classes avec des instances externes.
Incomplétude	**Les erreurs circulaires**	Définitions circulaires.
	Les erreurs de regroupement	Omission de la partition des sous classes.
		Omission de la partition exhaustive des sous classes.
Redondance	**Syntaxe**	Définitions identiques des classes (instances).
		Redondance des relations d'héritage entre les classes (instances).
	Sémantique	Erreurs grammaticales.

Généralement, il existe quatre approches qui ont été présentées dans la littérature pour l'évaluation des ontologies:

- Gold standard: une comparaison suivant un repère d'étalonnage en utilisant l'ontologie de référence gold standard.
- Évaluation basée sur les données: évaluation suivant le degré d'ajustement entre l'ontologie et l'ensemble de données décrivant le domaine d'intérêt. Par conséquent, cette approche nécessite des mécanismes de traçabilité et la réingénierie pour décrire les rapports concrets entre les entités de l'ontologie et l'ensemble de données.
- Évaluation basée sur les critères: une évaluation qualitative suivant les caractéristiques de l'ontologie indépendamment du domaine d'application. Les critères les plus connus peuvent être regroupés comme montrés dans le tableau (7.2).

Tableau 7.2 Critères des ontologies (Lozano-Tello et Gómez-Pérez, 2004)

Auteur	Année	Nombre de critères	Objectif
Gruber	95	6	Critères de conception
Uschold &Grüninger	96	3	Critères de conception
Noy & Hafner	97	28	Critères de conception
Hovy	97	36	Comparer des ontologies linguistiques
Uschold	98	10	Identifier les rôles de l'ontologie dans les applications

Cependant, cette approche est basée sur l'expérience du développement des ontologies et de ce fait elle présente l'inconvénient d'être assujettie aux préjugés des évaluations extrêmes.

- *L'évaluation basée sur les tâches*: l'accomplissement d'une tâche donnée pilote le choix d'une ontologie (Brank, Grobelnik et Mladenić, 2005).

Le tableau (7.3) donne un sommaire d'une évaluation décomposée en plusieurs couches ou une évaluation par niveau (lexicologique, hiérarchique, sémantique, contextuelle, syntaxique, architecturelle) selon l'approche.

Tableau 7.3 Approches d'évaluation des ontologies (Brank, Grobelnik et Mladenić, 2005)

Niveau d'évaluation	Evaluation			
	Golden Standard	Application	Données	Critères
Lexique, vocabulaire, concept, données	✔	✔	✔	✔
Hiérarchie, taxonomie	✔	✔	✔	✔
Autres relations sémantiques	✔	✔	✔	✔
Contexte, application		✔		✔
Syntaxique	✔			✔
Structure, architecture, conception				✔

Les différentes approches proposées dans la littérature consistent à consulter des experts qui utilisent leurs avis ainsi que leurs expériences dans le processus d'évaluation. Cependant, elles présentent l'inconvénient d'être assujetties aux préjugés des évaluations extrêmes. De plus, le choix d'une approche appropriée dépend des critères utilisés et du domaine modélisé.

Afin de surmonter ces inconvénients, nous proposons une approche d'évaluation qui se base sur la propriété de correction, de complétude et de décidabilité. En effet, la cohérence, la subsomption[10] et l'instanciation peuvent accentuer les caractéristiques principales de la hiérarchie conceptuelle aussi bien que l'inférence ontologique. Elles permettent de prendre des décisions au sujet de la consistance, la structuration de l'ontologie et la complétude. La logique descriptive est parfaitement adaptée à cette situation. Elle a une sémantique formelle basée sur la logique et équipée de procédures de preuve décidables. De plus, la logique descriptive offre plusieurs avantages, entre autres:

- La propriété de correction (soundness), de complétude (completeness) et de décidabilité.
- La possibilité de transformer une représentation descriptive en une représentation de la logique des prédicats du premier ordre, l'inverse en général n'est pas possible.
- L'efficacité du raisonnement par classification.
- Possède une sémantique bien définie.
- La facilité de modélisation des bases de données et les ontologies.
- La dualité expressivité versus complexité.
- Deux niveaux pour représenter la connaissance: la terminologie et la description du monde.
- La subsomption et l'instanciation sont les opérations qui sont à la base du raisonnement terminologique.

Comme montré à la figure (7.1), le système d'inférence descriptive utilisé pour vérifier la consistance, la correction et la complétude de l'ontologie est basé sur le moteur d'inférence RacerPro (**R**enamed **A**Box and **C**oncept **E**xpression **R**easoner) (Haarslev *et al.*, 2011). Ainsi, nous pouvons considérer l'outil de raisonnement en tant que système expert distribué basé sur des structures de faits et de règles de type si-alors.

Il faut noter que l'ontologie enrichie $O = (C, H_C, R_C, H_R, I, R_I, A)$ est considérée en deux parties:

- La partie extensionnelle C, R_C ou $\mathcal{T}$-Box: la représentation et la manipulation des concepts et les rôles dans un niveau terminologique. Les concepts et les rôles peuvent être primitifs ou définis. Les rôles définissent les relations entre les instances des concepts.
- La partie intensionnelle I, R_I ou $\mathcal{A}$-Box: la représentation et la manipulation des individus dans un niveau factuel ou assertionnel.

Les concepts, les rôles et les individus obéissent aux principes suivants:

- Une description structurée avec une sémantique formelle.

[10] Un concept subsumant est un concept plus général.

- Des hiérarchies de connaissances H_C et H_R.
- Une relation de subsomption pour exhiber un ordre bien fondé sur la hiérarchie H_C.
- Une relation d'instanciation pour tester la validité de l'assertion $A(o), \forall A \in C \wedge o \in I$ (Napoli, 1997).

Le protocole DIG (Bechhofer, Möller et Crowther, 2003) est utilisé pour connecter le système de Data Mining au système d'inférence. De cette façon, nous pouvons repérer des connaissances terminologiques à partir de l'ontologie. Par conséquent, le système d'inférence permettant de vérifier la cohérence est divisé en trois composantes principales: le moteur d'inférence, la terminologie $\mathcal{T}$-Box et la description du monde $\mathcal{A}$ -Box. Dans chacune de ces composantes, les connaissances sont déclarées sous forme de règles ou de faits. Les règles sont liées aux opérations terminologiques (la subsomption et l'instanciation).

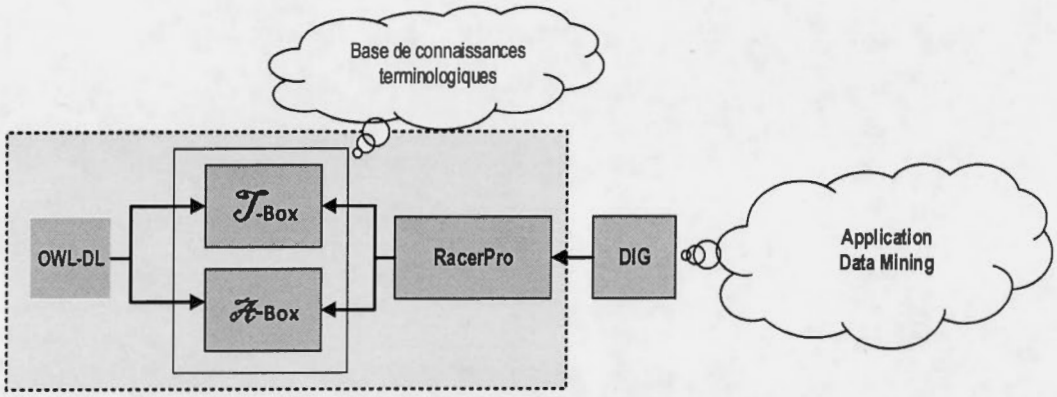


Figure 7.1 Le système d'inférence descriptive

7.2.1 Le moteur d'inférence

Le moteur d'inférence RacerPro fournit plusieurs règles d'inférence qui déduisent la connaissance implicite à partir de la connaissance représentée explicitement dans l'ontologie enrichie. Il existe quatre opérations qui sont à la base du raisonnement terminologique dans le moteur d'inférence RacerPro: le test de subsomption, de satisfiabilité, d'instanciation; de consistance de la base terminologique. Ces règles d'inférence sont décidables et avec une faible complexité. Elles permettent de transformer les artefacts de l'ontologie sous forme de règles descriptives pour appuyer le raisonnement formel. Les règles sont classées par catégorie et arrangées selon le niveau de complexité.

Comme nous l'avons mentionné en introduction de ce chapitre, le système d'inférence RacerPro est basé sur la logique descriptive $\mathcal{SHOIN}$. Il utilise la méthode des tableaux sémantiques comme procédures de décision dans le processus de raisonnement. Il adopte également les sémantiques des points fixes afin d'éviter les définitions dans la terminologie qui contiennent des cycles, c'est-à-dire, la terminaison de réécriture (Baader, 2003). De plus, le système d'inférence RacerPro offre plusieurs techniques d'optimisation avec des outils de preuve:

- L'architecture RacerPro comprend plusieurs techniques pour garantir une bonne performance de la recherche, en particulier, le chainage arrière de la dépendance dirigée et le branchement sémantique.
- La vérification de la consistance conceptuelle.
- La recherche des concepts inconsistants dans la terminologie $\mathcal{T}$-Box.
- La détermination des parents et des enfants d'un concept.
- La vérification de la consistance de la description du monde $\mathcal{A}$-Box.
- Tester la description du monde dans $\mathcal{A}$-Box et $\mathcal{T}$-Box.
- Retrouver le subsumé et le subsumant dans la terminologie $\mathcal{T}$-Box ainsi que la description du monde $\mathcal{A}$-Box.
- Calculer les types directs d'individus (Haarslev et Möller, 2003).

L'annexe (14) montre la connexion TCP/IP avec le serveur RacerPro.

7.2.2 La terminologie $\mathcal{T}$-Box

Cette composante contient les axiomes terminologiques qui décrivent la relation entre les concepts et les rôles. Ainsi, RacerPro déclare tous les axiomes terminologiques qui ont la forme suivante:

$$A \subseteq B(r_2 \subseteq r_2) \vee A \equiv B(r_1 \equiv r_2)(\forall A, B \in O\ , \forall r_1, r_2 \in R_O))$$

Les axiomes du premier type sont appelés les inclusions alors que les axiomes du second type sont appelés les égalités.

Comme nous l'avons précisé précédemment, la subsomption est l'inférence de base sur les expressions conceptuelle dans RacerPro, typiquement notée $A \subseteq B$. Afin de vérifier ce type de relation, le moteur d'inférence considère les relations définies dans la terminologie de l'ontologie enrichie CRISP-DM-OWL.

Plusieurs règles peuvent se présenter dans la terminologie $\mathcal{T}$-Box:

- **R1:** $(A \equiv B) \leftrightarrow (B \subseteq A \wedge A \subseteq B)$.
- **R2:** $\neg(A \equiv B) \leftrightarrow (A \cap B \subseteq \perp)$.
- **R3:** $(A \subseteq B) \leftrightarrow ((A \cap \neg B)^I = \phi)$.
- **R4:** $(A \equiv B) \leftrightarrow ((A \cap \neg B)^I = \phi \wedge (\neg A \cap B)^I = \phi)$.
- **R5:** $\neg(A \equiv B) \leftrightarrow ((A \cap B)^I = \phi)$.
- **R6:** $(A^I = \phi) \rightarrow (A \subseteq \perp)$.
- , etc. $\forall A, B \in$ CRISP-DM-OWL.

Pour des raisons de lisibilité, nous ne pouvons pas présenter toutes les règles.

La figure (7.2) montre le traitement de la cohérence en utilisant le raisonnement terminologique de la subsomption pour les axiomes terminologique $\mathcal{T}$-Box de l'ontologie enrichie.

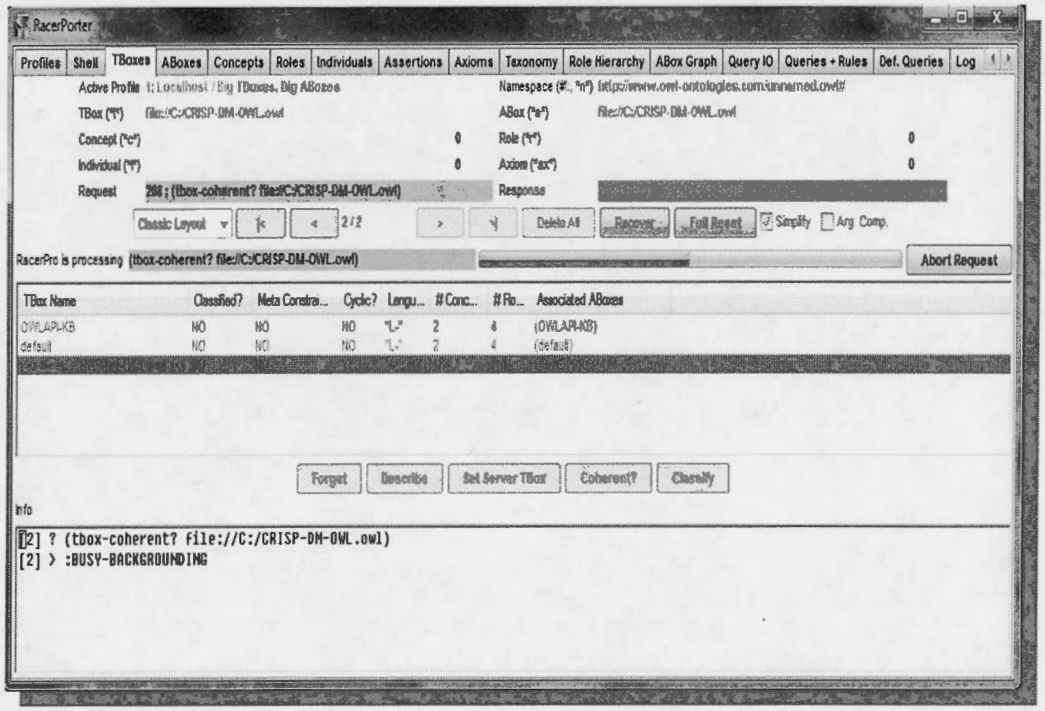


Figure 7.2 Le raisonnement terminologique RacerPro de la terminologie $\mathcal{T}$-Box

7.2.3 La description du monde $\mathcal{A}$-Box

La deuxième composante de la base de connaissances terminologiques est la description du monde $\mathcal{A}$-Box. Les entités dans ce monde peuvent prendre la forme suivante:

$A(o_1), r(o_2, o_3), (\forall A \in \text{CRISP-DM-OWL},\ \forall o_1, o_2, o_3 \in I, \forall r \in R_{CRISP-DM-OWL})$

Le premier type est appelé assertion des concepts. Le deuxième type est appelé assertion de rôles. Dans la description du monde ($\mathcal{A}$-Box), RacerPro déclare toutes les instances I qui sont interconnectées avec l'ensemble des instances R_I.

Le moteur d'inférence RacerPro utilise le raisonnement terminologique d'instanciation pour déterminer si un objet o est une instance d'un concept C (ou la validité de l'assertion $C(o)$).

Plusieurs règles d'instanciation peuvent se présenter dans la description du monde $\mathcal{A}$-Box:

- **R1:** $(B(o) \wedge B \subseteq A) \rightarrow A(o)$.
- **R2:** $((A(o) \wedge B(o)) \wedge (\neg(A \subseteq C)) \wedge (\neg(B \subseteq C)) \wedge (((and\ A \wedge B) \subseteq C)) \rightarrow C(o)$.
- **R3:** $r(o_1, o_2) \wedge ((all\ r\ A))(o_2) \rightarrow A(o_1)$.
- **R4:** $r(o_1, o_2) \wedge ((A \subseteq B) \wedge (all r B)(o_2) \rightarrow A(o_1)$.
- **R5:** $((1 \leq i \leq n), A(o_i) = B(o_i)) \rightarrow (A \equiv B)$.
- , etc. $\forall A, B, C \in$ CRISP-DM-OWL, $\forall o_1, o_2 \in I$.

La figure (7.3) montre le traitement de la cohérence en utilisant le raisonnement terminologique d'instanciation dans la description du monde $\mathcal{A}$-Box.

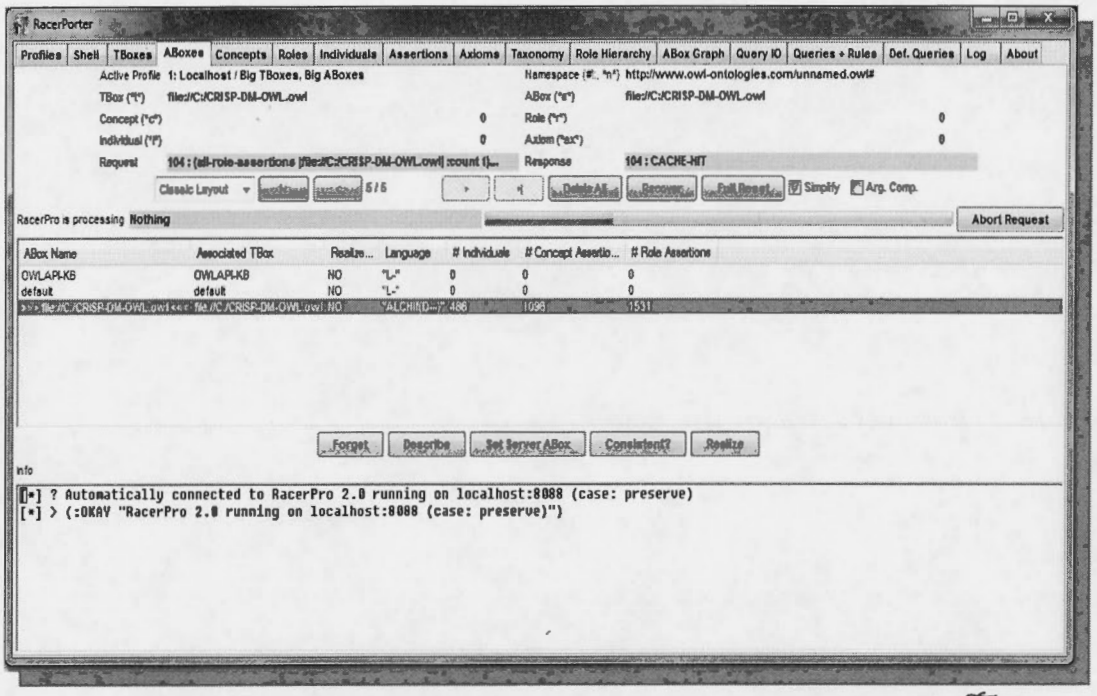


Figure 7.3 Le raisonnement terminologique RacerPro de la description du monde $\mathcal{A}$-Box

Les déductions de la subsomption, l'instanciation et la cohérence sont réalisées mécaniquement, plutôt que manuellement. Elles sont très importantes car elles permettent de:

- Déterminer les subsumants et les subsumés,
- Vérifier les relations implicites et inattendues entre les concepts et instances,
- Tester la consistance conceptuelle dans la terminologie $\mathcal{T}$-Box et la validité des assertions dans la description du monde $\mathcal{A}$-Box,
- Éviter les définitions dans la terminologie qui contiennent des cycles,
- Vérifier les incohérences et les points fixes.

En résumé, notre approche d'évaluation est indépendante de la conceptualisation du domaine modélisé et considère les caractéristiques principales de la structuration de l'ontologie et sa population (concepts, instances, axiome, relation, etc.) (Djellali, 2013h), (Djellali, 2014i).

Nous allons présenter, dans le prochain paragraphe un outil de visualisation pour représenter et comparer les différentes parties de la structure ontologique.

7.3 La visualisation

La méthode de visualisation présente l'ontologie CRISP-DM-OWL comme une structure d'un graphe abstrait. Cette structure graphique basée sur la profondeur et les propriétés est inférée par trois APIs (Application Programming Interface). Premièrement, l'API SAX[11] (Simple Api for Xml) (Su Cheng et Krishna Rao, 2007) est utilisée pour analyser (En anglais: parser) l'ontologie, et plus précisément, pour extraire les définitions des concepts et les axiomes de restriction. C'est une interface pilotée par les événements implémentant un mécanisme de notification (en anglais: callback) pour reconnaître les blocs de construction syntaxique dans le document OWL-DL. Ces événements sont de type:setDocumentLocator, startDocument , endDocument, startElement, etc. et représentent le cœur de l'analyse pour générer la structure ontologique. Deuxièmement, les fonctionnalités utiles pour les diagrammes concrets, telles que les options pour les couleurs, les polices, les fonds, les lignes de style, l'aperçu et les packages Apache sont basés sur l'API GRAPPA[12] (GRAPh Package) (Barghouti, Mocenigo et Lee, 1997). Ce dernier est sous licence AT&T Labs Research (Anciennement: Southwestern Bell Corporation). Finalement, l'API RacerPro[13] (Haarslev *et al.*, 2011) est utilisé pour distinguer les artéfacts ontologiques et pour souligner les concepts inconsistants.
L'outil de visualisation prend en charge les descriptions graphiques dans le langage textuel (dot). Par défaut, le processus de mise en page SAX analyse le fichier d'entrée (.owl) et génère un format de sortie (.dot) (ANNEXE P). Ce langage a une syntaxe lisible qui décrit les données du graphe, y compris les sous graphes et les apparences des éléments (c'est-à-dire la couleur, l'étiquette, etc.). La figure (7.4) montre une grammaire abstraite définissant le langage dot. Les terminaux sont indiqués en gras et les non terminaux sont en italique; les caractères littéraux sont donnés par des apostrophes. Les crochets (et) indiquent le groupement lorsque nécessaire; les parenthèses entourent les éléments facultatifs; les virgules séparent les solutions et les flèches représentent les relations de subsomption entre les artéfacts. Le code OWL-DL correspondant est illustré à la figure (7.5).

```
digraph G {
graph [ fontname = "Helvetica-Oblique",fontsize = 20, label = "\n\n\n\n TextMiner: CRISP-DM-OWL.OWL",
 size = "1,0" ];node [       label = "\N",shape = polygon,
                    sides = 4, distortion = "0.0", orientation = "0.0", skew = "0.0", color = lightskyblue1,
                    style = filled, fontname = "Helvetica-Outline" ];
```

[11] http://www.saxproject.org/
[12] http://www2.research.att.com/~john/Grappa/
[13] http://www.sts.tu-harburg.de/~r.f.moeller/racer/

```
graph [bb= "0,0,1189,1274"];
Root->Technique
Technique->Assosiation
Technique->Selection
Technique->Clustering
Clustering->Density
Clustering->Graph
}
```

Figure 7.4 La grammaire abstraite .dot

```
<?xml version="1.0"?>
<rdf:RDFxmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" ,
xmlns:xsd=http://www.w3.org/2001/XMLSchema# ,xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:owl=http://www.w3.org/2002/07/owl#, xmlns="http://www.owl-ontologies.com/unnamed.owl#"
xmlns:p1=http://www.owl-ontologies.com/assert.owl#,xml:base="http://www.owl-
ontologies.com/unnamed.owl">
<owl:Ontology rdf:about=""/>
<owl:Class rdf:about="# Technique "/>
<owl:disjointWith>
<rdfs:subClassOf>
<owl:Class rdf:about="# Assosiation "/> </rdfs:subClassOf>
<rdfs:subClassOf>
<owl:Class rdf:about="# Selection "/> </rdfs:subClassOf>
<owl:Class rdf:about="# Clustering "/> </rdfs:subClassOf>
<owl:Class rdf:about="# Density "/>
</rdfs:subClassOf>
<owl:Class rdf:about="# Graph "/> </rdfs:subClassOf>
</rdfs:subClassOf>
</owl:disjointWith>
</owl:Class>
```

Figure 7.5 Le code OWL-DL des techniques Data Mining

Étant donné que la visualisation de toutes les connexions entre les artéfacts ontologiques peut être une source de confusion, nous fournissons à l'utilisateur la possibilité de filtrer les propriétés et récupérer la partie demandée de l'ontologie. Par exemple, le package VG(DMpntology()) permet de représenter la structure globale de l'ontologie tandis que le package VL(DMpntology()) permet de visualiser la structure locale. L'annexe (O) illustre le comportement de ces packages.

L'outil de visualisation peut aider les utilisateurs à explorer et comprendre la structure ontologique et offre plusieurs avantages, entres autres:

- L'outil de visualisation permet de sauvegarder les vues des hiérarchies dans divers formats graphiques tels que dot, GraphML, GXL, GML, etc.
- Il permet à l'utilisateur d'explorer les artéfacts ontologiques pour identifier les inconsistances, les associations et les tendances des connaissances.
- L'outil de visualisation aide les utilisateurs à prendre des décisions solides en se basant sur les artéfacts visualisés. La visualisation des données de cette manière peut soutenir les ontologistes engagés à contrôler les effets des changements et surveiller le processus de la maintenance.
- Il aide l'utilisateur à se concentrer sur les parties pertinentes de l'ontologie.
- Il compare les différentes parties de la structure ontologique afin d'assurer la cohérence des connaissances.
- Il révèle les artéfacts ontologiques dans plusieurs niveaux de détails, à partir d'un large aperçu de la structure sous-jacente (visualisation locale et visualisation globale).

7.4 La documentation

La documentation liée aux artéfacts de l'ontologie enrichie est particulièrement importante non seulement pour améliorer sa clarté mais également pour faciliter l'utilisation, la maintenance et la réutilisation. La documentation des résultats est générée avec les outils Doxygen[14] et javadoc dédiés à cette fin (Laramee, 2011).

Dans le prochain paragraphe nous présentons l'organisation de l'architecture logicielle, les caractéristiques techniques du système et le protocole de déploiement des modules que nous venons d'énumérer.

[14] http://www.doxygen.org/

7.5 Implémentation

Le système Data Mining fournit une interface de programmation modulaire orientée-objet. Ainsi, le programmeur peut utiliser des mécanismes orientés-objet pour interagir avec les composantes, ce qui est contraire aux systèmes Data Mining traditionnels, où le développeur interagit avec les composantes en utilisant des outils graphiques GUI (Graphical User Interface).

Nous avons organisé les fonctionnalités dans plusieurs modules qui peuvent être couplés. Dans ce contexte, le terme module fait référence à des classes ou des sous classes. L'architecture du système est divisée en trois parties principales:

- *Une bibliothèque de composantes:* c'est un package qui forme les blocs d'implémentation des modules. Les classes fournies dans cette bibliothèque incluent les outils pour le prétraitement, l'indexation, le repérage, la sélection des variables, le clustering, la sélection des modèles, l'alignement, la mise à jour, la visualisation, les protocoles de construction et d'assemblage. Le module peut fonctionner indépendamment du système en utilisant une interface bien définie et normalisée.
- *Le conteneur d'implémentation*: le système implémente une unité de base (Conteneur) avec laquelle les différents modules peuvent être intégrés. Le conteneur permet à une variété de versions du même module d'être produites. Il a la capacité de faire face à toutes les variations prévues dans l'exécution et l'utilisation. Il fournit des mécanismes de synchronisation pour l'écoute et une collection d'outils consultés via des interfaces graphiques permettant de simplifier le processus de composition des requêtes en guidant les utilisateurs par la méthodologie Data Mining.
- *L'assembleur*: fournit un mécanisme d'assemblage glisser-déposer (de l'anglais: drag-and-drop) pour permettre aux utilisateurs de réorganiser les modules directement dans un méta module. L'utilisateur peut réorganiser l'agencement des modules sur une palette. Les interactions entre les modules peuvent être très complexes. Cette complexité réside dans le nombre de modules qui peuvent être utilisés dans une chaîne Data Mining. Le modèle vectoriel est utilisé pour garantir la communication entre les modules (Djellali, 2012).

Les caractéristiques techniques de notre système sont les suivantes:

- *Le serveur IIS*[15] *(Internet Information Services)*: est un serveur Web supportant les technologies: HTTP, HTTPS, FTP, FTPS, SMTP, etc. Il permet d'héberger les applications ASP.Net (Active Server Pages), le serveur de messagerie SMTP (Simple Mail Transfer Protocol) et le système de gestion de bases de données SQL (Structured Query Language) Server. Cette application permet

[15] http://www.iis.net/

de gérer les fichiers JNLP (Java Network Launch Protocol) et les différentes versions des modules exécutables (Java Archive).

- Le serveur SQL[16]: est un SGBD pour stocker et récupérer les données demandées par les utilisateurs et les applications Data Mining. Il supporte les données structurées et semi-structurées, y compris les formats de supports numériques pour les images, le texte, bit map, audio, vidéo et autres données multimédias. Il assure une communication facile entre le niveau Data Mining et le niveau de stockage.
- *Le serveur File Zilla*[17]: est un serveur FTP (File Transfer Protocol) rapide et fiable avec beaucoup de fonctionnalités utiles et une interface intuitive. Il supporte l'alimentation et la mise à jour à distance du système Data Mining.
- *SMTP*[18] *(Simple Mail Transfer Protocol)*: est un protocole TCP/IP utilisé pour envoyer et recevoir des messages pour les utilisateurs du système.
- RACER Pro [19] (Renamed ABox and Concept Expression Reasoner): est un serveur de raisonnement automatique implémentant la logique de description SHOIN. Il permet de vérifier la cohérence d'un ensemble de descriptions terminologiques de l'ontologie OWL-DL et de trouver les relations de subsomption et d'instanciation implicites.

7.5.1 Déploiement

Nous avons utilisé le protocole JWS (Carter *et al.*, 2009),(Kristensen, 2011) pour déployer le système en ligne. Grâce à ce protocole de déploiement, il est beaucoup plus facile de déployer le système sur plusieurs plateformes (Windows, Linux, Mac OS, Solaris, CP/M, etc.). Le module de déploiement utilise les informations dans le fichier de déploiement TextMiner.JNLP (JNLP de anglais: Java Network Launching Protocol) pour télécharger les différents fichiers JAR, les icônes d'installation, l'écran de démarrage, etc.
Le protocole déploiement améloire la sécurité et empêche les attaques malveillantes (les comportements antisociaux, hacker, cracker, hacktiviste, spywares, spamming, etc.). Il supporte plusieurs méthodes pour interagir avec le système d'exploitation telles que la lecture/écriture des fichiers et l'accès à la presse papiers. L'annexe (I) montre les différentes instructions utilisées dans le fichier JNLP.

[16] http://www.microsoft.com/sqlserver/
[17] http://filezilla-project.org/
[18] http://technet.microsoft.com/
[19] http://www.racer-systems.com/

Conclusion

Dans ce chapitre, nous avons présenté le processus de mise à jour, le modèle calculable et la méthode d'évaluation pour vérifier la cohérence de l'ontologie enrichie. Le plug-in Protégé OWL est utilisé pour représenter le modèle calculable de l'ontologie enrichie. Il fournit des classes et des méthodes pour interroger et manipuler l'ontologie OWL-DL et pour vérifier le raisonnement DL. Le modèle calculable choisi permet d'exprimer une grande variété de la connaissance et fournit des mécanismes efficaces pour faire des inférences décidables basées sur la logique descriptive. Cette dernière est utilisée pour représenter la connaissance sous une forme bien structurée. Elle a une sémantique formelle basée sur la logique, et est équipée de procédures décidables.

Afin de vérifier la cohérence de l'ontologie enrichie, nous avons choisi RacerPro comme un outil de raisonnement pour notre approche car il offre plusieurs fonctionnalités et plusieurs éditeurs graphiques tels qu'OilEd et RacerPorter (RACER Interactive Client Environment). Il offre aussi une manipulation du raisonnement symbolique utilisé pour vérifier la complétude et la correction de l'ontologie enrichie. Il permet d'affirmer les tautologies et les axiomes, tester si les concepts et instances ne sont pas contradictoires, repérer et évaluer une base de connaissances terminologiques et en tirer des déductions. La cohérence et le raisonnement de subsomption et d'instanciation sont utilisés pour vérifier l'inconsistance logique des concepts ainsi que les relations de subsomption et d'instanciation implicites.

La modularité est le style architectural utilisé pour concevoir et mettre en œuvre notre système. En adoptant Cette métaphore, nous avons amélioré la gestion de la complexité, la réutilisation et la division de la conception des tâches pour le développement parallèle. Puisque la communication et la coordination est traitée sous une forme vectorielle, la portabilité, l'extensibilité et la flexibilité du système sont très élevées. Le protocole de déploiement choisi facilite la portabilité et les mises à jour des différents modules téléchargés.

Dans le prochain chapitre, nous concluons ce mémoire et évoquons les perspectives de recherche que nous envisageons.

CHAPITRE VIII

CONCLUSIONS ET PERSPECTIVES

Résumé: Ce chapitre présente les résultats dégagés par chacune des étapes de recherche. Dans un premier temps, nous présentons une description des processus réalisés, en particulier, l'intégration des connaissances, la sélection des variables, le clustering, la sélection des modèles, l'alignement et la vérification de la consistance. Puis, le bilan du travail se focalise sur l'intégration de la connaissance corporative, l'apprentissage machine et la maintenance des ontologies. La troisième partie se focalise sur la contribution de ce travail. Enfin, ce chapitre ne peut se conclure sans introduire les directives futures de recherche, qui constitue la dernière étape de cette étude. En effet, les futures directions incluent la méta modélisation et la gestion distribuée de la connaissance.

8.1 Conclusions

L'objectif principal de l'intégration des connaissances est de structurer l'information et d'augmenter la performance de repérage en fournissant un entreposage bien structuré. Cependant, l'intégration des connaissances fait face à l'hétérogénéité de données. L'utilisation de l'ontologie est une approche possible pour surmonter ce problème. En effet, l'ontologie fournit une compréhension partagée pour supporter l'interopérabilité. Dans ce contexte, nous avons conçu un système d'intégration de la connaissance corporative qui se base sur l'indexation vectorielle dérivée des cooccurrences des termes dans le corpus d'apprentissage. Ces termes sont ensuite utilisés pour accéder aux ressources en utilisant une table d'indexation représentée par un fichier inversé. Le système d'intégration utilise une ontologie partagée pour expliciter la sémantique. Cette ontologie est typiquement indiquée en tant qu'élément de la conception de la mémoire corporative. L'utilisation de l'ontologie dans la mémoire corporative est bénéfique parce qu'elle permet d'accéder à la signification. Ceci fournit une flexibilité et une autonomie aux utilisateurs de la mémoire corporative. De ce fait, l'ontologie facilite l'intégration automatique des connaissances corporatives. Cependant, l'ontologie est une représentation structurée des connaissances dynamiques. Les changements du domaine, les changements des spécifications de la conceptualisation, la représentation de

la conceptualisation et les besoins descendants exigent des modifications de l'ontologie. Par conséquent, nous avons besoin d'un appui pour la gestion des changements dans l'ontologie.

De nombreuses approches ont été développées pour maintenir les ontologies dans plusieurs domaines. La plupart d'entre elles fournissent un soutien limité pour l'ensemble des activités du processus de maintenance. Dans ces approches, la structure de données est compliquée et la construction n'est pas triviale. Elles négligent la structure globale contenue dans l'ontologie et fournissent seulement un soutien limité pour aider à générer l'ontologie. La précision et le rappel dans ces approches ne satisfont pas les demandes des utilisateurs. Elles nécessitent des experts pour la conception et les modifications bruitent les données. De plus, l'apprentissage axiomatique est inexploré et elles ne sont donc pas appropriées aux problèmes avec une grande hétérogénéité.

Nous avons constaté que de nombreuses approches se sont concentrées sur des types limités de la connaissance dans l'ontologie et négligent les autres. Cependant, peu d'entre elles traitent explicitement toute l'information disponible dans l'ontologie. De plus, l'évaluation des ontologies demeure un problème important dans la majorité des approches précédentes et le choix d'une méthode appropriée dépend des critères utilisés. En tirant profit de ces dernières approches, nous avons conçu un ensemble intégré de modules pour supporter la maintenance des ontologies en utilisant l'apprentissage machine, le traitement automatique du langage naturel ainsi que la recherche d'information. L'apprentissage de l'ontologie se base sur l'acquisition de la connaissance, et plus spécifiquement, l'acquisition de la connaissance à partir des textes disponibles dans la mémoire corporative. Pour garantir une indexation efficace des connaissances corporatives, le prétraitement de données est essentiel. Notre approche utilise plusieurs étapes de prétraitement, en particulier, la suppression des chiffres et les signes de ponctuation, la suppression des mots fonctionnels et la troncature pour réduire l'espace de représentation. Les documents textuels sont représentés en utilisant le modèle vectoriel. Avec cette représentation, chaque document est considéré comme un vecteur représenté à l'aide de la fréquence des termes et la fréquence inverse du document (Term Frequency-Inverse Document Frequency). Cependant, nous avons constaté que la performance du clustering est étroitement liée à la taille de l'espace vectoriel VSM (Vector Space Model). D'une part, la malédiction de la dimensionnalité est très coûteuse et beaucoup d'algorithmes d'apprentissage obtiennent une performance significativement plus faible. D'autre part, la représentation vectorielle VSM est très simple mais ne prend pas en compte la succession des mots dans le texte.

La malédiction de la dimensionnalité influence l'efficacité de l'algorithme de clustering Fuzzy ART puisque celui-ci consomme plus de temps et les frontières de décision seront bruitées. Comme étape de prétraitement dans la chaîne Data Mining, la sélection des variables diminue non seulement le coût informatique des ressources mais améliore également de manière significative le taux de reconnaissance des modèles appris à partir de l'ensemble d'apprentissage. Ce processus permet de choisir les variables

pertinentes à partir des variables originales par l'élimination du bruit. Les algorithmes de sélection des variables peuvent être classés dans deux catégories. Le modèle d'emballage permet de choisir les variables pertinentes à partir de l'espace d'indexation basés sur les algorithmes de Data Mining et le modèle de filtrage qui utilise un critère de filtrage pour transformer l'espace original en un espace réduit. Comparé au modèle de filtrage, le modèle d'emballage peut non seulement réduire les dimensions de l'espace des variables mais il peut améliorer la performance d'apprentissage ainsi que la capacité de généralisation.

Pour dériver une indexation optimale de l'espace de représentation vectoriel dans un espace réduit, le processus d'apprentissage utilise le modèle d'emballage basé sur décomposition en valeurs singulières tronquées.

Afin d'induire les changements candidats dans le processus de maintenance de l'ontologie et pour passer en revue la collection de documents, nous avons utilisé le modèle de clustering de la théorie de résonance adaptative floue. L'apprentissage machine de la théorie de résonance adaptative permet d'automatiser le processus d'acquisition de la connaissance pour identifier les changements candidats. Il permet ainsi d'éliminer le processus laborieux d'extraction de la connaissance impliqué dans le processus de maintenance de l'ontologie. Le modèle découvert dans l'ensemble d'apprentissage est utilisé pour classifier et/ou prévoir le comportement de nouveaux exemples et pour déduire les changements candidats. La sélection des modèles basés sur la validation croisée double a été utilisée pour évaluer la capacité de généralisation et pour éviter les biais dans les résultats. Le modèle connexionniste sélectionné améliore de manière significative la précision et le rappel dans le système de repérage de la base d'apprentissage. Il projette les vecteurs-documents d'entrée sur une grille multidimensionnelle suivant le paramètre de résonance. Ainsi, cette projection facilite l'extraction et l'exploration des modèles cachés. De plus, l'architecture connexionniste de la théorie de résonance adaptative offre plusieurs avantages, tel que:

- L'échantillonnage: l'ensemble d'apprentissage peut avoir une grande taille et les méthodes d'apprentissage statique ne peuvent pas gérer la malédiction de l'échantillonnage. En revanche, la théorie de résonance adaptative floue peut s'exécuter avec un flux continu de longueur illimité en termes d'exemples d'apprentissage.
- La frontière de décision: le réseau connexionniste de la théorie de la résonance adaptative permet de rechercher la forme complexe des frontières de décision qui séparent les clusters. Ainsi, il peut induire effacement les changements dans le processus de maintenance.
- La convergence: l'initialisation typique et la configuration des paramètres réduisent le temps de calcul et améliorent la vitesse de convergence finale pour atteindre le voisinage de la solution.
- La généralisation: l'apprentissage par exemple est moins sensible à la présence de bruit. De ce fait, il améliore la performance et la généralisation.

- L'apprentissage dynamique: l'algorithme de la théorie de la résonance adaptative flou n'est pas sensible à l'ordre de présentation des vecteurs documents (élasticité - stabilité).
- La représentativité descriptive: le réseau connexionniste produit des résultats interprétables et utilisables (la compréhensibilité, la validité de l'étiquetage et la transparence de la relation entre le contenu et l'étiquette descriptive).
- L'approximation: le modèle connexionniste de la théorie de résonance adaptative généralise plusieurs types de modélisation en intelligence artificielle, à savoir, la régression, l'algorithme génétique, les arbres de décision, les fourmis artificielles, etc.

Pour trouver la correspondance entre les modèles cachés et les artéfacts ontologiques, le processus de maintenance utilise plusieurs processus individuels d'alignement. L'agrégation des alignements individuels forme un alignement faible et robuste.

Nous avons utilisé le plug-in OWL pour implémenter et mettre à jour l'ontologie. Ce plug-in offre une manière flexible pour l'encodage de l'ontologie et le traitement de la consistance. Dans ce contexte, l'expressivité et l'inférence sont des considérations importantes pour un langage d'implémentation.

Afin de soutenir l'expressivité au maximum tout en maintenant la complétude et la décidabilité computationnelle, nous avons utilisé la version OWL-DL pour représenter le modèle calculable. Ce modèle de représentation facilite l'interopérabilité en fournissant un vocabulaire additionnel avec une sémantique formelle. Il exploite les avantages de la logique descriptive, en particulier, la complétude, la correction et la décidabilité.

Le protocole de déploiement choisi pour déployer notre système est une technologie de distribution permettant la portabilité du code et le téléchargement automatique des mises à jour.

En résumé, dans cette thèse, nous avons présenté une vision de la maintenance des ontologies ainsi qu'une approche modulaire basée sur le Data Mining, l'apprentissage machine, le traitement automatique du langage naturel et la recherche d'information. Nous avons montré que le data Mining est une clef de voûte d'un tel système d'évolution et nous avons détaillé chaque étape dans le processus de maintenance.

8.2 Bilan du travail

L'essentiel de nos travaux peut se regrouper en trois grandes disciplines de recherche:

- Il s'agit de l'intégration de la connaissance corporative dans un dépôt structuré en offrant des mécanismes pour augmenter l'accessibilité durant le repérage des ressources.

En effet, l'intégration de la connaissance corporative est un défi crucial, les techniques du langage naturel et celles de la recherche d'information permettent d'améliorer le processus d'intégration. Les documents

disponibles dans la mémoire corporative fournissent un soutient pour implémenter l'outil d'extraction des modèles cachés.

Afin d'éviter d'induire en erreur le modèle d'indexation en définissant des corrélations inexistantes entre les documents, le dictionnaire négatif Glasgow et la troncature Porter sont les deux méthodes de prétraitement utilisées pour produire un flux de jetons. Ce dernier est ajouté à l'index dans le modèle d'indexation vectorielle pondérée par la fréquence des termes et la fréquence inverse du document. Dans cette représentation, la distribution des contenus est liée aux distributions des documents contenant un index particulier. L'indexation composée a été utilisée pour représenter les documents textuels dans un index contenant un seul fichier composé par segment. Cette structure optimisée encapsule les fichiers individuels d'index dans un seul fichier composé. De ce fait, elle consomme moins de descripteurs de fichiers et moins de ressources informatiques durant le processus d'indexation. En outre, l'architecture segmentée de l'indexation composée permet une utilisation efficace de l'espace disque, une recherche par mot-clé et des réponses rapides. Afin d'optimiser la structure de l'index, le module d'indexation sélectionne et fusionne les segments selon une politique d'alignement.

Le mécanisme de repérage prend en charge plusieurs processus de recherche avancée et offre plusieurs avantages, il s'agit notamment de la vitesse de recherche due à la structuration et l'optimisation d'indexation.

En résumé, le système d'intégration fournit aux utilisateurs un accès rapide et flexible à l'information à travers des mécanismes d'indexation et de repérage:

- La transformation de la requête dans un paquet de jetons pour extraire l'information pertinente.
- L'indexation dynamique des ressources et l'optimisation de l'indexation.
- La minimisation du coût de maintenance de l'index inversé.
- L'amélioration de la qualité de repérage des ressources corporatives (Djellali, 2013f), (Djellali, 2013g).

• Une seconde discipline est l'apprentissage machine qui fournit les modèles et les algorithmes pour l'extraction de la connaissance pertinente représentant les changements candidats dans le processus de maintenance des ontologies. Les propriétés les plus intéressantes sont issues de la sélection des variables, le clustering et la sélection des modèles.

Notre approche produit une structure de clustering descriptif en accélérant la convergence vers le voisinage de la solution sur la base des outils de prétraitement, la sélection des variables pertinentes, l'initialisation typique, la configuration des paramètres de l'architecture connexionniste et l'apprentissage en ligne. Les modèles cachés sont utilisés comme des marqueurs générés automatiquement pour alimenter le processus de mise à jour. Ainsi, en fonction de la collection de documents et leurs évolutions les modèles cachés

évoluent en conséquence. Cependant, la représentation textuelle des documents génère un espace hautement dimensionnel. Par conséquent, un critère robuste de sélection basé sur la décomposition en valeurs singulières tronquées a été utilisé pour réduire l'espace d'indexation. Cette approche d'emballage considère les biais de l'algorithme de la décomposition en valeurs singulières et l'effet du sous-ensemble des variables choisies. Elle supprime ainsi efficacement les variables redondantes et elle améliore la généralisation. En procédant ainsi, le clustering peut se concentrer seulement sur les variables pertinentes. En comparaison à d'autres approches d'emballage, nous avons choisi le modèle d'emballage de la décomposition en valeurs singulières tronquées pour les raisons suivantes:

- L'utilisation d'un espace réduit pour capter les relations terme-document.
- Une méthode de classification automatique avec corrélation.
- Une méthode de généralisation de la décomposition en valeurs propres.
- La capacité de la généralisation.
- Une modélisation vectorielle pour une indexation pertinente et un repérage performant.
- Une méthode de sélection des variables simple et automatique.
- Elle est capable de représenter et manipuler des grands corpus.

En conséquence, nous avons généré un espace de projection réduit en gardant seulement les variables pertinentes (Djellali, 2013c) et (Djellali, 2013h).

Nous avons utilisé une méthode non paramétrique pour sélectionner le modèle en minimisant l'erreur totale des estimations. La méthode repose sur un algorithme itératif de la validation croisée double, c'est-à-dire, dans chaque étape les modèles et leurs taux de reconnaissances correspondants sont sélectionnés par un apprentissage dynamique. La sélection du modèle par la validation croisée double est importante non seulement pour l'estimation du taux de reconnaissance mais aussi pour la sélection du modèle à partir de l'ensemble d'apprentissage. Il s'agit d'une méthode indispensable pour réduire la variance et ainsi améliorer la généralisation. De plus, l'estimation non paramétrique de la validation croisée double offre les avantages suivants, entre autre:

- L'évaluation de plusieurs modèles à partir de l'ensemble de données disponible.
- La validation de la robustesse du clustering de la théorie de résonance adaptative floue.
- La génération de plusieurs modèles et identification du meilleur modèle sur la base de statistiques de performance.
- L'estimation non biaisée de l'erreur de généralisation.
- La simplicité, la rapidité et la fiabilité.

En résumé, les techniques Data Mining proposées dans notre projet permettent d'améliorer le processus d'extraction des connaissances à partir des textes (Djellali, Meunier et Delisle, 2012).

- Une troisième discipline, dans laquelle peuvent s'inscrire nos travaux, est le recours à l'approche de maintenance des ontologies qui a fait l'objet de nombreux travaux dans des domaines tels que la mise à jour, l'alignement, la visualisation et le raisonnement automatique, mais constitue un axe de recherche assez récent dans le domaine de développement des ontologies. En effet, dans le processus de maintenance, les processus individuels d'alignement représentent le mécanisme fondamental permettant d'associer les étiquettes (substituer, insérer, supprimer) aux artéfacts de l'ontologie en exploitant la similitude entre les deux représentations. Ils déterminent les artefacts candidats et c'est à l'utilisateur d'accepter, rejeter ou changer les artefacts ontologiques.

Nous avons conclu que le choix de la distance est très important dans l'étape d'alignement. Afin d'utiliser autant d'information autant que possible, nous avons exploité des résultats composés de plusieurs processus individuels d'alignement. Ces derniers sont construits à partir des relations lexicographiques entre la description terminologique de l'ontologie et des étiquettes descriptives. L'agrégation des processus d'alignement forme un processus d'alignement fiable et robuste.

Le modèle calculable OWL-DL est utilisé pour représenter l'ontologie enrichie. Il prend en charge les utilisateurs qui souhaitent un maximum d'expressivité tout en gardant la complétude des calculs (la garantie de calculer toutes les conclusions) et la décidabilité (tous les calculs se terminent en un temps fini). En outre, nous avons choisi d'utiliser OWL-DL pour les raisons suivantes:

- OWL-DL accepte plusieurs formalismes de représentation pour supporter les définitions additionnelles.
- OWL-DL assure la consistance terminologique et détecte les contradictions.
- OWL-DL soutient la capacité d'exprimer et de raisonner.
- Le langage OWL-DL a une syntaxe et une sémantique claires.
- Compatible avec certaines normes et standards tels que XML Schema, RDF et DAML-OIL.

En raison de la correspondance entre OWL et la logique de description, l'inférence OWL-DL s'appuie fortement sur les outils de raisonnement automatique basés sur la logique de description $\mathcal{SHOIN}$. Par conséquent, les techniques de raisonnement RacerPro sont utilisées pour vérifier la cohérence de l'ontologie et les relations de subsomption (instanciation) implicites entre les entités. Les principales tâches d'inférence (en particulier l'instanciation et la subsomption) dans RacerPro sont décidables. De plus, le moteur d'inférence RacerPro comprend plusieurs techniques d'optimisation, en particulier, le retour en arrière de la dépendance dirigée et le branchement sémantique. Cela nous permet de faciliter le raisonnement et de réduire le coût informatique des ressources .

Le processus de vérification de la cohérence proposé permet de transformer les artefacts de l'ontologie dans une base de connaissance terminologique (la terminologie T-Box et la description du monde A-Box) pour appuyer le raisonnement formel. Le raisonnement de subsomption (instanciation) est utilisé pour vérifier l'inconsistance des concepts (instances) et les relations de subsomption (instanciation) implicites. La subsomption et l'instanciation permettent d'affirmer les tautologies et les axiomes, tester si les concepts (instances) ne sont pas contradictoires, repérer et évaluer une base de connaissances terminologiques et d'en tirer des déductions. Le protocole DIG est utilisé pour connecter les programmes des clients au moteur d'inférence RacerPro. De ce fait, les utilisateurs peuvent formuler des requêtes descriptives à partir de l'allocation de la base de connaissance terminologique DIG (Djellali, 2014i), (Djellali, 2014j).

Afin de représenter et comparer les différentes parties de la structure ontologique, nous avons utilisé un outil de visualisation basé sur la hiérarchie inférée par l'outil SAX. Il permet également de voir la collection des artefacts, supprimer les artéfacts inintéressants grâce aux mécanismes de filtrage, sélectionner un artéfact ontologique pour afficher les détails, etc.

En guise de conclusion, tout au long de notre travail nous avons fait appel à de multiples disciplines, en particulier, l'indexation, la sélection des variables, le clustering, la sélection des modèles, l'alignement, la visualisation et le raisonnement automatique basé sur la logique.

8.3 Contribution de ce travail

La conception de notre approche conceptuelle est motivée par le besoin des applications pilotées par les ontologies dans lesquelles les utilisateurs peuvent accéder et enrichir des ontologies. Notre approche fournit une solution pour intégrer la connaissance corporative et maintenir les ontologies. La modularité est le style architectural par lequel les modules sont conçus pour tenir compte de la réutilisation et du partage des modules. Cette métaphore modulaire fournit des gabarits configurables pour les nouvelles classes Data Mining ce qui permet l'extensibilité. Le système Data Mining fournit également des fonctionnalités étendues pour soutenir l'acquisition, l'indexation, le repérage, et la mise à jour, ce qui lui permet d'être utilisé ainsi dans les applications Data Mining. Le plan d'assemblage pour enrichir l'ontologie est exécuté en passant des sous requêtes aux différents modules du système Data Mining. Ce dernier génère la traduction finale sous forme d'un code calculable de l'ontologie enrichie. L'interface graphique permet à l'ingénieur de l'ontologie d'inspecter les diverses étapes dans le plan de maintenance de l'ontologie et de superviser son exécution. Cette démarche incrémentale composée de transformations successives des modèles permet de conserver la trace des divers modèles construits. Par conséquent, toute la connaissance disponible au sujet des rapports concrets entre les entités de l'ontologie et l'ensemble de données est stockée dans la mémoire corporative permettant ainsi de préserver l'historique de l'évolution.

De nombreux modèles ont été développés ces dernières années dans le contexte de la maintenance des ontologies. Nos travaux s'inscrivent dans cette dynamique mais il convient de préciser en quoi notre approche dans ce domaine est innovante:

- Mettre à disposition des concepteurs un outil pour l'apprentissage des ontologies.
- Proposer des patrons de conception qui constituent le patrimoine de la maintenance des ontologies.
- Les ontologues sont soutenus au cours de la maintenance grâce à des mécanismes pour l'extraction, l'alignement, l'inconsistance, la satisfiabilité, la reconnaissance d'instance, la détection des points fixes, le test de subsomption et l'implication des changements.
- La conservation des traces de divers modèles construits au cours du cycle de maintenance autorisant ainsi plus facilement les révisions.
- Tous les changements dans les entités de l'ontologie sont immédiatement visibles dans l'ontologie enrichie.
- La performance d'accessibilité aux connaissances corporatives.
- Pour maintenir l'ontologie à jour et améliorer graduellement ses connaissances, le système Data Mining traite l'information redondante, non pertinente et évolutive.
- La portabilité du système permet de perfectionner les outils et les techniques de maintenance existants.

L'approche d'apprentissage proposée est encourageante et originale car elle permet:

- L'exploitation des techniques Data Mining pour définir les parties de l'ontologie concernées par les changements.
- La représentation de l'enchaînement des transformations successives des ontologies dans la mémoire corporative.
- L'évaluation et amélioration du niveau de qualité du processus d'apprentissage des ontologies.
- La vérification de l'inconsistance.
- D'éviter les minima locaux durant l'apprentissage machine.

En résumé, l'approche visée par notre travail inclut l'ensemble des méthodes et des techniques nécessaires pour piloter la maintenance de l'ontologie. Par conséquent, le système de gestion de la mise à jour de l'ontologie est une plateforme pour modifier, versionner, faire des requêtes et pour supporter l'inférence terminologique.

8.4 Perspectives

Dans le but d'améliorer ce travail, deux suggestions peuvent être émises. La première consiste à tenir compte de la diversité des formats des ressources corporatives. En effet, la reconnaissance des ressources corporatives pourrait être améliorée. Tous les formats de ressources (HTML, PDF, DOC, RTF, XML, etc.) doivent être transformés avant l'analyse et l'indexation. La première étape du développement d'un tel système d'intégration est de comprendre les différents formats de fichiers et les méthodes utilisées pour stocker le texte brut. Toutefois, afin de procéder à l'extraction automatique, les fichiers doivent être traités de manière à identifier les métadonnées correspondantes. Les métas données, ou «données sur les données» fournissent des informations qui peuvent aider à comprendre les documents indépendamment de leur type. Ils représentent la sémantique de chaque information contenue dans la mémoire corporative, à savoir, la structure, le format, les associations, les contraintes appliquées sur les différentes associations, les attributs et les processus associés, etc. La deuxième étape consiste à utiliser des analyseurs (de l'anglais: Parser) afin de lire et écrire des documents dans un format spécifique.

Une deuxième idée est de proposer une architecture multi-agents (de l'anglais: Multi-Agent System ou MAS) pour l'intégration de la connaissance corporative et la maintenance de l'ontologie.

Aujourd'hui le système multi-agents est une technologie cruciale pour exploiter avec efficacité la connaissance distribuée. En effet, dans un monde dynamique et décentralisé, le système multi-agents propose une architecture distribuée qui peut être utilisée pour soutenir une gestion décentralisée de la connaissance. Il fournit une modélisation pour supporter la dynamique de la mémoire corporative et une structuration qui supporte la modularité, l'interopérabilité, l'autonomie, etc. De plus, cette modélisation de la gestion distribuée permet de partager la connaissance facilement. Dans cette perspective, nous envisageons de développer un système Data Mining basé sur une architecture multi-agent pour piloter l'intégration des connaissances et la maintenance des ontologies.

La majorité des approches de modélisation des systèmes multi-agent (MAS-CommonKADS, SODA, AALAADIN, AAII, Adelf, SABPO, AUML, Gaia, MaSE, Tropos, MESSAGE/UML, Prometheus, BDI agents, ROADMAP et FATMAS) font l'abstraction de l'environnement pendant l'analyse des besoins et la conception (Iglesias, Garijo et González, 1999). Cependant, la modélisation explicite de l'environnement est très importante pour comprendre les caractéristiques du système et les conditions de répartition. De ce fait, le système Data Mining basé sur la technologie agent devrait prendre en considération la variété de circonstances environnementales, à savoir, l'évolution et la dispersion de la connaissance, la gestion décentralisée, la répartition tâche/rôle, les interdépendances, les contraintes pour la coordination des activités, etc.

Dans ce contexte, le but de notre prochain travail est de proposer une architecture multi agents permettant d'intégrer la connaissance dynamique et l'enrichissement des ontologies. Cette perspective de la gestion de la connaissance distribuée est ainsi appropriée pour aborder la dynamique de l'environnement, l'intégration des connaissances et la maintenance des ontologies. Dans cette optique, la structuration de tel système d'intégration comme un système multi-agents possède plusieurs avantages:

- L'abstraction: la répartition des rôles des agents augmente le niveau d'abstraction et diminue la complexité d'intégration.
- La modularité: la structure du système est décomposée en un ensemble d'agents relativement indépendants dont il est possible de contrôler l'évolution, la reconfiguration, la réutilisation et la gestion de la complexité, etc.
- La flexibilité: les agents peuvent être ajoutés, modifiés et supprimés facilement grâce à la modularité et l'abstraction. De ce fait, l'architecture SMA a la capacité pour faire face à toutes les variations prévues dans l'exécution et l'utilisation.
- La rapidité de traitement: les traitements concurrentiels, le partage des tâches et la coordination entre les agents permettent de réduire le temps d'exécution et les ressources exigées, en particulier, dans les systèmes multi-agent pilotés par les ontologies.
- La fiabilité: l'interaction entre les agents pour résoudre efficacement leurs sous problèmes et la coordination de leurs activités permettent de trouver une solution au problème global.
- La planification: la structure SMA permet de piloter la conception du système, c'est-à-dire, une planification orientée tâche et rôles.
- La coordination: les systèmes multi-agents fournissent des mécanismes de négociation pour gouverner les interactions.
- La répartition: le style architectural des SMAs permet au concepteur de définir explicitement les conditions d'intégration des nouveaux agents et leurs positions dans le système (Zambonelli, Jennings et Wooldridge, 2003).

En résumé, nous croyons que l'approche multi-agents favorise un niveau d'abstraction convenable pour l'intégration des connaissances corporatives et la maintenance des ontologies. En outre, elle permet d'implémenter un système Data Mining flexible, robuste et ayant haute modularité.

ANNEXE A

DESCRIPTION DE L'ONTOLOGIE

URL: http://www.elmanahel.ca/ontology/crisp-dm-owl.owl

Comme illustré sur la figure (A1.1), l'ontologie CRISP-DM-OWL réserve une section pour représenter le concept CRISP-DM comme étant un modèle qui couvre toutes les phases d'exploration qui peuvent être appliquées dans les différents types d'activités de la fouille de données. Cette section est organisée en plusieurs phases, dont chacune elle-même composée de plusieurs tâches. Chaque tâche contient certaines activités plus détaillées et chaque activité peut avoir diverses sorties.

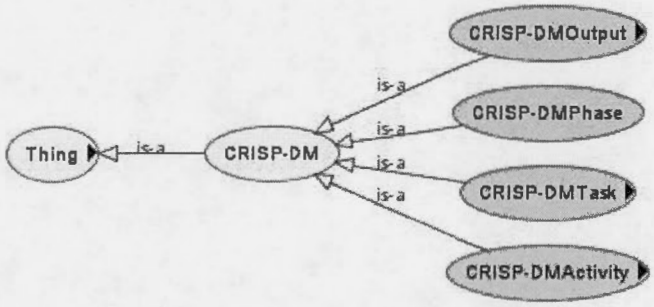


Figure A1.1 La méthodologie CRISP-DM

Les tâches de la compréhension de données «DataUnderstandingTask», le déploiement «DeploymentTask», la modélisation «ModelingTask», la préparation des données «DataPreparationTask», l'évaluation «EvaluationTask» et la compréhension d'affaires «BusinessUnderstandingTask» sont représentées dans la classe CRISP-DMTask.

Ces tâches aident les organismes à comprendre le processus Data Mining et fournissent une bonne planification pour bien gérer leurs projets. La figure (A1.2) montre les six tâches de la méthodologie CRISP-DM.

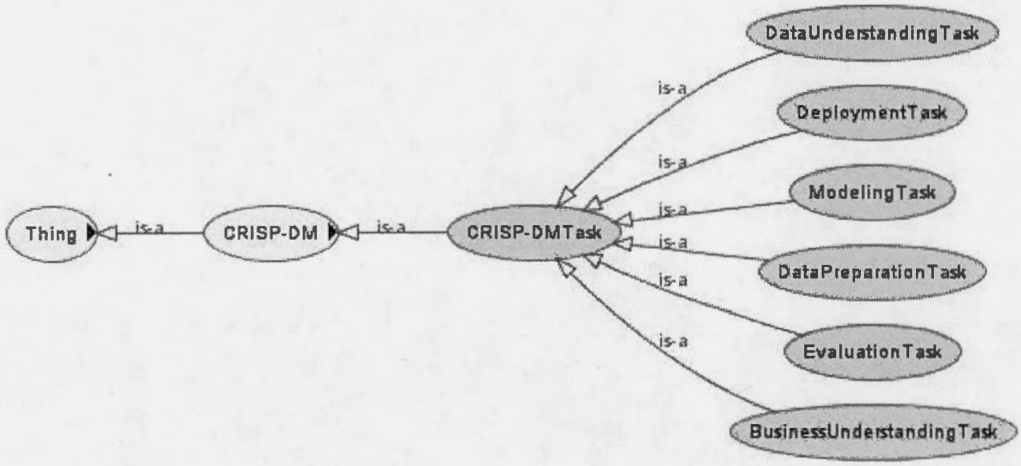


Figure A1.2 Les différentes tâches de la méthodologie CRISP-DM

Le concept *Techniques* (figure (A1.3)) est divisé en trois sous classes, à savoir, la modélisation «Modeling», la préparation des données «DataPrepartion» et la compréhension de données «DataUnderstanding».
Dans la méthodologie CRISP-DM, la compréhension de données aide à: familiariser les analystes avec la structure des données sous-jacente, identifier les problèmes de qualité de données, découvrir une vue globale sur les données et détecter les informations pertinentes pour former des hypothèses sur les modèles cachés. La phase de préparation de données couvre toutes les activités pour construire l'ensemble de données finales ou les données qui seront transmises aux outils Data Mining. La modélisation inclut le choix de la technique modélisation, la génération des tests, la création et l'évaluation des modèles.
La compréhension d'affaires «BusinessUnderstanding», l'évaluation «Evaluation» et le déploiement «Deployment» ne sont pas couverts par la connaissance du domaine Data Mining.

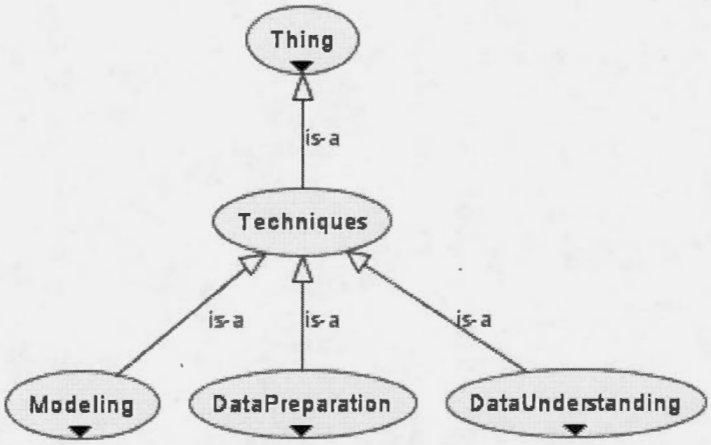


Figure A1.3 Les techniques Data Mining couvertes par l'ontologie

Le concept de compréhension de données «DataUnderstanding» subsume quatre concepts, y compris, la description de données «DataDescription», la collection de données initiales «DataCollection», la vérification de la qualité de données «DataQualityVerification» et l'exploration des données «DataExploration». Il décrit également les caractéristiques des données (les types d'attributs «AttributeType», le type de l'ensemble de données «DataSetType», les erreurs dans les données «GeneralErrors» et la façon dont les données sont collectées et exploitées, etc.). La figure (A1.4) illustre les différentes étapes de la compréhension de données.

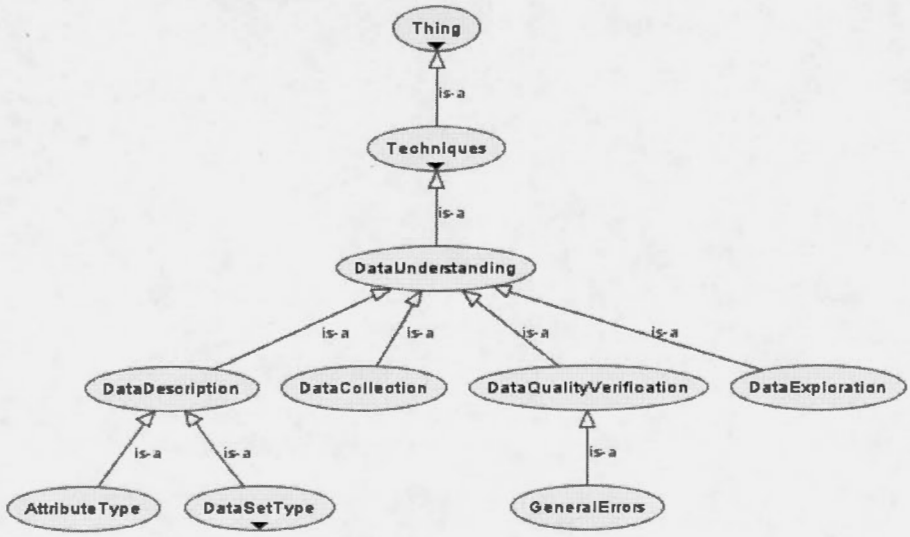


Figure A1.4 Les différentes étapes de la compréhension des données

Comme illustré à la figure (A1.5.), les cinq concepts subsumés par le concept préparation de données «DataPreparartion» sont:

- L'intégration des données «DataIntegration»: où plusieurs modèles de données sont combinés.
- La sélection des données «DataSelection»: l'extraction de l'information pertinente à partir des données.
- La construction des données «DataConstruction»: où les données sont converties en des formes appropriées pour la fouille en effectuant des opérations de synthèse ou d'agrégation.
- Le nettoyage des données «DataCleaning»: pour éliminer le bruit et l'inconsistance des données.
- Le formatage des données «DataFormatting»: où les données sont préparées pour la fouille.

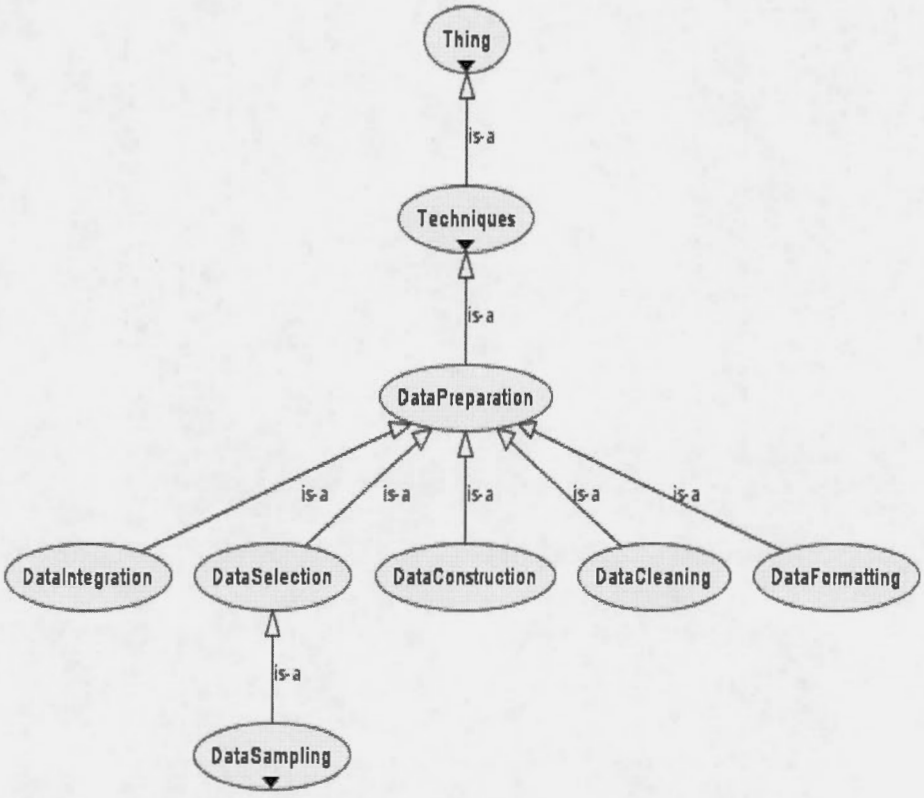


Figure A1.5 Les différentes étapes de la préparation des données

La figure (A1.6) montre les concepts subsumés dans la sous section de modélisation «modeling», y compris, la tâche «task », l'algorithme «algorithm», le programme «program», le modèle «model» et «suite». Ces concepts sont expliqués ci-dessous:

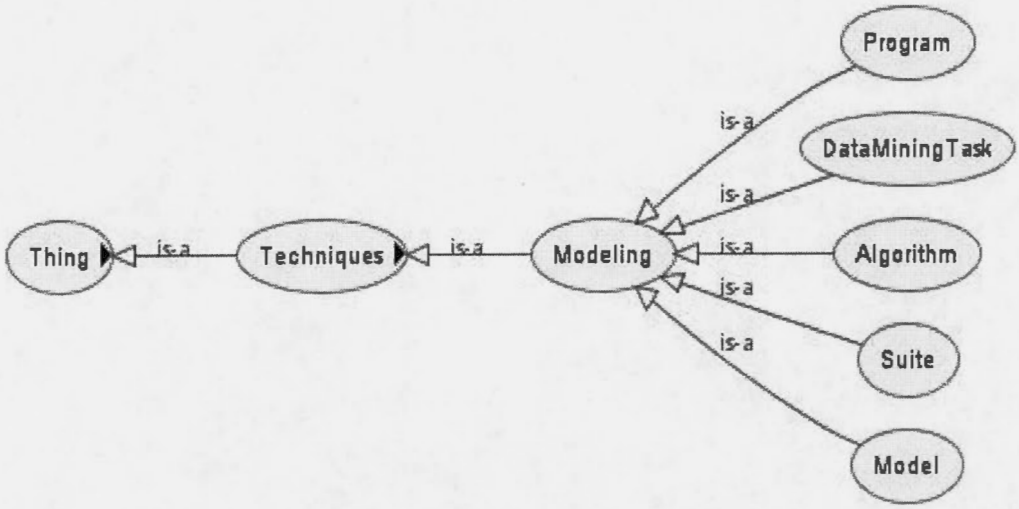


Figure A1.6 La modélisation CRISP-DM

- La tâche Data Mining «DataMiningtask »: est la technique d'apprentissage machine telle que la classification, la régression, l'association et le clustering.
- L'algorithme Data Mining «algorithm»: est le code calculable qui accepte les formes d'apprentissage et génère le modèle caché. Il peut prendre plusieurs formes:
  - L'algorithme de classification «ClassificationAlgorithm» attribue les formes d'apprentissage à une ou plusieurs catégories prédéfinies en se basant sur un apprentissage supervisé. Ce dernier traite les relations entrée-sortie représentées dans l'ensemble de l'apprentissage. La figure (A1.7) montre les algorithmes les plus populaires dans cette catégorie: l'algorithme bayesien «BayesianAlgorithm», l'induction des règles «RuleInductionAlgorithm», les k plus proches voisins «kNearestNeiborAlgorithm», les réseaux connexionnistes «NeronalNetworkAlgorithm» et l'arbre de décision«DecisionTreeAlgorithm».

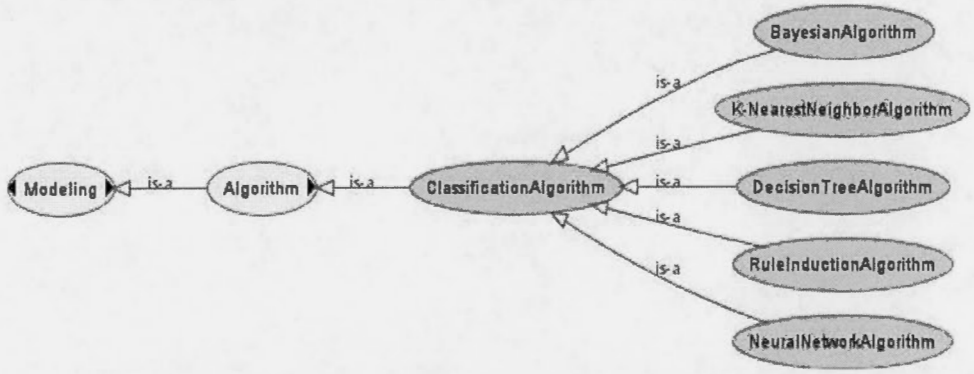


Figure A1.7 Les différents algorithmes de la classification

- L'algorithme de clustering «ClusteringAlgorithm» vise à regrouper un ensemble de formes sans des étiquettes dans un ensemble de groupes disjoints en utilisant un processus exploratoire non supervisé. Beaucoup d'algorithmes ont été proposés pour l'analyse des clusters. ces algorithmes sont montrés à la figure (A1.8).

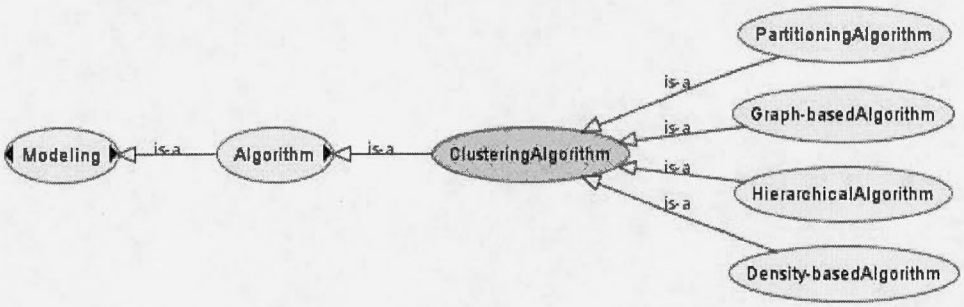


Figure A1.8 Les différents algorithmes du clustering

- Le clustering de partitionnement «PartitioningAlgorithm»: l'algorithme de partitionnement construit des partitions (clusters) de données, où chaque cluster optimise un critère de groupement.
- Le clustering hiérarchique «HierachicalAlgorithm»: les algorithmes hiérarchiques créent une décomposition hiérarchique des objets. Nous pouvons distinguer entre les algorithmes agglomératifs (bottom-up) et les algorithmes de division (top-down). Dans les algorithmes agglomératifs, les clusters sont fusionnés selon une mesure de distance. Le clustering peut s'arrêter quand tous les objets sont dans un seul groupe ou la valeur d'un seuil donné est atteinte.

  Les algorithmes de division divisent les objets de données disjoints dans des groupes à chaque étape.
- le clustering piloté par la densité «Density-basedAlgorithm»: ces algorithmes groupent les objets selon une fonction objective de densité spécifique. La densité est habituellement définie comme le nombre d'objets dans un voisinage particulier de données.
- Le clustering graphique «Graph-basedAlgorithm»: ces algorithmes se concentrent sur des données spatiales, c.-à-d. les données qui représentent la structure géométrique des objets dans l'espace de représentation. L'objectif de ces algorithmes est de quantifier l'ensemble de données dans un certain nombre de cellules en cherchant la meilleure coupure du graphe, celle qui optimise une

fonction de coût prédéfinie. L'optimisation des fonctions de coût est calculée en cherchant les vecteurs propres de la matrice du graphe.

- L'algorithme de la régression «RegressionAlgorithm» peut être utilisé pour modéliser la relation de prédiction entre plusieurs variables indépendantes et une variable dépendante ou une variable de réponse (figure A1.9).

Figure A1.9 Les différents algorithmes de la régression

- L'algorithme d'association «AssociationAlgorithm» est une méthodologie utile pour découvrir les rapports cachés dans un ensemble d'apprentissage.

- Le modèle Data Mining «Model»: c'est le résultat de l'application de l'engin Data Mining. La sortie peut être un modèle descriptif ou prédictif.

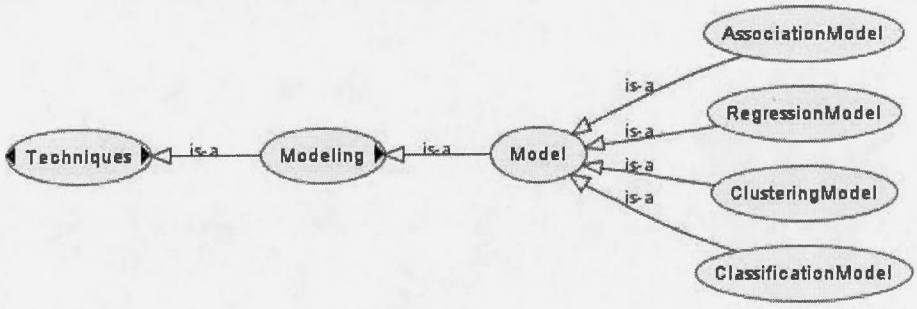


Figure A1.10 Les différents modèles Data Mining

- Le programme Data Mining «program»: est le codage direct de l'algorithme Data Mining.

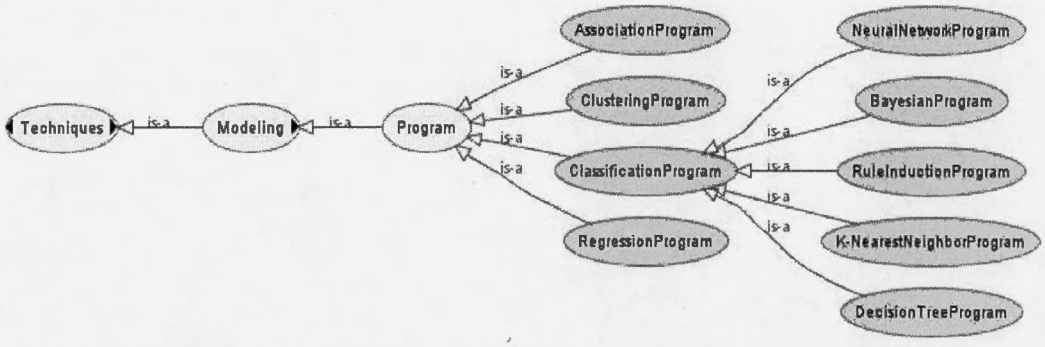


Figure A1.11 Les différents programmes Data Mining

- Data Mining suite: est un ensemble de paquetages intégrés contenant des programmes Data Mining. Chaque programme utilise un algorithme Data Mining pour réaliser un but particulier.

Tous les concepts peuvent être encore divisés pour faire une hiérarchie plus détaillée de concepts, en particulier, la classification, l'association, la régression et le clustering. La figure ci-dessous illustre une vue partielle permettant de se concentrer sur la partie modélisation de l'ontologie.

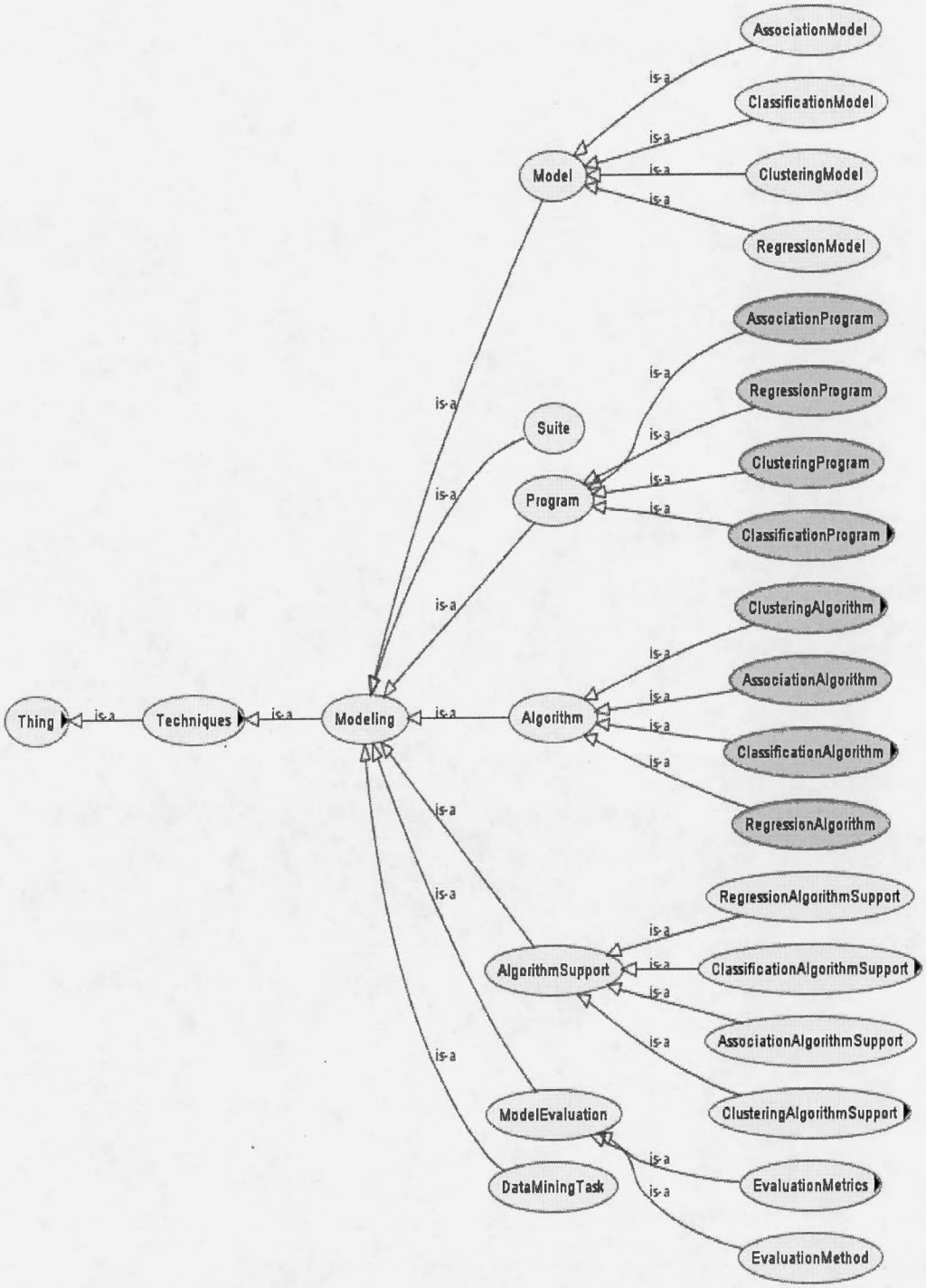


Figure A1.12 Les différentes techniques de la modélisation Data Mining

ANNEXE B

LE CODE OWL-DL DE L'ONTOLOGIE CRISP-DM-OWL

```
<?xml version="1.0"?>
<rdf:RDF
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
 xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
 xmlns:owl="http://www.w3.org/2002/07/owl#"
 xmlns="http://www.owl-ontologies.com/unnamed.owl#"
 xmlns:p1="http://www.owl-ontologies.com/assert.owl#"
 xml:base="http://www.owl-ontologies.com/unnamed.owl">
 <owl:Ontology rdf:about=""/>
 <owl:Class rdf:ID="DeploymentTask">
 <owl:equivalentClass>
 <owl:Class>
 <owl:intersectionOf rdf:parseType="Collection">
  <owl:Restriction>
  <owl:hasValue>
  <CRISP-DMPhase rdf:ID="P6_Deployment">
  <includesTask>
   <DeploymentTask rdf:ID="T3_Produce_Final_Report">
   <generatesOutputs>
   <FinalReportOutput rdf:ID="O1_Final_Report">
   <generatedByTask rdf:resource="#T3_Produce_Final_Report"/>
   <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
   >&lt;p style="margin-top: 0">
 Final report
 &lt;/p></name> <description
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
   >&lt;p style="margin-top: 0">
 The final report is used to summarize the project and its results.
 &lt;/p>
 &lt;p style="margin-top: 0">
 Contents:
 &lt;/p>
 &lt;p style="margin-top: 0">
 * Summary of Business Understanding: background, objectives and success
 criteria.
 &lt;/p>
 &lt;p style="margin-top: 0">
 * Summary of data minin process.
```

```
&lt;/p>
&lt;p style="margin-top: 0">
* Summary of data mining results.
&lt;/p>
&lt;p style="margin-top: 0">
* Summary of results evaluation.
&lt;/p>
&lt;p style="margin-top: 0">
* Summary of deploymeny and maintenance plans.
&lt;/p>
&lt;p style="margin-top: 0">
* Cost/benefit analysis.
&lt;/p>
&lt;p style="margin-top: 0">
* Conclusions for the business.
&lt;/p>
&lt;p style="margin-top: 0">
* Conclusions for future data mininig.
&lt;/p></description>
  </FinalReportOutput>
  </generatesOutputs>
  <includedInPhase rdf:resource="#P6_Deployment"/>
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
Produce final report
&lt;/p></name> <description
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
At the end of the project, the project leader and his team write up a
final report. It depends on the deployment plan, if this report is only
a summary of the project and its experience or if this report is a
final
presentation of the data mining results.
&lt;/p></description>
  <generatesOutputs>
  <FinalReportOutput rdf:ID="O2_Final_Presentation">
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
Final presentation
&lt;/p></name> <generatedByTask
rdf:resource="#T3_Produce_Final_Report"/>
  <description rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
The representation report normally contains a subset of the information
contained in the Final report, but structured in a different way.
&lt;/p></description>
  </FinalReportOutput>
  </generatesOutputs>
  </DeploymentTask>
 </includesTask>
 <includesTask>
  <DeploymentTask rdf:ID="T4_Review_Project">
  <description rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
This task assesses what went right and what went wrong, what was done
well and what needs to be improved.
```

```
&lt;/p></description>
  <generatesOutputs>
  <ProjectReviewOutput rdf:ID="O1_Experience-Documentation">
  <description rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
This section summarize important experiences made during the project.
In
ideal projects, experience documentation covers also any reports that
have been written by individual project members during the project
phases and their tasks.
&lt;/p></description>
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
Experience documentation
&lt;/p></name> <generatedByTask rdf:resource="#T4_Review_Project"/>
  </ProjectReviewOutput>
  </generatesOutputs>
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
Review project
&lt;/p></name> <includedInPhase rdf:resource="#P6_Deployment"/>
  </DeploymentTask>
 </includesTask>
 <description rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >&lt;p style="margin-top: 0">
Creation of the model is generally not the end of the project. Even if
the purpose of the model is to increase knowledge of the data, the
knowledge gained will need to be organized and presented in a way that
the customer can use it. It often involves applying
&amp;quot;live&amp;quot; moels within
an organization's decision making processes, for example in real-time
personalization of Web pages or repeated scoring of maketing databases.
However, depending on the requirements, the deployment phase can be as
simple as generating a report or as complex as implementing a
repeatable
data mining process across the enterprise. In many cases it is the
customer, not the data analyst, who carries out the deployment steps.
However, even if the analyst will not carry out the deployment effort
it
is important fot the customer to understand up front what actions need
to be carried out in order to actually make use of the created models.
&lt;/p></description>
 <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >&lt;p style="margin-top: 0">
Deployement
&lt;/p></name>
 <includesTask>
  <DeploymentTask rdf:ID="T1_Plan_Deployment">
  <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
Plan deployment
&lt;/p></name> <includedInPhase rdf:resource="#P6_Deployment"/>
  <description rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >&lt;p style="margin-top: 0">
This task takes the evaluation results and concludes a strategy for
deployment of the data mining resluts into the business.
&lt;/p></description>
```

```
   <generatesOutputs>
   <PlanDeploymentOutput rdf:ID="O1_Deployment_Plan">
   <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
   >&lt;p style="margin-top: 0">
 Deployment plan
 &lt;/p></name> <generatedByTask rdf:resource="#T1_Plan_Deployment"/>
   <description rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
   >&lt;p style="margin-top: 0">
 This section specifies the deploymwent of the data mining results.
 &lt;/p>
 &lt;p style="margin-top: 0">
 Topics to be covered:
 &lt;/p>
 &lt;p style="margin-top: 0">
 * Summary of deployable resluts (derived from Next Steps erport).
 &lt;/p>
 &lt;p style="margin-top: 0">
 * Description of deployment plan.
 &lt;/p></description>
   </PlanDeploymentOutput>
   </generatesOutputs>
   </DeploymentTask>
  </includesTask>
  <includesTask>
   <DeploymentTask rdf:ID="T2_Plan_Monitoring_and_Maintenance">
   <generatesOutputs>
   <PlanMonitoringAndMaintenanceOutput
rdf:ID="O1_Monitoring_and_Maintenance_Plan">
   <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
   >&lt;p style="margin-top: 0">
 Monitoring and maintenance plan
 &lt;/p></name> <description
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
   >&lt;p style="margin-top: 0">
 he monitoring and maintenance plan specifies how the deployed results
 are to be maintained.
 &lt;/p>
 &lt;p style="margin-top: 0">
 Topics to be covered:
 &lt;/p>
 &lt;p style="margin-top: 0">
 * Overview of results deployment and indication of which resluts may
 require updating(and why).
 &lt;/p>
 &lt;p style="margin-top: 0">
  For each deployed reslut:
 &lt;/p>
 &lt;p style="margin-top: 0">
  * Descriptioin of how updating will be triggered (regular updates,
 trigger event, performance monitoring).
 &lt;/p>
 &lt;p style="margin-top: 0">
  * Description of how updating will be performed.
 &lt;/p>
 &lt;p style="margin-top: 0">
 * Summary of the results updating process.
 &lt;/p></description>
```

```
    <generatedByTask rdf:resource="#T2_Plan_Monitoring_and_Maintenance"/>
    </PlanMonitoringAndMaintenanceOutput>
    </generatesOutputs>
    <description rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >&lt;p style="margin-top: 0">
  Monitoring and maintenance are important issues if the data mining
  result becomes part of the day-to-day business and its environment. A
  careful perparation of a maintenance strategy helps to avoid
 unnecessary
  long period of incorrect usaeg of data mining results. In order to
  monitor the deployment of the data mining results, the project needs a
  detailed plan on the monitoring process. This plan takes inito account
  the specific type of deployment.
  &lt;/p></description>
    <includedInPhase rdf:resource="#P6_Deployment"/>
    <name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >&lt;p style="margin-top: 0">
  Plan monitoring and maintenance
  &lt;/p></name>
    </DeploymentTask>
   </includesTask>
   </CRISP-DMPhase>
   </owl:hasValue>
   <owl:onProperty>
   <owl:ObjectProperty rdf:ID="includedInPhase"/>
   </owl:onProperty>
   </owl:Restriction>
   <owl:Restriction>
   <owl:onProperty>
   <owl:ObjectProperty rdf:ID="generatesOutputs"/>
   </owl:onProperty>
   <owl:allValuesFrom>
   <owl:Class rdf:ID="DeploymentOutput"/>
   </owl:allValuesFrom>
   </owl:Restriction>
   <owl:Restriction>
   <owl:someValuesFrom>
   <owl:Class rdf:about="#DeploymentOutput"/>
   </owl:someValuesFrom>
   <owl:onProperty>
   <owl:ObjectProperty rdf:about="#generatesOutputs"/>
   </owl:onProperty>
   </owl:Restriction>
  </owl:intersectionOf>
  </owl:Class>
  </owl:equivalentClass>
  <owl:disjointWith>
  <owl:Class rdf:ID="DataUnderstandingTask"/>
  </owl:disjointWith>
  <owl:disjointWith>
  <owl:Class rdf:ID="DataPreparationTask"/>
  </owl:disjointWith>
  <owl:disjointWith>
  <owl:Class rdf:ID="ModelingTask"/>
  </owl:disjointWith>
  <owl:disjointWith>
  <owl:Class rdf:ID="BusinessUnderstandingTask"/>
```

```
</owl:disjointWith>
<rdfs:subClassOf>
<owl:Class rdf:ID="CRISP-DMTask"/>
</rdfs:subClassOf>
<owl:disjointWith>
<owl:Class rdf:ID="EvaluationTask"/>
</owl:disjointWith>
</owl:Class>
<owl:Class rdf:ID="Model">
<owl:disjointWith>
<owl:Class rdf:ID="ModelEvaluation"/>
</owl:disjointWith>
   <owl:disjointWith>
   .....
   .....
   .....
   .....
   .....
   .....
   .....
   .....
   .....

<rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 ></rdfs:comment>
 </RecordsSampling>
 <DataSelectionActivity rdf:ID="A5_Select_Different_Data_Subsets">
 <description rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >&lt;p style="margin-top: 0">
 Select different data subsets (e.g., different attributes, only data
 which meet certain conditions).
 &lt;/p></description>
 </DataSelectionActivity>
</rdf:RDF>

<!-- Created with Protege (with OWL Plugin 2.2 beta, Build 291)
http://protege.stanford.edu -->
```

ANNEXE C

L'INDEXATION

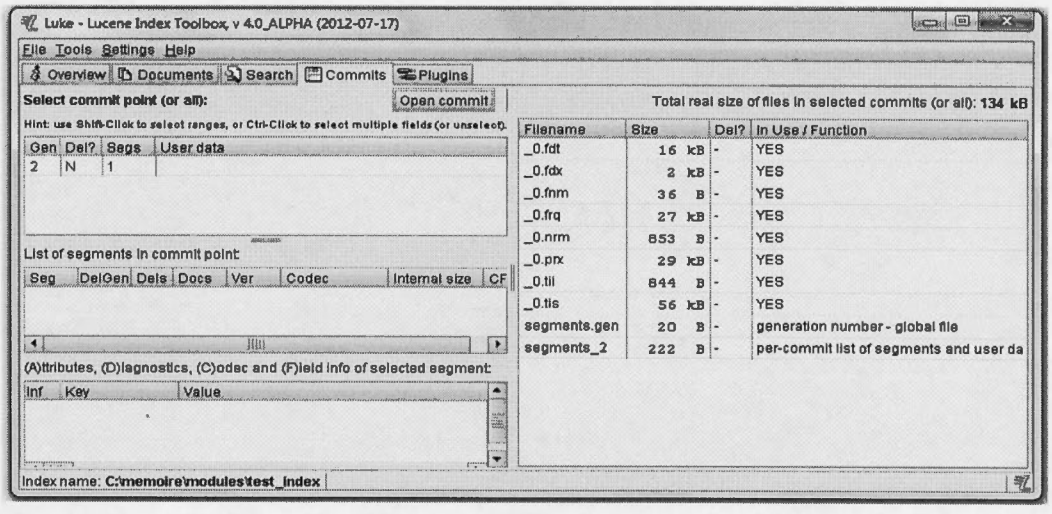


Figure A3.1 L'indexation multi fichiers

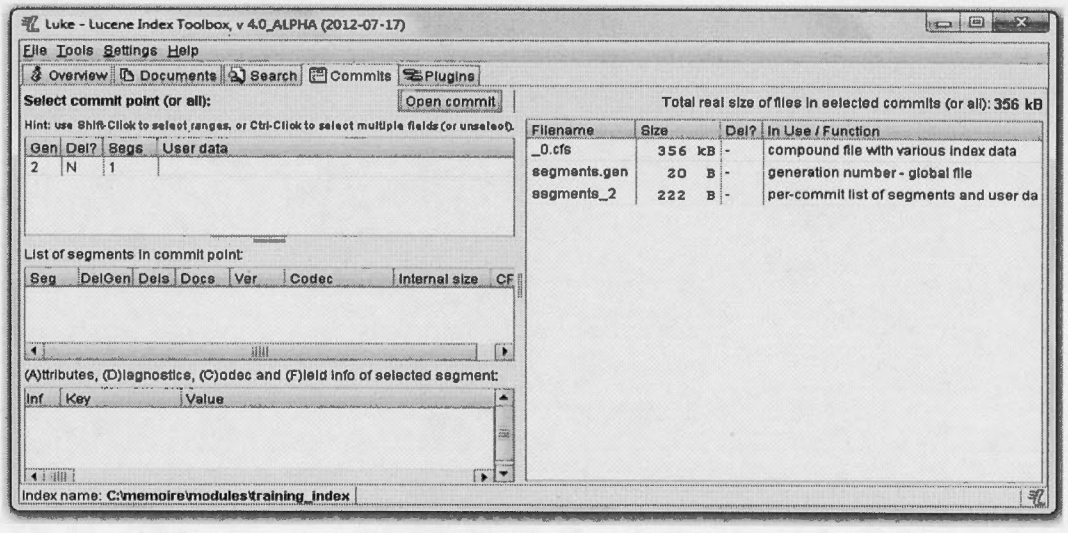


Figure A3.2 L'indexation composée

ANNEXE D

LA STRUCTURE DE L'INDEX

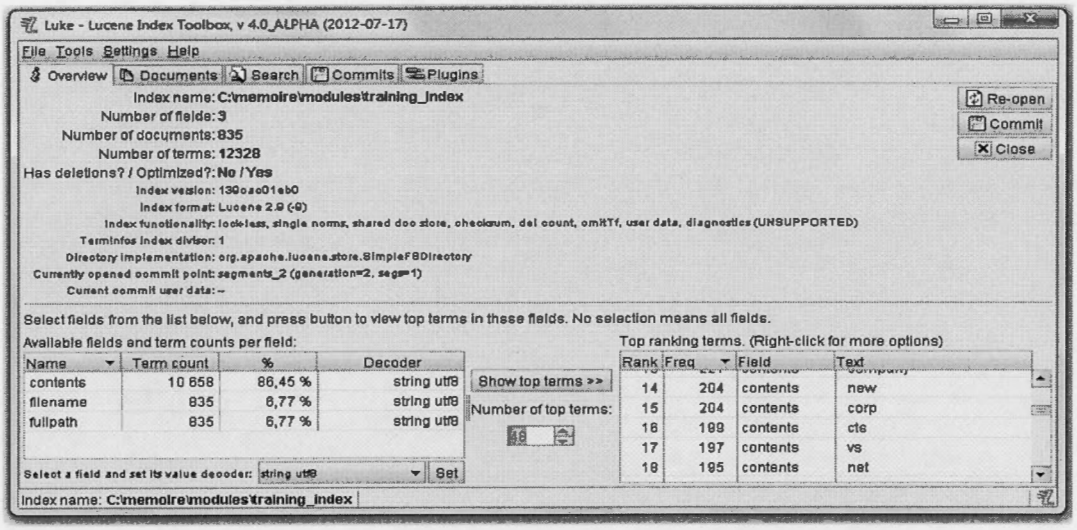


Figure A4.1 L'index d'apprentissage

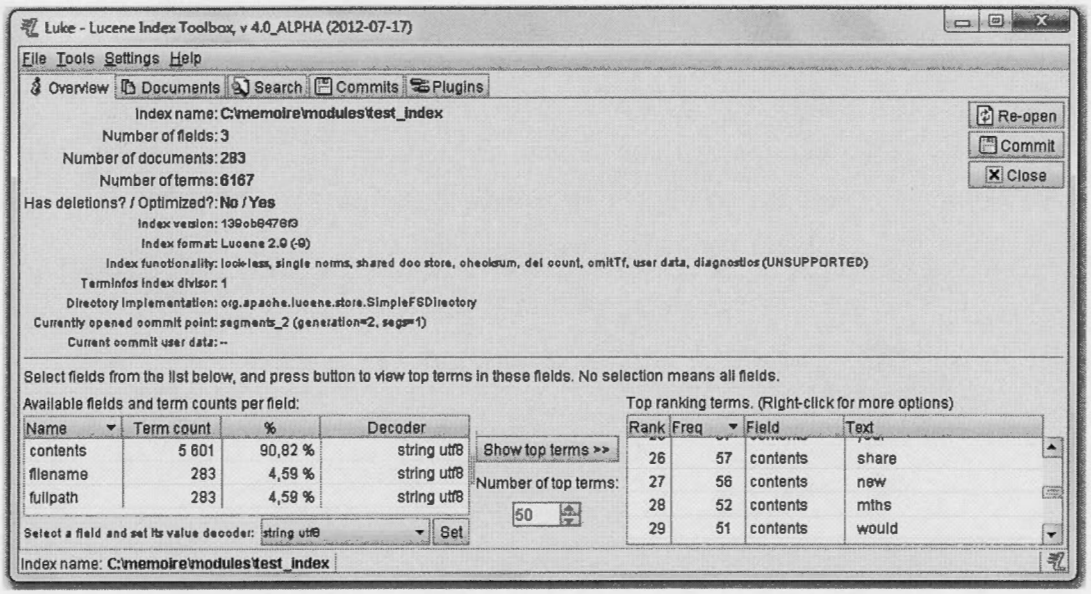


Figure A4.2 L'index de test

ANNEXE E

LES OUTILS D'ADMINISTRATION SYSTEME

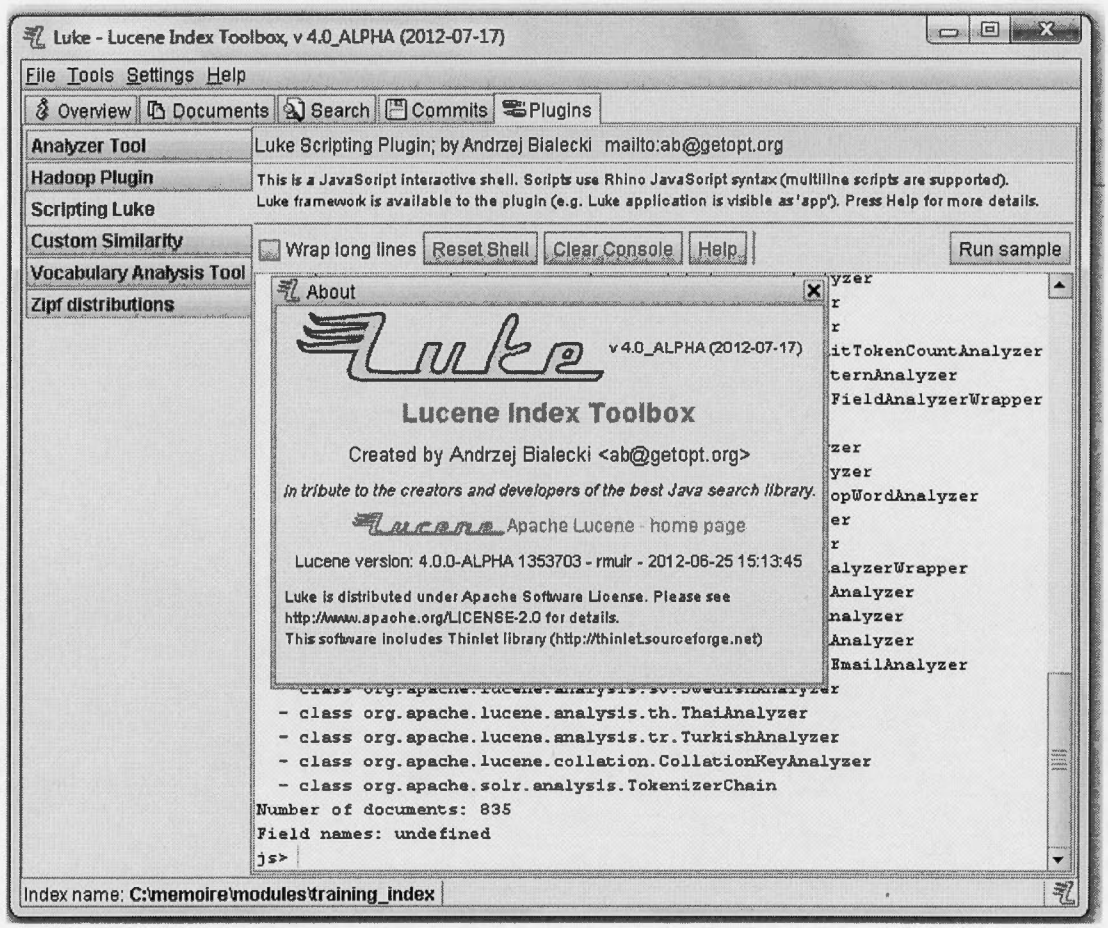


Figure A5.1 L'outil LUKE

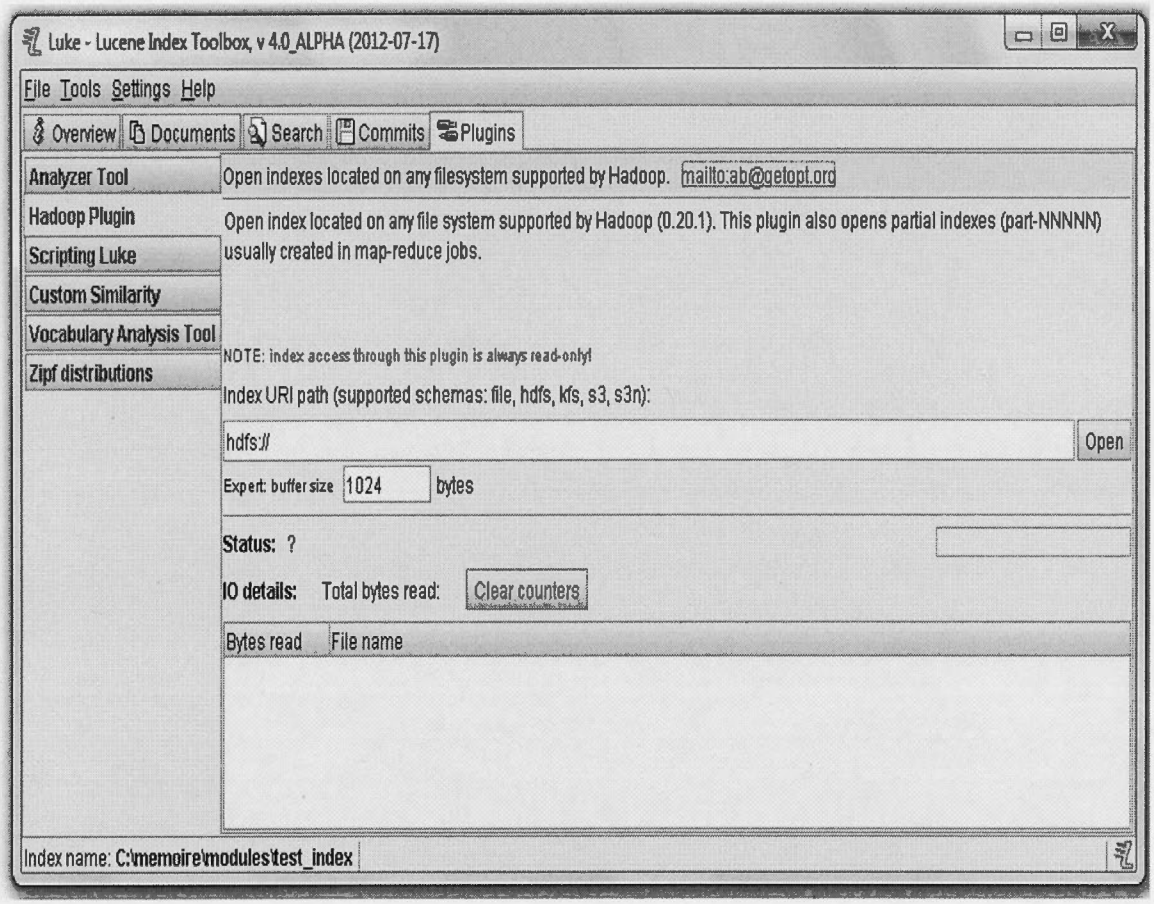


Figure A5.2 L'outil HADOOP

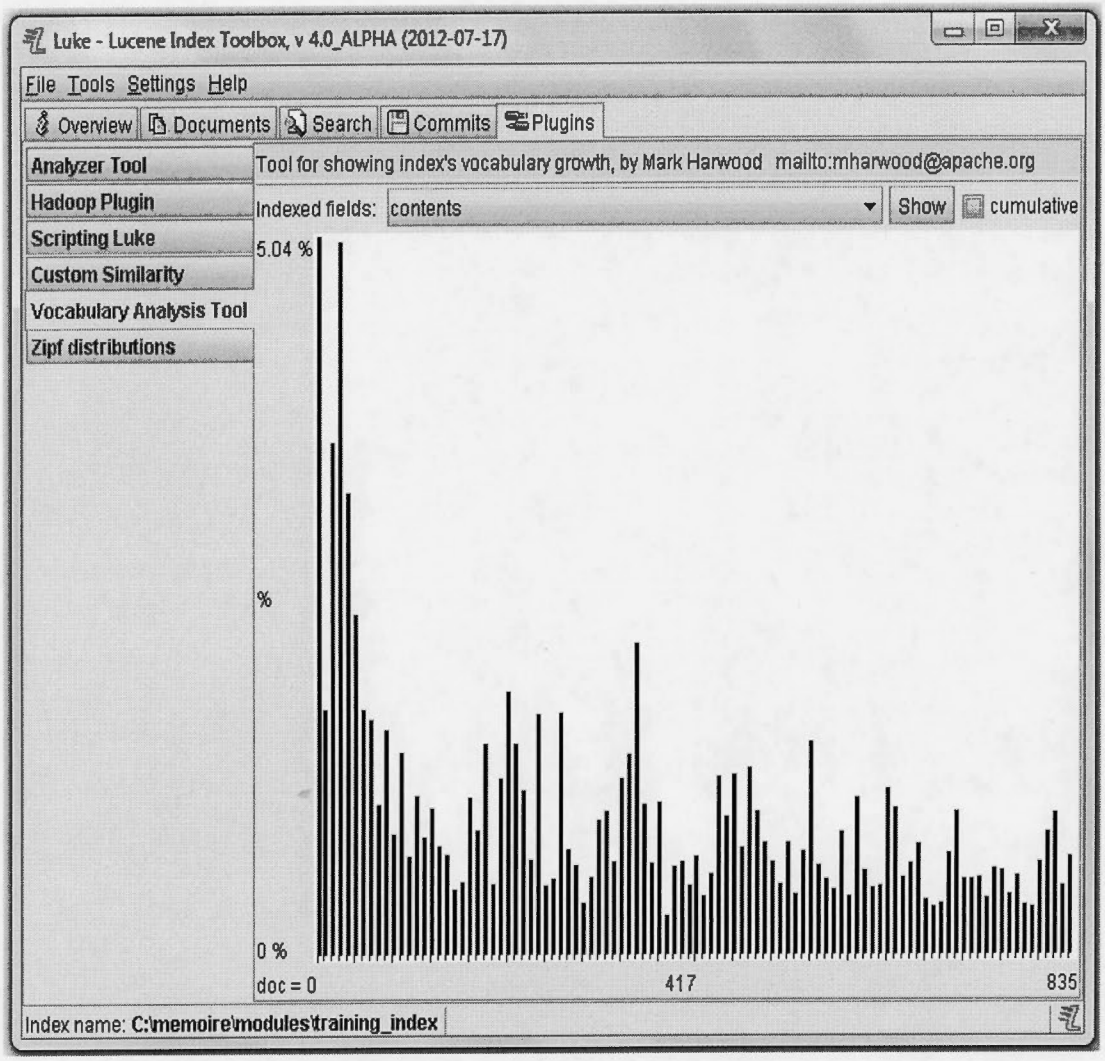


Figure A5.3 L'outil d'analyse du vocabulaire

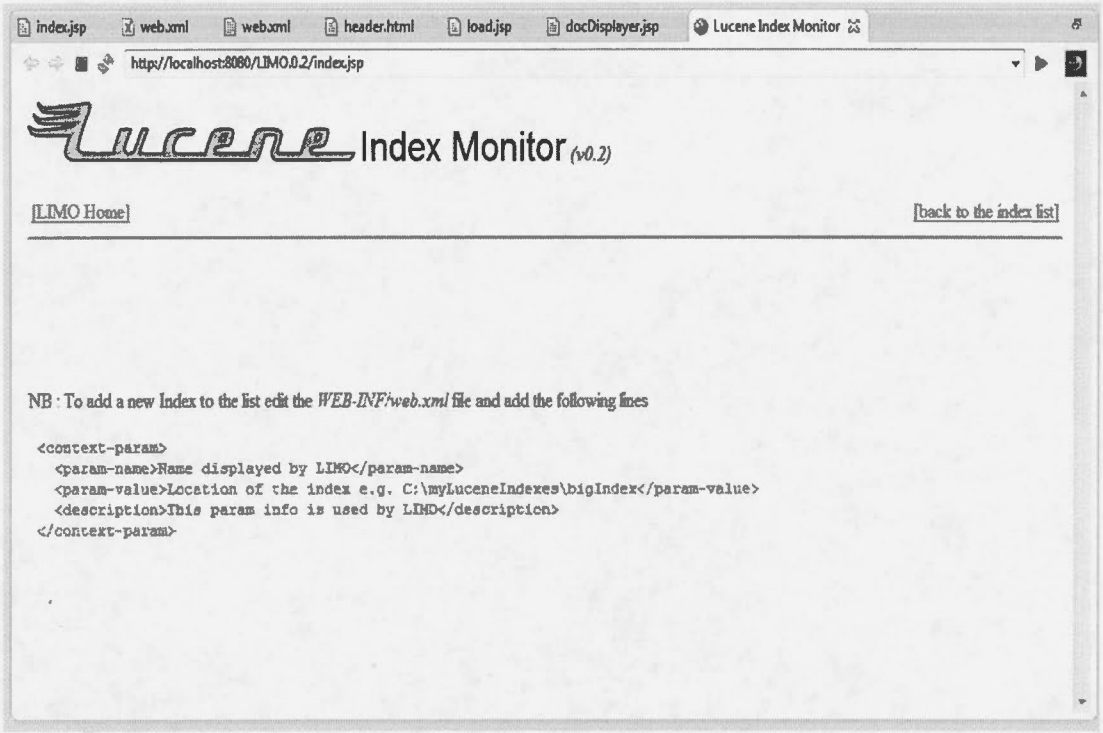


Figure A5.4 L'outil LIMO

ANNEXE F

LE REPERAGE

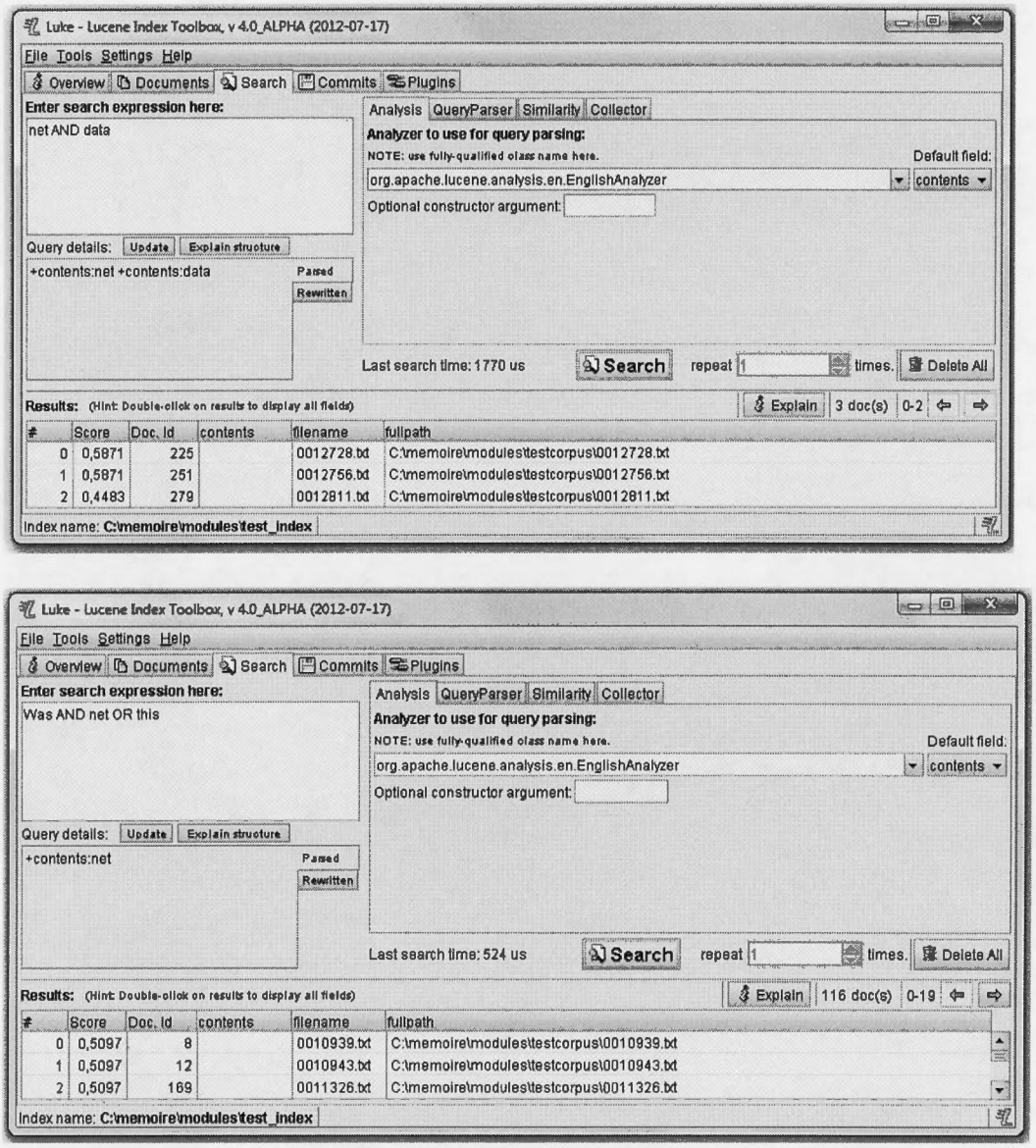


Figure A6.1 Le repérage multicritères

ANNEXE G

LA SÉLECTION

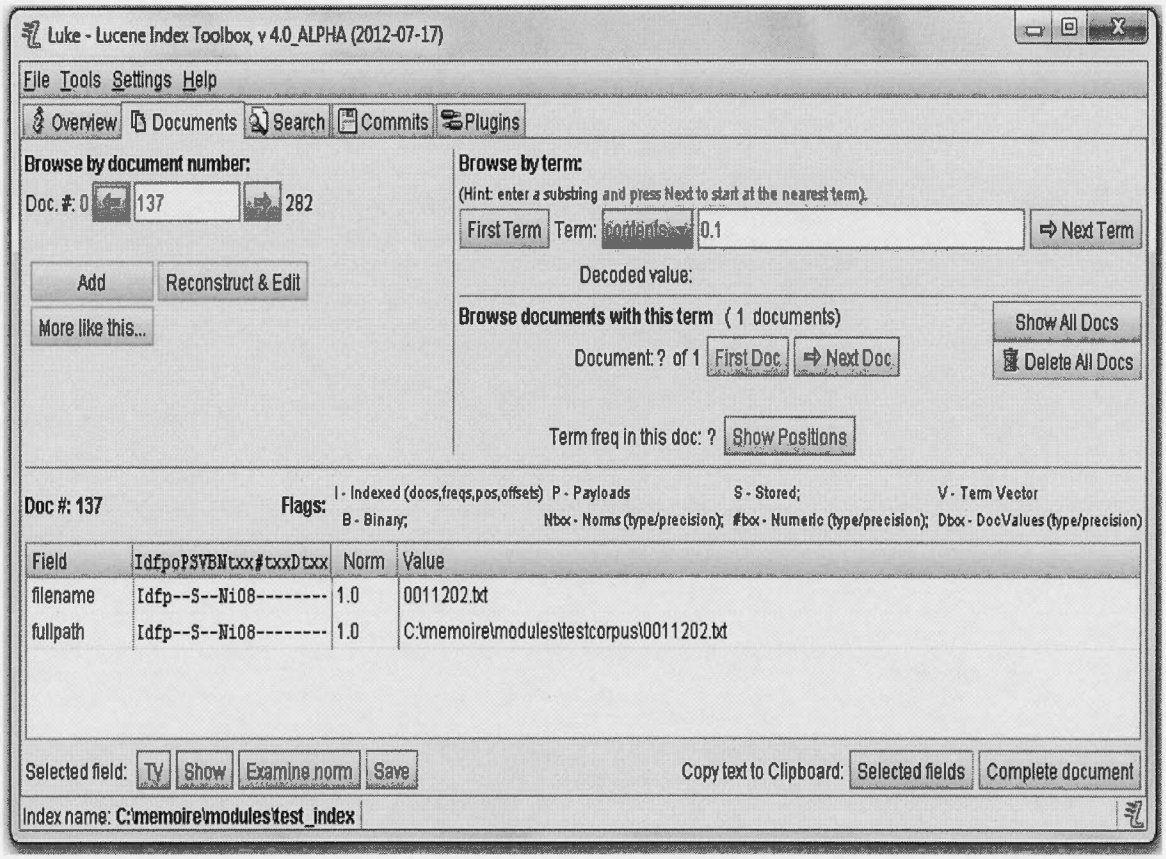


Figure A7.1 La sélection document/terme/champs

ANNEXE H

LE DEPLOIEMENT DU SYSTEME DATA MINING

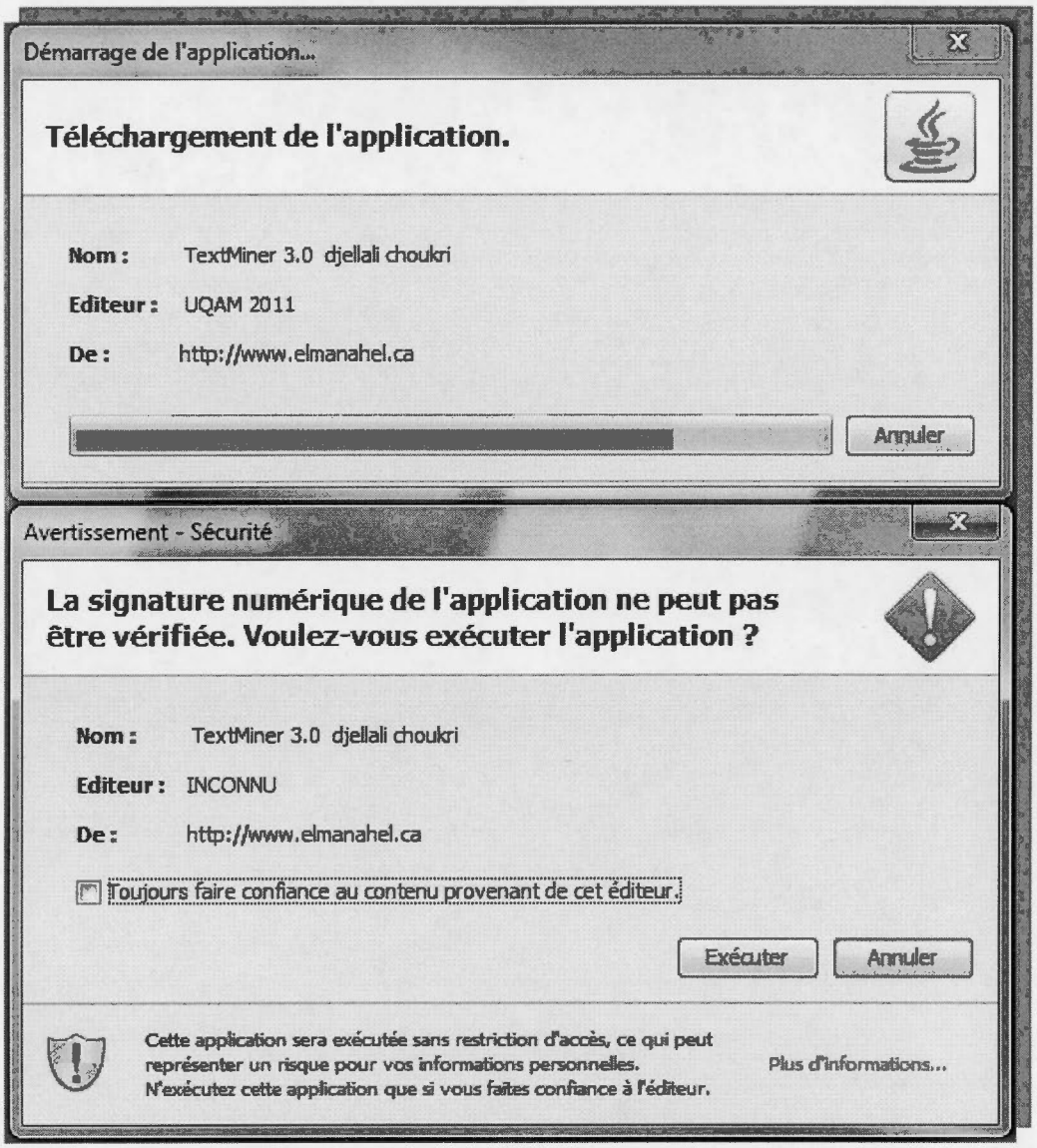


Figure A8.1 Le déploiement du système Data Mining

ANNEXE I

LE DEPLOIEMENT JNLP

```
<?
header("Content-Type: application/x-java-jnlp-file; name= textminer.jnlp");
header("Content-disposition: attachment; filename= TextMiner.jnlp");?>
<jnlp spec="1.0+" codebase="http://www.elmanahel.ca/TextMiner/" href=" TextMiner.jnlp">

 <information>
 <title>TextMiner 1.0 </title>
 <vendor> LANCI </vendor>
 <description kind='one-line'>
 Test of the JNLP file API - 'always allow'
 </description>
 <!-- Web-start will attempt* to associate this
 file extension/type with our application.
 * it will ask the user, in any case -->
<resources os="Windows">
<j2se version="1.5+"
 href="http://java.sun.com/products/autodl/j2se"
 initial-heap-size="128m"
 max-heap-size="1024m" />
<jar href="altersig.jar" main="true" download="eager" />
<jar href="data.jar" />
<jar href="lib/jsl.jar" />
<jar href="lib/looks-2.0.4.jar" />
 <extension name="unAutresource" version="1" href="/ TextMiner.jnlp"/>
</resources>
 <association
 extensions="zzz"
 mime-type="text/sleepytime" />
 <shortcut online='false'>
 <desktop/>
 </shortcut>
 </information>
 <!-- This example also works with no permissions at all,
 though the behaviour (in regards to prompts) is slightly
 different. To observe the behaviour of the sandboxed
```

```
example, simply delete the entire <security /> element. -->
<security>
<all-permissions/>
</security>
<resources>
<property
name='jnlp.file.extension'
value="txt"/>
<j2se version='1.2+' />
<jar href='http://www.elmanahel.ca/TextMiner/TextMiner.jar' main='true' />
</resources>
<application-desc main-class='pilote.TextMiner' />
</jnlp>
<application-desc main-class="asig.load.Boot">
<argument>-jconsole disable</argument>
  <argument>show</argument>

</application-desc>
```

Figure A9.1 Le code JNLP

ANNEXE J

Spécifications des exigences d'un module (adapté de la norme IEEE 830)

1) Objectifs: ce document permet de détailler les fonctionnalités du module: Troncature.

1.1) Portée: Fichier et Corpus.

2) Contexte de l'application: Text Mining, Web sémantique et repérage.

2.1) OBJECTIF: réduire le nombre de mots dans une représentation vectorielle.

3) Description générale du module

3.1) Vue d'ensemble des fonctions du produit: troncature (fichier et corpus).

3.2) Description des utilisateurs: utilisateurs.

4) Le diagramme de séquences acteurs/module:

Diagramme de séquence acteur/système du cas d'utilisation: Troncature fichier.

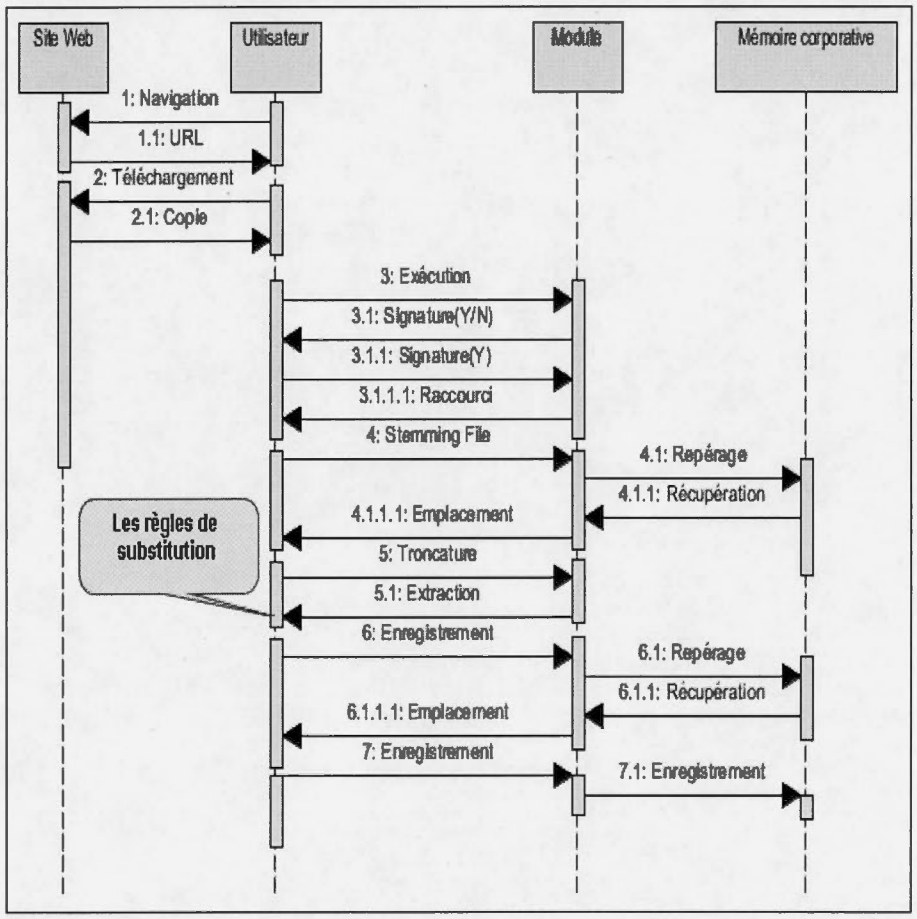
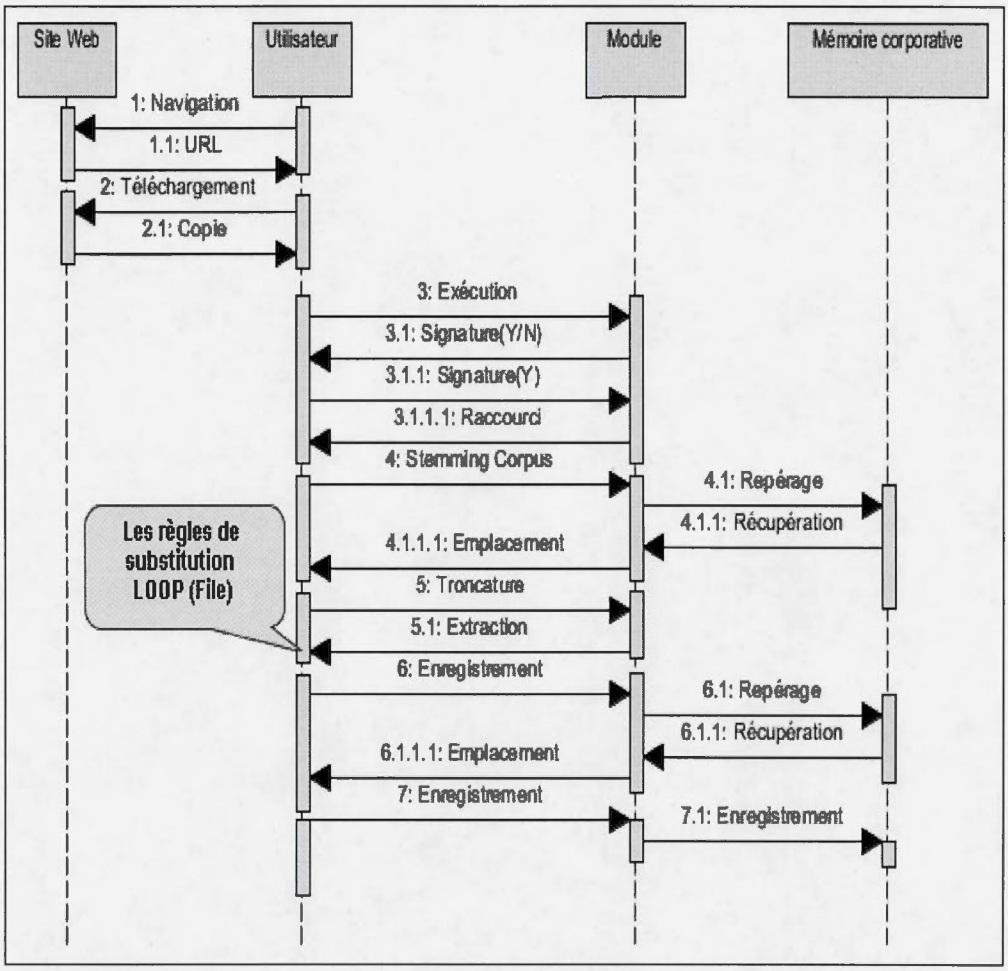


Diagramme de séquence acteur/module du cas d'utilisation: Troncature corpus.



5) Exigences non fonctionnelles

5.1) Déploiement: (Windows, Linux, Windows NT, Mac OS, Solaris, CP/M, Mandriva, SUSE, ubuntu, etc.).

5.2) Hébergement: JAR,HTML et JNLP (le serveur Apache et IIS).

5.3) Signature numérique: keystore TextMiner -alias choukri, LANCI 2011, 06 MOIS.

6) Informations complémentaires

6.1) Caractéristiques: La modularité, la portabilité, la sécurité, la flexibilité, l'extensibilité, etc.

.....

ANNEXE K

Spécifications des exigences d'un module (adapté de la norme IEEE 830)

1) Objectifs: ce document permet de détailler les fonctionnalités du module: Motfontionnel.

1.1) Portée: Fichier et corpus.

2) Contexte de l'application: Text Mining, Web sémantique et repérage.

2.1) OBJECTIF: Le filtrage des mots fonctionnels.

3) Description générale du module

3.1) Vue d'ensemble des fonctions du produit: Filtrage (fichier et corpus).

3.2) Description des utilisateurs: utilisateurs.

4) Le diagramme de séquences acteurs/module:

Diagramme de séquence acteur/système du cas d'utilisation: Filtrage d'un fichier.

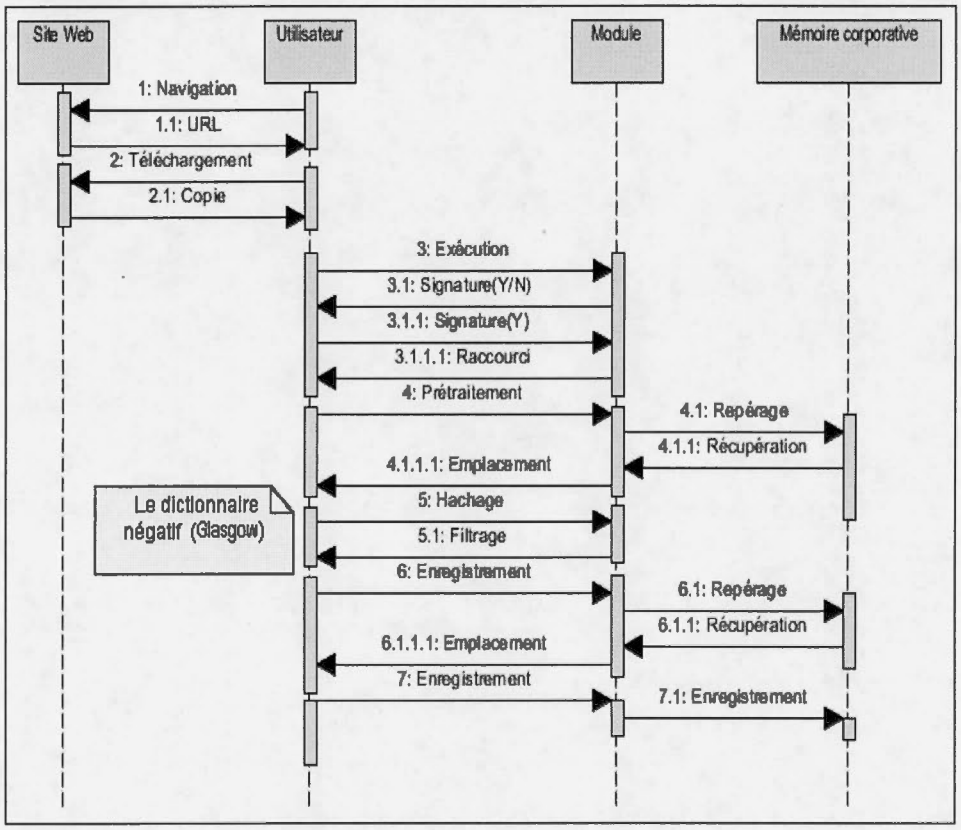


Diagramme de séquence acteur/système du cas d'utilisation: Filtrage d'un corpus.

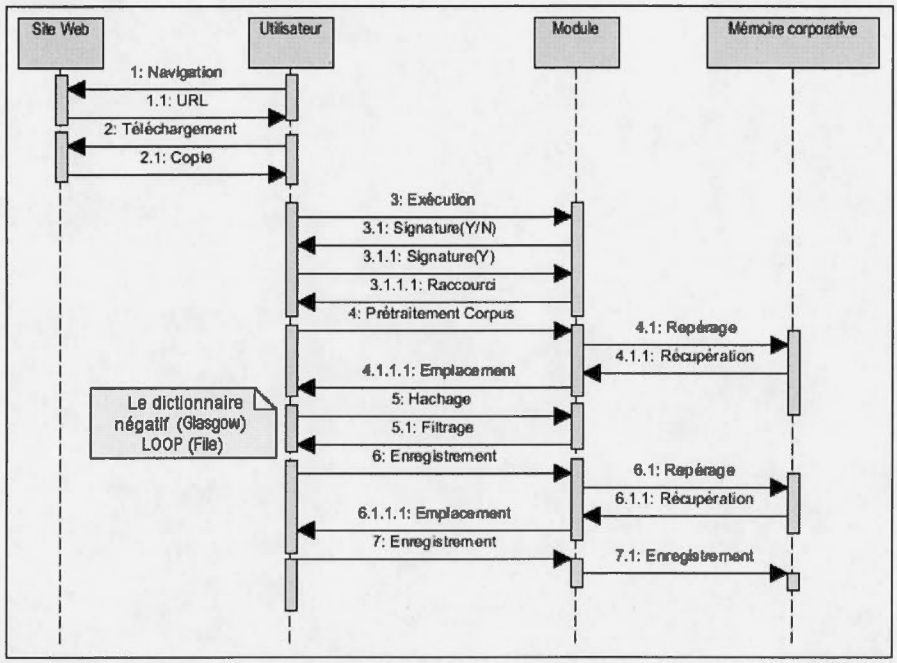
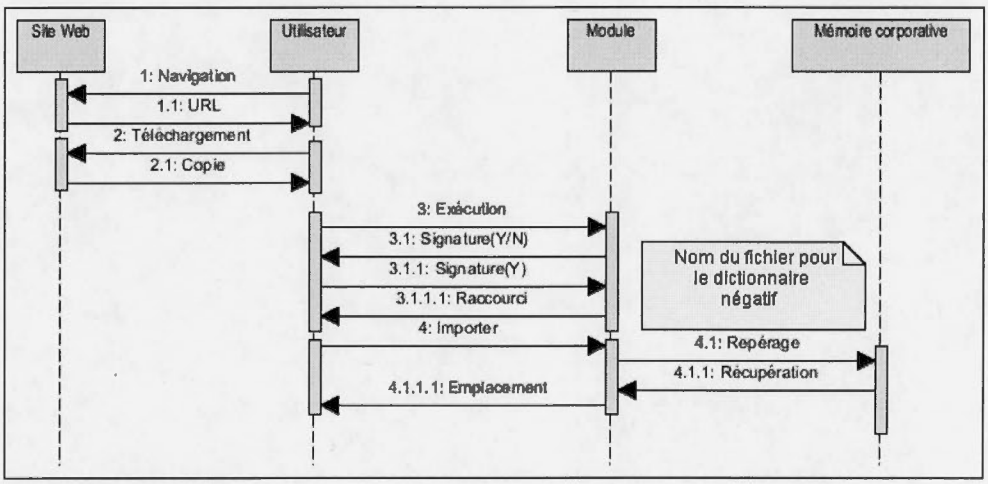


Diagramme de séquence acteur/système du cas d'utilisation: Importer le dictionnaire négatif.



5) Exigences non fonctionnelles

5.1) Déploiement: (Windows, Linux, Windows NT, Mac OS, Solaris, CP/M, Mandriva, SUSE, ubuntu,..).

5.2) Hébergement: JAR, HTML, et JNLP (le serveur Apache et IIS).

5.3) Signature numérique: keystore TextMiner -alias choukri, LANCI 2011, 06 MOIS.

6) Informations complémentaires

6.1) Caractéristiques: La modularité, la portabilité, la sécurité, la flexibilité, l'extensibilité, etc.

ANNEXE L

Spécifications des exigences d'un module (adapté de la norme IEEE 830)

1) Objectifs: ce document permet de détailler les fonctionnalités du module: Clustering

1.1) Portée: Corpus d'apprentissage.

2) Contexte de l'application: Data Mining et apprentissage machine.

2.1) OBJECTIF: apprentissage / test de l'architecture connexionniste Fuzzy ART.

3) Description générale du module

3.1) Vue d'ensemble des fonctions du produit: matrices des poids.

3.2) Description des utilisateurs: utilisateurs.

4) Le diagramme de séquences acteurs/module:

Diagramme de séquence acteur/système du cas d'utilisation: Apprentissage.

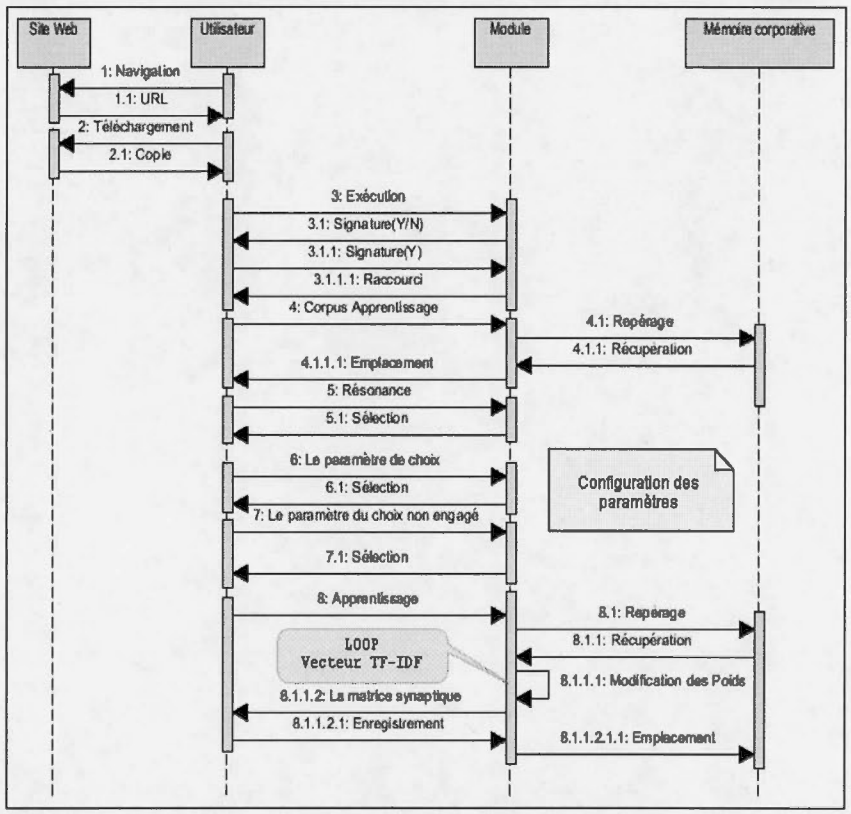
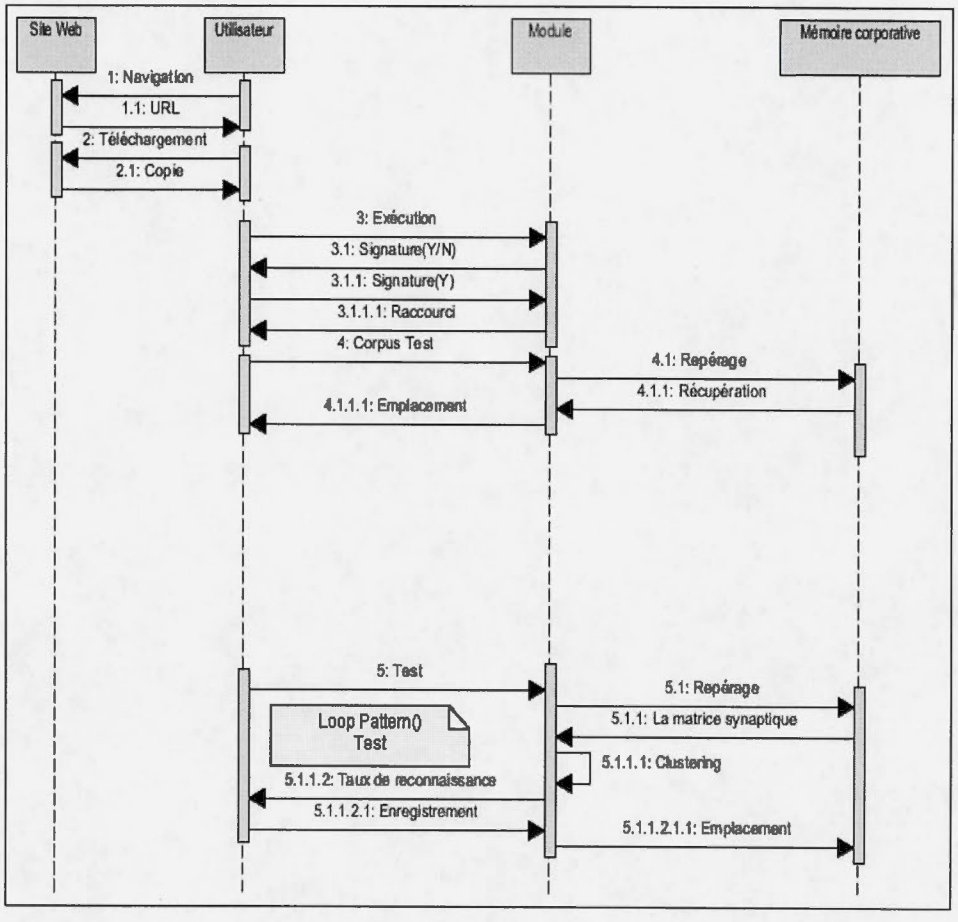


Diagramme de séquence acteur/système du cas d'utilisation: Reconnaissance.



5) Exigences non fonctionnelles

5.1) Déploiement: (Windows, Linux, Windows NT, Mac OS, Solaris, CP/M, Mandriva, SUSE, ubuntu, etc.).

5.2) Hébergement: JAR, HTML et JNLP (le serveur Apache et IIS).

5.3) Signature numérique: keystore TextMiner -alias choukri , LANCI 2011, 06 MOIS.

6) Informations complémentaires

6.1) Caractéristiques: La modularité, la portabilité, la sécurité, la flexibilité, l'extensibilité, etc.

.....

ANNEXE M

LE CODE OWL-DL DES ALGORITHMES DATA MINING

```
<?xml version="1.0"?>
<rdf:RDFxmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" ,
xmlns:xsd=http://www.w3.org/2001/XMLSchema# ,xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
 xmlns:owl=http://www.w3.org/2002/07/owl#, xmlns="http://www.owl-ontologies.com/unnamed.owl#"
 xmlns:p1=http://www.owl-ontologies.com/assert.owl#,xml:base="http://www.owl-
ontologies.com/unnamed.owl">
<owl:Class rdf:ID="Density-basedAlgorithm">
 <rdfs:subClassOf>
  <owl:Class rdf:ID="ClusteringAlgorithm"/>
 </rdfs:subClassOf>
 <owl:disjointWith>
  <owl:Class rdf:ID="PartitioningAlgorithm"/>
 </owl:disjointWith>
 <owl:disjointWith>
  <owl:Class rdf:ID="Graph-basedAlgorithm"/>
 </owl:disjointWith>
 <owl:disjointWith>
  <owl:Class rdf:ID="HierarchicalAlgorithm"/>
 </owl:disjointWith>
 </owl:Class>
 <owl:Class rdf:ID="AssociationAlgorithmSupport">
 <owl:disjointWith>
  <owl:Class rdf:ID="RegressionAlgorithmSupport"/>
 </owl:disjointWith>
   <owl:disjointWith>
   .....
   .....
   .....
  <owl:Class rdf:ID="ClusteringAlgorithmSupport"/>
 </owl:disjointWith>
 <rdfs:subClassOf>
  <owl:Class rdf:about="#AlgorithmSupport"/>
 </rdfs:subClassOf>
<owl:disjointWith>
```

```
 <owl:Class rdf:ID="ClassificationAlgorithmSupport"/>
</owl:disjointWith>
<owl:disjointWith>
 <owl:Class rdf:ID="ClassificationCriteria"/>
</owl:disjointWith>
</owl:Class>
 <owl:Class rdf:ID="Table"/>
</rdfs:subClassOf>
</owl:Class>
```

Figure A13.1 Le code OWL-DL des algorithmes Data Mining

ANNEXE N

LA CONNEXION TCP-IP RACERPRO

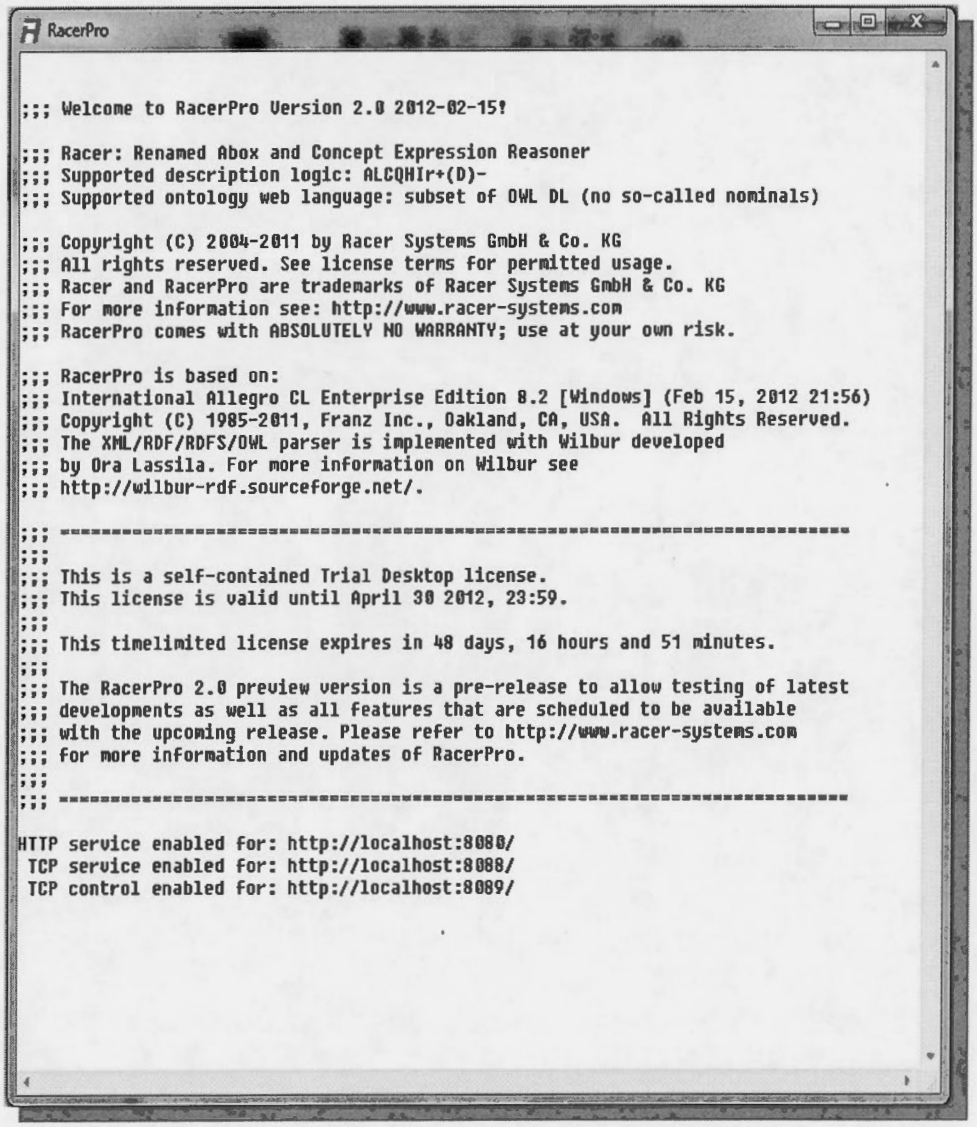


Figure A14.1 La connection TCP/IP avec le serveur RacerPro

ANNEXE 0

Spécifications des exigences d'un module (adapté de la norme IEEE 830)

1) Objectifs: ce document permet de détailler les fonctionnalités du module: Visualisation.

1.1) Portée: Ontologie, clustering, etc.

2) Contexte de l'application: Text Mining, Web sémantique et repérage.

2.1) OBJECTIF: Visualisation des ontologies.

3) Description générale du module

3.1) Vue d'ensemble des fonctions du produit: Visualisation locale et visualisation globale.

3.2) Description des utilisateurs: utilisateurs.

4) Le diagramme de séquences acteurs/module:

Diagramme de séquence acteur/système du cas d'utilisation: Visualisation locale.

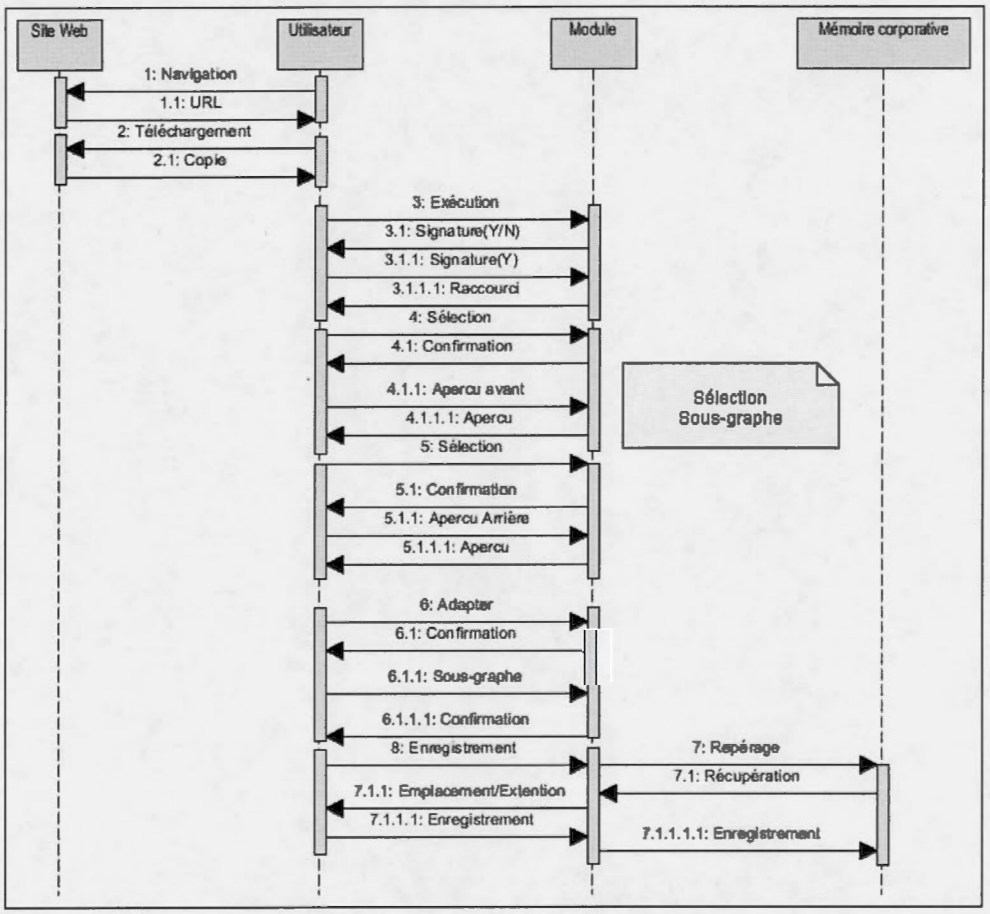
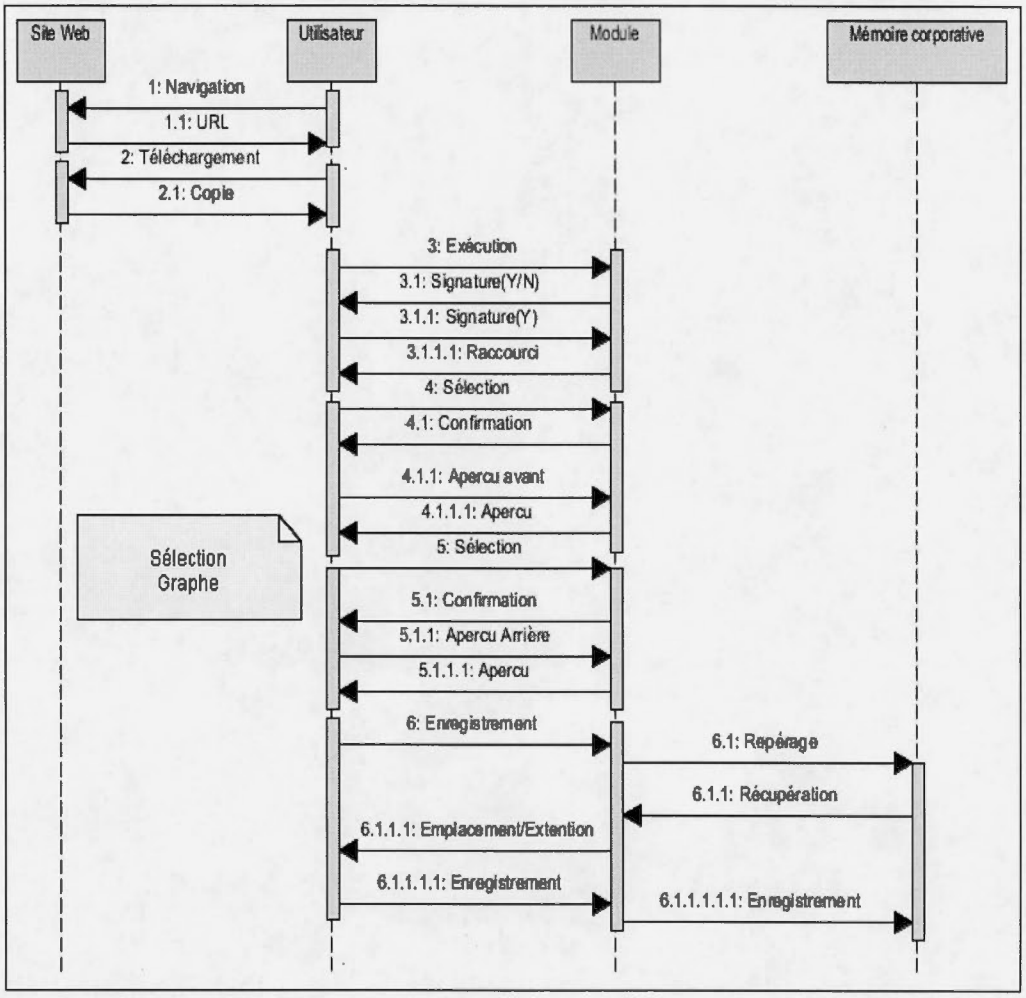


Diagramme de séquence acteur/système du cas d'utilisation: Visualisation Globale.



5) Exigences non fonctionnelles

5.1) Déploiement: (Windows, Linux, Windows NT, Mac OS, Solaris, CP/M, Mandriva, SUSE, ubuntu, etc.).

5.2) Hébergement: JAR, HTML et JNLP (le serveur Apache et IIS).

5.3) Signature numérique: keystore TextMiner -alias choukri , LANCI 2011, 06 MOIS.

6) Informations complémentaires

6.1) Caractéristiques: La modularité, la portabilité, la sécurité, la flexibilité, l'extensibilité, etc.

.....

ANNEXE P

LE DOT CRISP-DM-OWL

```
digraph G {graph [      fontname = "Helvetica-Oblique",
            fontsize = 20,
            label = "\n\n\n\n CRISP-DM-OWL.OWL",
            size = "1,0" ];
      node [      label = "\N",
            shape = polygon,
            sides = 4,
            distortion = "0.0",
            orientation = "0.0",
            skew = "0.0",
            color = lightskyblue1,
            style = filled,
            fontname = "Helvetica-Outline" ];
      graph [lp= "594,114"];
      graph [bb= "0,0,1189,1274"];
Ontologie_Datamining.OWL-DL->Root
Root->Technique
Technique->Assosiation
Technique->Selection
Technique->Clustering
Clustering->Density
Clustering->Graph
Graph->Node
vertice [color=gold1]
Node->vertice
Graph->Edg
Clustering->Hierarchical
Clustering->Partitioning
Technique->Regression
Linear [color=gold1]
Regression->Linear
Multiple [color=gold1]
Regression->Multiple
Polynomial [color=gold1]
Regression->Polynomial
Logistic [color=gold1]
Regression->Logistic
Proportional [color=gold1]
Regression->Proportional
Technique->Classification
Classification->Bayesian
Classification->Neighbor
Classification->Neural
Neural->Activation
Exponential [color=gold1]
Activation->Exponential
Signum [color=gold1]
```

```
Activation->Signum
Sigmoid [color=gold1]
Activation->Sigmoid
Hyperboloic [color=gold1]
Activation->Hyperboloic
Neural->Network
Network->Connection
Connection->Weight
Network->Layer
Input [color=gold1]
Layer->Input
Ouput [color=gold1]
Layer->Ouput
Hidden [color=gold1]
Layer->Hidden
Layer->Neurone
Formel [color=gold1]
Neurone->Formel
Radiale [color=gold1]
Neurone->Radiale
SigmaPi [color=gold1]
Neurone->SigmaPi
Neural->Architecture
Architecture->Feedforward
Feedforward->Error
Feedforward->Sensitivity
Feedforward->Target
Feedforward->Backpropagation
Backpropagation->Adaline
Backpropagation->Madaline
Backpropagation->MLP
Backpropagation->RBF
Backpropagation->Convolutional
TDNN [color=gold1]
Convolutional->TDNN
Feedforward->Gradient
Architecture->Recurrent
Elman [color=gold1]
Recurrent->Elman
Jordan [color=gold1]
Recurrent->Jordan
Classification->Induction
Induction->Tree
ID3 [color=gold1]
Tree->ID3
C45 [color=gold1]
Tree->C45
CART [color=gold1]
Tree->CART
Tree->Node
Tree->Link
Tree->Branch
Tree->Leaf
Tree->Descendent
Tree->Split
Tree->Branching
```

```
Tree->Ratio
Tree->Inpurity
Tree->Misclassification
Tree->Purity
Tree->Greedy
Tree->Twoing
Tree->Length
Tree->Confidence
Tree->Pruning
Tree->Merging
Technique->Prediction
Technique->Description
Technique->Summarization
Root->Task
Task->Preparation
Preparation->Cleaning
Preparation->Construction
Preparation->Formating
Preparation->Integration
Integration->Warehousing
Integration->Federation
Integration->Wrapper
TSIMMIS [color=gold1]
Wrapper->TSIMMIS
Infomaster [color=gold1]
Wrapper->Infomaster
Integration->Ontology
Ontology->Evolution
Ontology->Mapping
Ontology->Construction
Construction->Aquisition
Construction->Specification
Construction->Conceptualization
Construction->Reuse
Construction->Alignment
Construction->Maintenance
Construction->Versioning
Construction->Evaluation
GoldStandard [color=gold1]
Evaluation->GoldStandard
DataDriven [color=gold1]
Evaluation->DataDriven
Evaluation->Criteria
Conciseness [color=gold1]
Criteria->Conciseness
completeness [color=gold1]
Criteria->completeness
Expandability [color=gold1]
Criteria->Expandability
Sensitiveness [color=gold1]
Criteria->Sensitiveness
Construction->Documentation
Integration->Memory
Task->Learning
Learning->Rate
Learning->Unsupervized
Unsupervized->Competitive
```

```
Unsupervized->Adaptative
Learning->Supervized
Supervized->Simulated
Simulated->Annealing
Annealing->Trajectory
Annealing->Schedule
Schedule->Stochastic
Schedule->Deterministic
Annealing->Model
Model->Boltzmann
Annealing->State
Annealing->Energy
Annealing->Transition
Annealing->Temperature
Annealing->Configuration
Annealing->MonteCarlo
Simulated->Genetic
Genetic->Parent
Genetic->Score
Genetic->Offspring
Genetic->Mating
Genetic->Replication
Genetic->Gene
Gene->Population
Gene->Chromosome
Genetic->Reproduction
Reproduction->Crossover
Reproduction->Cut
Reproduction->Mutation
Genetic->Selection
Selection->Filter
Selection->Feature
Selection->Fitness
Supervized->Exploration
Supervized->Optimization
Optimization->Colony
Colony->Pheromone
Colony->Evaporation
Colony->Desirability
Colony->Stagnation
Supervized->Reinforcement
Supervized->Association
Task->Recognition
.................
.................
.................
.................
```

CONTRIBUTIONS SCIENTIFIQUES

Nos travaux ont été proposés pour répondre aux nouveaux défis lies à l'apprentissage des ontologies, notamment pour l'intégration des connaissances corporative d'une part, et d'autre part pour la prise en compte de l'hétérogénéité des connaissances.

Il s'agit d'une approche conceptuelle issue d'une recherche multidisciplinaire:

- L'intégration de la connaissance corporative,
- L'apprentissage machine,
- L'évolution des ontologies.

Dans cet objectif, j'ai participé à plusieurs conférences internationales à travers des collaborations scientifiques dont je rapporte ici une synthèse.

Dans les annexes (17-28), je présente mes productions scientifiques dans un ordre chronologique.

ANNEXE Q

EGC-M 2012

L'annexe décrit l'article: A new approach to the evolution of Data Mining ontology. Publié en 2012 dans la conférence EGC-M'2012: The 3rd International Conference on the Extraction and Management of Knowledge - Maghreb. November 12, 15, 2012, Hammamet, Tunisia.

Résumé: L'article décrit une approche hybride qui utilise l'apprentissage machine, le traitement automatique du langage naturel et la recherche d'information pour enrichir les ontologies. Afin d'éviter d'induire en erreur le modèle d'indexation vectorielle, le dictionnaire négatif Glasgow et la troncature Porter sont les deux méthodes de prétraitement utilisées pour supprimer le bruit. Le clustering de la théorie de la résonance adaptative floue est utilisé pour découvrir les changements candidats. Notre approche utilise un processus d'alignement pour comparer les changements candidats et les entités de l'ontologie. En raison de la correspondance entre OWL et la logique de description, l'inférence Racer Pro est utilisée pour vérifier la cohérence de l'ontologie enrichie.

EGC-M'2012 Proceedings of The 3rd International Conference on the Extraction and Management of Knowledge - Maghreb.
Pages 100 – 107.

URL: https://oraprdnt.uqtr.uquebec.ca/pls/public/gscw031?owa_no_site=21&owa_no_fiche=58

ANNEXE R

ICCIT 2013

L'annexe décrit l'article: Truncated singular value decomposition for semantic-based data retrieval. Publié en 2013 dans la conférence ICCIT 2013: The Third International Conference on Communications and Information Technology. American University of Beirut. June 19-21, 2013 – Beirut, Lebanon.

Résumé: L'article décrit une nouvelle approche pour l'indexation et le repérage des ressources textuelles. L'indexation vectorielle des documents génère un espace hautement dimensionnel. Par conséquent, un critère robuste de sélection basé sur la décomposition en valeurs singulières tronquées a été utilisé pour réduire l'espace d'indexation. Cette approche d'emballage considère les biais de l'algorithme de la décomposition en valeurs singulières et l'effet du sous-ensemble des variables choisies. Ainsi, elle supprime efficacement les variables redondantes et elle améliore la généralisation.

Keywords: semantic analysis; variable selection; clustering; indexation; learning, retrieval; truncated singular value decomposition.

ICCIT '2013 Proceedings of the 2013 International Conference on Communications and Information Technology. Pages 61 – 66.
DOI:10.1109/ICCITechnology.2013.6579523.

URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6579523

ANNEXE S

ANT 13

L'annexe décrit l'article: A new on-line digital conceptual model oriented corporate memory constructing: Taking Data Mining models as a case. Publié en 2013 dans la conférence: ANT 13, the 4th International Conference on Ambient Systems, Networks and Technologies. June 25-28, 2013, Halifax, Nova Scotia, Canada.

Résumé: L'article décrit une nouvelle approche conceptuelle pour l'intégration de la connaissance corporative en utilisant l'indexation composée et une ontologie spécifiant la sémantique. La structure optimisée de l'indexation composée consomme moins de descripteurs de fichiers et moins de ressources computationnelles durant le processus d'indexation. De plus, le mécanisme de repérage prend en charge plusieurs processus de recherche avancée et offre plusieurs avantages, il s'agit notamment de la vitesse de recherche due à la structuration et l'optimisation d'indexation.

Procedia Computer Science 19 (2013) 977 – 983.

Keywords: preprocessing, integration, corporate memory, data mining, ontology, information retrieval, machine learning.

URL: http://www.sciencedirect.com/science/article/pii/S1877050913007448

ANNEXE T

IEEE/ACM ASONAM 2013

L'annexe décrit l'article: Enhancing text clustering model based on Truncated Singular Value Decomposition, Fuzzy ART and cross Validation. Publié en 2013 dans la conférence ASONAM 2013, The 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining.

Niagara Falls, Canada, August 25-28, 2013.

Résumé: L'article décrit une nouvelle approche pour le clustering textuel. Afin de sélectionner les indexes pertinents, nous avons appliqué la décomposition en valeurs singulières tronquées ou TSVD (Truncated Singular Value Decomposition). La validation croisée double a été utilisée pour sélectionner le modèle en minimisant l'erreur totale des estimations. Cette méthode est importante non seulement pour l'estimation du taux de reconnaissance mais aussi pour la sélection du modèle à partir de l'ensemble d'apprentissage. Il s'agit d'une méthode indispensable pour réduire la variance et ainsi améliorer la généralisation.

Keywords: Learning, Data Mining, NLP, TSVD, variable selection, semantic analysis, model selection.

ASONAM 2013 Proceedings of The 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining.

Pages 1078-1083.

URL: http://dl.acm.org/citation.cfm?id=2500317

ANNEXE U

ACM RACS '13

L'annexe décrit l'article: Using Hamming Similarity to Map Ontology Learning: A New Data Mining System. Publié en 2013 dans la conférence ACM RACS '13 Research in Adaptive and Convergent Systems. Concordia University, Montreal, QC, Canada, October 1-4, 2014.

Résumé: L'article décrit une nouvelle approche pour l'apprentissage des ontologies utilisant l'indexation vectorielle, la sélection des variables et le clustering. Afin d'éviter d'induire en erreur le modèle de clustering, la décomposition en valeurs singulières tronquées est utilisée pour sélectionner les variables pertinentes. Le modèle connexionniste de la théorie de la résonance adaptative floue est utilisé pour découvrir les changements candidats. Pour associer ces derneirs aux artéfacts ontologiques de base, notre approche utilise un processus d'alignement basé sur la distance de Hamming.

RACS '13 Proceedings of the 2013 Research in Adaptive and Convergent Systems.

Pages 82-87.

URL: http://dl.acm.org/citation.cfm?id=2513232

ANNEXE V

ACM RACS '13

L'annexe décrit l'article: A new on-line digital conceptual model oriented corporate memory constructing: taking unstructured text as a case. Publié en 2013 dans la conférence ACM RACS '13 Research in Adaptive and Convergent Systems. Concordia University, Montreal, QC, Canada, October 1-4, 2014.

Résumé: L'article décrit une nouvelle approche conceptuelle pour l'intégration de la connaissance corporative utilisant l'indexation et la sélection des variables. Notre approche tire profit des processus implémentés dans le cadre de cette thèse (sélection des variables et indexation composée). L'étalonnage des modules montre que le processus de la sélection des variables améliore la performance de repérage.

RACS '13 Proceedings of the 2013 Research in Adaptive and Convergent Systems.
Pages 88-93.
URL: http://dl.acm.org/citation.cfm?id=2513233

ANNEXE W

EUSPN 2013

L'annexe décrit l'article: A New Data Mining System for Ontology Learning Using Dynamic Time Warping Alignment as a Case. Publié en 2013 dans la conférence EUSPN 2013: The 4th International Conference on Emerging Ubiquitous Systems and Pervasive Networks.
October 21-24, 2013, Niagara Falls, Ontario, Canada.

Résumé: L'article décrit une nouvelle approche pour l'apprentissage des ontologies utilisant plusieurs processus, entre autres:

- L'indexation vectorielle basée sur la fréquence des termes et la fréquence inverse du document.
- Le modèle d'emballage de la décomposition en valeurs singulières tronquées.
- Le clustering de la théorie de la résonance adaptative floue.
- L'alignement de la Déformation Temporelle Dynamique pour associer les modèles cachés aux artéfacts ontologiques de base.

EUSPN 2013, Proceedings of The International Conference on Emerging Ubiquitous Systems and Pervasive Networks. Procedia Computer Science 21 (2013) 75 – 82.
URL: http://www.sciencedirect.com/science/article/pii/S1877050913008053

ANNEXE Y

ACM C3S2E '14

L'annexe décrit l'article: A new conceptual model for dynamic text clustering Using unstructured text as a case. Publié en 2014 dans la conférence ACM C3S2E '14: The Seventh International C* Conference on Computer Science & Software Engineering. Concordia University, Montreal, QC, Canada, August 4-6, 2014.

Résumé: L'article décrit une nouvelle approche pour le clustering dynamique utilisant la sélection des variables et l'architecture connexionniste de la théorie de la résonance adaptative floue. La sélection des variables implique un processus d'optimisation combinatoire basé sur la décomposition en valeurs singulières. La génération été décrite comme une fonction objective qui vise à maximiser le cumul de variances proportionnelles aux valeurs singulières. Le clustering Fuzzy ART est utilisé pour grouper les vecteurs-documents sur la base leur distances. De plus, l'initialisation typique et la configuration de l'architecture connexionniste ont été utilisées pour réduire le temps de calcul et pour chercher une convergence rapide vers le voisinage de la solution.

Proceedings of the 2014 International C* Conference on Computer Science & Software Engineering
Article No. 13, DOI:10.1145/2641483.2641538

URL: http://dl.acm.org/citation.cfm?id=2641538

ANNEXE X

EUSPN 2014

L'annexe décrit l'article: A Semantic-based Variables Selection for Ontology Learning Taking Jaccard Alignment as Case. Publié en 2014 dans la conférence EUSPN 2014: The 5th International Conference on Emerging Ubiquitous Systems and Pervasive Networks.
September 22-25, 2014, Halifax, Nova Scotia, Canada.

Résumé: L'article décrit une méthode de sélection des variables pour l'apprentissage des ontologies. La décomposition en valeurs singulières tronquées est utilisée chercher les corrélations entre les indexes. De plus, l'indice de chevauchement Jaccard est la stratégie adoptée pour comparer les étiquettes les artéfacts de l'ontologie. Cette stratégie consiste essentiellement en l'utilisation de la technique de distance pour chercher l'alignement optimal entre les deux représentations.

EUSPN 2014, Proceedings of the International Conference on Emerging Ubiquitous Systems and Pervasive Networks. Procedia Computer Science 37 (2014) 56-63.

URL: http://www.sciencedirect.com/science/article/pii/S1877050914009776

ANNEXE Z

EUSPN 2014

L'annexe décrit l'article: A New Distributed Expert System to Ontology Evaluation. Publié en 2014 dans la conférence EUSPN 2014: The 5th International Conference on Emerging Ubiquitous Systems and Pervasive Networks. September 22-25, 2014, Halifax, Nova Scotia, Canada.

Résumé: L'article décrit un système distribué pour l'évaluation des ontologies utilisant la logique descriptive $\mathcal{SHOIN}$. Le système permet de vérifier le raisonnement terminologique et la description du monde et, entre autres, d'évaluer la cohérence, la subsomption, l'instanciation, les cycles et les points fixes. Notre approche d'évaluation est indépendante de la conceptualisation du domaine modélisé et considère les caractéristiques principales de la structuration de l'ontologie et sa population (concepts, instances, axiomes, relations, etc.).

EUSPN 2014, Proceedings of the International Conference on Emerging Ubiquitous Systems and Pervasive Networks. Procedia Computer Science 37 (2014) 48-55.

URL: http://www.sciencedirect.com/science/article/pii/S1877050914009764

ANNEXE XX

JADT10

L'annexe décrit l'article: Analyse des variations entre partitions générées par différentes techniques de classification automatique de textes. Publié en 2010 dans la conférence JADT10, 10th International Conference on Statistical Analysis of Textual Data SAPIENZA - University of Rome (Italy) 9,11 June 2010.

Résumé: L'article décrit une étude basée sur une comparaison du comportement de différents algorithmes de clustering. Les corrélations globales et locales ont été utilisées pour évaluer la performance de généralisation et comparer les différents algorithmes de clustering.

Mots clés: clustering, corrélation, partition, ART1, K-means, SOM, EM.
JADT10 Proceedings of The 10th International Conference on Statistical Analysis of Textual Data Italy 9,11 June 2010.
Pages 37 –48.

URL: http://lexicometrica.univ-paris3.fr/jadt/jadt2010/allegati/JADT-2010-0037-0048_183-Chartier.pdf

ANNEXE ZZ

ACFAS 10

L'annexe décrit l'article: Un système Data Mining en-ligne pour la maintenance ontologique d'une mémoire corporative DM. Publié en 2010 dans la conférence: 78e Congrès de l'Acfas. May 10-14, 2010, Montréal, Canada.

Résumé: L'article décrit un système Data Mining en-ligne pour la maintenance ontologique d'une mémoire corporative DM.

Mots clés: Data Mining, traitement automatique du langage naturel, apprentissage machine, recherche d'information, intégration, ontologie, mémoire corporative.

URL: http://www.umontreal.ca/acfas2010/

BIBLIOGRAPHIE

Abecker, A., A. Bernardi, K. Hinkelmann, O. Kuhn et M. Sintek. 1998. «Toward a technology for organizational memories». *Intelligent Systems and Their Applications, IEEE*, vol. 13, no 3, p. 40-48.

Agirre, Eneko, Olatz Ansa, Eduard Hovy et David Martínez. 2000. «Enriching very large ontologies using the WWW». *arXiv preprint cs/0010026*.

Alexandru-Lucian, Ginsca, et Adrian Iftene. 2010. *Using a genetic algorithm for optimizing the similarity aggregation step in the process of ontology alignment*: *Roedunet International Conference (RoEduNet), 2010 9th* (24-26 June 2010). 118-122 p.

Alter, O., P.O. Brown et D. Botstein. 2000. «Singular value decomposition for genome-wide expression data processing and modeling». *Proceedings of the National Academy of Sciences*, vol. 97, no 18, p. 10101-10106.

Baader, Franz. 2003. *The description logic handbook: theory, implementation, and applications*. Cambridge, Angleterre: Cambridge University Press, xvii, 555 p.

Bahi, J. M., Christophe Guyeux et A. Makhoul. 2010. *Efficient and Robust Secure Aggregation of Encrypted Data in Sensor Networks*: *Sensor Technologies and Applications (SENSORCOMM), 2010 Fourth International Conference on* (18-25 July 2010). 472-477 p.

Baowen, Xu, Lu Jianjiang et Huang Gangshi. 2003. *A constrained non-negative matrix factorization in information retrieval*: *Information Reuse and Integration, 2003. IRI 2003. IEEE International Conference on* (27-29 Oct. 2003). 273-277 p.

Barghouti, N., J. Mocenigo et W. Lee. 1997. *Grappa: A graph package in java*. Springer, 336-343 p.

Bechhofer, S., R. Möller et P. Crowther. 2003. *The DIG description logic interface: DIG/1.1*.

Beneventano, D., S. Bergamaschi, F. Guerra et M. Vincini. 2003. «Synthesizing an integrated ontology». *Internet Computing, IEEE*, vol. 7, no 5, p. 42-51.

Beneventano, Domenico, Nikolai Dahlem, Sabina El Haoum, Axel Hahn, Daniele Montanari et Matthias Reinelt. 2008. «Ontology-driven semantic mapping». In *Enterprise Interoperability III*, p. 329-341: Springer.

Bennani, Y. 2001. «Systèmes d'apprentissage connexionnistes: Sélection de variables». *Revue d'intelligence artificielle*, vol. 15, no 3-4, p. 303-316.

Berkhin, Pavel. 2006. «A survey of clustering data mining techniques». In *Grouping multidimensional data*, p. 25-71: Springer.

Berners-Lee, J. Hendler et O. Lassila. 2001. «The SemanticWeb.». *ScientificAmerican, 284(5):34–43,*.

Berzal, F., et N. Matín. 2002. «Data mining: concepts and techniques by Jiawei Han and Micheline Kamber». *ACM Sigmod Record*, vol. 31, no 2, p. 66-68.

Bhamidipati, N. L., et S. K. Pal. 2007. «Stemming via Distribution-Based Word Segregation for Classification and Retrieval». *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 37, no 2, p. 350-360.

Bin, Zhao, J. Kwok, Wang Fei et Zhang Changshui. 2009. *Unsupervised Maximum Margin Feature Selection with manifold regularization: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on* (20-25 June 2009). 888-895 p.

Blomqvist, Eva. 2007. «Ontocase-a pattern-based ontology construction approach». In *On the Move to Meaningful Internet Systems 2007: CoopIS, DOA, ODBASE, GADA, and IS*, p. 971-988: Springer.

Bo, Wang, Jia Yan, Han Yi et Han Weihong. 2009. *Effective Feature Selection on Data with Uncertain Labels: Data Engineering, 2009. ICDE '09. IEEE 25th International Conference on* (March 29 2009-April 2 2009). 1657-1662 p.

Borst, W.N. 1997. *Construction of engineering ontologies for knowledge sharing and reuse*: Universiteit Twente p.

Boukhadoum, Mounir. 2010. «Introduction au traitement de l'information par réseaux neuroniques ». *UQAM , 9310.*

Brank, Janez, Marko Grobelnik et Dunja Mladenić. 2005. «A survey of ontology evaluation techniques».

Braun, A. C., U. Weidner et S. Hinz. 2011. *Support vector machines, import vector machines and relevance vector machines for hyperspectral classification — A comparison: Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), 2011 3rd Workshop on* (6-9 June 2011). 1-4 p.

Carpenter, G.A., S. Grossberg et D.B. Rosen. 1991. «Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system». *Neural networks*, vol. 4, no 6, p. 759-771.

Carter, C., A. El Rhalibi, M. Merabti et M. Price. 2009. *Homura and Net-Homura: The creation and web-based deployment of cross-platform 3D games: Ultra Modern Telecommunications & Workshops, 2009. ICUMT '09. International Conference on* (12-14 Oct. 2009). 1-8 p.

Chao, Ke, et Zhong Shangping. 2012. *Kernel target alignment for feature kernel selection in universal steganographic detection based on multiple kernel SVM: Instrumentation & Measurement, Sensor Network and Automation (IMSNA), 2012 International Symposium on* (25-28 Aug. 2012). 222-227 p.

Cha, S.-H. 2007. "Comprehensive survey on distance/similarity measures between probability density functions." City 1(2): 1.

Chartier, J.F., J.G. Meunier et C. Djellali. 2010. «Analyse des variations entre partitions générées par différentes techniques de classification automatique de textes». *JADT*, vol. 10th International Conference on Statistical Analysis of Textual Data

Chen, G., et al. (2002). "Evaluation and comparison of clustering algorithms in analyzing ES cell gene expression data." *Statistica Sinica* 12(1): 241-262.

Chiang, S.-S. C.-I. Chang. 2001. Discrimination measures for target classification. *Geoscience and Remote Sensing Symposium, 2001. IGARSS'01.* IEEE 2001 International, IEEE.

Choi, Namyoun, Il-Yeol Song et Hyoil Han. 2006. «A survey on ontology mapping». *ACM Sigmod Record*, vol. 35, no 3, p. 34-41.

Chung-Hsien, Wu, Hsia Chi-Chun, Lee Chung-Han et Lin Mai-Chun. 2010. «Hierarchical Prosody Conversion Using Regression-Based Clustering for Emotional Speech Synthesis». *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 18, no 6, p. 1394-1405.

CHRYSAFIS, T. 2003. On-Line Analytical Processing. *Second International Student Spring Symposium on Contemporary Topics In IT (CTIT),* Thessaloniki, Greece,28-1 March 2003.

Cimiano, Philipp, Andreas Hotho et Steffen Staab. 2005. «Learning Concept Hierarchies from Text Corpora using Formal Concept Analysis». *J. Artif. Intell. Res.(JAIR)*, vol. 24, p. 305-339.

Dandach, S. H., R. Carli et F. Bullo. 2010. *Accuracy and decision time for a class of sequential decision aggregation rules*: *Decision and Control (CDC), 2010 49th IEEE Conference on* (15-17 Dec. 2010). 4777-4782 p.

Deepa, T. and D. M. Punithavalli 2012. A GLFES and DFT Technique for feature selection in High-dimensional Imbalanced Dataset, IJCSE.

Dieng, R., O. Corby, A. Giboin et M. Ribière. 1998. «Methods and tools for corporate knowledge management».

Djellali, Choukri. 2012. «TextMiner: un système Text Mining en-ligne » *colloque Informatique Cognitive TELUQ, UQAM 2012*.

———. 2013a. «Enhancing text Clustering model based on Truncated Singular Value Decomposition and Fuzzy ART and Cross
Validation». *2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, p. 1078-1083.

———. 2013d. *Truncated singular value decomposition for semantic-based data retrieval*: *ICCIT, 2013 Third International Conference on Communications and Information Technology , Digital Information Management & Security* (19-21 June 2013). 61-66 p.

Djellali, Choukri. 2013e. «A New Data Mining System for Ontology Learning Using Dynamic Time Warping Alignment as a Case». *Procedia Computer Science*, vol. 21, p. 75-82.

———. 2013f. «A New Digital Conceptual Model Oriented Corporate Memory Constructing: Taking Data Mining Models as a Case». *Procedia Computer Science*, vol. 19, p. 977-983.

———. 2013g. *A new on-line digital conceptual model oriented corporate memory constructing: taking unstructured text as a case*: *Proceedings of the 2013 Research in Adaptive and Convergent Systems*. ACM, 88-93 p.

———. 2013h. *Using hamming similarity to map ontology learning: a new data mining system: Proceedings of the 2013 Research in Adaptive and Convergent Systems*. ACM, 82-87 p.

———. 2014i. A New Distributed Expert System to Ontology Evaluation." Procedia Computer Science 37: 48-55.

———. 2014j. A Semantic-based Variables Selection for Ontology Learning Taking Jaccard Alignment as Case." Procedia Computer Science 37: 56-63.

———. 2014k. A new conceptual model for dynamic text clustering Using unstructured text as a case. Proceedings of the 2014 International C* Conference on Computer Science & Software Engineering, ACM.

Djellali, Choukri, Jean-Guy Meunier et Sylvain Delisle. 2012. *A new approach to the evolution of Data Mining ontology: EGC-M 2012: The 3rd International Conference on the Extraction and Management of Knowledge - Maghreb* (November 12-15, 2012
Hammamet, Tunisia). 100-107 p.

Doan, AnHai, Jayant Madhavan, Pedro Domingos et Alon Halevy. 2002. *Learning to map between ontologies on the semantic web: Proceedings of the 11th international conference on World Wide Web*. ACM, 662-673 p.

Duda, Richard O., Peter E. Hart et David G. Stork. 2001. *Pattern classification*, 2nd. New York ; Toronto: Wiley, xx, 654 p.

Ehrig, Marc, et York Sure. 2004. «Ontology mapping–an integrated approach». In *The Semantic Web: Research and Applications*, p. 76-91: Springer.

Faatz, A., et R. Steinmetz. 2002. «Ontology enrichment with texts from the WWW». *Semantic Web Mining*, p. 20.

Faure, David, et Thierry Poibeau. 2000. *First experiments of using semantic knowledge learned by ASIUM for information extraction task using INTEX: Proceedings of the ECAI Workshop on Ontology Learning*. Citeseer.

Frakes, William B., Ricardo A. et Baeza-Yates. 2000. «Information Retrieval: Data Structures & Algorithms.». *Prentice-Hall*

Gandon, Fabien, Rose Dieng-Kuntz, Olivier Corby et Alain Giboin. 2002. «Web sémantique et approche multi-agents pour la gestion d'une mémoire organisationnelle distribuée». *Journées Ingénierie des Connaissances*, p. 15-26.

Gang, Cheng, Wang Fei, Lv Haiyang et Zhang Yinling. 2011. *A new matching algorithm for Chinese place names: Geoinformatics, 2011 19th International Conference on* (24-26 June 2011). 1-4 p.

Gomez-Perez, A., et D. Manzano-Macho. 2003. «A survey of ontology learning methods and techniques». *OntoWeb Deliverable D*, vol. 1, p. 5.

Gómez-Pérez, Asunción. 1996. «Towards a framework to verify knowledge sharing technology». *Expert Systems with Applications*, vol. 11, no 4, p. 519-529.

———. 1999. «Evaluation of taxonomic knowledge in ontologies and knowledge bases».

Gruber, T.R. 1995. «Toward principles for the design of ontologies used for knowledge sharing». *International journal of human computer studies*, vol. 43, no 5, p. 907-928.

Guarino, N. 1998. *Formal ontology in information systems: proceedings of the first international conference (FOIS'98), June 6-8, Trento, Italy*: Ios Pr Inc p.

Guohua, Shen, Huang Zhiqiu, Zhu Xiaodong, Wang Lei et Xiang Gaoyou. 2007. *Using Description Logics Reasoner for Ontology Matching*: *Intelligent Information Technology Application, Workshop on* (2-3 Dec. 2007). 30-33 p.

Haarslev, V., K. Hidde, R. Möller et M. Wessel. 2011. «The RacerPro knowledge representation and reasoning system». *Semantic Web*.

Haarslev, Volker, et Ralf Möller. 2003. *Racer: A Core Inference Engine for the Semantic Web*: *EON*.

Haase, P., et Y. Sure. 2004. «State-of-the-art on ontology evolution».

Hacene, M. R., A. Napoli, P. Valtchev, Y. Toussaint et R. Bendaoud. 2008. *Ontology Learning from Text Using Relational Concept Analysis*: *e-Technologies, 2008 International MCETECH Conference on* (23-25 Jan. 2008). 154-163 p.

Hahn, Udo, et Stefan Schulz. 2000. «Towards very large terminological knowledge bases: a case study from medicine». In *Advances in Artificial Intelligence*, p. 176-186: Springer.

Haifeng, Li, Zhang Keshu et Jiang Tao. 2004. *Minimum entropy clustering and applications to gene expression analysis*: *Computational Systems Bioinformatics Conference, 2004. CSB 2004. Proceedings. 2004 IEEE* (16-19 Aug. 2004). 142-151 p.

Hall, L. O., I. B. Ozyurt et J. C. Bezdek. 1999. «Clustering with a genetically optimized approach». *Evolutionary Computation, IEEE Transactions on*, vol. 3, no 2, p. 103-112.

Hatcher, E., O. Gospodnetic et M. McCandless (2004). Lucene in action, Manning Publications

Heins, L.G., et D.R. Tauritz. 1995. «Adaptive resonance theory (ART): an introduction». *unpublished, http://web. mst. edu/~ tauritzd/art/artintro. html.*

Heping, Li, Liu Jie et Zhang Shuwu. 2011. *Hierarchical Latent Dirichlet Allocation models for realistic action recognition*: *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on* (22-27 May 2011). 1297-1300 p.

Huan, Liu, et Yu Lei. 2005. «Toward integrating feature selection algorithms for classification and clustering». *Knowledge and Data Engineering, IEEE Transactions on*, vol. 17, no 4, p. 491-502.

Ichise, R. 2009. «Evaluation of similarity measures for ontology mapping». *New Frontiers in Artificial Intelligence*, p. 15-25.

Iglesias, Carlos A, Mercedes Garijo et José C González. 1999. *A survey of agent-oriented methodologies*: *Intelligent Agents V: Agents Theories, Architectures, and Languages: 5th International Workshop, ATAL'98 Paris, France, July 4–7, 1998 Proceedings*. Springer, 317-330 p.

Issac, B., et W. J. Jap. 2009. *Implementing spam detection using Bayesian and Porter Stemmer keyword stripping approaches*: *TENCON 2009 - 2009 IEEE Region 10 Conference* (23-26 Jan. 2009). 1-5 p.

Jain, A., et D. Zongker. 1997. «Feature selection: Evaluation, application, and small sample performance». *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 19, no 2, p. 153-158.

Jiannan, Wang, Li Guoliang et Fe Jianhua. 2011. *Fast-join: An efficient method for fuzzy token matching based string similarity join*: *Data Engineering (ICDE), 2011 IEEE 27th International Conference on* (11-16 April 2011). 458-469 p.

Jiehan, Zhou, et R. Dieng-Kuntz. 2004. *Manufacturing ontology analysis and design: towards excellent manufacturing*: *Industrial Informatics, 2004. INDIN '04. 2004 2nd IEEE International Conference on* (26-26 June 2004). 39-45 p.

Jyh-Jong, Wei, Chang Chuang-Jan, Chou Nai-Kuan et Jan Gwo-Jen. 2001. «ECG data compression using truncated singular value decomposition». *Information Technology in Biomedicine, IEEE Transactions on*, vol. 5, no 4, p. 290-299.

Kalfoglou, Yannis, et Marco Schorlemmer. 2003a. «IF-Map: An ontology-mapping method based on information-flow theory». In *Journal on data semantics I*, p. 98-127: Springer.

———. 2003b. «Ontology mapping: the state of the art». *The knowledge engineering review*, vol. 18, no 1, p. 1-31.

Kietz, Joerg-Uwe, Alexander Maedche et Raphael Volz. 2000. *A method for semi-automatic ontology acquisition from a corporate intranet*: *Workshop "Ontologies and text.*

Kim, H.-J., S.-G. Lee 2000. A semi-supervised document clustering technique for information organization. *Proceedings of the ninth international conference on Information and knowledge management*, ACM.

Kiu, C.C., et C.S. Lee. 2006. «Ontology mapping and merging through OntoDNA for learning object reusability». *JOURNAL OF EDUCATIONAL TECHNOLOGYAND SOCIETY*, vol. 9, no 3, p. 27.

Klein, Michel Christiaan Alexander. 2004. *Change management for distributed ontologies*: p.

Kristensen, T. 2011. *The Dynamic Content Management system*: *Information Technology Based Higher Education and Training (ITHET), 2011 International Conference on* (4-6 Aug. 2011). 1-8 p.

Laramee, R.S. (2011). Bob's Concise Introduction to Doxygen, Technical report, The Visual and Interactive Computing Group, Computer Science Department, Swansea University, Wales, UK, 2007.(available online)

Le Capitaine, Hoel. 2009. «Opérateurs d'agrégation pour la mesure de similarité. Application à l'ambiguïté en reconnaissance de formes». Université de La Rochelle.

Lei, Wang, Shen Chunhua et R. Hartley. 2011. *On the Optimality of Sequential Forward Feature Selection Using Class Separability Measure*: *Digital Image Computing Techniques and Applications (DICTA), 2011 International Conference on* (6-8 Dec. 2011). 203-208 p.

Li, Sheng, Heping Hu et Xian Hu. 2006. *An ontology mapping method based on tree structure: Semantics, Knowledge and Grid, 2006. SKG'06. Second International Conference on*. IEEE, 87-87 p.

Liu, A. X., Shen Ke et E. Torng. 2011. *Large scale Hamming distance query processing: Data Engineering (ICDE), 2011 IEEE 27th International Conference on* (11-16 April 2011). 553-564 p.

Lozano-Tello, Adolfo, et Asunción Gómez-Pérez. 2004. «Ontometric: A method to choose the appropriate ontology». *Journal of Database Management*, vol. 2, no 15, p. 1-18.

Lu, C.T., M. Shukla, S.H. Subramanya et Y. Wu. 2007. *Performance evaluation of desktop search engines*. IEEE, 110-115 p.

Maedche, A., et S. Staab. 2001. «Ontology learning for the semantic web». *Intelligent Systems, IEEE*, vol. 16, no 2, p. 72-79.

Maedche, Alexander, et Steffen Staab. 2000. *Discovering conceptual relations from text: Ecai*. 27 p.

Missikoff, M., R. Navigli et P. Velardi. 2002. «Integrated approach to web ontology learning and engineering». *Computer*, vol. 35, no 11, p. 60-63.

Mokris, I., et L. Skovajsova. 2008. *Document space dimension reduction by Latent Semantic Analysis and Hebbian neural network: Intelligent Systems and Informatics, 2008. SISY 2008. 6th International Symposium on* (26-27 Sept. 2008). 1-4 p.

Moser, T., K. Schimper, R. Mordinyi et A. Anjomshoaa. 2009. *SAMOA - A Semi-Automated Ontology Alignment Method for Systems Integration in Safety-Critical Environments: Complex, Intelligent and Software Intensive Systems, 2009. CISIS '09. International Conference on* (16-19 March 2009). 724-729 p.

Napoli, Amedeo. 1997. «Une introduction aux logiques de descriptions».

Pereira, F. C. and B. J. Gross 1994. Natural language processing, MIT Press.

Qu, Zhiming. 2009. *Application of Fuzzy Clustering and DM in Information Extraction of Machine Learning: Web Mining and Web-based Application, 2009. WMWA '09. Second Pacific-Asia Conference on* (6-7 June 2009). 3-6 p.

Ribière, M., et N. Matta. 1998. «Virtual enterprise and corporate memory». *INRIA, Sophia Antipolis,(project ACACIA)*.

Rui, Xu, et D. Wunsch, II. 2005. «Survey of clustering algorithms». *Neural Networks, IEEE Transactions on*, vol. 16, no 3, p. 645-678.

Rust, J. (1997). "Using randomization to break the curse of dimensionality." *Econometrica: Journal of the Econometric Society:* 487-516.

Salton, G., A. Wong et C.S. Yang. 1975. «A vector space model for automatic indexing». *Communications of the ACM*, vol. 18, no 11, p. 613-620.

Sebastiani, F. 2002. «Machine learning in automated text categorization». *ACM computing surveys (CSUR)*, vol. 34, no 1, p. 1-47.

Shen, Yanfen. 2007. «A formal ontology for Data Mining: principles, design and evolution: thesis presented at University of Quebec at three rivers. ».

Shibata, N., Y. Kajikawa et I. Sakata. 2010. *How to measure the semantic similarities between scientific papers and patents in order to discover uncommercialized research fronts: A case study of solar cells*: *Technology Management for Global Economic Growth (PICMET), 2010 Proceedings of PICMET '10:* (18-22 July 2010). 1-6 p.

Shihan, Yang, et Wu Jinzhao. 2010. *Mapping Relational Databases into Ontologies through a Graph-based Formal Model*: *Semantics Knowledge and Grid (SKG), 2010 Sixth International Conference on* (1-3 Nov. 2010). 219-226 p.

Shin, K., et al. (2011). Consistency Measures for Feature Selection: A Formal Definition, Relative Sensitivity Comparison, and a Fast Algorithm. *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*.

Somol, P., P. Pudil, J. Novovičová et P. Paclık. 1999. «Adaptive floating search methods in feature selection». *Pattern recognition letters*, vol. 20, no 11, p. 1157-1163.

Stojanovic, L., et al. (2002). User-driven ontology evolution management. *Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web,* Springer: 285-300.

Stojanovic, Nenad, Gregoris Mentzas et Dimitris Apostolou. 2006. *Semantic-Enabled Agile Knowledge-based e-Government*: *AAAI Spring Symposium: Semantic Web Meets eGovernment.* 132-134 p.

Stuhler, E., G. Platsch, M. Weih, J. Kornhuber, T. Kuwert et D. Merhof. 2011. *Multiple discriminant analysis of SPECT data for alzheimer's disease, frontotemporal dementia and asymptomatic controls*: *Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC), 2011 IEEE* (23-29 Oct. 2011). 4398-4401 p.

Su Cheng, Haw, et G. S. V. R. Krishna Rao. 2007. *A Comparative Study and Benchmarking on XML Parsers*: *Advanced Communication Technology, The 9th International Conference on* (12-14 Feb. 2007). 321-325 p.

Sun, Y., C. F. Babbs et E. J. Delp. 2005. *A Comparison of Feature Selection Methods for the Detection of Breast Cancers in Mammograms: Adaptive Sequential Floating Search vs. Genetic Algorithm*: *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the* (01-04 Sept. 2005). 6536-6539 p.

Tapia, J. E., et C. A. Perez. 2013. «Gender Classification Based on Fusion of Different Spatial Scale Features Selected by Mutual Information From Histogram of LBP, Intensity, and Shape». *Information Forensics and Security, IEEE Transactions on*, vol. 8, no 3, p. 488-499.

Berners-Lee, T. Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by Its Inventor, HarperSanFrancisco (1999), ISBN 0-06-251587-X.

Uschold, Mike, et Michael Gruninger. 1996. «Ontologies: Principles, methods and applications». *Knowledge engineering review*, vol. 11, no 2, p. 93-136.

Visser, P. R., et al. (1997). An analysis of ontology mismatches; heterogeneity versus interoperability. *AAAI 1997 Spring Symposium on Ontological Engineering,* Stanford CA., USA.

Wache, Holger, Thomas Voegele, Ubbo Visser, Heiner Stuckenschmidt, Gerhard Schuster, Holger Neumann et Sebastian Hübner. 2001. *Ontology-based integration of information-a survey of existing approaches: IJCAI-01 workshop: ontologies and information sharing.* Citeseer, 108-117 p.

Walsh, J.P., et G.R. Ungson. 1991. «Organizational memory». *Academy of management review*, p. 57-91.

Wei, J.-J., et al. (2001). "ECG data compression using truncated singular value decomposition." *Information Technology in Biomedicine, IEEE Transactions on* 5(4): 290-299.

Weiwei, Zhuang, Ye Yanfang, Chen Yong et Li Tao. 2012. «Ensemble Clustering for Internet Security Applications». *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 42, no 6, p. 1784-1796.

Xiao-Li, Dong, Gu Cheng-Kui et Wang Zheng-Ou. 2006. *Notice of Violation of IEEE Publication Principles
A Local Segmented Dynamic Time Warping Distance Measure Algorithm for Time Series Data Mining: Machine Learning and Cybernetics, 2006 International Conference on* (13-16 Aug. 2006). 1247-1252 p.

Xin-xing, Jing, Zhan Ling, Zhao Hong et Zhou Ping. 2010. *Speaker recognition system using the improved GMM-based clustering algorithm: Intelligent Computing and Integrated Systems (ICISS), 2010 International Conference on* (22-24 Oct. 2010). 482-485 p.

Xing, Sun, Zhou Hua, Liao Hongzhi, Liang Zhihong et Liu Junhui. 2008. *Study on Integration Methods for Project Management System Based on Ontology: Wireless Communications, Networking and Mobile Computing, 2008. WiCOM '08. 4th International Conference on* (12-14 Oct. 2008). 1-6 p. En ligne. <http://ieeexplore.ieee.org/ielx5/4677908/4677909/04679214.pdf?tp=&arnumber=4679214&isnumber=4677909>.

Xingfu, Zhang, et Ren Xiangmin. 2011. *Two Dimensional Principal Component Analysis based Independent Component Analysis for face recognition: Multimedia Technology (ICMT), 2011 International Conference on* (26-28 July 2011). 934-936 p.

Yildiz, Burcu (2006). Ontology evolution and versioning, Technical Report, TU Vienna

Yong, Zhang, Fan Bin et Xiao Long-bin. 2008. *Web Page Classification Based on a Least Square Support Vector Machine with Latent Semantic Analysis: Fuzzy Systems and Knowledge Discovery, 2008. FSKD '08. Fifth International Conference on* (18-20 Oct. 2008). 528-532 p.

Zaman, A. N. K., P. Matsakis et C. Brown. 2011. *Evaluation of stop word lists in text retrieval using Latent Semantic Indexing: Digital Information Management (ICDIM), 2011 Sixth International Conference on* (26-28 Sept. 2011). 133-136 p.

Zambonelli, Franco, Nicholas R Jennings et Michael Wooldridge. 2003. «Developing multiagent systems: The Gaia methodology». *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 12, no 3, p. 317-370.

Zargayouna, H., et S. Salotti. 2004. «Mesure de similarité sémantique pour l'indexation de documents semi-structurés». *12ème Atelier de Raisonnement à Partir de Cas*, vol. 189.

Zhang, B. S. N. Srihari 2003. Properties of binary vector dissimilarity measures. Proc. JCIS Int'l Conf. Computer Vision, Pattern Recognition, and Image Processing.

Zhanli, Sun, Huang De-Shuang, Cheung Yiu-ming, Liu Jiming et Huang Guang-Bin. 2005. «Using FCMC, FVS, and PCA techniques for feature extraction of multispectral images». *Geoscience and Remote Sensing Letters, IEEE*, vol. 2, no 2, p. 108-112.

Zhao, J., G.Y. Wang, Z.F. Wu, H. Tang et H. Li. 2002. *The study on technologies for feature selection.* IEEE, 689-693 vol. 682 p.

Zhiwei, Lin, Wang Hui et S. McClean. 2012. «A Multidimensional Sequence Approach to Measuring Tree Similarity». *Knowledge and Data Engineering, IEEE Transactions on*, vol. 24, no 2, p. 197-208.

Zhou, Jiehan, et Rose Dieng-Kuntz. 2004. *Manufacturing ontology analysis and design: towards excellent manufacturing: Industrial Informatics, 2004. INDIN'04. 2004 2nd IEEE International Conference on.* IEEE, 39-45 p.

Zighed, D. A. 2013. Comparison of Proximity Measures: A Topological Approach. Advances in Knowledge Discovery and Management, Springer: 43-58p.

Zongker, D., et A. Jain. 1996. *Algorithms for feature selection: An evaluation.* IEEE, 18-22 vol. 12 p.