République Algérienne Démocratique et Populaire
Université Abou Bakr Belkaid
Faculté des sciences
Département d'informatique

Mémoire de magister en « informatique »
Option « Intelligence Artificiel et Aide à la Décision »
Intitulé :

Construction semi-automatique d'ontologies à partir de textes arabes

Présenté par :

BENAISSA Bedr-Eddine

Devant le Jury :

M^r BESSAID Abdelhafid	Professeur, université de Tlemcen	Président
M^r CHIKH Mohamed Amine	Maitre de Conférences A, université de Tlemcen	Encadreur
M^r RETERI Sidi Mohamed	Professeur, université de Tlemcen	Examinateur
M^me. DIDI Fedoua	Maitre de Conférences A, université de Tlemcen	Examinatrice
M^r ABDERRAHIM Med. El Amine	Maitre de Conférences B, université de Tlemcen	Invité

Année Universitaire
2011/2012

Je dédie ce mémoire

A mes parents pour leur amour inestimable, leur confiance, leur soutien, leurs sacrifices et toutes les valeurs qu'ils ont su m'enseigner.

A la mère de mes enfants pour son soutien moral, et ses encouragements.

A mes beaux parents qu'ils trouvent ici le témoignage de mon respect.

A mes enfants, mes frères et mes sœurs, j'espère que la vie leurs réserve le meilleur.

A un ami qui m'est cher Amine Zouaoui et à sa famille.

A toute ma famille ainsi qu'à mes amis.

Remerciements

Je remercie, tout d'abord, Dieu tout puissant de m'avoir donné le courage, la force et la patience d'achever ce modeste travail.

Je tiens tout d'abord à exprimer ma profonde gratitude à Monsieur **Chikh Amine**, mon encadreur et Monsieur **Abderrahim Amine** mon co-encadreur, Professeurs à la Faculté des Sciences de l'Ingénieur de l'Université de Tlemcen, de m'avoir encadré avec un intérêt constant et une grande compétence ainsi pour l'intérêt qu'ils ont bien voulu porter à mon travail.

J'exprime ma profonde reconnaissance à **Zouaoui Amine** professeur à l'université de Sidi Belabbes pour sa disponibilité, son soutien, ses conseils, les discussions fructueuses que nous avons eu et pour et les encouragements qui m'ont permis de mener à bien ce travail.

Que Messieurs les jurys, trouvent ici l'expression de mes remerciements les plus sincères d'avoir accepté d'examiner ce mémoire.

J'exprime également mes remerciements à monsieur Benaissa Tedjini, mon père, professeur docteur en linguistique arabe à l'université Tlemcen dont la compagnie en contexte professionnel est réellement enrichissante, et aussi d'avoir évalué les résultats finaux.

Je remercie toutes les personnes qui ont participé de manière directe ou indirecte à la concrétisation de ce travail.

Sommaire

INTRODUCTION GENERALE **1**

PROBLEMATIQUE 2

OBJECTIF 2

MOTIVATION POUR UNE APPROCHE D'ONTOLOGIE LEXICALE 3

PLAN DU MEMOIRE 3

Chapitre 1 : Traitement automatique du langage naturel (TALN)

2. LES DIFFERENTS NIVEAUX D'ANALYSE EN TALN **6**

2.1. L'ANALYSE D'UN SYSTEME TALN 6

2.1.1. Analyse morphologique *7*

2.1.2. Analyse syntaxique *8*

2.1.3 Analyse sémantique *9*

2.1.4 Analyse contextuelle *9*

2.2. LE SENS 10

2.3. LE PROBLEME DU SENS 10

3. COMPREHENSION ET FORMALISMES DE REPRESENTATIONS DIVERSES **12**

3.1. LA COMPREHENSION D'UN TEXTE 13

3.2. LE SENS ET SA REPRESENTATION 13

3.3. LES LOGIQUES 13

3.4. LES GRAPHES CONCEPTUELS 14

3.5. STRUCTURES DE TRAITS (SDT) 15

4. REPRESENTATION DES CONNAISSANCES LINGUISTIQUES **16**

4.1. LES LEXIQUES 16

4.2. LES GRAMMAIRES FORMELLES 17

4.3. LES MOTS CLEFS D'UN TEXTE 18

5. CONNAISSANCES DU MONDE (CM) ET CONNAISSANCES LINGUISTIQUES (CL) **18**

5.1 METHODOLOGIE D'IDENTIFICATION DES CONNAISSANCES ENCODEES DANS LE LEXIQUE 18

5.1.1 Encodage des Connaissances du Monde (CM) *18*

5.1.2 Comment les connaissances sont lexicalisées ? *19*

5.1.3. Dictionnaires et connaissances lexicalisées *20*

a/ Détermination du genre prochain (hyperonymie) 21

b/ Détermination des différences spécifiques 21

5.2 FOUILLE DES CONNAISSANCES DANS LES LIENS LEXICAUX 22

6. LES OUTILS DU TRAITEMENT AUTOMATIQUE DE LA LANGUE (TAL) ARABE 22

6.1. ANALYSEURS MORPHOLOGIQUE 23

6.1.1. L'analyseur morphologique à états finis de Beesley 2001 (Xerox) 24

6.1.2 L'analyseur morphologique de Buckwalter : Aramorph 24

6.1.3 L'analyseur morphologique Sebawi de Darwish 24

6.2. LES PART OF SPEECH TAGGERS: 25

6.3. LE TAGGER APT DE KHOJA 25

7. CONCLUSION 26

Chapitre 2 : Les ontologies

1. INTRODUCTION 27

2. LA NOTION ONTOLOGIE 27

2.1. L'ORIGINE DES ONTOLOGIES 27

2.2. QU'EST CE QU'UNE ONTOLOGIE ? 28

2.3. POURQUOI LES ONTOLOGIES ? 30

2.4. LA REPRESENTATION DES CONNAISSANCES ET LES ONTOLOGIES 31

2.5. LES CONSTITUANTS D'UNE ONTOLOGIE 33

2.5.1. Les connaissances et domaines de connaissance 33

2.5.2. Les concepts et les relations 33

a. Concepts 33

b. Relations 36

2.6. LES FORMALISMES DE REPRESENTATION [GAE02] 38

2.6.1. Les formalismes logiques 38

2.6.2. Les réseaux sémantiques 39

2.6.3. Les schémas (Frame) 41

2.6.4. Les scripts 42

3. CONSTRUCTION D'UNE ONTOLOGIE 42

3.1. LE CYCLE DE VIE DES ONTOLOGIES 43

3.1.1. Evaluation des besoins 46

3.1.2. Conceptualisation [FUR02] 46

3.1.3. Ontologisation 47

3.1.4. Opérationnalisation 48

3.2. L'EVALUATION ET L'EVOLUTION D'UNE ONTOLOGIE 49

3.3. LA FUSION DES ONTOLOGIES 49

3.4. METHODOLOGIE ET OUTILS DE CONSTRUCTION D'ONTOLOGIES 50

4. CLASSIFICATION DES ONTOLOGIES 51

5. CONCLUSION 54

Chapitre 3 : Ontologie WordNet (Modèle de notre axe de recherche)

1. HISTORIQUE ET ORIGINE 55

2. PRESENTATION DE WORDNET 55

3. CONCEPTION & STRUCTURE DE WORDNET 56

3.1. SYNSET 57
3.2. ORGANISATION 58
3.3. LA MATRICE LEXICALE 59

4. LES RELATIONS DANS WORDNET 59

4.1. SYNONYMIE 60
4.2. ANTONYMIE 61
4.3. L'HYPERONYMIE / HYPONYMIE 61
4.4. MERONYMIE 62

5. LES VERBES DANS WORDNET (RESEAU SEMANTIQUE) 63

6. L'HYPONYMIE ENTRE LES VERBES 63

7. POLYSEMIE 64

8. ARABIC WORDNET (AWN) 65

8.1. L'ECRITURE ARABE [BLA06] 65
8.2. DESCRIPTION D'AWN 67
8.3. CONSTRUCTION D'ARABIC WORDNET (AWN) 68
8.4. L'INTERFACE UTILISATEUR 70

9. CONCLUSION 71

Chapitre 4 : Etat de l'art - Apprentissage ontologique (Extraction des connaissances à partir des textes)

INTRODUCTION 72

PARTIE I : COMPARAISON ENTRE DIFFERENTS SYSTEMES & APPROCHES

1. LES SYSTEMES D'APPRENTISSAGES ONTOLOGIQUES 73

2. LES SIX DIMENSIONS DE COMPARAISON 75

2.1. LES ELEMENTS A APPRENDRE 76
2.1.1. Les termes 76
2.1.2. Les concepts 76
2.1.3. Les instances 77
2.1.4. Les relations entre concepts 77

2.1.5. Les axiomes ... *77*
2.1.6. Les Méta-connaissances ... *77*
2.2. LES SOURCES D'APPRENTISSAGES ... 78
2.2.1. Les sources réutilisables (Ontologie de base) ... *78*
2.2.2. Les entrées ... *78*
2.3. LE PRETRAITEMENT ... 79
2.4. LES METHODES D'APPRENTISSAGES ... 80
2.4.1. Approches d'apprentissage ... *80*
a. L'approche statistique ... 80
b. L'approche logique ... 81
c. Les approches linguistiques ... 82
d. Les approches basées sur les Patrons (Pattern matching approaches) ... 83
e. Les approches heuristiques ... 84
f. Stratégie d'apprentissage Multiples ... 84
2.4.2. Les tâches d'apprentissage ... *84*
2.4.3. Le degré d'automatisation ... *86*
2.5. LES RESULTATS ... 86
2.6. L'EVALUATION ... 87
PARTI II : APPRENTISSAGE ONTOLOGIQUE ... 88
TECHNIQUES ET APPROCHES ... 88
1. INTRODUCTION ... 88
2. CLASSIFICATION DES SOURCES D'APPRENTISSAGE ... 88
3. UN PROCESSUS D'APPRENTISSAGE CONSENSUEL ... 89
4. METHODES D'EXTRACTION DES TERMES (LEXICAUX) ... 90
4.1. EXTRACTION DES FUTURS CONCEPTS ... 90
4.2. OUTILS D'EXTRACTION ... 91
4.2.1. Méthodes statistiques ... *91*
4.2.2. Méthodes à base de dictionnaires (notre axe de recherche) ... *92*
4.3. EXTRACTION DE RELATION ... 92
4.4. RELATIONS TAXONOMIQUES : ... 92
CONCLUSION ... 94

Chapitre 5 : Approche Adoptée - Conception et Implémentation

1. INTRODUCTION 95

1.1. ONTOLOGIE LEXICALE 95
1.2. OBJECTIF 95

2. CONCEPTION DE L'APPROCHE 96

2.1 HYPOTHESE DE BASE 97
2.2. DICTIONNAIRE, GRAPHE DES SYNONYMES ET CLUSTERING 98
2.2.1. Dictionnaire source 98
2.2.2. Patrons lexico-syntaxiques 101
2.2.3. Sous graphe de synonymies 105
2.2.4. Sous graphe de synonymie et matrice d'adjacence 107
2.2.5. Markov Cluster aLgorithm (MCL) & Clustering 109
2.2.6. Regroupement (clustering) en synsets 110
2.2.7. Détails sur l'algorithme MCL 111

3. IMPLEMENTATION 112

3.1 PLATEFORME DE DEVELOPPEMENT 113
3.2 RESSOURCES UTILISES 113
3.2.1. Outils linguistiques. 113
3.2.2. Ressources lexicales. 113
3.3. ARCHITECTURE GENERALE DE L'APPLICATION 114
3.4 INTERFACES DE RECHERCHES 115
3.4.1. Interfaces de la phase de prétraitement 115
3.4.2. Interfaces de la phase de traitement 117

CONCLUSION GENERALE 126

BIBLIOGRAPHIE 128

Sommaire des figures

FIG. 1 : HIERARCHIE DES NIVEAUX D'ANALYSE DES LANGUES NATURELLES 6

FIG. 2 : ARCHITECTURE GENERALE DU TALN. 7

FIG. 3 : ARBRE SYNTAXIQUE DE L'EXEMPLE PRECEDANT 8

FIG. 4 : CALCUL DU SENS 10

FIG. 5 : DIFFERENTES CATEGORIES DE LA SEMANTIQUE 11

FIG. 6 : GRAPHE CONCEPTUEL DE "يوجد الكراس الأحمر فوق الطاولة" 14

FIG. 7 : GRAPHE CONCEPTUEL DE" يوجد الدفـتر فوق الطاولة " 14

FIG. 8 : EXEMPLE DE GRAPHE CONCEPTUEL RESULTANT DE LA JOINTURE DE DEUX INFORMATIONS. 14

FIG. 9 : EXEMPLE D'UNE SDT 15

FIG. 10: REPRESENTATION D'UNE STRUCTURE DE TRAITS PAR UN ARBRE. 15

FIG. 11 : EXEMPLE DE REPRESENTATION D'UN MOT DANS LE LEXIQUE 16

FIG. 12 : RELATION IS_A - EXTRAITE DE L'ARBRE DE PORPHYRE 21

FIG. 13 : EXTRAIT DU TABLEAU DE TRANSLITERATION ARABE DE BUCKWALTER 24

FIG. 14 : CONCEPTUALISATION D'UNE ONTOLOGIE 29

FIG. 15 : LE TRIANGLE SEMANTIQUE (OGDEN ET RICHARDS, 1923) 34

FIG. 16 : EXEMPLE DE LA RELATION « PARTIE-DE » 37

FIG. 17 : EXEMPLE DE RESEAU SEMANTIQUE UTILISANT LA RELATION " EST-UN " 39

FIG. 18 : EXEMPLE DE RESEAU SEMANTIQUE UTILISANT LA RELATION " SORTE-DE " 39

FIG. 19 : HIERARCHIE DU GRAPHE DE SOWA (TOP-LEVEL) : TREILLIS 40

FIG. 20 : ELEMENTS CARACTERISANT UN SCHEMA [MEL07] 42

FIG. 21 : ETAPES POUR LA CONSTRUCTION DES ONTOLOGIES [MEL07] 43

FIG. 22 : LE CYCLE DE VIE D'UNE ONTOLOGIE [FUR02] 45

FIG. 23 : PROCESSUS DE CONCEPTUALISATION 47

FIG. 24 : CONSTRUCTION D'UNE ONTOLOGIE OPERATIONNELLE. [FUR02] 49

FIG. 25 : TYPE D'ONTOLOGIE SELON GUARINO 52

FIG. 26 : DIFFERENTS TYPES D'ONTOLOGIES SELON LE DEGRE DE FORMALITE 53

FIG. 27 : DIFFERENTS TYPES D'ONTOLOGIES [MIZ 97] 53

FIG. 28 : RESSOURCES DESCENDANCES DE WORDNET 56

FIG. 29 : EXEMPLE DE SOUS HIERARCHIE DANS WORDNET CORRESPONDENT AU CONCEPT "CAR". [BAZ05] 58

FIG. 30 : PRINCIPALES RELATIONS SEMANTIQUES DANS WORDNET. [BAZ05] 60

FIG. 31 : REPRESENTATION DES VOYELLES ARABES 65

FIG. 32 : MAPPING DE SUMO VERS WORDNET(S) 67

FIG. 33 : LA TAXONOMIE DES SIX DIMENSIONS DE COMPARAISON 75

FIG. 34. CLASSIFICATION DE *MAEDCHE* : SOURCES D'APPRENTISSAGES 89

FIG. 35 : HIERARCHIE VISEE PAR NOTRE APPROCHE D'ONTOLOGIE LEXICALE 95

FIG. 36. STRUCTURE DU DICTIONNAIRE DE L'APPROCHE « EL-GHANNYE » 99

FIG. 37. SOUS GRAPHE DE G_D : RELATION ENTRE UN VERBE ENTREE ET SES VERBES DEFINISSANTS 100

FIG. 38 : GRAPHE G_S DU VERBE « صَفْصَفَ » 107

FIG. 39 : ARCHITECTURE DE LA SOLUTION 114

FIG. 40 : BASES DE DONNEES ET TABLES UTILISEES 122

FIG. 41 : INTERFACE DE PRETRAITEMENT : ETAPE 1 116

FIG. 42 : INTERFACE DE PRETRAITEMENT : ETAPE 2 ..... ERREUR ! SIGNET NON DEFINI.

FIG. 43 : INTERFACE DE TRAITEMENT : ETAPE 1 117

FIG. 44 : INTERFACE DE TRAITEMENT : ETAPE 2 118

FIG. 45 : GDL DU GRAPHE DE SYNONYMIE D'UNE PARTIE DE L'ENTREE VERBALE « صَرَّدَ » 119

Sommaire des tableaux

TABLEAU 2 : QUELQUES RELATIONS DANS WORDNET 60

TABLEAU 3. STATISTIQUE SUR WORDNET (JUILLET 2008) 63

TABLEAU 4 : VOYELLES DIACRITIQUES POSSIBLES SUR « بر » ET SUR « علـم » 66

TABLEAU 6 : SYSTEMES PROPOSES ET SELECTION DU CADRE DE L'ETUDE DE COMPARAISON 74

TABLEAU 7 : DEFINISSANTS DU VERBE « صَفَقَ » 102

TABLEAU 8 : PATRONS MORPHOLOGIQUES 103

TABLEAU 9 : DEFINISSANTS DU VERBE « صَفَحَ » 103

TABLEAU 10 : DEFINISSANTS DU VERBE « صَهَرَ » 104

TABLEAU 11 : DEFINISSANTS DES VERBES « هَادَ » ET « هَابَ » 104

TABLEAU 12 : DEFINISSANTS DU VERBE « هَاشَ » 105

TABLEAU 13 : DEFINISSANTS DU VERBE « صَفْصَفَ » 106

TABLEAU 14 : DEFINISSANTS DU VERBE « زَقْزَقَ » 106

TABLEAU 15 : DEFINISSANTS DU VERBE « أَطْعَمَ » 106

TABLEAU 16 : DISTRIBUTION DES ENTREE VERBALES DANS LE DICTIONNAIRE «AL GHANNYE » 115

TABLEAU 17 : EVALUATION VERBE « حَلَّل » AVEC r = 1.6 121

TABLEAU 18 : EVALUATION VERBE « اِمْتَحَنَ » AVEC r = 1.6 122

TABLEAU 19 : EVALUATION VERBE « أَفَقَ» AVEC r =1.6 122

Introduction générale

L'informatique : traitement *automatique* de l'information, est devenue aujourd'hui une science attrayante pour la plupart des personnes, on parle aussi de la science de l'information car elle est au service des exigences des hommes. Elle a pris une place considérable, dans la vie humaine, en s'ingérant dans tous les domaines : l'enseignement, médicale, économie etc.

L'information est un ensemble de données et de faits. Cette collection de données est constituée de tout ce qui perçu par les cinq sens (l'œil, l'odorat, le toucher, l'ouïe, le goût) de l'être humain. Ce dernier, et devant ce gigantesque tas d'informations auquel il doit faire face, son seul moyen de l'exploiter, est bien sûr, *le langage* qu'il soit écrit ou parlé. Le langage a une capacité unique de s'adapter, de tout représenter même le langage lui-même. On dit que le langage est son propre **métalangage**, un langage qui sert à décrire une langue. Par exemple :
Dans la phrase :

- «في الكراس »
- في : "حرف جر"، الكراس: " اسم مجرور "

On utilise un métalangage, puisqu'on utilise la langue pour parler de la langue.

Beaucoup de chercheurs essayent sans relâche de cerner l'étude le langage naturel, mais les problèmes issue de son traitement sont assez complexes du fait que les modèles de représentation classiques (logique, règles de production, réseaux sémantiques,…) se sont révélés insuffisants pour modéliser correctement les concepts linguistes.

N'oublions pas aussi que le langage naturel est bâti autour d'une grammaire, donc un ensemble de règles qu'il est possible de transgresser pour engendrer de nouveaux sens ou de nouveaux effets. Bien que, beaucoup de recherches tentent, de découvrir s'il n'existe pas une certaine systématicité dans cette manière de transgression, et ainsi, de mettre en évidence de nouvelles règles qui régissent le langage naturel.

En voyant l'accroissement flambant des documents numériques et textuels les chercheurs essayaient de trouver des solutions techniques afin de contrôler cette masse d'information. L'apparition des *métadonnées*[1] est le fruit de leurs études. Ce nouveau principe

1- Métadonnées : En informatique, se sont des données qui nous renseignent sur des objets numériques (textes, images, fichiers son, vidéo). Elles sont utiles au repérage et restitution de ces documents enfouis dans un ensemble d'information.

de représentation de données est considéré comme la clé des nouveaux systèmes informations et des services qui en résultent. Citons par exemple le datamining (fouille de données, extraction de connaissances à partir de données), et workflow (flux d'informations au sein d'une organisation, comme la transmission automatique de documents entre des personnes.)

Cette nouvelle approche, trouve son apogée dans les réseaux internet qui se voit enrichie par les apports techniques dans le domaine des stockages et de recherches documentaires électronique, donnant naissance à des langages structurés comme SGML, XML, RDF. Ce nouveau développement a réussi à donner une valeur ajoutée à la recherche sémantique et par conséquent aux développements des thésaurus et leurs successeurs les ontologies.

Problématique

La construction d'ontologies à partir de textes constitue un sous-domaine à part entière de l'ingénierie des ontologies. Dans le contexte du Web sémantique, ces ontologies servent essentiellement à l'annotation sémantique[1] de ressources et à la structuration de bases de connaissances.

Les travaux dans ce domaine existent déjà pour d'autres langues comme l'anglais ou le français, malheureusement pour l'arabe les choses ne font que commencer. Nous essayons donc de dégager une méthodologie de construction d'ontologie pour la langue arabe.

Cette problématique constitue en effet un nouvel enjeu important aussi bien pour le Traitement Automatique des Langues que pour l'Ingénierie des Connaissances. Les systèmes automatisés de traitement de l'information fonctionnant dans des domaines de connaissances spécialisés ne peuvent être efficaces que s'ils reposent sur des ressources termino-ontologiques, construites pour le domaine et l'application concernés.

Objectif

L'architecture générale d'un système TAL montre qu'il a besoin de deux types de sources de connaissances, l'une organise les connaissances encodées dans la langue (connaissances linguistiques) et l'autre stocke les connaissances générales du monde (extralinguistiques), la première prend la forme d'une ontologie linguistique dans la quelle on

1- Annotation sémantique a pour objectif de formaliser l'interprétation qui peut être faite des textes sous la forme de métadonnées attachées aux textes ou à certains de leurs segments.

trouve les classes de mots structurés en hiérarchie, et la deuxième est une base de connaissance ou ontologie générale tel que Cyc de D. Lenat et R. Guha [BAN03].

L'objectif du présent travail est de présenter différentes participations pratiques et théoriques « de base » à la création d'une ontologie, plus précisément une ontologie lexicale dans une perspective d'utilisation en TAL. Ceci pour inciter différents chercheurs de parts leurs disciplines, à investir leurs travaux autour de cette problématique. L'idée de la construction d'une ontologie[1] en utilisant des textes arabes passe nécessairement par les mêmes étapes de construction d'une ontologie classique.

Motivation pour une approche d'ontologie Lexicale

Avec l'apparition des ontologies et la notion de concepts pour la gestion des connaissances, le besoin d'une indexation sémantique s'est vite ressenti pour gérer les documents selon leurs contenus.

Le développement des applications informatiques actuelles est en plein croissance. L'empressement à parfaire et adapter leurs interfaces de dialogues homme-machine pour une utilisation moins désagréable et moins contraignante s'est fait vite ressentit, d'où l'intrusion inévitable du langage naturel computationnelle (linguistique informatique). Ce denier axe de recherche permet d'établir des théories qui régissent les différentes parties impliquées dans un système de traitement automatique du langage naturel (TAL). Par exemple, la reconnaissance de la parole, la correction orthographique et grammaticale des énonces, la compréhension automatique du langage naturel (CALN), la génération automatique des textes (GAT), la synthèse de la parole, …etc.

Les applications de ces champs de recherches sont de plus en plus en croissance, et elles s'étalent sur des applications les plus classiques de l'intelligence artificielle telle que les systèmes à question et réponse, la traduction automatique et les systèmes d'interrogations de bases de données en langage naturel, aux applications les plus récentes tel que les applications émergentes avec le Web sémantique.

Plan du mémoire

Ce mémoire est composé de cinq chapitre ; après une introduction générale, on expose dans le premier chapitre le domaine de recherche de notre sujet de travail, i.e. le

1 Ontologie : «Petit Larousse», le terme ontologie, qui est obtenue en juxtaposant les racines grecques Ontos (être) et logos (science).

traitement automatique du langage naturel et l'ontologie, on souligne trois aspects : la sémantique, les formalismes de représentations de connaissances linguistiques et les ontologies, aussi on donne un état de l'art de ce paradigme dans les systèmes TAL.

Le deuxième chapitre est consacré entièrement à la description de la structure ontologique. L'accent est mis surtout sur la représentation des connaissances en citant les différents types de formalismes de représentation et par la suite les différentes étapes de construction d'une ontologie. Le chapitre est clôturé par l'exposition d'une classification des ontologies.

Dans le troisième chapitre on a présenté l'ontologie lexicale WordNet, considéré comme model de référence de notre approche, par ses différentes catégories syntaxiques : les noms, les verbes, les adjectifs et les adverbes. On a pris en détail les verbes seulement, et on a exposé le système de synonymie. Les grandes lignes de l'implémentation de WordNet sont présentées en bref, afin de nous servir plus tard dans le chapitre cinq où on détaille notre base lexicale.

Le quatrième chapitre présente un état de l'art des recherches de références dans le domaine des ontologies. Ce dernier est sectionné en deux parties. La première relate un cadre de comparaison des différents systèmes existants et achevé par les différentes méthodes d'apprentissages. Dans la deuxième partie, on présente les différentes techniques d'apprentissages ontologiques en mettant l'accent sur l'extraction des termes (lexicaux).

Le cinquième chapitre est consacré à la présentation de l'approche adoptée pour construire les concepts de l'ontologie en se basant sur la relation de synonymie pour les verbes de la langue arabe. Ainsi nous définissons les principes de base nécessaires à la compréhension de l'approche adoptée et nous passons par la suite à la présentation de l'implémentation réalisée et des résultats obtenus et nous concluons ce chapitre par une évaluation de la méthode la méthode adoptée, sur ses résultats et sur son utilité pour divers domaines.

Nous terminons notre mémoire par une conclusion générale, dans laquelle on expose le jugement final de notre approche et de son efficacité.

Chapitre 1
Traitement automatique du langage naturel
(TALN)

1. Introduction

Un bouleversement considérable s'est apparu dans les années 90 : ordinateurs personnels standardisés, avec des capacités de stockage et de traitement en progression exponentielle, ainsi que l'apparition du Web qui a marqué l'apogée technologique en informatique. Dans tout ce changement est née « l'ingénierie linguistique ». La linguistique appelée aussi sciences du langage, est l'étude scientifique des langues naturelles de l'espèce humaine.

Les textes constituent la masse d'information la plus présente sur le Web (le son et les images sont plus récents). Ainsi toute contribution au classement, au traitement des documents textuels et l'extraction de l'information devient une préoccupation principale. C'est dans cette perspective que l'ingénierie linguistique se met ainsi au service de la "fouille de textes" où on remarque la domination des méthodes statistiques sur les méthodes symboliques.

Pour distinguer la langue humaine, on parle actuellement des "langues naturelles", contrairement aux "langues artificielles" ou "formelles" que sont les langages de programmation informatique ou la logique mathématique.

« On regroupe sous le vocable de traitement automatique du langage naturel (TALN) l'ensemble des recherches et développements visant à modéliser et à reproduire, à l'aide de machines, la capacité humaine à produire et à comprendre des énoncés linguistiques dans des buts de communication » Véronis (2001) ; Tellier (2010) ; Yvon (2010)

Le traitement automatique du langage, récemment à la croisée de la linguistique, de l'informatique et de l'intelligence artificielle, voit ses applications, ses programmes et beaucoup de techniques informatiques, au service du langage humain en vue d'appréhender le sens des données en langage naturel. Une compréhension de haut niveau pour ce raisonnement humain a était longtemps recherchée et considérée comme le but extrême des premiers travaux.

Ce chapitre présente ce que peut être un traitement automatique du langage naturel TALN, son architecture, ses niveaux d'analyse du langage traité et ses différents formalismes de représentation de connaissances et du sens sont exposés. Un aperçu d'horizon sur les différents systèmes ou outils TALN, développé pour la langue arabe sera traité à la fin de ce chapitre.

2. Les différents niveaux d'analyse en TALN

2.1. L'analyse d'un système TALN

A ce niveau, deux études formelles ont été menées. L'une peu ancienne, au niveau de la morphologie et de la syntaxe, et l'autre beaucoup plus récente au niveau de la sémantique et de la pragmatique linguistique. A noter qu'on confond souvent la *sémantique lexicale*, qui explique le sens d'unités individuelles, et la *sémantique propositionnelle* qui étudie le sens d'énoncés dans son ensemble et à qui on peut lui donner une valeur de vérité.

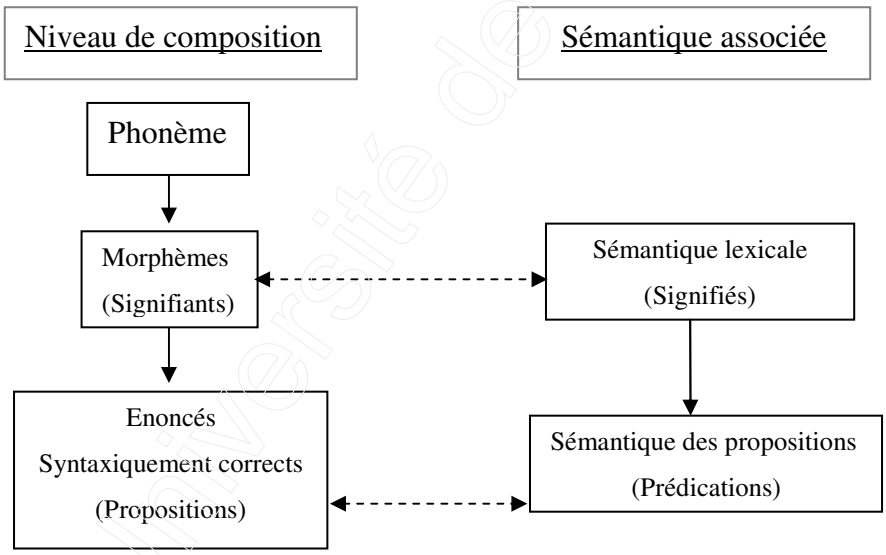


Fig. 1 : Hiérarchie des niveaux d'analyse des langues naturelles

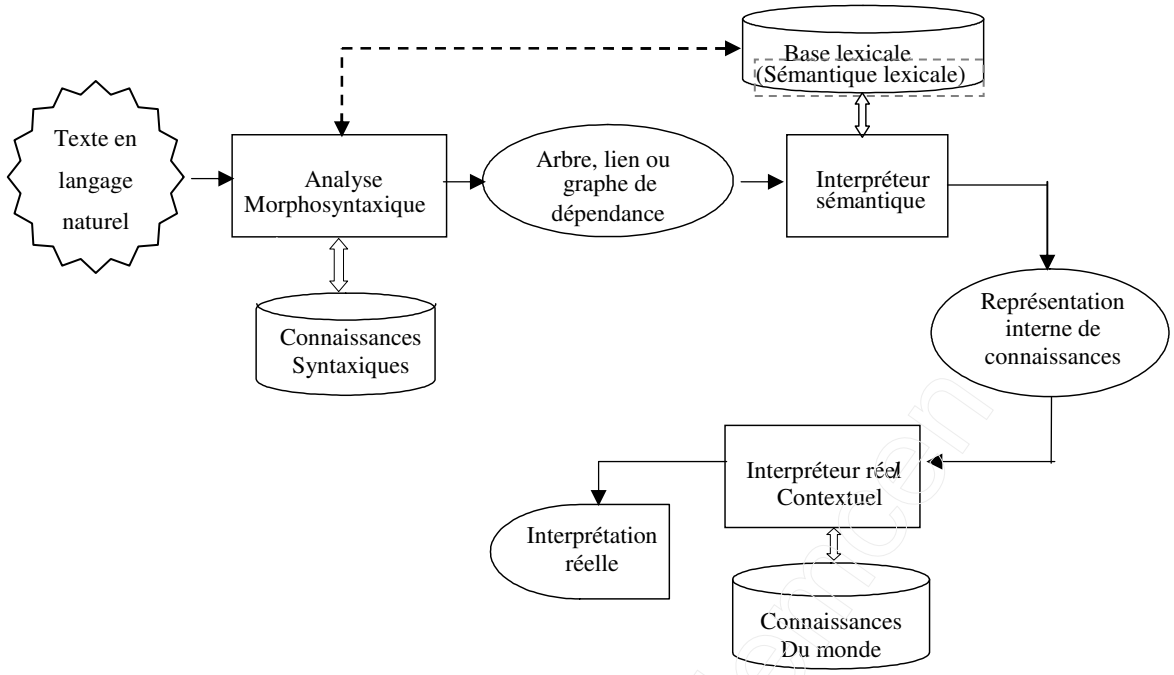


Fig. 2 : Architecture générale du TALN.

2.1.1. Analyse morphologique

La morphologie : interprète comment les mots sont structurés et quels sont leurs rôles dans la phrase. Cette analyse consiste à une segmentation du texte en unités élémentaires auxquelles sont attachées des connaissances dans le système : une fois cette segmentation effectuée, ce n'est plus le texte qui est manipulé, mais une liste ordonnée d'unités. Pour le traitement d'un texte numérique : on part d'une chaîne de caractères typographiques, et on essaie de la segmenter de manière à ce que chaque partie corresponde à une unité classée dans le système.

Exemple : soit la chaîne de caractères « **يأكل عـــمر التفاحة .** »

La segmentation se fera de la manière suivante :

U1 = يأكل

U2 = عمر

U3 = التفاحة

Maintenant, on pourra associer toutes sortes d'informations aux U_i (i = 1, 2, 3, ...), comme par exemple : U2 = عمر

Informations morpho-syntaxiques : nom propre, masculin, singulier.

Informations sémantiques : animé, humain, prénom ...

U1 = **يأكل**

Forme lemmatisée : **أكل**

Informations morpho-syntaxiques : verbe (فعل) , passé (ماضي), indicatif , 3ème personne, singulier, constructions : transitif, ...

Idem pour U3…

Remarque : il y a des phénomènes (concernant le choix et le statut des unités) qui sont répertoriés de longue date par les linguistes : qui conduisent à s'interroger sur la notion de mot : élisions[1], amalgames, flexions, dérivations, compositions, ...

2.1.2. Analyse syntaxique

C'est une partie de la grammaire qui traite la manière dont les mots peuvent se combiner pour former des propositions et de l'enchaînement des propositions entre elles. Cela consiste à associer, à la chaîne découpée en unités, une représentation des groupements structurels entre ces unités ainsi que des relations fonctionnelles qui unissent les groupes d'unités (voir Fig.3).

Reprenons l'exemple précédant : « يأكل عمر التفاحة . », et sa représentation morphologique:

U1 = يأكل U2 = عمر U3 = التفاحة

Le résultat de l'analyse syntaxique pourra être par exemple l'arbre suivant :

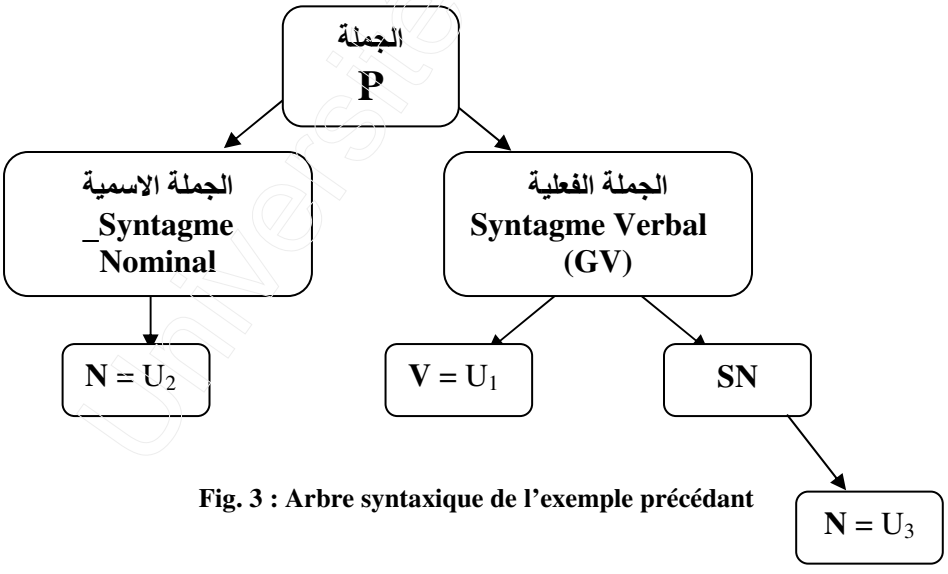


Fig. 3 : Arbre syntaxique de l'exemple précédant

1 - **élision** : nom féminin singulier (grammaire) suppression de la voyelle finale devant un mot commençant par une voyelle ou un 'h' muet, **...en arabe :**

- ترخيم : حذف الحرف الأخير أو أكثر بعد أداة النداء مثال : فاطم في فاطمة.

- حذف : إسقاط بعض أجزاء الكلمة أو الجملة أو التفعيلة لعلة.

- إدغام : إدخال الحرف في الآخر مثال : ("وَمَن يَّعْمَلْ مثقال ذرة")

P = « يأكل عمر التفاحة »

SN = عمر

SV = يأكل التفاحة

SN = التفاحة

N = عمر

V = يأكل

N = التفاحة

2.1.3 Analyse sémantique

Le niveau sémantique est encore beaucoup plus complexe à décrire et à formaliser que les niveaux précédemment énoncés. De ce fait, peu d'outils de traitement reste opérationnel ou du moins, concernent des applications très réduites où l'analyse sémantique se limite à un domaine parfaitement étroit ; par contre, il reste beaucoup à apprendre sur la manière de construire en grandeur réelle des analyseurs sémantiques généraux qui couvriraient la totalité de la langue arabe et seraient indépendants d'un domaine d'application particulier.

La phrase est l'unité d'analyse principale que prend en charge le traitement sémantique afin de représenter sa partie significative. Ces phrases, dont l'analyseur sémantique doit décrire le sens, se composent d'un certain nombre de mots identifiés par l'analyse morphologique, et regroupés en structures par l'analyse syntaxique. Ces mots et ces structures constituent autant d'indices pour le calcul du sens : *on pourrait dire, que le sens résulte de la double-donnée du sens des mots et du sens des relations entre ces mots.*

2.1.4 Analyse contextuelle

La phrase traitée hors contexte, c'est-à-dire isolé de son texte, n'a peut être pas le même sens que dans son contexte. L'analyse sémantique de la phrase isolée, nous amène à représenter la partie de la signification des mots dans cette phrase, elle n'épuise donc pas ce que l'on peut appeler la signification complète d'un texte, à savoir les relations existantes entre les phrases du texte telles que l'humain l'appréhende lors d'un processus de compréhension. C'est ainsi qu'intervient l'analyse contextuelle qui consiste à trouver la signification "réelle" des phrases liées aux conditions positionnelles et contextuelles d'utilisation des mots.

2.2. Le sens

Le sens est partout dans le traitement automatique des langues : il faudrait parler des aspects :

→ Lexicaux (quels liens existent entre les mots et leurs sens ?),

→ Syntaxiques (quel sens est porté par les structures dans lesquelles ces mots interviennent?),

→ Sémantiques bien sûr (comment sont représentées, obtenues et traitées des significations ?)

→ Contextuelles (quelles sont les influences des connaissances sur le monde et la situation pour déterminer le sens ?)…

2.3. Le problème du sens

Qu'est ce que le mot « sens » ? Tout le monde répondra à première vue que c'est « approfondir un peu », c'est-à-dire aller plus loin que "le sens d'un terme, que veut-il évoquer ?". Plusieurs interprétations du sens du mot "sens" peuvent exister. Toutes ces définitions dévoilent le flou qui couvre ce domaine, mais permettent aussi de souligner une différence entre le sens fondamental et le sens interprété, lié également à la prise en considération ou non du contexte [JPM-00]. En effet, une grande partie des travaux en intelligence artificielle et surtout en traitement automatique des langues suppose (implicitement ou non) la possibilité de calculer un sens littéral (qui relève de ce qui est alors appelé sémantique), puis de l'interpréter selon les connaissances générales sur le monde de référence, le contexte et les caractéristiques des interlocuteurs (on parle alors de contextuel).

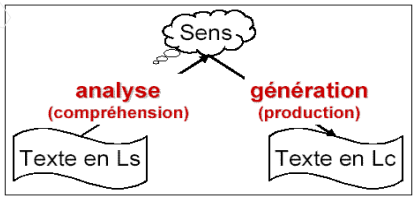


Fig. 4 : Calcul du sens

Bien entendu, cela pose la question de l'existence d'un niveau linguistique indépendant, que certains remettent en cause en arguant de l'impossibilité de séparer l'interpréteur de la chose interprétée. D'autres contestent l'existence des acceptations énumérées dans les dictionnaires pour défendre le sens littéral… Cette hypothèse est si

commode pour les traitements automatiques qu'elle est à peu près systématique même si sa validité psychologique reste incertaine. Mais, même ici, on trouvera un certain flou dans les catégories possibles ; ainsi peut-on distinguer (sans qu'il s'agisse le moins du monde d'une partition), voir figure 5 :

Fig. 5 : Différentes catégories de la sémantique

— La sémantique veri-conditionnelle précise les conditions de vérité de l'expression traitée (on parle aussi parfois de référence virtuelle…).
— La sémantique intentionnelle voit une expression comme l'ensemble des propriétés théoriques que possèdent les concepts correspondants.
— La sémantique extensionnelle décrit une expression comme l'ensemble des objets ou situations du monde que cette expression désigne (on parle aussi de sémantique dénotationnelle ou référentielle).
— La sémantique componentielle cherche à décomposer les mots en éléments de sens plus primitifs, puis étudie leurs possibilités de combinaison.
— La sémantique procédurale décrit le sens d'une expression comme l'ensemble des actions à effectuer pour trouver l'objet désigné.
— La sémantique argumentative (liée aux notions de supposition et de présupposition) dépasse la description d'actes de langage isolés pour étudier leurs enchaînements dans le discours et les connecteurs correspondants.

Néanmoins, une question retient l'attention des chercheurs : « est-il possible de décomposer la notion de sens ? ». Plusieurs points de vue opposés ont résulté du débat de cette question. Par exemple, W. Chafe défend l'idée que le sens est unitaire et ne peut se

décomposer. D'autres soutiennent l'idée que le sens global peut se décomposer en divers éléments, étudiés séparément. Leech, quand à lui, distingue sept formes fondamentales liées au sens (en considérant ou non le contexte et en faisant la distinction entre le sens intentionnel et le sens interprété) : conceptuel, connotatif, stylistique, affectif, réfléchi, collocatif, thématique dont il a apporté des éléments partiels.

3. Compréhension et formalismes de représentations diverses

La compréhension littérale d'un texte nécessite divers types de connaissances (modèle de la langue, modèle de la tâche, éventuellement état de la tâche, historique du dialogue et modèle utilisateur).

L'utilité de construire un module de compréhension nous donne l'avantage d'en extraire ce que nous appelons le « *sens utile* » d'un texte (informations nécessaires pour l'application). Si on situe la compréhension par rapport à un système de commandes, le *sens utile* permet de construire sa commande.

La représentation sémantique peut être vue comme la fonction de transformation d'une représentation primaire vers une autre représentation interprétable par le contrôleur de dialogue d'un système interactif.

Dans la littérature informatique, une multitude de formalismes de représentations sémantiques est proposée pour la représentation interne d'une phrase, afin d'en révéler le sens. Nous pouvons entre autre citer :

Les logiques (la logique des propositions, la logique des prédicats ou la logique modale), par exemple, le démonstrateur AGS (Audiotel Guide des Services) du CNET utilise la logique du premier ordre pour représenter le sens d'un énoncé.

Les graphes conceptuels (Sowa), appelés aussi réseaux sémantiques ou graphes de Sowa, ont été développés par Sowa. Les logiques et les graphes de Sowa sont surtout utilisés dans le domaine de l'ingénierie des connaissances linguistiques.

Les structures de traits et les ensembles d'attributs sont très courants dans les interfaces homme-machine. Le choix d'un formalisme dépend de ses propriétés et caractéristiques selon l'objectif recherché. Notons que pour des serveurs dialoguant avec des bases de données, la représentation sémantique doit permettre de générer une requête de type SQL (Structured Query Language) pour interroger la base de données. De ce fait, les types de représentations sémantiques les mieux adaptés sont les structures de traits et les ensembles d'attributs. [BOU-02]

3.1. La compréhension d'un texte

Informatiquement parlant, comprendre un texte ou un énoncé, implique sa transformation en une structure de données exploitable par la machine. C'est cette structure que nous appelons le sens du texte. Mais pour pouvoir faire cette transformation, le module de compréhension (voir figure 4) utilise de nombreuses connaissances linguistiques (lexique[1], grammaire, etc.)

A la suite de cette partie, on propose plusieurs formalismes de représentations du sens d'un texte ainsi que les connaissances utiles au processus de compréhension. Puis nous étudions quelques stratégies de compréhension typiques des systèmes de dialogue.

3.2. Le sens et sa représentation

Nous allons nous intéresser, dans cette section, aux principaux formalismes permettant la représentation interne d'une phrase, afin d'en dégager le sens. La représentation du sens d'un texte ou d'un énoncé est donc la structure obtenue en sortie du module de compréhension. Une description de la logique, des graphes de SOWA, des structures de traits et des attributs seront explicités dans cette section. Les deux premiers sont surtout utilisés dans le domaine de l'ingénierie des connaissances linguistiques, les deux autres sont très courants dans les interfaces homme-machine.

Bon nombre de ces formalismes sont presque identique à la logique des prédicats du premier ordre. Le choix d'un formalisme est donc avant tout accrédité à l'expert pour exprimer ces connaissances et aux algorithmes d'interprétation utilisés. On peut aussi associer au sein d'un même système plusieurs formalismes.

3.3. Les logiques

Plusieurs approches logiques ont vu le jour comme la logique des propositions, la logique des prédicats ou la logique modale. Notons qu'aucune logique n'a réussi à représenter une phrase de façon complète. Mais elles peuvent nous satisfaire comme dans le cas particulier des serveurs vocaux interactifs : comme le démonstrateur AGS (Audiotel Guide des Services) du CNET qui utilise d'ailleurs la logique du premier ordre pour représenter le sens d'un texte ou d'un énoncé.

1 - Lexique : Ensemble de mots constituant une langue.

3.4. Les graphes conceptuels

Les graphes conceptuels ou réseaux sémantiques, développés par Sowa, est une représentation graphique composé d'arcs orientés et de deux types de nœuds:

- Les nœuds représentant les entités (concepts) notés par des rectangles.
- Les nœuds représentant les relations notées par des ovales.

Les arcs relient deux nœuds de nature différente. Les entités sont définies par un type et un marqueur. Le marqueur peut désigner un objet en particulier (noté par le signe #, suivi d'un numéro référençant l'objet en question) ou au contraire un générique (noté par le signe *).

Un des intérêts de ce formalisme est que l'on peut très facilement ajouter des connaissances à un graphe : c'est le procédé de jointure de plusieurs graphes.

Exemple : [BOU-02], Représentons les phrases :

—" الكراس الأحمر يوجد فوق الطاولة "

—" الدفـتر يوجد فوق الطاولة "

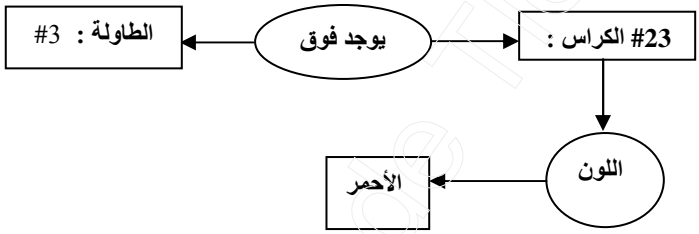


Fig. 6 : Graphe conceptuel de" يوجد الكراس الأحمر فوق الطاولة "

Fig. 7 : Graphe conceptuel de" يوجد الدفـتر فوق الطاولة "

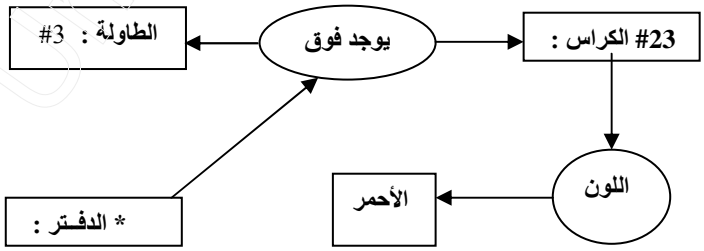


Fig. 8 : Exemple de graphe conceptuel résultant de la jointure de deux informations.

La logique classique peut représenter les graphes conceptuels. Ces dernières permettent de réaliser des inférences, déductions et autres opérations permises par la logique.

3.5. Structures de traits (SDT)

Une structure de traits (SDT) est un ensemble de couples (traits) [attribut = valeur] dont la valeur peut être un entier, un réel, une chaîne ou une autre SDT. C'est donc une structure récursive. Une SDT peut aussi être représentée par un arbre.

Exemple : [BOU-02]

Représentons la phrase :

– أريد قطارا ينطلق من الجزائر يوم الاثنين على الساعة 06 سا ويصل إلى تلمسان على الساعة 13 سا .

Fig. 9 : exemple d'une SDT

Cette SDT est équivalente avec l'arbre suivant :

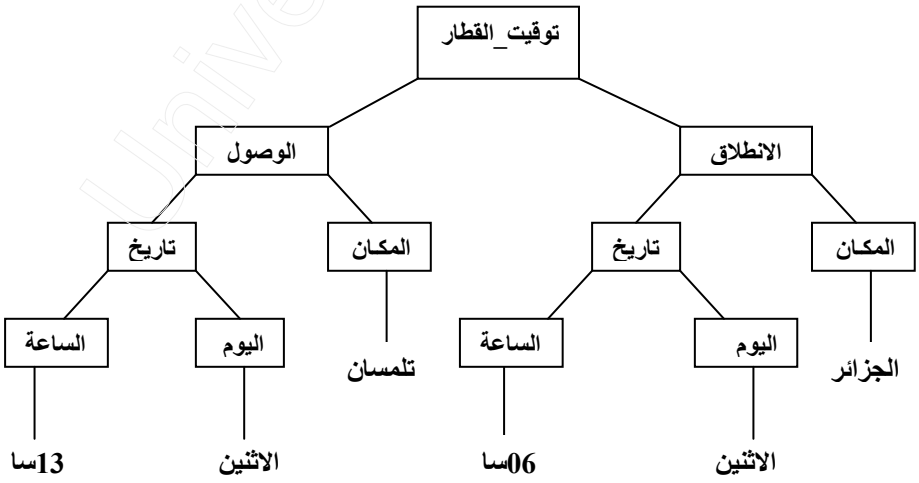


Fig. 10: Représentation d'une structure de traits par un arbre.

Ce formalisme permet de faire des opérations aussi bien que la disjonction ou l'unification (i.e. union de deux SDT). L'analyseur ALPES employé dans le système de dialogue de l'équipe GEOD (CLIPSIMAG) utilise des STD pour représenter le sens des énoncés [BOU-02].

4. Représentation des connaissances linguistiques

En clair, un système de compréhension doit disposer d'un modèle de la langue et un modèle de la tâche (c'est-à-dire un vocabulaire propre à la tâche, comme la syntaxe, la grammaire…), toutes ces connaissances sont enveloppées dans un formalisme afin que ce système puisse les identifier et comprendre un texte. Soulignons deux manières d'aborder l'analyse d'un texte :

- **soit on essaie de le comprendre en se servant des règles de syntaxe et de grammaire.**
- **soit on ne tient pas compte de sa représentation syntaxique mais uniquement des éléments porteurs de sens (appelés aussi concepts).**

4.1. Les lexiques

Un module de compréhension a besoin de connaître les mots (lexies) pour pouvoir analyser une phrase. Ainsi nous pourrons dire que les lexiques sont comme des dictionnaires permettant de décrire un vocabulaire.

Il existe plusieurs types de lexiques : certains ne contiennent que le vocabulaire, d'autres indiquent le genre, nombre et autres particularités du mot. Les mots du lexique peuvent être représentés sous la forme d'une structure de traits.

Exemple : représentons le mot '**السفن**' [BOU-02].

$$\begin{pmatrix} \text{Forme orthographique} = \textbf{السفن} \\ \text{Catégorie lexicale} = \textbf{اسم} \\ \text{Accord} = \begin{pmatrix} \text{genre} = \textbf{مؤنث} \\ \text{nombre} = \textbf{جمع} \end{pmatrix} \end{pmatrix}$$

Fig. 11 : Exemple de représentation d'un mot dans le lexique

Ce type de lexique est utilisé si l'on désire tenir compte des accords (entre l'article et le nom par exemple).

Maintenant, considérons un texte comme une suite de concept, au niveau de l'analyse le lexique peut nous renseigner dans quel(s) concept(s) se trouve le mot, afin d'éviter de donner toutes les caractéristiques du mot (citées dans Fig.11 ci-dessus).

Exemple : supposons que le mot '**السفن**' fasse partie du concept " **التـنقل** " transport [BOU-02].

السفن : وسيلة للتـنقل على البحر

L'entrée du lexique correspondant aux deux mots sera alors de la forme :

السفن : البحر

السفن : تـنقل

Le type et le contenu du lexique dépendent donc énormément de la stratégie de compréhension utilisée et du formalisme choisi pour représenter le sens d'un énoncé.

4.2. Les grammaires formelles

Les grammaires formelles permettent de décrire la syntaxe du langage.

On peut représenter une grammaire par un quadruplet (V_a, V_t, R, S) où

- V_a est le vocabulaire auxiliaire,
- V_t le vocabulaire terminal,
- R l'ensemble des règles
- et S l'axiome.

Il existe plusieurs types de grammaire rangés selon la classification de ***Chomsky***. On peut citer entre autres les grammaires régulières et les grammaires hors contexte. Exemple d'une grammaire régulière (représentable par un automate fini) :

Soient :

V_a = {P =>, SN (syntagme nominal), SV (syntagme verbale), Dét (déterminant), N (nom)},

V_t = {**ال, سفينة, أبحرت**} et S l'axiome.

Les règles sont les suivantes :

S :	P	=>	SN + SV
		SN	→ Dét + N
		SV	→ V
		Dét	→ **ال**
		N	→ **سفينة**
		V	→ **أبحرت**

Cette grammaire permet de former la phrase " **السفينة أبحرت** ".

4.3. Les mots clefs d'un texte

En lisant une phrase, Il suffit donc de comprendre quelques mots « clefs » pour pouvoir en extraire le sens. En général on ne tient pas compte de leur emplacement dans la phrase. L'analyse par mots « clefs » ne prend pas compte ni de la syntaxe de la phrase ni des mots qui ne sont pas considérés comme "clefs". Le principe des mots clefs ne peut être utilisé qu'avec des textes relativement simples. Ce sont les mots nécessaires et suffisants à la compréhension d'un texte.

5. Connaissances du monde (CM) et connaissances linguistiques (CL)

5.1 Méthodologie d'identification des connaissances encodées dans le lexique

Faisons tout d'abord le point sur la problématique d'encodage des CM afin de présenter une méthodologie permettant d'établir des CM encodées dans la langue.

5.1.1 Encodage des Connaissances du Monde (CM)

Notons que la grande partie de nos CM n'est pas encodée dans la langue. Prenons un exemple pour mieux illustrer cette notion. Soit la phrase : « **الثلج أبيض** », nous savons précisément que la neige est blanche. Nous savons aussi par exemple qu'une voiture est plus lourde – en général – qu'une bicyclette, de même si on parle d'une cuisine, on trouve une table et des ustensiles, Ces connaissances ne sont pas encodées dans la langue, n'ont pas leurs équivalents linguistiques. Bien entendu, nous pouvons les exprimer à l'aide de la langue: en formulant des énoncés ou en écrivant des textes. On peut dire que la grande partie de nos CM n'est pas encodée dans la langue. Si par contre, on arrive à trouver cette information écrite dans la langue arabe (dans un dictionnaire, dans une encyclopédie,…) alors on peut dire que c'est une connaissance linguistique (CL). Ainsi dire, une connaissance C est encodée dans une langue L signifie pour nous qu'il existe une parallèle entre X et une connaissance (règle) qui fait partie de L, en tant que système. [Ban-03].

Il est important de savoir que les connaissances lexicales (qui sont un sous ensemble des connaissances linguistiques) sont généralement des connaissances du monde, encodées dans un lexique. L'encodage linguistique de certaines connaissances est conceptuellement vraisemblable : par exemple, tout le monde sait ce que c'est qu'un restaurant et s'attendra, en fait, à ce que la définition de la lexie « **المطعم** » contienne le sens « **أَكْلٌ** », sans pour autant croire qu'on y trouvera formellement la lexie « **أكل** » : cette définition peut aussi contenir une autre lexie (par exemple « **غذاء** ») qui inclut ce même sens.

Il existe aussi des connaissances dont l'encodage linguistique est conceptuellement imprévisible : par exemple, pour parler d'un remerciement dont le degré est élevé, on peut dire des phrases comme [Ban-03] :

— **شكرا جزيلا.**

— **شكرته بحرارة.**

Mais les phrases :

— **شكرته جزيلا.**

— **شكرا بحرارة.**

seraient étranges (ou du moins incorrecte) en arabe.

Deux questions importantes que R. BANGHA a souligné, « pourquoi telle ou telle partie de nos connaissances sont encodées dans la langue et pas d'autres ? » et « Dans quelle mesure ce qu'est le monde actuellement (ou ce qu'il a été auparavant) justifie les choix d'encodage et dans quelle mesure c'est arbitraire ? »

Dans sa thèse, Robert BANGHA (2003) souligne l'idée que c'est parce que la connaissance est écrite dans une langue et qu'elle est importante et plus présente dans nos esprits. Reprenons l'exemple du restaurant « **المطعم** » : il y a des employés de cuisine, cette connaissance est également encodée dans la langue arabe à travers les lexies « **المطعم** » et «**طباخ**» et les liens lexicaux qui les unissent : notamment, « **المطعم** » est un actant[1] de « **طباخ** » – et « **طباخ** » est aussi un actant de « **المطعم** ».

D'un point de vue conceptuel, la situation est similaire par exemple : un guichet d'informations. Précisément, on y trouve un employé qui donne des renseignements, mais pas de lexie en arabe pour désigner cet employé « **المكلف بمكتب الاستعلامات** ». L'encodage lexical d'une connaissance peut fortifier les connaissances du monde à propos de ce à quoi on se réfère.

5.1.2 Comment les connaissances sont lexicalisées ?

Bien entendu, nous soutenons l'idée que la langue encode une grande quantité de nos connaissances du monde mais elle le fait de façon fortuite à partir de nos CM.

Pour commencer, il faut recenser tous les concepts et toutes les entités qui sont lexicalisées dans un certain domaine afin de relever l'ensemble des mots (qui ont un sens), « les lexies » qui marquent ce domaine. Prenons un exemple concret, celui de l'école :

1 En linguistique, le terme d'**actants** désigne les constituants syntaxiques imposés par la valence de certaines classes lexicales (comme le verbe**,** principalement, mais aussi le nom, l'adjectif, la préposition...).

« **المدرسة** ». Lorsqu'on parle d' « **المدرسة** », on peut tout de suite penser à des lexies comme « **طالب** », « **مدير** », « **ناجح** », « **الشهادة** », « **الدرس** », « **القلم** », « **القسم** », « **المرقد** », «**كشف** » [**النقاط**], « **الناظر** », « **المراقب** », « **نقاط** », « **مؤدب** », « **مربي** », « **معلم** », « مدرس », « **تدريس** », « **تلميذ** », « **الدخول** », « **دراسي** », « **غياب** », etc.

Le fait qu'il existe des lexies encodées dans la langue arabe désignant par exemple des personnes qui étudient dans des écoles : « **تلميذ** » ou « **طالب** », on parle ici de connaissances lexicalisées. Une résultante qui semble triviale, d'avoir des mots ayant du sens pour des personnes qui étudient dans une école, mais d'autres optiques[1] ne le voient pas du même angle. Si on compare les deux lexies : « **المدرسة** » et « **المرآب** » ; dans le premier, on peut trouver des écoliers « **التلاميذ** », et dans le second, on peut trouver des voitures « **السيارات** ». Ces deux connaissances sont conceptuellement comparables mais elles ne sont pas encodées de la même façon en arabe.

Le mot « **التلاميذ** » est lexicalisée et désigne des personnes qui étudient à l'école, mais la seconde « **المرآب** » ne l'est pas, car il n'existe pas de mots propres aux voitures garées dans un parking. Ici nous soulignons l'encodage imprévisible des connaissances. Bien sur, il est clair que la définition de la lexie « **المرآب** » doit contenir des éléments relatifs au stationnement des voitures. Cette comparaison nous laisse penser que la présence de la lexie « **التلاميذ** » n'est pas une exigence absolue. [Ban-03]

5.1.3. Dictionnaires et connaissances lexicalisées

Il est clair que pour définir des mots, on a toujours recours à des dictionnaires. Mais souvent les définitions ont plusieurs acceptations qui ne sont pas identiques – même si elles sont souvent semblables par rapport à la référence choisie, par exemple : « **زَيَّنَ** » est synonyme de « جَمَّلَ ». Bien qu'il existe des difficultés concernant la conception des définitions des lexies, Robert BANGHA croit qu'il est possible de surmonter ces difficultés en s'appuyant sur une lexicologie explicative et combinatoire (LEC) suivant des critères précis. Trois étapes qui ont le plus de rapport avec l'encodage des CM :

→ Déterminer le genre prochain,

→ Déterminer les différences spécifiques

→ Prendre en considération la pertinence linguistique.

1 - Optique saussurienne (**Ferdinand de Saussure**, linguiste suisse (1857-1913)), la langue est considérée comme un système relativement autonome et arbitraire – et non pas comme un simple reflet, un simple encodage du monde ou de nos CM

La définition hypéronymiques (genre prochain et différence spécifique) est souvent privilégiée en lexicographie. [Ban-03].

a/ Détermination du genre prochain (hyperonymie)

Le genre prochain ou hyperonymie « *est un concept dont les traits sémantiques sont partagés par les concepts qui lui sont immédiatement subordonnés* ». La relation fondamentale appelée IS_A, souligne l'appartenance d'une connaissance à une autre, et marque clairement cette notion en Intelligence artificielle.

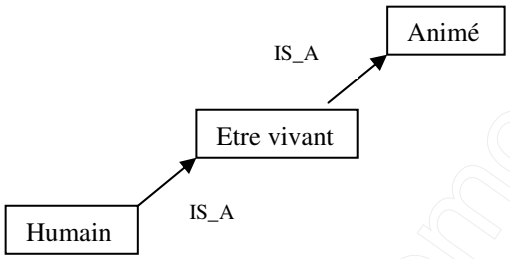


Fig. 12 : relation IS_A - extraite de l'arbre de Porphyre

Ainsi, si on remarque bien les définitions lexicographiques, on se rend compte que cette relation (IS_A) est présente et encode des connaissances similaires. Le genre prochain est la composante centrale de la définition de la lexie : il en est la paraphrase minimale, l'hyperonyme le plus proche. Par exemple, dans un dictionnaire, on trouve la définition d'un restaurant « **المطعم**» :

Is A

المَطْعَم :هو مكان تقدم فيه المأكولات والمشروبات للزبائن

Cette définition nous indique que « **مكان »** est le genre prochain de « **المطعم**». Cela signifie que nos connaissances du monde qui indiquent qu'un restaurant est un endroit sont encodées dans la langue à travers le genre prochain de « **المطعم**». [Ban-03]

b/ Détermination des différences spécifiques

Prenons un petit exemple d'oiseaux : le chardonneret et la huppe et soit les lexies suivantes « **الحسون** » ، « **الهدهد** » :

Premièrement, tous deux sont des oiseaux, de taille moyenne, le premier aux plumages bariolé et l'autre aux plumes roussâtres, etc. Parmi tout ce que nous savons à leur sujet, que faut-il encore inclure dans leur définition ? Malgré toute l'importance du genre prochain, il ne peut pas suffire à lui seul pour définir une lexie. C'est pour cette raison que l'on doit inclure également dans la définition un ensemble de composantes que l'on appelle les différences

spécifiques : elles permettent, d'une part, de faire une distinction sémantique entre les lexies qui ont le même genre prochain, d'autre part, de caractériser la combinatoire sémantique d'une lexie. [Ban-03]

5.2 Fouille des connaissances dans les liens lexicaux

Reprenons l'exemple de la figure 12, on peut dire que le concept «Animé» est l'hyperonyme du concept «Etre vivant», mais aussi on peut associer une relation d'hyponymie, en occurrence le concept « Humain » est hyponyme du concept «Etre vivant ». Remarquons par la suite, que nos connaissances du monde ne nous permettent pas d'envisager la façon dont seront encodées ces connaissances (d'un point de vue purement conceptuel) dans la langue sans connaître vraiment la langue arabe. Illustrons tout ça par un deuxième exemple, le cas de ceux qui empruntent des livres d'une bibliothèque [Ban-03]:

— القارئ للمكتبة

— المستخدم للمكتبة

Et non pas (ça serait étrange):

— المستأجر للمكتبة

6. Les outils du traitement automatique de la langue (TAL) Arabe

Le TAL est un domaine de savoir et de méthodes élaborées autour de diverses préoccupations. Beaucoup de concepts et de techniques régissent de son étude et il se trouve à l'intersection de multiples disciplines : l'informatique théorique, la logique, la linguistique, l'Intelligence Artificielle, mais aussi les neurosciences, les statistiques, etc. Pour mieux cerner le TAL considérons à juste exemple l'énoncé :

— هَتَفَ الجُمْهُورُ بِدُخُولِ اللاَّعِبِينَ المَلْعَبَ.

Remarquons l'enchainement des opérations qu'il faudra suivre pour réussir la compréhension complète et automatiquement de cette phrase. Il nous faudra :

1. Segmenter cette phrase en unités lexicales (mots) ;

— « هَتَفَ » ، « الجُمْهُور » ، « بِدُخُول » ، « اللاَّعِبِينَ »، « المَلْعَبَ » .

2. Identifier les composants lexicaux, et leurs propriétés : c'est l'étape de traitement lexical ;

— "هتف" : فعل ماضي ، منصوب بالفتحة الظاهرة على آخره ، . . .

— "الجمهور" : فاعل مرفوع بالضمة الظاهرة على آخره ،

— **بـ : حرف جرٍ .**

— **الملعبَ : مفعول به ، منصوب بالفتحة الظاهرة على آخره.**

3. Identifier des constituants (groupe) de plus haut niveau, et les relations (de dominance) qu'ils entretiennent entre eux : c'est l'étape de traitement syntaxique ;

— **دخول : اسم مجرور بالكسرة الظاهرة على آخره ، وهو مضاف**

— **اللاعبين مضاف إليه مجرور بالياء والنون.**

4. Bâtir une représentation du sens de cette phrase, en associant à chaque concept évoqué un objet ou une action dans un monde de référence (réel ou imaginaire) : c'est l'étape de traitement sémantique ;

— Le public a acclamé les joueurs dès leurs entrer dans le stade

— **يُحيِّ الجُمْهُور اللاَعِبِينَ عند دُخُولِهم المَلْعَبَ.**

5. reconnaître enfin la fonction de l'énoncé dans le contexte particulier de la situation dans lequel il a été produit : c'est l'étape de traitement pragmatique ;

— **هَتَفَ الجُمْهُور بشدّةٍ بِدُخُولِ اللاَعِبِينَ الجزائريين المَلْعَبَ يوم 12 جوان 2010.**

L'analyse morphologique, est la première étape d'un traitement linguistique de données textuelles. Le challenge à la réalisation des analyseurs morphologique et d'étiquetage grammatical non ambigu de corpus est devenu d'actualité.

6.1. Analyseurs morphologique

L'analyse morphologique est très développée pour les langues latines. Mais ce n'est pas le cas pour la langue arabe par manque de ressources linguistiques (eg. corpus, lexique de base, segmenteurs de textes en phrases,…). C'est pour cette raison, que la majorité des travaux se sont basés sur l'étiquetage morphologique s'appuyant sur des méthodes d'apprentissage et une légère analyse morphologique, par exemple khoja 2001 ; Diab et al. 2004.

L'analyseur morphologique consiste après segmentation du texte, à étudier la forme d'un mot pris isolement (sans contexte) et à déduire les informations dérivationnelles et inflectionnelles. Ainsi, l'analyseur doit générer pour le mot traité une ou plusieurs solutions morphologique décrites par les informations suivante : les suffixes, les préfixes, le radical, la forme canonique (lemme) ainsi que d'autres informations comme le genre grammatical (féminin, masculin), le nombre (singulier, pluriel) ou le temps (verbe conjugué au présent, au passé parfait,...etc.).

Le premier essai d'analyse de la langue arabe a été proposé par David Cohen dans les débuts des années soixante. (1960).

6.1.1. L'analyseur morphologique à états finis de Beesley 2001 (Xerox)

Beesley a développé un analyseur morphologique arabe utilisant les outils de Xerox de modélisation de langage à état fini. Cet analyseur donne pour chaque mot toutes ces listes de caractéristiques morphologiques possibles. Cet analyseur de Beesley trouve son utilisation comme composante d'aide à l'apprentissage dans un système de traitement de langage naturel, plus large.

6.1.2 L'analyseur morphologique de Buckwalter : Aramorph

Développé par Tim Buckwalter [Buc-04] en langage Perl pour le compte du **Linguistic Data Consortium, Université de Pennsylvanie (LDC)** ***actuellement sous Java. Cet Analyseur baptisé «*** Aramorph » est c***onsidéré comme «*** **la ressource lexicologique la plus respecté de son genre».** Le texte analysé en entrée doit être transformé en code ASCII (translitération) avant traitement et le résultat d'analyse doit retranscrit en arabe (translitération inverse) afin d'être compris.

Transliteration	Arabic Windows	Unicode Value and Unicode Name
'	C1	U+0621 ARABIC LETTER HAMZA
|	C2	U+0622 ARABIC LETTER ALEF WITH MADDA ABOVE
>	C3	U+0623 ARABIC LETTER ALEF WITH HAMZA ABOVE
&	C4	U+0624 ARABIC LETTER WAW WITH HAMZA ABOVE
<	C5	U+0625 ARABIC LETTER ALEF WITH HAMZA BELOW
}	C6	U+0626 ARABIC LETTER YEH WITH HAMZA ABOVE
A	C7	U+0627 ARABIC LETTER ALEF
b	C8	U+0628 ARABIC LETTER BEH

Fig. 13 : Extrait du tableau de translitération arabe de Buckwalter

L'analyseur n'accepte pas du texte en arabe avec de l'alphabet romain dans le même document ; Un problème peut être rencontrée lorsque par exemple, le texte contient des étiquettes de Part of Speech ou marqueurs XML en en alphabet romain.

6.1.3 L'analyseur morphologique Sebawi de Darwish

Développé par Darwish [DAR-02] en une seule journée, Sebawi est un analyseur morphologique de la langue arabe. C'est un analyseur de surface utilisé dans des applications de recherches d'information. Il réalise seulement la recherche des racines

possible d'un mot arabe. Cet analyseur morphologique arrive dans 84% des cas à trouver avec succès la racine.

6.2. Les Part of speech Taggers:

L'étiquetage grammatical (en français) ; consiste à donner une étiquette (tag) à chaque mot analysé du corpus, décrivant sa fonction grammaticale dans une phrase donnée :

- Texte original :

Nous sommes allées à Sidi- BelAbbès faire un stage pédagogique.

Texte étiqueté :

Nous/PRO:PER sommes/VER:pres allées/VER:pper à/PRP/à Sidi- BelAbbès /NAM …

L'étiquetage grammatical peut avoir une ou plusieurs solutions possibles pour un mot analysé. C'est pour cette raison qu'il faut penser à supprimer l'ambigüité, tout en considérant le contexte dans lequel le mot apparait.

6.3. Le tagger APT de Khoja

Arabic Part-of-Speech Tagger (APT) a été développé par Shereen Khoja [KHO-01], [KHO-03]. Cette méthode se base sur une combinaison de techniques statistiques et des techniques à base de règles. Les étiquettes (Tags) du « tagset » (l'ensemble des étiquettes définies) de l'APT sont initialement dérivées des étiquettes du « tagset » du corpus BNC (British National Corpus), mais qui ont étaient modifiées en prenant en compte quelque concepts de la grammaire traditionnelle arabe.

La raison de cette modification est que la langue arabe possède ses propres systèmes syntaxique, sémantique et morphologique qui rendent difficile l'adaptation aux « tagset » des langues indo-européennes. Le « Tagset » contient 131 étiquettes qui vont être assignées aux mots analysées. Un corpus de 50.000 mots issu des articles de presse saoudienne « Al Jazzira » a était utilisé pour l'apprentissage de l'étiqueteur.

7. Conclusion

Le chapitre, ainsi exploré, donne une plate forme pour la construction d'un système de traitement du langage naturel. Ce système nécessite plusieurs ressources lors des différentes phases d'analyses (Dictionnaire, ontologie).

Notons que les outils mis à notre disposition pour le traitement de la langue arabe sont appauvrit par rapport à d'autres langues telles que le français ou l'anglais. Les outils, non disponibles gratuitement à présent, permettent d'aborder les corpus textuel arabes dans une perspective d'extraction des connaissances.

Chapitre 2

Les ontologies

1. Introduction

L'avènement des ontologies a actionné considérablement de nombreux domaines notamment la recherche d'information avec ses variations sémantiques et lexicales, la gestion des connaissances, les systèmes coopératifs, le e-commerce, et bien sur le Web sémantique.

Notons aussi que les ontologies viennent s'ajouter à cet amalgame de recherches et apportent un plus aux systèmes experts et surtout aux systèmes à base de connaissances(SBC), les successeurs des systèmes experts et répondaient aux besoins d'une coopération et à un dialogue entre le système et l'utilisateur humain. Citons quelques systèmes à titre d'exemple : les systèmes d'aide à la décision, système d'enseignement par ordinateur (EAO) et plus particulièrement la recherche d'information sur le web. Les SBC offraient une représentation des connaissances d'un domaine d'une part et des connaissances de raisonnement qui manipulent et utilisent ces connaissances du domaine d'autre part. L'idée de cette séparation modulaire était de construire au mieux et plus rapidement des systèmes à bases de connaissances en réutilisant le plus possible des composants génériques, que ce soit au niveau du raisonnement ou des connaissance du domaine [BAC03].

C'est dans cette perspective que les chercheurs ont proposé de développer ces connaissances sur la spécification d'une *ontologie*. Ce second chapitre, donne un aperçu horizontal sur la notion d'ontologie et ses constituants, ainsi qu'une description des besoins auxquels elle peut répondre et ces différents domaines d'applications. Nous soulignerons aussi une présentation des différentes classifications des ontologies et en fin, une description des langages utilisés pour les manipuler.

2. La notion ontologie

2.1. L'origine des ontologies

L'ontologie, un mot grec de racine, est composée d'onto (le participe présent du verbe *être*) qui est l'étude de l'être en tant qu'être et logos qui signifie *discours*. Son premier

sens trouve son origine en philosophie depuis *ARISTOTE*, où l'ontologie est l'étude des propriétés générales de ce qui existe.

L'apparition du terme « Ontologie » dans l'informatique a vu le jour au début des années 1990, grâce au projet d'ARPA Knowledge Sharing Effort (Effort de partager la connaissance) [GRU91]. Selon son sens philosophique, une ontologie est une explication systématique de l'être [GOM04].

2.2. Qu'est ce qu'une ontologie ?

En 1991, Neches et ses collègues donne une, des premières, définition du terme « ontologie » en citant : *« An ontology defines the basic terms and relations comprising the vocabulary of a topic area as well as the rules for combining terms and relations to define extensions to the vocabulary». A* savoir :

«Une ontologie définit les termes et les relations de base du vocabulaire d'un domaine ainsi que les règles qui indiquent comment combiner les termes et les relations de façon à pouvoir étendre le vocabulaire ».

Cette dernière définition nous dicte une méthode d'élaboration d'une ontologie, le repérage des termes de base et les relations entre les termes, ensuite relever les règles qui régissent de cette relation, et en fin donner des définitions à ces termes et ces relations. Le résultat, est une ontologie incluant non seulement des termes qui y sont explicitement définis, mais aussi des termes qui peuvent être créés par déduction en utilisant des règles [GOM99].

Une deuxième définition, au sens strict, est donnée en juin 1993, par Gruber, et qui est la plus citée en informatique plus précisément en intelligence artificielle (IA) : *« An ontology is an explicit specification of conceptualization.»* à savoir : «*Une ontologie est une spécification explicite d'une conceptualisation* ».

En 1995, Guarino donna une troisième définition de l'ontologie, et clarifia la définition de Gruber en énonçant que : *«Les ontologies sont des spécifications partielles et formelles d'une conceptualisation commune*».

Quand à la notion de « partielles », la conceptualisation n'est jamais formalisée logiquement en sa totalité, du fait d'ambiguïtés ou du fait qu'aucune représentation de leur sémantique n'existe dans le langage de représentation d'ontologies choisi. Le terme « commune » du fait qu'une ontologie partage un savoir consensuel (reconnue par une communauté) [FUR02].

En 1997, Borst la reformula légèrement, « *an ontology is a formal, specification of a shared conceptualization»,* puis Studer et ses collègues reprennent les deux définitions de Gruber et de Borst, en 1998, les fusionnent et énoncent la définition suivante :
« *an ontology is a formal, explicit specification of a shared conceptualization*»[OSC05].

Le terme « conceptualisation » oriente les ontologies sur l'aspect *sémantique* c'est-à-dire le sens des termes et que la littérature philosophique désigne par les *intensions*, par opposition aux *extensions*. La conceptualisation est représentée dans un langage, suivant lequel l'ontologie prendra la forme d'une théorie logique (ensemble de formules logiques) ou d'un réseau sémantique.

- Le terme « spécification explicite » oriente, par contre, les ontologies sur le côté syntaxique. Une spécification explicite signifie que les concepts, ainsi que les contraintes qui s'y rapportent sont explicitement définis.

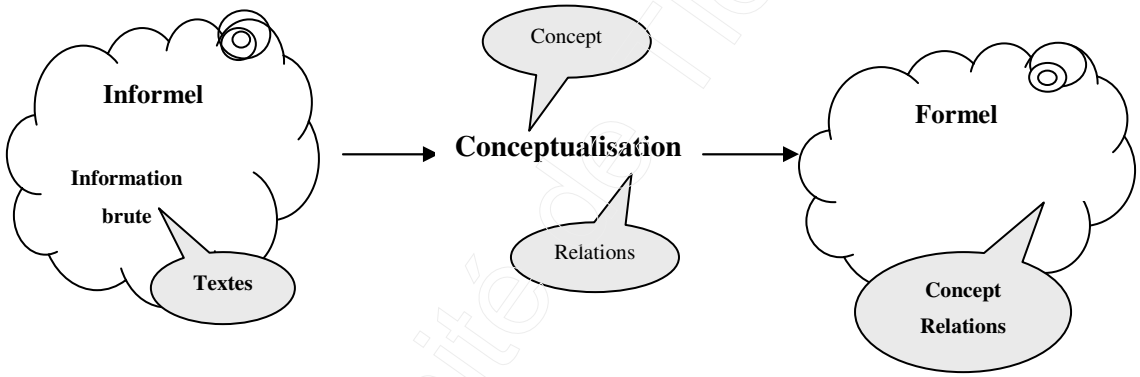


Fig. 14 : Conceptualisation d'une Ontologie

D'autres définitions d'ontologie viennent s'ajouter pour mieux cerner cette notion. On note souvent qu'une ontologie définit un vocabulaire partagé pour aboutir à une compréhension commune d'un domaine donné. Effectivement Dans un sens large, on peut adopter pour la notion d'ontologie la caractérisation suivante [USC98] :

". . . An ontology may take a variety of forms, but will necessarily include a vocabulary of terms and some specification of their meaning (i.e. definitions)".

Cette définition met en évidence les différentes formes que l'ontologie peut prendre. Mais souligne aussi que cette dernière, doit inclure nécessairement un vocabulaire de termes et une spécification de leur signification. L'ontologie inclut des définitions et une indication

de la façon dont les concepts sont reliés entre eux, les liens imposant collectivement une structure sur le domaine et contraignant les interprétations possibles des termes.
Exemples de formes d'ontologies : – glossaires – terminologies – thesaurus – Lexicons.

2.3. Pourquoi les ontologies ?

Et quelles sont les raisons de développer des ontologies ?

Une issue pour offrir des structures d'un espace commun, partagé, nécessaire pour une véritable intégration sémantique des sources d'informations. Une description précise d'un univers du discours et affirmé dans une langue utilisée pour le raisonnement. [CHR09]

En réponse, nous dirons que l'une des motivations principales du développement d'ontologie est de communiquer et partager de la compréhension commune d'un domaine entre des humains ou des systèmes. Plus précisément, c'est une solution pour fournir des bases communes d'un domaine, partagées, nécessaires pour une réelle assimilation sémantique des sources d'information comme une description précise d'un monde de discours.

Ce type d'ontologies est considéré comme étant formel. Les ontologies formelles sont, en général, explicites et manipulables par la machine, ce qui permet de gérer la sémantique et l'inférence à travers un moteur de raisonnement [DIB04]. Le développement des ontologies a d'autres objectifs, citons ici quelques unes que nous considérons les plus importantes [Mer09] :

- Les ontologies offre la possibilité de réutilisation du savoir sur un domaine : cette caractéristique était parmi l'une des causes principales qui ont manœuvré la recherche sur les ontologies ces dernières années.

- Expliciter ce qui est considéré comme implicite sur un domaine ; Les spécifications formels du savoir sur un domaine sont, de surcroît, utiles pour les nouveaux utilisateurs qui doivent apprendre seulement la signification des termes du domaine.

- Distinguer le savoir sur un domaine du savoir opérationnel est une autre finalité des ontologies. La tâche de configuration d'un produit final à partir de ses constituants, peut être décrit en considérant les spécifications essentielles et faire un programme qui réalisera cette configuration indépendamment des produits et de leurs composants [WRI98].

- Possibilité d'analyser le savoir sur un domaine dès que la spécification des termes du domaine est faite. L'analyse formelle des termes est extrêmement précieuse aussi bien pour la réutilisation des ontologies existantes, que pour leur extension [FIK00]. De façon plus générale, pour [Usc96], les avantages à tirer d'une ontologie peuvent être réparties en trois catégories :

  → Communication : une fois les termes fixés, et le vocabulaire et la grammaire bien définie, elles permettent de répondre au besoin de communication entre personnes-personnes, personnes- systèmes et systèmes-systèmes ;

  → Interopérabilité : Une seul seule connaissance commune et partagée, pour la compréhension des concepts, elles permettent de faciliter l'interopérabilité des systèmes et la réutilisation des sources de connaissances ;

  → Amélioration logicielle : au niveau de la spécification (compréhension partagée des problèmes), fiabilité (tests de consistance [semi-automatique]) et de la réutilisabilité (fixe la structure des connaissances).

2.4. La représentation des connaissances et les ontologies

Partant d'une pensée, enrichie par une idée conceptuelle et élaborer dans des laboratoires, les ontologies sont considérées comme des modules logiciels s'insérant dans les systèmes d'information et leur apportant une dimension sémantique qui leur faisait défaut auparavant. [FUR02]

On ne peut parler d'ontologie, sans évoquer la représentation des connaissances, puisque l'ontologie est le fruit de cette représentation au sein des systèmes informatiques. Si nous reprenons la définition de T. Gruber 93, « … une spécification explicite d'une conceptualisation … », nous remarquons que dans le processus de construction d'une ontologie, on doit tout d'abord réaliser la première phase qu'est la conceptualisation. Ce travail nécessite une identification, dans un corpus, les connaissances spécifiques au domaine de connaissances à représenter. « A conceptualisation is an abstract, simplified view of the world that we wish to represent for some purpose » [GRU 93].

Les ontologies sont formelles car ils sont logiquement exprimées, et partielles car une conceptualisation ne peut pas toujours être entièrement formalisée dans un cadre logique, du fait d'ambigüités ou du fait qu'aucune représentation de leur sémantique n'existe dans le langage de représentation d'ontologies choisi. (Revoir la définition de Guarrino 95 plus haut)

J. NOBÉCOURT présente dans [NOB 00] que les formalismes opérationnels présentent une faible tolérance d'interprétation, ce qui oblige à passer directement d'une ontologie informelle à une ontologie totalement formelle et non-ambiguë. De plus, on doit souvent modifier la base de connaissances[1] au cours de son élaboration, ce qui entraine des incohérences et des modifications plus larges. [FUR02]

Ainsi, pour minimiser cette incohérence, il est nécessaire d'élaborer des modèles semi-formels, partiellement cohérents, correspondant à une conceptualisation semi-formelle. C'est ce qu'on appelle l'ontologie conceptuelle semi-formelles, et le processus de spécifications de en question est appelé ontologisation [KAS00].

Une ontologie doit représenter des connaissances formelles et opérationnelles pour son utilisation dans des systèmes. En effet, une ontologie n'est pas opérationnelle, dans la mesure où elle ne fait pas référence à des mécanismes de raisonnement. Le langage cible doit donc permettre de représenter les différents types de connaissances (connaissances terminologiques, faits, règles et contraintes) et de manipuler ces connaissances à travers des mécanismes adaptés à l'objectif opérationnel du système conçu. Ce processus de traduction est appelé opérationnalisation. [FUR02]

FRÉDÉRIC FÜRST dans [FUR02] découpe le processus général de représentation des connaissances en 3 phases :

La conceptualisation : identification des connaissances issue d'un corpus représentatif du domaine. Ce travail doit être mené par un expert du domaine, assisté par un ingénieur de la connaissance ;

L'ontologisation : formalisation, autant que possible, du modèle conceptuel obtenu à l'étape précédente. Ce travail doit être mené par l'ingénieur de la connaissance, assisté de l'expert du domaine ;

L'opérationnalisation : transcription de l'ontologie dans un langage formel et opérationnel de représentation de connaissances. Ce travail doit être mené par l'ingénieur de la connaissance. En fin, on peut définir l'ontologie en Ingénierie des Connaissances comme un ensemble d'objets identifiés et reconnus comme existant dans le domaine.

1 - Une base de connaissances contient les connaissances utilisées dans un Système à Base de Connaissances. Ces connaissances sont formalisées et des mécanismes permettent de gérer la base pour consulter des connaissances ou en ajouter.

2.5. Les constituants d'une ontologie

2.5.1. Les connaissances et domaines de connaissance

En ingénierie des Connaissances, le terme de connaissance a un sens limité : ne sont considérées que les connaissances (au sens large) susceptibles d'être formalisées, c'est-à-dire les connaissances peu ou amplement techniques [BAC00] : « For knowledge-based systems, what exists is exactly that which can be represented » [GRU 93].

On ne peut manipuler automatiquement des connaissances que si leurs sens est largement consensuel. Plus précisément, on peut considérer qu'il n'y a connaissance que si l'information présente dans la machine prend un sens pour l'utilisateur, c'est-à-dire qu'il peut établir un lien entre cette information et celles qu'il possède déjà, et ce sens doit être le même pour tous les utilisateurs [CHA01]. C'est dans cette théorie que naissent les modèles de type réseaux sémantiques qui restent une conséquence de ce lien sémantique.

En conclusion, la construction d'une ontologie doit se faire à partir d'un champ de connaissances bien délimité par un objectif opérationnel clair, et portant sur des connaissances objectives dont la sémantique puisse être exprimée rigoureusement et formellement. Partant de là, plusieurs types d'ontologies peuvent être distingués en fonction des différents objectifs opérationnels recensés. En analysons les définitions de J. Charlet, B. Bachimont et R. Troncy, trois caractéristiques principales, nous permettent de préciser les constituants d'une ontologie en tant qu'objet informatique :

- Les concepts
- Les propriétés
- Les relations

2.5.2. Les concepts et les relations

a. Concepts

Plusieurs définitions ont été données pour ce terme, par exemple : « *Un concept est une représentation générale et abstraite d'une réalité. Le terme concept vient du participe passé latin «conceptus » du verbe « concipere », qui signifie « contenir entièrement», «former en soi »*[1].

"Le *concept* du temps."

"Le *concept* de l'espace."

1 - fr.wikipedia.org/wiki/Concept

Un concept est défini comme étant une notion généralement exprimée par un terme, ou plus généralement par un signe. Il représente un groupe d'objets ou d'entités qui ont en commun un ensemble de caractéristiques et qui nous permettent de les reconnaître comme faisant partie de ce groupe [GAN02].

Dans [REN07], A. Renouf définit plus formellement une ontologie comme étant :

- un ensemble de concepts ;
- un ensemble de relations entre ces concepts ;
- un ensemble d'axiomes (transitivité, réflexivité, symétrie des relations…) ;

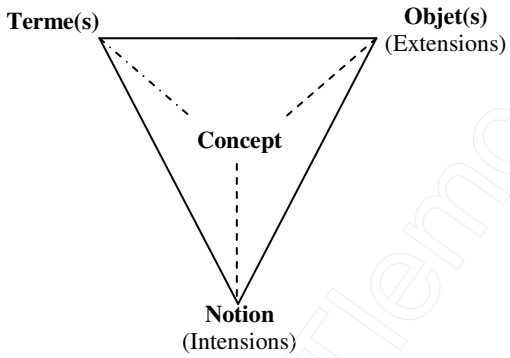


Fig. 15 : Le triangle sémantique (Ogden et Richards, 1923)

Le(s) terme(s) : expriment le concept dans la langue. Le terme en linguistique est un *élément lexical* qui permet d'exprimer le concept en langue naturelle, pouvant admettre des synonymes.

La signification du concept, appelée également « notion » ou « intension » (avec un 's' !) du concept. Notion : également appelée intension du concept, contient *la sémantique du concept*, exprimée en termes de propriétés et attributs, et de contraintes. L'(es) objet(s) dénotés par le concept, appelé(s) également « réalisation » ou « extension » du concept regroupe *les objets manipulés* à travers le concept ; ces objets sont appelés instances du concept. [FUR02]

Par exemple :

Le terme « **طاولة** » renvoie à la fois à la notion de table comme objet de type « **أثاث** » possédant «**لوحة_مسطحة** » et« **أعمدة** », et à l'ensemble des objets de ce type.

Prenons l'exemple du concept ÉTOILE :

– Son terme est le nom commun « النجم ».

– Sa notion est « **نقطة مضيئة في السماء، ليلا** »

– Son extension est justement : « **جميع النقاط المضيئة في السماء، ليلا**» que l'on peut découvrir dans un ciel nocturne … dégagé !

Soulignons qu'on peut trouver des concepts sans extension, ce sont les concepts génériques comme par exemple le concept « **الحقيقة** » qui a le sens de « **كل ما هو صحيح** », cette notion est abstraite. [FAL01] souligne l'existence des concepts partageant la même extension mais pas leur intension et portent le même terme. Ceci correspond à *des points de vue* différents sur un même concept. Mais l'exception ne fait pas la règle, l'exemple du concept de « **طاولة** », montre que cette notion ne peut se définir qu'en utilisant d'autres concepts comme « **أثاث** », « **لوحة_مسطحة** » et « **أعمدة** » et à l'ensemble des objets de ce type. [FUR02]

- La subsomption

La subsomption désigne une relation hiérarchique, entre des concepts.
En logique classique, la subsomption est proche de la relation « est impliqué par », ou encore « contient » en logique ensembliste.
Par exemple le concept « **الإنسان** » subsume le concept « **رجل** ».
En général, un concept C1 *subsume* un concept C2 si toute propriété sémantique de C1 est aussi une propriété sémantique de C2, c'est-à-dire que C2 est plus spécifique que C1. L'extension d'un concept est forcément plus réduite que celle d'un concept qui le subsume. Son intension est par contre plus riche.

- La généricité

Un concept est générique s'il n'admet pas d'extension.
Exemple : « **الحقيقة** » est un concept générique.

- L'identité

En 1994, N. GUARINO propose cette propriété et souligne qu'un concept porte une *identité* si cette propriété permet de conclure quant à l'identité de deux instances de ce concept. Cette propriété peut porter sur des attributs du concept ou sur d'autres concepts.
Exemple : le concept « **تلميذ** » porte une propriété d'identité liée au numéro d'élève, deux élèves étant identiques s'ils ont le même numéro.

- La rigidité

Proposé par N. GUARINO en 1994, un concept est rigide si toute extension de concept en reste extension dans toutes les connaissances du monde possibles.
Exemple 1 : « **الإنسان** » est un concept rigide, « **تلميذ** » est un concept non rigide ;

Exemple 2 (Voir fig.12) : « Etre vivant » est un concept rigide. Mais « Humain » ne l'est pas. Car « Humain » est une instance de « Etre vivant ».

L'anti-rigidité : Un concept est anti-rigide si toute instance de ce concept est essentiellement définie par son appartenance à l'extension d'un autre concept.

Exemple 3 : « **تلميذ** » est un concept anti-rigide car « **تلميذ** » est avant tout « **الإنسان** ».

- L'unité

N. GUARINO (1994), un concept est un concept unité si, pour chacune de ses instances, les différentes parties de l'instance sont liées par une relation qui ne lie pas d'autres instances de concepts. Par exemple : les deux parties d'un couteau « **السكين** », « **المِقْبض** » et « **النَّصْل** » sont liées par une relation « **تَوْصِيل** » qui ne lie que cette « **النَّصْل** » et ce « **المِقْبض** ».

Remarque :

Pour la suite de notre étude, on emploie, tout au long du manuscrit, le mot concept pour désigner l'intension de concept et le mot instance pour désigner un élément de l'ensemble constituant l'extension de concept.

Propriétés :

- L'équivalence : Deux concepts sont équivalents s'ils ont la même extension;
- La disjonction (l'incompatibilité) : Deux concepts sont disjoints si leurs extensions sont disjointes. *Exemple* : « **رجل** » et « **امرأة** » ; « **الليل** » et « **النهار** ».
- La dépendance : Un concept C1 est dépendant d'un concept C2, si pour toute instance de C1, il existe une instance de C2 qui ne soit ni partie ni constituant de l'instance de C2. *Exemple* : « **الأب** » est un concept dépendant de « **الإبْن** » et inversement.

b. Relations

Les concepts peuvent être reliés entre eux à l'aide des propriétés décrites ci-dessus, mais il existe d'autres liens représentés par des relations autonomes [FUR02]. Par exemple, la relation « **يكتب** » lie une instance du concept « **إنسان** » et une instance du concept « **نص** », dans cet ordre.

Les concepts (respectivement instance) peuvent être reliés entre eux par des relations au sein d'une ontologie. Une relation est définie comme une notion de lien entre les entités, souvent exprimé par un terme ou un symbole littéral ou autre. En général, ces liens sont classés en deux catégories : *les liens hiérarchiques* et *les liens sémantiques*.

→ La structure hiérarchique : reprend la structure d'Hyperonymie/Hyponymie. Elle lie un élément supérieur, dit l'hyperonyme, et un élément inférieur, dit l'élément hyponyme, ayant les mêmes propriétés que le premier élément avec au moins une en plus. Dans certains cas, le couple (*Hyperonymie*, *Hyponymie*) s'interprète par (*Type*, *Sous-Type*). L'hyperonyme englobe l'hyponyme. On pourra alors écrire «HYPONYME est une sorte de HYPERONYME ».

Exemple :

« **الأب** » est une sorte « **الإنسان** »

donc,

« **الإنسان** » est l'hyperonyme de « **الأب** ».

→ La relation sémantique : c'est liaison entre les concepts à travers un lien, appelée souvent « Partie-de» ou « Partie_Tout », ce qui correspond à la structuration de HOLONYMIE/MERONYMIE. La relation « partie-Tout » est différente de celle d'hyponymie par le fait qu'un hyperonyme impose ses propriétés à ses hyponymes, par contre le TOUT dispose des propriétés qui ne sont pas obligatoirement transmises à ses parties.

Exemple : Dans le corps humain, « **الرأس** » et « **الساق** » font partie du « **الجسم** » mais elles ne disposent pas des mêmes propriétés. « **الرأس** » n'est pas une sorte de « **الجسم** » (voir la Figure 16).

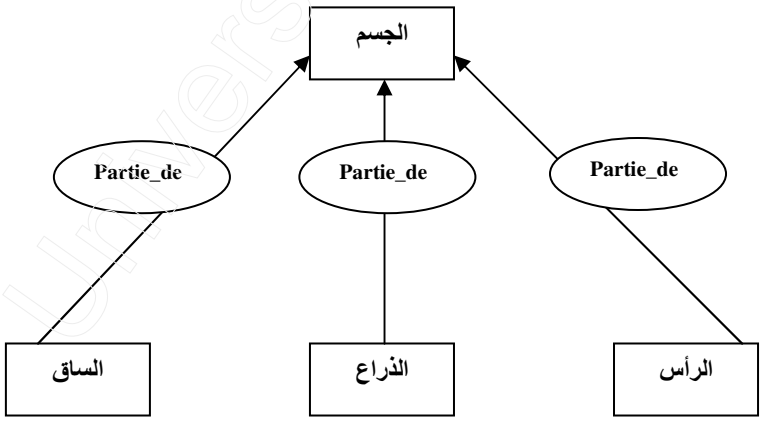


Fig. 16 : Exemple de la relation « Partie-de »

Notons aussi, que les relations peuvent avoir des propriétés. Essentiellement ces dernières peuvent être algébriques (symétrie, réflexivité, transitivité). Mais aussi des propriétés de cardinalité, en général, ces relations sont binaires.

Comme par exemple :

← "الحاسوب " - " يحتوي_على_الأقل" - "قرص_صلب "

← "المنـزل" - " يحتوي_على_الأقل " - "بـاب "

← "إنسان" - " عنده ما بين صفر واثنان" - "ساق"

- *Les propriétés liant deux relations*

- L'incompatibilité : Deux relations sont incompatibles si elles ne peuvent lier les mêmes instances de concepts.

Exemple : les relations «**تكون حمراء**» et «**تكون خضراء** » sont incompatibles ;

- L'inverse : Deux relations binaires sont inverses l'une de l'autre si, quand l'une lie deux instances I1 et I2, l'autre lie I2 et I1.

Exemple : les relations « **أب** » et « إبن » sont inverses l'une de l'autre ;

- L'exclusivité : Deux relations sont exclusives si, quand l'une lie des instances de concepts, l'autre ne lie pas ces instances, et vice-versa. L'exclusivité entraîne l'incompatibilité.

Exemple : « **يحتوي_على** » et « **لا يحتوي_على** » sont exclusives.

2.6. Les formalismes de représentation [GAE02]

Une ontologie, a besoin d'être représentée formellement et explicite (GARINO 1995). Plus encore, une ontologie doit représenter l'aspect sémantique des relations liant les concepts. A cet effet, de nombreux formalismes ont été développés pour représenter les connaissances, de la logique des prédicats jusqu'aux langages sophistiqués basés sur des structures de données. [GAE02]

2.6.1. Les formalismes logiques

Dans ce formalisme, Une base de connaissances rassemble un ensemble d'axiomes décrivant une situation, sur lesquels des règles d'inférence opèrent et produisent de nouvelles formules valides. Celles-ci constituent alors de nouveaux états de choses dans la base. Le langage de programmation Prolog en est un exemple.

2.6.2. Les réseaux sémantiques

En I.A., hommage à QUILLIAN qui fut le premier à développer de tels réseaux en tant que modèles de la mémoire associative humaine. Un réseau sémantique est un modèle de représentation, sous forme de graphe, du contenu sémantique des concepts. Les nœuds du graphe représentent des objets (concepts, situations, événements, etc.) et les arcs expriment des relations entre ces objets. Ces relations peuvent être des liens " sorte - de " exprimant la relation d'inclusion ou des liens " est-un " représentant la relation d'appartenance.
Exemple : on peut dire que : Volkswagen est une marque de voiture (voir la figure 17).

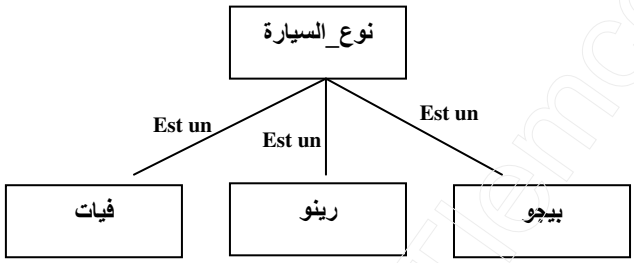


Fig. 17 : Exemple de réseau sémantique utilisant la relation " Est-un "

Exemple : On peut dire que « الصقر » *<est un>* « طائر_الجوارح » qui *<est_une_sorte de>* «طيور» (Voir la figure 18).

Fig. 18 : Exemple de réseau sémantique utilisant la relation " sorte-de "

Remarque :

« Une ontologie peut être considérée comme un réseau sémantique, puisqu'elle regroupe un ensemble de concepts décrivant complètement un domaine, reliés les uns aux autres par des relations, taxonomiques (hiérarchisation des concepts) et sémantiques. »

Parmi les réseaux sémantiques, les plus répandus pour la conceptualisation des ontologies, on trouve *les graphes conceptuels* dont le but fondamental est d'être " un système de logique hautement expressif, permettant une correspondance directe avec la langue naturelle " [SOW92]. Ils sont basés sur la logique pour la représentation des connaissances.

Note : Actuellement, la théorie des graphes conceptuels de [SOWA84], représentant les relations sémantiques, constitue le formalisme le plus répandu pour conceptualiser les ontologies.

Mis au point par John F. Sowa, pour modéliser une ontologie de haut niveau [SOW00]. Un graphe conceptuel est un graphe étiqueté, biparti, connexe et fini. Les sommets représentent les entités, attributs, états ou évènements. Chaque sommet est typé. Ces types sont ordonnés dans une structure de treillis orienté du plus spécifique au plus général avec des relations "sorte-de ". Le langage CGIF (Conceptual Graph Interchange Form) a été développé pour définir des graphes conceptuels. Exemple d'ontologie « Top-Level » de Sowa, treillis :

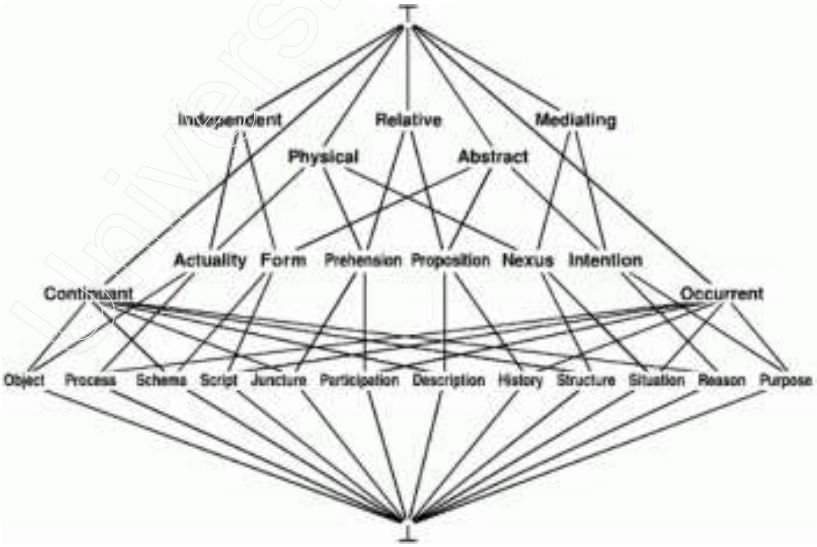
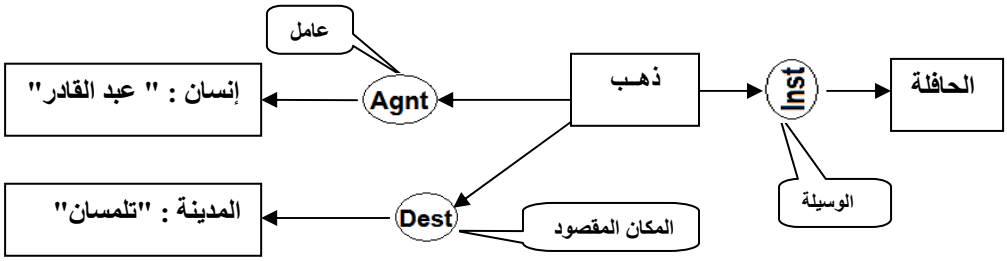


Fig. 19 : Hiérarchie du graphe de Sowa (Top-Level) : Treillis

Exemple : Graphe conceptuel à quatre concepts de la phrase :

ذَهَبَ عبدالقادر إلىَ تلمسان بالحافلة ←



2.6.3. Les schémas (Frame)

Les schémas, connus beaucoup plus sous son terme anglais « Frame », sont apparue en 1932 dans les études psychologiques. Plus tard, les schémas ont été introduits en intelligence artificielle par **Minsky** afin de résoudre les problèmes de la vision par ordinateur. Parmi les langages de représentation des connaissances à base de frame : KRL (1977) et KL-One (1985). Les frames ont été defines par MINSKY [MIN75] par: *"A frame is a data-structure for representing a stereotyped situation, like being in a certain kind of living room, or going to a child's birthday party. Attached to each frame are several kinds of information. Some of this information is about how to use the frame. Some is about what one can expect to happen next. Some is about what to do if these expectations are not confirmed. We can think of a frame as a network of nodes and relations. The "top levels " of a frame are fixed, and represent things that are always true about the supposed situation. The lower levels have many terminals - "slots " that must be filled by specific instances or data. Each terminal can specify conditions its assignments must meet (assignments themselves are usually smaller "sub-frames "). Simple conditions are specified by markers that might require a terminal assignment to be a person, an object of sufficient value, or a pointer to a sub-frame of a certain type. More complex conditions can specify relations among the things to several terminals ".*

Ce qui se traduit par :

> *« Un frame est une structure de données représentant une situation stéréotypée, comme se trouver dans un certain type de salon ou se rendre à un fête d'anniversaire d'un enfant. Divers types d'informations sont associés à chaque frame. Certaines d'entre elles concernent l'utilisation de ce frame. D'autres portent sur ce que l'on s'attend à ce qu'il arrive par la suite. D'autres encore portent sur ce qu'il faut faire si ces attentes ne sont pas confirmées […]. »*

Donc on peut dire qu'un schéma est une structure de données complexe décrivant une situation ou un objet standard. Un schéma, comme le montre la figure 20, est caractérisé par des attributs, des facettes et des relations.

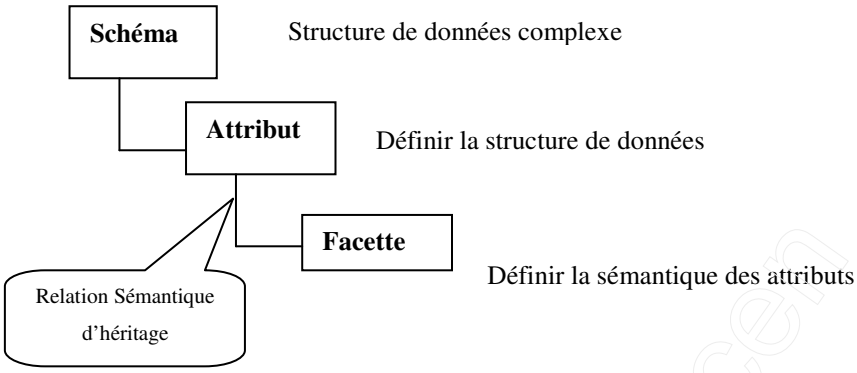


Fig. 20 : Eléments caractérisant un Schéma [MEL07]

2.6.4. Les scripts

La notion de " script " (ou scénario) a été introduite par Schank et Abel, sur le modèle des schémas pour le traitement du langage naturel. Ils ont défini un script par : " *A script is a structure that describes appropriate sequences of events in a particular context. A script is made up of slots and requirements about what can fill those slots. The structure is an interconnected whole, and what is in one slot affects what can be in an other. Scripts handle stylized everyday situations. They are not subject to much change, nor do they provide the apparatus for handling totally novel situations. Thus, a script is a predetermined, stereotyped sequence of actions that defines a well-known situation. Scripts allow for new references to objects within them just as if these objects had been previously mentioned; objects within a script may take* "the" *without explicit introduction because the script has already implicitly introduced them* " [SCH88].

Un script est donc une structure de données regroupant des connaissances relatives à une situation et qui permet de combiner des représentations.

3. Construction d'une ontologie

Soulignons qu'on ne peut construire une ontologie généralisant tous les domaines, pour la simple raison, que beaucoup de termes n'ont pas le même sens d'un domaine à un autre [BAC00]. De ce principe on peut que dire que pour construire une ontologie il faut la restreindre dans un domaine bien précis. Ainsi, les connaissances traduites par une ontologie sont à véhiculer à l'aide des deux principaux éléments suivants : *Concepts* et *Relations.*

Il existe trois méthodes pour la construction d'une ontologie [MEL07] :

- La Méthode manuelle : Les experts créent une nouvelle ontologie d'un domaine ou développent une ontologie déjà présente, l'ontologie WordNet[1].
- La Méthode automatique : L'ontologie est construite par des techniques d'extraction des connaissances : Des concepts et leurs relations sont extraits des bases de connaissances et ensuite vérifiés par les inférences.
- La Méthode mixte (Semi-automatique) : Les ontologies sont construites par des techniques automatiques mais elles permettent d'étendre des ontologies qui ont été construites manuellement comme la base des connaissances (Cyc[2]).

3.1. Le cycle de vie des ontologies

Le cycle de vie de l'ontologie permet d'identifier les différentes étapes de construction d'une ontologie. Uschold et Grüninger dans [USC96], ont mis au point une méthode, qui d'ailleurs est la plus connue, pour la création d'ontologie.

La méthode est générique, c'est pourquoi, ses étapes sont considérées comme la base d'un processus standard de construction. On remarque quatre phases fondamentales (Voir la figure 21) :

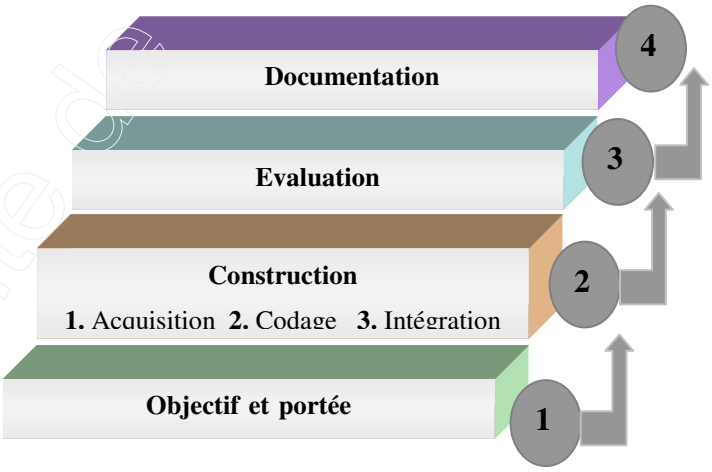


Fig. 21 : Etapes pour la construction des ontologies [MEL07]

Phase 1 : Objectif et portée : Identifier d'une façon générale l'objectif, la portée et les limitations de l'ontologie à construire.

Phase 2 : Construction : Cette étape prend un temps considérable et reste la plus difficile à réaliser, elle comporte :

1 - http://wordnet.princeton.edu/
2 - http://www.opencyc.org

- L'acquisition des connaissances :

Définir les concepts et les relations entre eux, sans ambiguïtés. Parmi les formes d'acquisition de connaissances qui existent, nous soulignons la forme d'analyse informelle de texte pour définir les concepts généraux ou bien sous forme d'une analyse formelle afin de définir les structures des connaissances [GOM97].

- Le codage :

Après la formalisation des concepts et leurs relations acquises, le codage permet de représenter l'ontologie dans un langage formel. La formalisation de l'ontologie peut être de différents degrés [BRI04] :

Très informel : L'ontologie est exprimée dans le langage naturel ;

Semi-informel : Elle s'exprime dans une forme structurée du langage naturel;

Semi-formel : Elle est exprimée dans un langage artificiel défini formellement;

Rigoureusement formel : l'ontologie est exprimée dans un langage formel utilisant une sémantique formelle avec des théorèmes et preuves.

- L'intégration :

Permettre la réutilisation des concepts déjà définis dans des ontologies existantes.

Phase 3 : Evaluation : Grüber 1993, repris par (Corcho et al. 2002) a proposé quelques critères pour l'évaluation d'une ontologie :

• La clarté : les concepts d'une ontologie doivent fournir le sens attendu de termes;

• La cohérence : l'ontologie ne doit permettre que les inférences qui sont en accord avec les définitions afin de ne pas créer de contradictions.

• L'extensibilité : Réutilisation de l'ontologie, sans remettre en cause le vocabulaire précédemment conçu ;

• Le biais d'encodage minimum1 (minimal encoding bias) : la spécification de l'ontologie doit être aussi indépendante que possible du choix d'une représentation.

1 - On parle aussi de déformation d'encodage minimale : une déformation d'encodage apparaît lorsque les choix d'une représentation sont faits uniquement pour une commodité d'implémentation. Ceci doit être évité car les agents partageant la connaissance peuvent être implémentés dans différents systèmes de représentation et styles de représentation.

• L'engagement ontologique minimal1 (minimal ontological commitment) : l'objectif est de permettre la spécialisation des spécifications d'une ontologie donnée selon des besoins réels ; c'est-à-dire l'expressivité maximum de chaque terme.

Phase 4 : Documentation : permet de renseigner les ontologies, leurs concepts importants ainsi que leurs objectifs. Ontolingua représente l'un des éditeurs le plus performant qui peut nous aider à réaliser cette étape en mettant à notre disposition des documentations formelles et informelles.

Par la suite, R.DIENG et al. dans [DIE01], affinent les étapes d'Uschold et Grüninger, et proposent un cycle de vie inspiré du génie logiciel, Il comprend une étape initiale d'évaluation des besoins, une étape de construction, une étape de diffusion, et une étape d'utilisation. Après chaque utilisation significative, l'ontologie et les besoins sont réévalués et l'ontologie peut être étendue et, si nécessaire, en partie reconstruite.

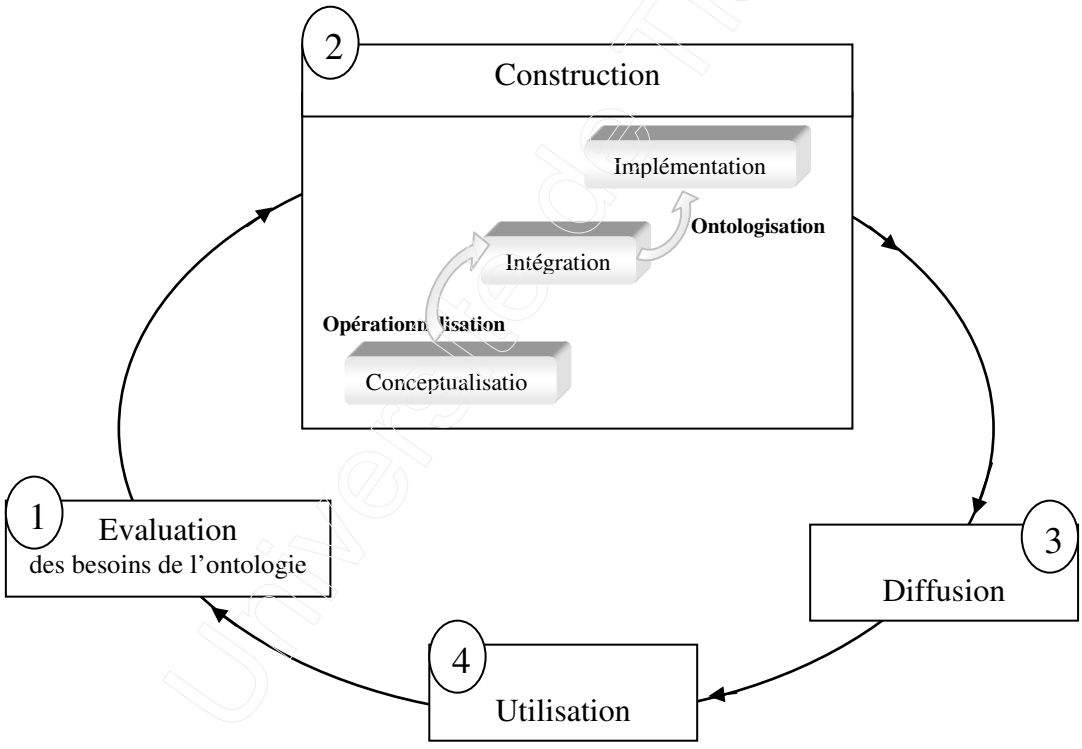


Fig. 22 : Le cycle de vie d'une ontologie [FUR02]

1 - Une ontologie doit faire aussi peu d'affirmations que possible sur ce qui a été modélisé afin de pouvoir instancier et spécialiser l'ontologie suivant les besoins.

3.1.1. Evaluation des besoins

Le cycle Commence par la phase d'évaluation des besoins, la finalité observée par la construction d'une ontologie se résume en 3 points [USC 95] :

L'objectif opérationnel : il est essentiel de souligner l'usage opérationnel de l'ontologie à travers des diagrammes de cas d'utilisation ;

- Le domaine de connaissance : il faut le cerner aussi clairement que possible, et le découper si nécessaire en termes de connaissances du domaine, connaissances de raisonnement, connaissances de haut niveau (communes à plusieurs domaines) ;
- Les utilisateurs : doivent être ciblés et identifiés autant que possible, pour ne pas s'éloigner de l'objectif opérationnel et de la granularité du formalisme choisie qui indique les propriétés d'intérêt pour l'utilisateur.

Après cette phase, les besoins sont définis, ainsi l'étape de la construction de l'ontologie peut démarrer, en commençant par la phase de conceptualisation.

3.1.2. Conceptualisation [FUR02]

Cette étape, permet de faire ressortir, à partir des données brutes, d'un corpus[1] l'ensemble des concepts et les relations entre eux décrivant ainsi les connaissances informelles.

Pour commencer, la première tâche est de trier les connaissances se trouvant dans le corpus spécifique (séparer les termes spécifique du domaine de ceux qui sont présents juste pour l'expression du domaine). La seconde tâche vient ensuite, pour spécifier ces concepts, ces relations, les propriétés des concepts et des relations, les règles et les contraintes, etc. c'est à-dire, tout l'aspect conceptuel. Le choix d'usage de l'ontologie peut donc être indiqué dans cette étape.

Notons qu'un texte ou corpus, n'a un sens que lorsqu'il est lu par un expert[2] [BAC99]. C'est pour cela que ce travail doit être mené par un expert du domaine, assisté par un ingénieur de la connaissance.

1 - Le corpus est souvent composé de textes d'auteurs extraits d'œuvres littéraires ou d'ouvrages critiques, etc.
2 - Le terme expert désignant justement ici une personne pour qui le corpus fait sens.

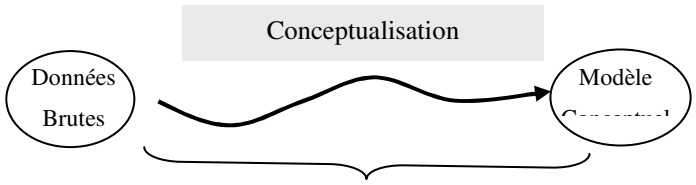


Fig. 23 : Processus de conceptualisation

En achevant cette étape, on obtient un ensemble de termes désignant les entités d'un domaine de connaissances (relatif au corpus). Ainsi, la conceptualisation est couronnée par un modèle informel, donc sémantiquement ambiguë et exprimé en langage naturel.

La fouille des connaissances d'un domaine peut se reposer à la fois sur l'analyse de documents et sur l'interview d'experts du domaine. Ces activités doivent être raffinées au fur et à mesure que la conceptualisation se dévoile. [FUR02]

3.1.3. Ontologisation

La deuxième phase est *l'ontologisation*. C'est une formalisation partielle de connaissances, sans perte d'information (formalisme générique avec une sémantique claire), du modèle conceptuel obtenu dans l'étape de conceptualisation.

Cette formalisation partielle va faciliter la représentation de l'ontologie dans un langage totalement formel et opérationnel. [FUR02]

Mais, essayer de préserver toutes les connaissances du domaine, nous amène à ajouter, à l'ontologie, des connaissances que nous ne pouvons formalisées c'est-à-dire dont la sémantique est ambiguïté. Cependant, le modèle ainsi obtenu est *semi-formel.*

Une ontologie semi-formelle n'est pas autorisée à être utilisée dans un système de base de connaissances. En revanche, une ontologie, contenant toutes les connaissances d'un domaine, constitue le support idéal de communication et de partage des connaissances.

Le résultat final d'une ontologisation est une hiérarchie de concepts, de relations, mais aussi d'attributs des concepts.

USCHOLD (1996) conseille de construire ces hiérarchies de bas en haut, pour donner ainsi la priorité aux concepts de bas niveau réellement utilisés dans le domaine, par rapport aux concepts qui ne sont souvent qu'ajouter artificiellement pour bâtir la hiérarchie.

Par exemple, pour construire une ontologie en rapport avec « الهندسة », les concepts de « النقطة » et « خط » vont, entre autres, être pris en compte, mais le concept

« مجموعة_من_النقط» peut être ajouté, concept qui subsume « النقطة » et « خط », et ceci uniquement pour structurer l'ontologie. Cette étape d'ontologisation donne un résultat en deux parties :

– *Une partie formelle* avec une sémantique claire ou du moins consensuelle ;

– *Une partie informelle*, avec une sémantique fixée *a priori* et donc exprimée dans un langage naturel ou semi-structuré.

Cette ontologisation est évalué par les critères d'évaluations de T.Grüber (1993), mentionné plus haut à l'étape d'évaluation du model de Uschold et Grüninger.

B. BACHIMONT propose de commencer à bâtir d'abord une ontologie différentielle[1] en organisant les concepts à l'aide des quatre principes suivant [BAC01] :

→ Communauté avec le père ou principe de similarité : un concept partage l'intension de son concept père ;

→ Différence avec le père ou principe de différence : l'intension d'un concept est différente de celle de son concept père, sinon il n'y aurait pas besoin de définir le concept fils ;

→ Communauté avec les frères ou principe de sémantique unique : une propriété est commune aux concepts frères issus du même concept père mais s'exprime différemment pour chaque frère, exemple : Les concepts « رجل » et «امرأة » portent la propriété « جنس » héritée de leur concept père «إنسان», mais cette propriété vaut « ذكر » chez « رجل » et « أنثى » chez « امرأة » ;

→ Différence avec les frères ou principe d'opposition : les frères doivent tous être incompatibles, sinon il n'y aurait pas besoin de tous les définir.

3.1.4. Opérationnalisation

Appelée parfois *représentation*, est une transcription de l'ontologie dans un langage formel et opérationnel de représentation de connaissances (*possède une syntaxe et une sémantique*). L'opérationnalisation consiste à donner la main à une machine pour manipuler les connaissances d'une ontologie.

Ce travail doit être mené par l'ingénieur des connaissances pour traduire le model conceptuel structuré, issu de la phase d'ontologisation, dans un langage semi formel de représentation par exemple, les langages à base des frames, le modèle des graphes conceptuels (traitant des opérations telles que la jointure, la projection, etc.) ou les logiques de description (opérations de subsomption). L'ontologie opérationnelle est implantée en machine au sein d'un système manipulant le modèle de connaissances utilisé via le langage opérationnel choisi. A l'étape de diffusion l'ontologie est testée par rapport au contexte d'usage pour lequel

1 - Ontologie différentielle : permet de préciser le sens des concepts (concepts sémantiques) de manière non ambigüe

elle a été construite. Une fois cette étape achevée, l'ontologie, en question, peut être mise à la disposition des utilisateurs. [FUR02]

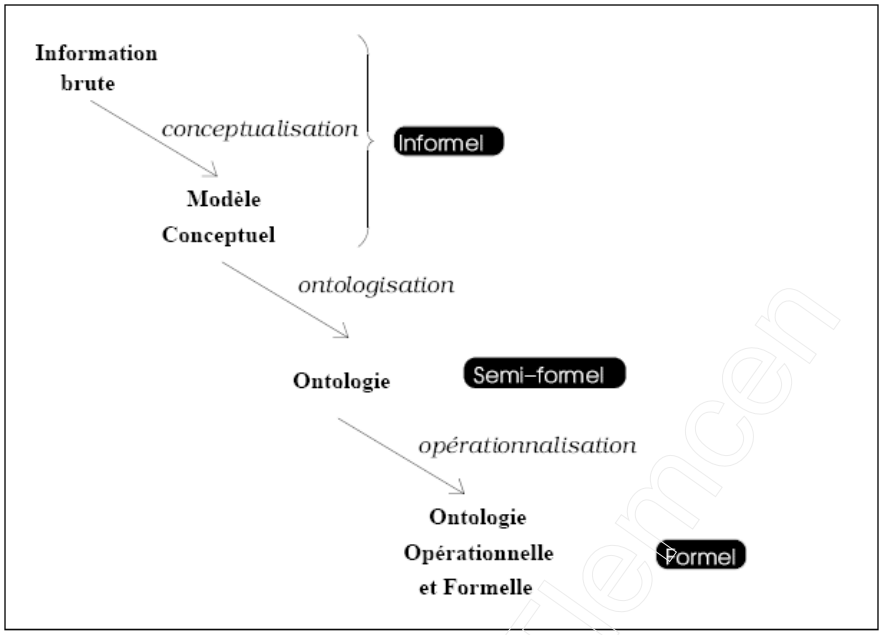


Fig. 24 : Construction d'une ontologie opérationnelle. [FUR02]

3.2. L'évaluation et l'évolution d'une ontologie

M. GRUNINGER et M.S. FOX proposent d'utiliser des questions de compétences permettant de tester l'ontologie. Ainsi cette phase, d'évaluation d'une ontologie se fait a priori par des tests correspondants à l'objectif opérationnel de l'ontologie. [GRU95]

La validation formelle consiste à vérifier l'absence de cycle, c'est-à-dire de définition en boucle, s'il n'y a pas redondance de concepts ou de relations, si chaque hiérarchie est bien connexe[1] [GOM 96].

La validation sémantique permet de contrôler que la structure des hiérarchies est correcte vis-à-vis des principes différentiels utilisés.

Si l'évaluation de l'ontologie a échoué, il faut donc remonter jusqu'à l'étape de conceptualisation, afin d'éviter des modifications qui ne respectent pas la sémantique du domaine, et faire évoluer l'ontologie.

3.3. La fusion des ontologies

Un domaine de recherche encore peu exploré. La fusion consiste de construire une nouvelle ontologie et l'intégrer dans une ontologie déjà existante. Soit un simple alignement

1 - hiérarchie connexe : il n'y a pas de concept ou de relation isolé des autres et donc sans aucun sens

dans le cas où aucune partie n'est commune aux ontologies, soit une véritable fusion [NOY02]. La fusion ne peut se faire que si les deux ontologies présentent des entités conceptuelles (concepts ou relations) communes afin de pouvoir les exprimer et les identifier dans un même formalisme.

3.4. Méthodologie et outils de construction d'ontologies

Beaucoup de méthodologies et d'outils ont été développés dans le but de créer des ontologies. Nous allons cités à titre d'information, que les plus importants c'est-à-dire ceux qui offrent des implémentations de méthodologies, bien qu'ils offrent un processus d'ontologisation mais ils restent beaucoup à faire du coté de conceptualisation.

Méthodologies :

→ Approche de construction d'ontologie de domaine à partir de grandes ontologies (SENSUS, Cyc, AKT,…) : Construire des ontologies de domaine à partir des grandes ontologies déjà existantes.

→ Méthode de Uschold et King's [USC95] : Basé sur l'expérience acquise lors du développement de l'ontologie d'Entreprise, the enterprise ontology.

→ La méthodologie On-To-Knowledge[1] (OTK) : Donc, la méthodologie On-To- Knowledge propose de construire une ontologie en tenant compte de comment l'ontologie va être utilisée par l'application plus tard (cette méthodologie est très dépendantes de l'application).

→ METHONTOLOGY : Cette méthodologie développée A.GOMEZ-PEREZ au sein du groupe d'ontologie à l'université polytechnique de Madrid. METHONTOLOGY est né dans les travaux de software development process [IEEE96] et dans knowledge engineering methodologies [GOM97] et [WAT86].

Outils :

- TERMINAE, qui intègre des outils linguistiques, permet la visualisation des résultats des extracteurs de candidats-termes Lexter et/ou Syntex développés par D. Bourigault.Ces concepts doivent ensuite être triés par un expert et organisés hiérarchiquement, puis la sémantique du domaine est précisée à travers des axiomes.

- DOE (DIFFERENTIAL ONTOLOGY EDITOR) 1ère version en 2002, dernière version (v2.0) 2008. DOE n'est pas complet, il est simplement un éditeur

1 - http://www.ontoknowledge.org.

d'ontologie qui applique la méthodologie des principes différentiels de Bruno Bachimont.

- ODE (ONTOLOGY DESIGN ENVIRONMENT), construction des ontologies au niveau connaissances (Méthodologie METHONTOLOGY).

- ONTOEDIT (ONTOLOGY EDITOR), est indépendant de tout formalisme. Il permet l'édition des hiérarchies de concepts et de relations et l'expression d'axiomes algébriques portant sur les relations, et de propriétés telles que la généricité d'un concept.

- PROTEGE2000 issu du modèle des frames, c'est un outil non lié à des formalismes de représentation. Il permet aussi l'édition, le contrôle, la visualisation et l'extraction d'ontologie à partir des textes.

- ONTOLINGUA qui constitue en fait une extension du langage KIF (KNOWLEDGE INTERCHANGE FORMAT). Ce serveur d'édition permet la fusion d'ontologie. L'ontologie est immédiatement représentée dans un formalisme.

- OILED (OIL EDITOR) est un éditeur d'ontologie s'inspirant du formalisme OIL. Un éditeur de petites ontologies avec un moteur d'inférence pour tester la cohérence de l'ontologie construite.

Plusieurs méthodes en Ingénierie Ontologique mais encore loin d'être complètes des méthodes du Génie Logiciel. Il reste beaucoup à faire pour unifiés toutes ces méthodes. Cette situation justifie les travaux de recherche en cours dans différentes équipes de laboratoires.

L'article, dans [FER99], présente un premier état de l'art, relativement complet, sur ces méthodes de développement. Ainsi que l'article [AUS00] qui exploite des résultats de travaux menés par le groupe français TIA (Terminologie et IA) pour proposer une méthode basée sur l'analyse de corpus.

4. Classification des ontologies

Les ontologies peuvent être de nature très diverses. Afin de mieux s'y retrouver, un certain nombre de classifications ont été proposées.

La plus courante des classifications d'ontologies est la classification selon l'objet de conceptualisation [PSY03]. On peut ainsi distinguer sept catégories [GOM99b] :

- Les ontologies de représentation des connaissances : les ontologies de représentations des connaissances sont utilisées pour formaliser un modèle de représentation des connaissances. On peut par exemple citer l'exemple de l'ontologie de *frame* [GRU93], qui définit les primitives de représentation des langages à base de frames (classes, instances, slots, facettes, etc.).

- les ontologies supérieures (aussi appelées ontologies de haut niveau):

Une ontologie de haut niveau décrit des concepts très généraux comme l'espace, le temps, la matière, les objets, les événements, les actions, etc. Ces concepts ne dépendent pas d'un problème ou d'un domaine particulier. Ces ontologies doivent être, du moins en théorie, consensuels à de grandes communautés d'utilisateurs [GUA98]. Des exemples d'ontologies de haut niveau sont « Upper Cyc ».

- Les ontologies génériques (méta-ontologie) : Elles contiennent des concepts généralistes, mais moins abstraits que ceux contenus dans les ontologies de haut niveau. On pourra réutiliser l'ontologie dans plusieurs domaines [PSY03]. Un exemple d'une telle ontologie : SUMO (Suggested Upper Merged Ontology), une autre ontologie générique a été développée dans cette classe citons : Wordnet.

- Les ontologies de tâches [MIZ03] : Ce type d'ontologie sert à modéliser les tâches d'un problème ou d'une activité donnée. Ce type d'ontologie est utile pour décrire la structure d'une tâche de résolution de problème de manière indépendante du domaine concerné.

- Les ontologies de domaine : Elles sont réutilisables à l'intérieur d'un domaine donné et modélisent le vocabulaire à l'intérieur de ce domaine [GOM99b]. La plupart des ontologies existantes sont des ontologies de domaine [PSY03].

- Les ontologies de tâches-domaine : ce sont des ontologies de tâches spécifiques à un certain domaine. Un exemple d'une telle ontologie est celui d'une ontologie des termes liés à la planification chirurgicale. [GOM99b]

- Les ontologies d'application. Il s'agit du type d'ontologie le plus spécifique [PSY03]. Les concepts que l'on trouve dans ce genre d'ontologies modélisent les concepts d'un domaine particulier dans le cadre d'une application donnée.

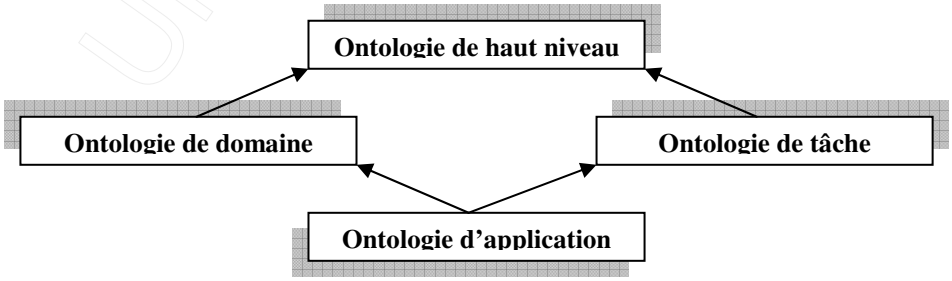


Fig. 25 : Type d'ontologie selon Guarino

On peut aussi classer les ontologies selon le niveau de formalisme du langage que l'on utilise pour les modéliser. Uschold et Gruninger proposent [USC96] à ce niveau quatre types

d'ontologies (Figure 26) :

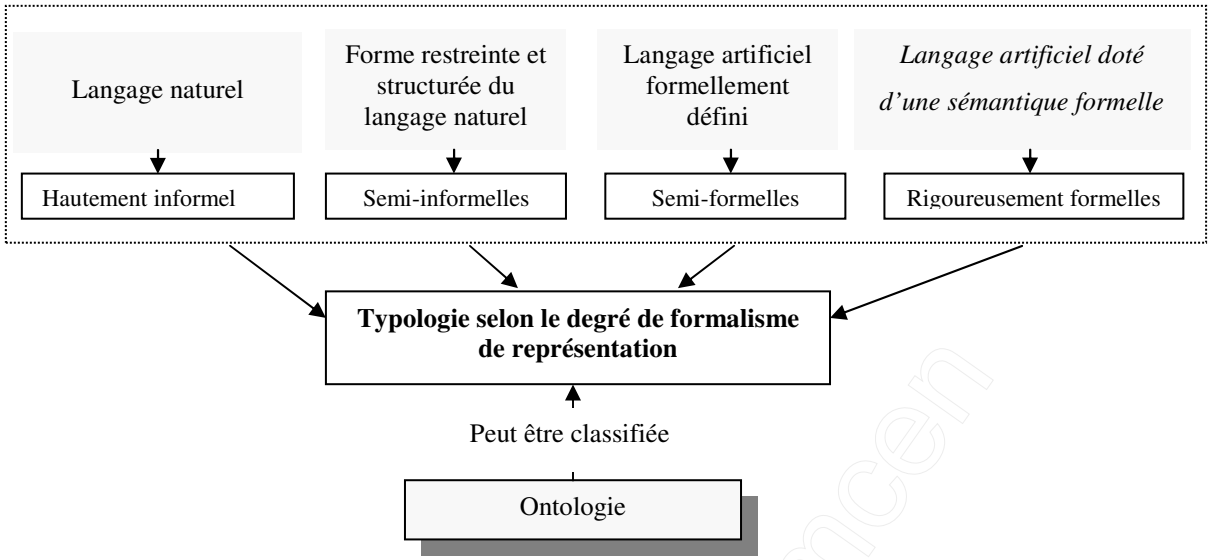


Fig. 26 : Différents types d'ontologies selon le degré de formalité

- Les ontologies informelles, exprimées en langage naturel.
- Les ontologies semi-informelles, écrites dans un langage naturel, mais sous une forme limitée et structurée, permettant d'augmenter la clarté et la lisibilité.
- Les ontologies semi-formelles, exprimées dans un langage artificiel défini de manière formelle.
- Les ontologies strictement formelles, définies elles aussi dans un langage artificiel, mais avec des théorèmes et des preuves sur des propriétés de l'ontologie, telles que la robustesse ou la complétude.

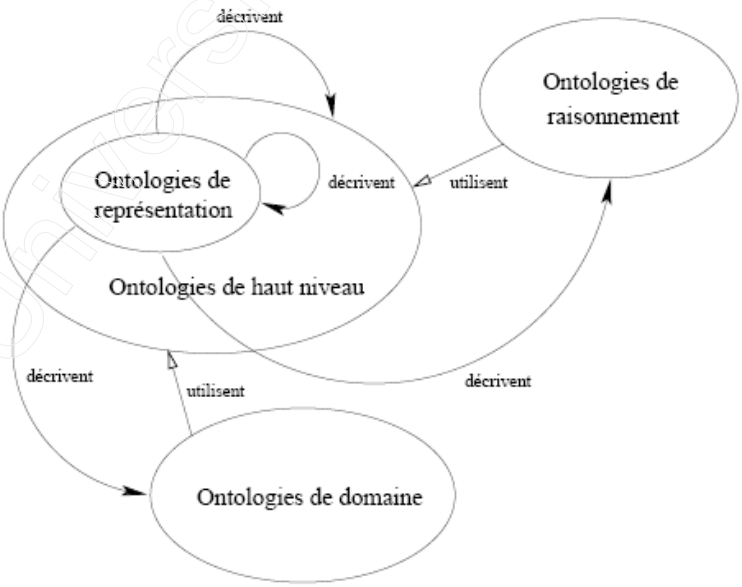


Fig. 27 : Différents types d'ontologies [MIZ 97]

5. Conclusion

Dans ce chapitre, nous avons précisé les différentes définitions de la notion d'ontologie. Nous avons aussi souligné les différents éléments dont elle est constituée et une description des besoins auxquels elle répond, ainsi que les différents domaines d'applications.

Les méthodes et les outils de construction ont été bien détaillés, car ils ont une relation importance avec le thème de notre thèse. La création d'ontologie est expliquée par la représentation d'un cycle de vie d'une ontologie. Le chapitre est achevé par une classification des ontologies.

Le chapitre suivant sera consacré à la description de la base lexicale WordNet, désormais appelée ontologie lexicale, [GRA04] énonce « … *un lexique peut néanmoins servir à l'élaboration d'une ontologie*… ». Cette dernière sera notre modèle de construction d'ontologie.

Chapitre 3

Ontologie WordNet

Modèle de notre axe de recherche

1. Historique et origine

Machiavel a dit un jour en politique, « la fin justifie les moyens ». L'absence d'un dictionnaire électronique facilement accessible, a fait de WordNet un projet d'étude (développement manuel) au début des années 1980 à l'Université Princeton par une équipe de psycholinguistes et de linguistes, qui se sont basées sur la mémoire lexicale humaine, sous l'impulsion de **G. Miller**. Peu temps après, il a émergé pour devenir ***une ressources lexicale électronique de la langue anglaise***, comprenant plus de 200.000 de mots de classe ouvertes ainsi que plus de 115.000 ensemble de synonymes. [**MIL93**]

Actuellement plusieurs réalisations descendantes de WordNet existent (différentes langues), parmi eux EuroWordNet (EWN, 1996) et ArabicWordNet (AWN), ce dernier est construit selon les méthodes développées pour EuroWordNet. L'approche EuroWordNet maximise la compatibilité à travers les WordNet(s) et se concentre sur un encodage manuel des concepts les plus complexes et les plus importants. [**VOS98**]

Deux éléments ont participé au succès de WordNet :

- La maturité du projet rendue possible grâce à un travail de plus de dix ans.
- Le libre accès aux sources du projet tant pour consultation que pour la modification ainsi que la possibilité de redistribution du produit modifié.

2. Présentation de WordNet

Qu'est-ce WordNet ? Un dictionnaire[1] ? Un thésaurus[2] ? Les dictionnaires contiennent généralement des connaissances sur des lexies[3] alors que les encyclopédies[4]

1 - Dictionnaire : Recueil des mots d'une langue, des termes d'une science, d'un art, rangés par ordre alphabétique, avec leur signification. [*www.mediadico.com*]

2 - Thésaurus : une liste de termes sur un domaine de connaissances, reliés entre eux par des relations synonymiques, hiérarchiques et associatives. Le thésaurus constitue un vocabulaire normalisé. [www.fr.wikipedia.org]

3 - Lexie : C'est une suite de caractères formant une unité sémantique, *un mot*, et pouvant constituer une entrée de dictionnaire. [www.fr.wikipedia.org]

4 - Une encyclopédie peut prendre la forme d'un livre ou plusieurs livres. Elle se présente souvent comme une collection d'articles traitant chacun un thème. [www.fr.wikipedia.org]

contiennent des connaissances éparses, du monde, sur la surface de la terre. Quant aux thésaurus, leur structure est bâti autour des concepts et aident l'utilisateur à acquérir l'unité lexicale la plus appropriée lorsqu'il a un concept à rechercher. WordNet n'est ni un dictionnaire classique ni un thésaurus : il est en fait, un arrangement des traits de chacune de ces deux ressources lexicales. [**FEL98**]

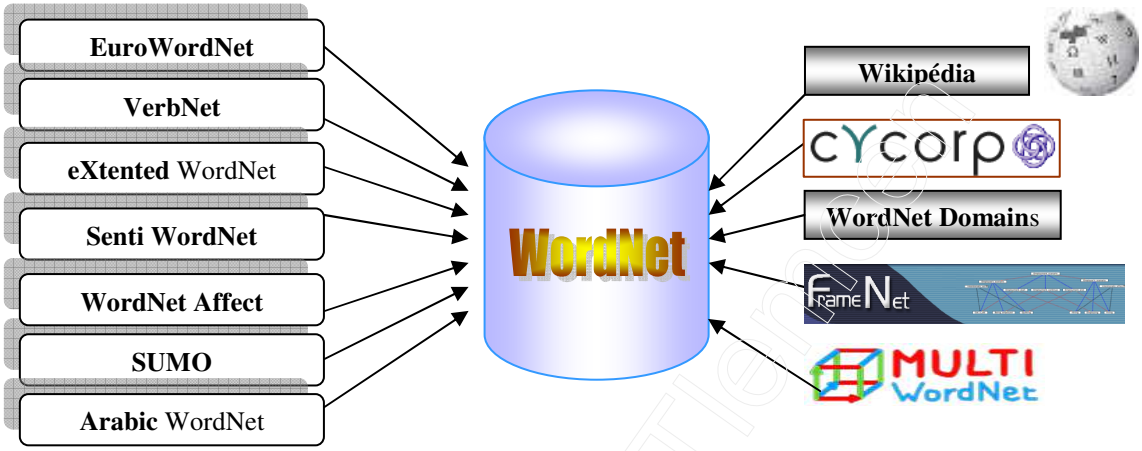


Fig. 28 : Ressources descendances de WordNet

(Liste non exhaustive)

3. Conception & Structure de WordNet

On peut considérer WordNet comme un graphe ou un réseau sémantique, souvent qu'on qualifie d'ontologie légère (Light Ontology), ou chaque nœud représente un concept du monde réel. La conception de WordNet est basée sur les théories de la représentation des connaissances mentales : mémorisation des mots et concepts d'une manière hiérarchique, en utilisant la relation d'inclusion (qui lie, par exemple, des triplets comme «animal », « oiseau», et « Chardonnet »). [**COL98**]

Exemple :

Un concept peut être un objet tel que « *Car* » une entité tel que « *Teacher* » ou un concept abstrait tel que « *art* ». Chaque nœud est constitué d'un ensemble de mots, où chacun représente le concept associé à ce nœud. Un nœud peut être vu comme un ensemble de mots dont chacun représente le même concept.

Exemple :

Le concept « *car* » est représenté par l'ensemble de mots {car, auto, automobile, motocar}.

Dans la terminologie de WordNet cet ensemble est appelé est nommé « Synset ». WordNet offre des descriptions détaillées et précises des mots. Leur structuration sur un axe ontologique a un fondement psychologique. Il résulte de cette approche qu'il arrive parfois que l'on rencontre plus de 20 sens pour un verbe, par exemple le verbe « ***give*** » a 27 sens.

3.1. SynSet

WordNet manipule les unités lexicales non pas par des mots mais par un ensemble de synonymes ou « Synset », groupes de mots ou de phrases qui expriment le même concept. Des différences de sens entre les membres d'un «Synset» se montrent dans différentes restrictions de sélection. Par exemple, « *rise* » (monter) et « *fall* » (tomber / descendre) peuvent choisir comme argument des entités abstraites comme « *temperature* » (température) et « *prices* » (prix).

Un « Synset » est accompagné d'une petite définition dite « ***gloss*** **»** qui décrit un concept du monde réel.

Exemple **:** les nœuds suivant correspondent aux différents sens de "mouse" dans WordNet :

1. Mouse -- (any of numerous small rodents typically resembling diminutive rats having pointed snouts and small ears on elongated bodies with slender usually hairless tails)
2. Shiner, black eye, mouse -- (a swollen bruise caused by a blow to the eye)
3. Mouse -- (person who is quiet or timid)
4. Mouse, computer mouse -- (a hand-operated electronic device that controls the coordinates of a cursor on your computer screen as you move it around on a pad ; on the bottom of the device is a ball that rolls on the surface of the pad ; "a mouse takes much more room than a trackball")
5. Sneak, mouse, creep, pussyfoot -- (to go stealthily or furtively; "…stead of sneaking around spying on the neighbour's house")
6. Mouse -- (manipulate the mouse of a computer)

Notons que WordNet met l'accent sur les liaisons entre les « Synset » (arc du graphe de la Figure 29) pour marquer sa valeur ajoutée faces aux dictionnaires traditionnaires. Chaque lien décrit une relation entre concept du monde réel. Par exemple, les relations telles que : « a spoke **is a part of** a wheel » ou « **a vehicle is a kind of** conveyance »

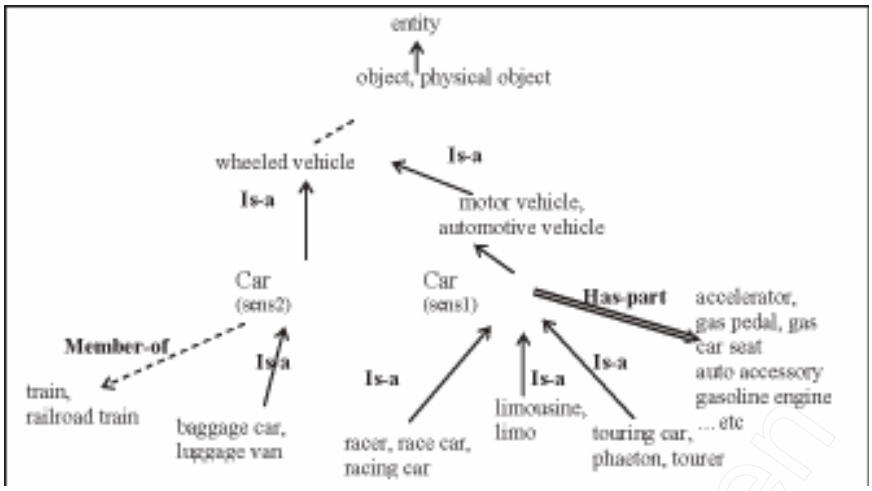


Fig. 29 : Exemple de sous hiérarchie dans WordNet correspondent au concept "car". [BAZ05]

3.2. Organisation

WordNet sépare les données en quatre (04) bases de données, organisées différemment les unes des autres, associées aux catégories de **noms**, **verbes**, **adjectifs** et **adverbes**. Les **noms** et **verbes** sont organisés en hiérarchies. Des relations d'hyperonymie («**est-un**») et d'hyponymie relient les « ancêtres » des noms et des verbes avec leurs «spécialisations ». Au niveau racine, ces hiérarchies sont organisées en types de base. [**FEL98**] Le réseau des noms est bien plus profond que les autres parties. Il faut noter que, les deux premiers niveaux de la hiérarchie des noms se constituent des concepts *abstraits* suivants :

• **ABSTRACTION**: ATTRIBUTE, MEASURE/QUANTITY/AMOUNT, RELATION, SET, SPACE, TIME…
• **HUMAN ACTION**: ACTIVITY, COMMUNICATION, DISTRIBUTION, INACTIVITY, JUDGMENT, LEANING, LEGITIMATION, MOTIVATION, PROCLAMATION, PRODUCTION, SPEECH ACT…
ENTITY: ANTICIPATION, CAUSAL AGENT, ENCLOSURE, EXPANSE, LOCATION, PHYSICAL OBJECT, SKY, SUBSTANCE, THING…
• **EVENT**: GROUP ACTION, NATURAL EVENT, MIGHT-HAVE-BEEN, MIGRATION, MIRACLE, NONEVENT, SOCIAL EVENT…
• **GROUP, GROUPING**: ASSOCIATION, BIOLOGICAL GROUP, PEOPLE, COLLECTION, AGGREGATION, COMMUNITY, ETHNIC GROUP, KINGDOM, MULTITUDE, POPULATION, RACE, RARE-EARTH ELEMENT…
• **PHENOMENON**: EFFECT/RESULT, LEVITATION, FORTUNE/CHANCE, REBIRTH, NATURAL PHENOMENON, PROCESS, PULSATION…
• **POSSESSION**: ASSETS, CIRCUMSTANCES, PROPERTY/MATERIAL POSSESSION, TRANSFERRED PROPERTY, TREASURE…
• **PSYCHOLOGICAL FEATURE**: COGNITION/KNOWLEDGE, FEELING, MOTIVATION/NEED…
• **STATE**: ACTION/ACTIVITY, EXISTENCE, STATE OF MIND, CONDITION, CONFLICT, DAMNATION, DEATH, DEGREE, DEPENDENCY, DISORDER, EMPLOYMENT, END, FREEDOM, ANTAGONISM, IMMATURITY, IMMINENCE, IMPERFECTION, INTEGRITY, MATURITY, OMNIPOTENCE, PERFECTION, PHYSIOLOGICAL STATE, RELATIONSHIP, STATE OF AFFAIRS, STATUS, TEMPORARY STATE, NATURAL STATE…

WordNet organisent les verbes en taxonomie. Plusieurs propriétés des verbes dépendent de la manière avec laquelle on peut combiner les arguments (sujet, objets direct, objets indirects) [**WID04**].

Notamment qu'elle relation utilise t – on ? « ***is a kind of*** » ?, WordNet fournit une relation équivalente à celle–ci c'est la relation « ***Manner Of*** » nommée : « **Troponymy** »

Les adjectifs et les adverbes sont organisés de paires d'antonymes[1] tels que « good, bad ». Bon nombre d'adjectifs en langue anglaise possède des antonymes, par opposition aux verbes et aux noms. Les adverbes sont souvent définis par les adjectifs dont ils dérivent. C'est ainsi ils héritent de la structure des adjectifs.

3.3. La matrice lexicale

Un sens peut être représenté par plusieurs mots. Et un mot à son tour peut désigner plusieurs sens.

Exemple : le mot « *word* » fait référence en même temps à une *expression* et à un *concept*. Pour éliminer cette ambiguïté : **« *word form* »** sera utilisé pour exprimer (la forme ou l'image) et **« *word meaning* »** sera utilisé dénoter le concept que porte ce mot « *word* ». [**FEL98**]

Dans la matrice lexicale (voir à droite), une entrée dans la cellule de la matrice E_{ij}, suppose le « *word form* » F_i est utilisée pour référencer le concept « *word meaning* » (concept) Mj. [FEL98]

Word Meanings	Word Forms				
	F1	F2	F3	…	Fn
M1	E_{11}	E_{12}			
M2		E_{22}			
M3			E_{33}		
…				…	
Mn					E_{mn}

Tableau 1 : Illustration des concepts de la matrice Lexical [FEL98]

4. Les relations dans WordNet

Deux relations fondamentales interviennent dans WordNet, notamment celle entre les « *word form* » appelés *relations lexicales* (par exemple : la synonymie), et celle qui associent les « *word meaning* » appelés relation sémantiques (par exemple : l'hyperonymie).

Remarquons que la majorité des relations dans WordNet sont des « synset » de la même catégorie, excepté les relations « *pertains to* » et « *attribute* » souvent utilisées entre les adjectifs et les noms. Le tableau 2 illustre un sous ensemble des relations dans WordNet qu'on détaillera par la suite.

1 - Antonyme : nom opposé, (exemple : *black* est antonyme de *White*).

Relation	Description	Exemple
Hypernym	Is a generalization of	Furniture is a hypernym of chai
Hyponym	Is a kind of	Chair is a hyponym of furniture
Troponym	Is a way to	Amble is a troponym of walk
Meronym	Is part/substance/member of	Wheel is a (part) meronym of a bicycle
Holonym	Contains part	Bicycle is a holonym of a wheel
Antonym	Opposite of	Ascend is an opposite of descend
Attribute	Attribute of	Heavy is a attribute of weight
Entailment	entails	Ploughing entails digging
Cause	Cause to	To offend causes to resent
Also see	Related verb	To lodge is related to reside
Similar to	Similar to	Dead is similar to assassinated
Participle of	Is participle of	Stored (adj) is the participle of "to store"
Pertainym of	Pertains to	Radial pertains to radius

Tableau 2 : Quelques relations dans WordNet

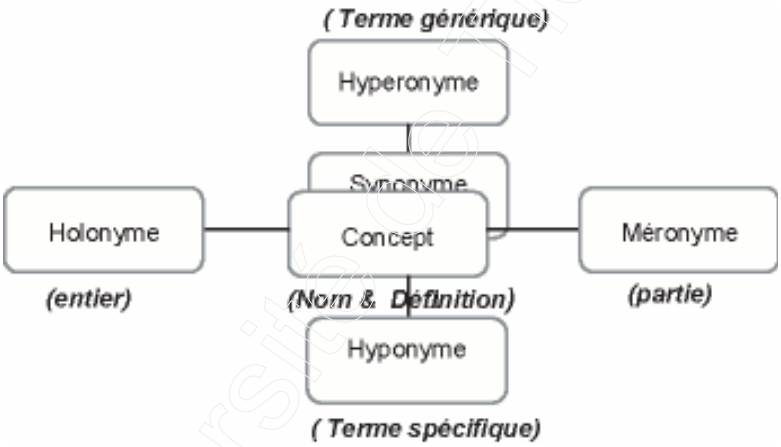


Fig. 30 : Principales relations sémantiques dans WordNet. [BAZ05]

Voilà, une définition des relations les plus importantes dans WordNet :

4.1. Synonymie

La synonymie est une relation liant deux concepts équivalents ou voisins (frêle / fragile). Il s'agit d'une relation symétrique. Christiane Fellbaum, dans [**FEL98**], énonce une définition, généralement attribuée à Leibniz, « *two expressions are synonymous if the substitution of one for the other never changes the truth value of a sentence in which the substitution is made.* » Ce qui ce traduit par : Deux expressions sont synonymes dans un contexte linguistique C, si la substitution de l'une pour l'autre en C, ne modifie pas la valeur de vérité de la phrase dans laquelle la substitution soit faite. Par exemple, la substitution « a

plank » pour « a board » groupé dans un même Synset {board, plank}, sera rarement substituer dans des contextes de menuiserie, bien qu'il y ait d'autres contextes comme un conseil d'administration où cette substitution serait totalement inapproprié. [**FEL98**]

4.2. Antonymie

Une autre relation familière est l'antonymie, qui s'avère être étonnamment difficile de définir. L'antonyme d'un mot « x » n'est pas toujours « Non x ». Par exemple, « riche » et « pauvre » sont des antonymes, mais de dire que quelqu'un n'est pas riche ne signifie pas qu'il doit être pauvre, de nombreuses personnes ne se considèrent ni riche ni pauvre. L'Antonymie, semble être une simple relation symétrique, mais elle est en faite assez complexe, et beaucoup de personnes trouvent des difficultés à reconnaître les antonymes quand ils les voient. **[FEL98]** Antonymie n'est pas une relation sémantique entre « *word meanings* » mais beaucoup plus une relation lexicale entre « *word forms* ». Par exemple, la signification {rise, descend} – (s'élever/descendre) et {fall, ascend} – (monter/tomber) peuvent être conceptuellement opposées, mais ils ne sont pas des antonymes; [rise, fall] sont des antonymes et aussi [asscend, descend]. Ces faits font ressortir la nécessité de distinguer les relations sémantiques entre les « word forms » et les relations sémantiques entre « word meaning ».

4.3. L'Hyperonymie / Hyponymie

Ce sont les relations, les plus importantes, dans notre travail. Elles représentent la relation « ***is a kind of***» ou tout simplement la relation « ***is-a*** ».

Elles sont réservées seulement pour les catégories Nom et Verbes qui se voient organisées sous forme d'une hiérarchie comportant un seul nœud racine. Les nœuds représentant les concepts les plus généraux sont les ancêtres des nœuds représentant les concepts les plus spécifiques. On dit alors que le concept le plus général « Subsume » celui du plus spécifique.

Exemple : dans la figure 29, « Entity » est le concept le plus hiérarchique des noms c'est la racine du nœud, il subsume le concept spécifique « wheeled vehicle » qui à son tour subsume le concept « Car ».

WordNet dispose de 9 hiérarchies pour les noms et 628 pour les verbes. L'hyperonymie représente la relation permettant d'avoir les ancêtres d'un « *word meanings* » à l'opposé de l'hyponymie qui fournit ses descendants. On note, aussi que WordNet permet

l'héritage multiple, ce qui induit qu'un concept peut avoir plusieurs hyperonymes. WordNet représente :

L'hyperonymie par le symbole : @ Exemple : Tree @ → plant

L'hyponymie par le symbole : ~ Exemple : plant ~ → Tree

4.4. Méronymie

C'est une relation liant un concept C1 à un concept C2 qui est en fait une partie de C1 (C1= Fleur / C2= Pétale), un de ses membres (Forêt / Arbre) ou une substance le constituant (vitre / verre). Donc la méronymie est interprétée de trois manières différentes :

Si X is a part of Y (Composant – Objet complet)

Si X is substance of Y (Matière – Objet)

Si X is a member of Y (Membre – collection)

Il existe d'autres relations que la relations que nous allons donner juste une bref définition, telles que :

- **Métonymie** (Holonymie) : relation liant un concept C1 à un concept C2 dont il est une des parties. C'est la relation inverse de la méronymie.
- **Implication** : relation lie un concept C1 à un concept C2 qui en découle (marcher/faire un pas).
- **Causalité** : relation liant un concept C1 à son effet (tuer / mourir).
- **Valeur** : relation liant un concept C1 (adjectif) qui est un état possible pour un concept C2 (pauvre / condition financière).
- **A pour valeur** : relation liant un concept C à ses valeurs (adjectifs) possibles (taille / grand). C'est la relation inverse de Valeur.
- **Voir aussi** : relation entre des concepts ayant une certaine affinité (froid / gelé).
- **Similaire à** : certains concepts adjectifs dont le sens est proche sont regroupés. Un Synset est alors désigné comme étant central au regroupement. La relation « Similaire à » lie un Synset périphérique au Synset central (moite / humide).
- **Dérivé de** : indique une dérivation morphologique entre le concept cible (adjectif) et le concept origine (froidement / froid).

WordNet contient approximativement 117798 mots de nom organisés approximativement en 82115 concepts (Synset) (tableau 1) jusqu'à juillet 2008. Puisque la majorité des noms communs parfois sont des noms propres, aucune tentative curieuse de les

exclure n'est faite. En termes d'exhaustivité le but de WordNet diffère un peu des dictionnaires standards des écoliers. C'est à l'organisation de ces informations que WordNet espère de l'innovation.

Réseau	Formes	Synsets	Paires mot-sens
Noms	117798	82115	146312
Verbes	11529	13767	25047
Adjectifs	21479	18156	30002
Adverbes	4481	3621	5580
TOTAL	155287	117659	206941

Tableau 3. Statistique sur WordNet (juillet 2008)

5. Les verbes dans WordNet (réseau sémantique)

Actuellement, WordNet contient plus de 25000 mots verbes. Les verbes sont divisés en 15 fichiers, selon un critère sémantique, presque chacun représente ce que les linguistes appellent domaine sémantique :

Les verbes de soins de corps et fonctions, changement, cognition, communication, compétition, consommation, contact, création, émotion, mouvement, perception, possession, interaction sociale et les verbes météorologiques.

Pratiquement tous les verbes dans ces fichiers dénotent des évènements et des actions, un autre fichier contient les verbes d'état, tel que *suffice*, *belong*, et *resemble*, qui ne peuvent pas être intégrés dans les autres fichiers. Les verbes dans ce dernier groupe font référence à un état, et ne constituent pas un domaine sémantique et ne partage aucune propriété sémantique.

Plusieurs fichiers prennent leurs noms des verbes les plus hauts, ou "*unique beginners*" qui sont en tête des groupes lexicaux sémantiquement cohérents.

Les verbes sont la catégorie syntaxique la plus importante du langage. Toutes les phrases anglaises doivent contenir au moins un verbe. Les linguistes ont longuement débattues la question de mettre le verbe comme pivot centrale de la phrase.

A cause de la complexité de ces informations, les verbes sont probablement la catégorie syntaxique la plus difficile à étudier.

6. L'hyponymie entre les verbes

La phrase modèle utilisée pour tester l'hyponymie entre les noms, « x est-un y » n'est pas convenable pour les verbes : *to amble is a kind of to walk* ce n'est pas une phrase correcte.

La distinction sémantique entre les verbes est différente des propriétés qui distinguent deux noms dans une relation hyponymique.

Les différentes élaborations qui distinguent un l'hyponyme du verbe de son père immédiat sont résumées dans la relation « de manière que », Fellbaum et Miller la surnommée « Troponymie ». La relation de troponymie entre deux verbes peut être exprimée par la formule *To V1 is to* V2 d'une certaine manière particulière.

7. Polysémie[1]

Bien que les phrases anglaises nécessitent des verbes et non pas nécessairement des noms, le langage a moins de verbes que de noms. Par exemple, le dictionnaire « *Collins English* » liste 43636 différents noms et 14190 différents verbes.

Les verbes sont polysémiques beaucoup plus que les noms : les noms en *Collins* ont une moyenne de 1.74 sens, alors que les verbes ont 2.11 sens.

La haute polysémie des verbes suggère que les sens des verbes sont plus flexibles que les sens des noms. Les verbes changent leurs sens selon les types d'arguments de nom avec lesquels se ils se produisent, alors que les sens de noms tendent à être plus stables avec les différents verbes.

Genter et France ont montré ce qu'ils appellent « la haute mutabilité des verbes », et conclurent que les sens des verbes sont plus facilement changeables parce qu'ils sont moins cohésifs que les sens de noms.

Les verbes les plus fréquemment utilisés (have, be, run, make, set, go, take, …) sont plus polysèmes et leurs sens dépendent légèrement des noms avec lesquels ils entrent en production. Par exemple, les dictionnaires différencient entre les sens de « *have* » dans les phrases comme « I have a Mercedes » et « I have a headache ». La différence est moins due à la polysémie de have que à la nature concrète ou abstraite de ses objets.

Dans le cas des verbes polysèmes comme « beat » (battre), les différences de sens sont déterminées par la sémantique des arguments du verbe plutôt que par les différentes élaborations d'un ou deux composants essentiels communs partagés par la majorité des sens de « beat ». Afin de réduire l'ambiguïté en WordNet, les synsets de verbes pourraient contenir des pointeurs de renvoi aux Synsets Noms qui contiennent des noms choisis par les verbes.

---

1 - Polysémie : est la qualité d'un mot ou d'une expression qui a deux voire plusieurs sens différents.
Exemple : «œil-de-bœuf » désigne une plante en botanique, un animal en zoologie, une pierre en géologie et une fenêtre ronde en architecture.

8. Arabic WordNet (AWN)

8.1. L'écriture arabe [BLA06]

L'arabe est une langue sémitique. Le système d'écriture de la langue arabe a 25 consonnes et trois voyelles longues **: « و، ا، ي »** : (OU, A, iii) on les appelle « **حروف العلّة** » qui sont écrites de droite à gauche et prennent différentes formes en fonction de leur position dans le mot. En plus des voyelles longues, l'arabe a des voyelles courtes qui ne font pas partie de l'alphabet, mais plutôt sont écrites comme des voyelles diacritiques (Fig.30 voyelles en verts) en haut ou en bas d'une consonne pour lui donner le son désiré et par conséquent de générer un mot dans un sens souhaité.

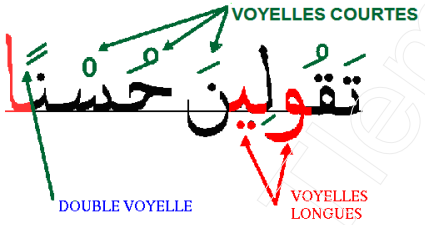


Fig. 31 : Représentation des voyelles arabes[1]

Le terme « arabe classique » renvoie la forme standard de la langue utilisée dans tous les écrits et entendu à la télévision, la radio et dans les discours publics et les serments religieux. Les textes sans voyelles sont considérées comme étant plus appropriée par la communauté de langue arabe puisque c'est la forme habituelle de la vie quotidienne des documents écrits et imprimés (livres, magazines, journaux, lettres, etc.)

Mais quand il s'agit du texte du « Coran », et plus généralement aux collections imprimées des livres scolaires et des dictionnaires arabes sur les supports en papier, les voyelles diacritiques apparaissent dans leurs intégralités. Comme on remarque très souvent dans des livres bien édités, des manuscrits ou bien certains textes imprimés la présence partielle ou au hasard des voyelles diacritiques sur les mots ambigus ou difficiles à lire.

Par exemple, un mot en arabe composé de deux lettres comme « **بر** » c'est à dire, «b» et «r», peut être très ambiguë, sans les voyelles diacritiques (voir Tableau 4), ou par exemple « **علم** ». Notamment pour un écrivain, il peut utiliser les signes diacritiques afin que les lecteurs puissent facilement résoudre toute ambiguïté.

1- **http://www.webarabic.com/portail/apprendre/index.php?rub=ecrire&page=3&section=les%20voyelles%20arabes**

Cependant, bien que la plupart des Arabes peuvent lire des textes avec des voyelles explicitement indiqué, moins ils peuvent écrire des textes en utilisant des voyelles diacritiques correctes. Ainsi, il est très difficile de compter sur des utilisateurs, quel que soit leur origine, pour entrer correctement des mots clés de recherche nécessitant des voyelles diacritiques. [BLA06]

Arabic word	Transliteration	POS	Meaning
بَرَ	**barr** short vowel 'a'	noun	land (as opposed to sea)
بَرَّ	**barr** short vowel '*a*'	adj	reverent, dutiful, kind
بُرُ	**burr** short vowel 'u'	noun	wheat
بِرُ	**birr** short vowel '*i*'	noun	reverence, kindness

Arabe	Translitération	PoS	Sens
عَلَم	'alam	n	flag
عِلْم	'ilm	n	science
عُلِمَ	ulima	v	known
عَلَّمَ	'allama	v	teach
عَلَم	'alam	a	famous

Tableau 4 : Voyelles diacritiques possibles sur « بر » et sur « علـم »

Pourtant, une mauvaise utilisation d'un seul signe diacritique fera un échec lors d'une requête, de recherche d'un document numérique par exemple, « **السكون** » qui indique que la consonne n'est pas suivi par une voyelle, ou la « **الشدّة** » (comme dans « **بـرّ** » 'barr' dans le tableau 4 et « **درّس** » darrasa dans le tableau 5), ce qui indique une double consonne {le premier n'est pas suivie d'une voyelle (ici « **السكون** ») et la seconde est suivie d'une voyelle}.

Arabic word	POS	Pattern	Meaning
دَرَسَ darasa	verb	فَعَلَ fa?ala	study
دَرَّسَ darrasa	verb	فَعَّلَ fa??ala	teach
دَرْس dars	noun	فَعْل fa?l	lesson
دِرَاسَة dira:sah	noun	فِعَالَة fi?a:lah	study
مُدَرِّس mudarris	noun	مُفَعِّل mufa??il	teacher
مَدْرَسَة madrasah	noun	مَفْعَلَة maf?alah	school
تَدْرِيس tadri:s	noun	تَفْعِيل taf?i:l	teaching
تَدَارَسَ tada:rasa	verb	تَفَاعَلَ tafa:?ala	discuss
دِرَاسِي dira:si	adj	فِعَالِي fi?a:li	educational

Tableau 5 : Dérivations de la racine (d r s)

Beaucoup de personnes ont tendance à faire des erreurs sur la position de certains signes diacritiques sur un mot. Cela peut poser un sérieux problème pour les systèmes de recherche d'information et les systèmes informatisés de ressources lexicales qui dépendent de l'entrée, bien formulée, de l'utilisateur. Sinon, on peut assister même à des rejets de requêtes d'utilisateurs.

En particulier, il peut y avoir un rejet total d'une nouvelle ressource lexicale solide tel qu'AWN à moins que cette nouvelle ressource suppose que la plupart des utilisateurs du discours arabe ne sont pas experts en écriture des voyelles diacritiques et par conséquent les ignorent complètement. Ces utilisateurs sont plus à l'aise quand à la lecture de textes sans signes diacritiques, dans les documents écrits de tous les jours y compris les contrats juridiques et commerciaux, journaux, livres ainsi que les dictionnaires sur des supports papiers ou numérisés. En conclusion, on peut dire qu'il est préférable de permettre aux utilisateurs d'entrer des mots en arabe sans voyelles diacritiques mais parallèlement permettre au système de retrouver ces mots avec des voyelles diacritiques pour les besoins de désambiguïsation. [BLA06]

8.2. Description d'AWN

L'Arabic WordNet est une base de données lexicale. Sa conception basé sur Princeton WordNet est construite suivant des méthodes développées pour EuroWordNet est reliée avec l'ontologie SUMO (Suggested Upper Merged Ontology). Arabic WordNet a été développé par DOI / REFLEX (2005-2007) [BLA06] (voir Figure 32).

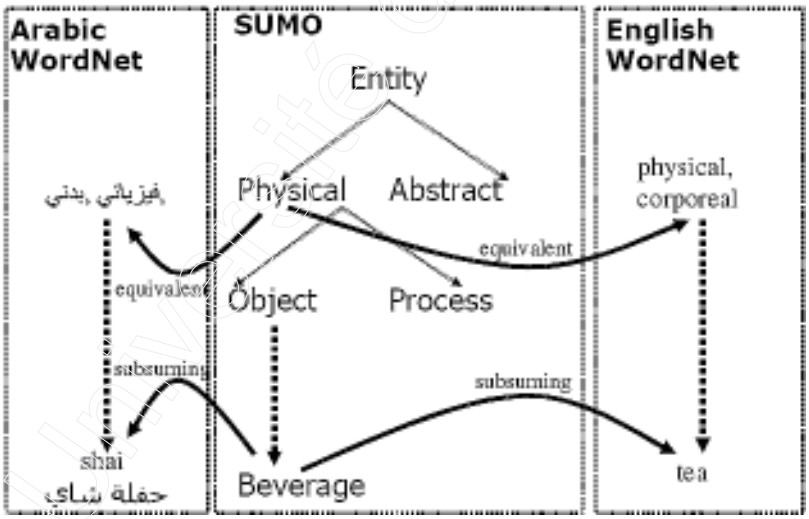


Fig. 32: Mapping de SUMO vers WordNet(s)

(Structure et organisation de l'AWN)

SUMO est en pleine expansion afin d'offrir un fondement solide pour la formalisation sémantique de l'Arabic WordNet (AWN). La base de données AWN[1] est disponible gratuitement.

L'ontologie Arabic WordNet contient 9228 concepts « Synsets » (6252 nominales et 2260 verbales, 606 adjectival, et 106 adverbales), contient 18,957 expressions et 1155 concepts nommés [**BLA06**] le fichier base de l'AWN sous format XML[2] contient les quatre balises :

- Item : Contient les concepts (Synsets), les classes et les instances de l'ontologie.
- Word : Contient les mots arabes vocalisés.
- Form : Contient les Racines des mots arabes « root ».
- Link : Contient les relations entre les concepts.

8.3. Construction d'Arabic WordNet (AWN)

AWN sélectionne les Synsets, en se basant sur des critères [**BLA06**] :

AWN doit être aussi dense que possible par des connexions de chaînes hyperonymie/hyponymie etc. En contreparties, La plupart des Synsets d'AWN doivent correspondre à leurs homologues du WordNet anglais et la topologie entière des deux Wordnets doit être similaire.

Pertinence : Donner la priorité aux concepts les plus fréquents. Ces critères incluront la fréquence des éléments lexicaux (en arabe et en anglais), PoS (Nom, verbes, adjectifs, adverbes…) et la fréquence des racines arabes dans leur corpus de référence respectifs.

Généralités : Les Synsets les plus préférés sont ceux des ontologies de hauts niveaux – WordNet. Pour assurer ces trois critères, deux façons de procéder :

- De l'anglais vers l'arabe : On sélectionne, pour chaque Synset anglais, toutes les variantes correspondantes en arabe.
- De l'arabe à l'anglais : Tous les sens d'un mot arabe doivent être trouvés dans le WordNet anglais, ainsi que chacun de ces sens il faut sélectionner ces Synsets correspondants en anglais.

1 - Disponible gratuitement dans le lien suivant : http://www.globalwordnet.org/AWN/

2 - La base sous format *XML* et une autre *MySQL disponible dans le lien suivant :* http://www.globalwordnet.org/AWN/DataSpec.html**c**

Ces deux étapes doivent être suivies tout au long de la construction d'AWN. Tous les Synsets AWN doivent être validé manuellement (et éventuellement verrouillé, lorsque toutes leurs variantes ont été trouvées), mais il convient d'exploiter, autant que possible, les ressources disponibles pour guider le processus de construction et de validation.

Une fois qu'un nouveau verbe arabe est ajouté à AWN, plusieurs possibilités d'extension sont à considérer : Les extensions des entrées verbales, y compris les dérivés verbaux1, les nominalisations et les noms verbaux2, etc. Nous considérons également les formes les plus productives comme les dérivés des pluriels brisés (**جموع التكسير**)3. Ceux-ci peut-être fait grâce à un ensemble de règles lexicales et morphologiques pour tirer un maximum de profit de ces extensions des itérations courtes seront effectuées. Pour construire AWN, il faut d'abord construire l'ensemble de la base de ses concepts (C.B.) à partir de l'ensemble de la base commune des concepts (CBCs4) d'EWN5 et BalkaNet6. La concentration est faite sur les termes les plus pertinents afin d'obtenir environ 1.000 Synsets nominal et 500 Synsets verbale.

La deuxième étape consiste en une extension verticale de haut en bas de la base des concepts. [**FAR05**] et [**DIA04**]. Certains prétraitements sont nécessaires pour la prochaine étape. Nous citons deux tâches, la préparation et extension.

La préparation : La préparation consiste au traitement des ressources disponibles bilingue et la compilation d'un ensemble de règles lexicales et morphologiques. De l'ensemble des dictionnaires bilingues disponibles, un dictionnaire bilingue homogène (HBIL) a été construit comprenant pour chaque information en entrée Arabe/Anglais, une paire de mot, la racine arabe est ajoutée manuellement, PoS (Part of speech), les fréquences relatives et les sources supportent l'appariement.

---

1 - Exemple : soit le radical ***ktb*** (**كتب**) « écrire », on peut former *les dérivés verbaux :*

- ***kataba*** (**كَتَبَ**) « écrire »,
- *i**kta-ta-ba*** (**اكتتب**) : « copier »

2 - Exemple : **EL**dhikr (**الذِكْر**) : Nom verbal tiré de la racine arabe **dhakara** (**ذَكَرَ**).

3 - Exemple : Arka:n (**أَرْكَانٌ**) : pluriels brisés de *roukn* (**رُكْنٌ**)

4 - CBCs: Set of Common Base Concepts (CBCs) from the 12 languages in EWN and BalkaNet. [TUF04]

5 - EWN : EuroWordNet est un projet visant à construire des ontologies similaires au projet WordNet de l'université de Princeton pour 8 langues européennes dont le français. [http://www.illc.uva.nl/EuroWordNet/]

6 - BalkaNet est un projet européen 2001-2004, visant à développer des Wordnets ***alignés*** pour la région des langues Balkans suivantes : Bulgare, Grec, Romanie, Serbe, Turc et d'étendre le WordNet Tchèque précédemment élaborée dans le projet EuroWordNet.

Dix sept (17) méthodes heuristiques sont utilisées pour le développement d'EWN et sont appliquées à HBIL [FAR05] pour dériver les mots candidats anglais/arabe par un simple mappage des Synsets. Pour chaque mappage, l'information attachée comprend le mot arabe et sa racine, le Synset anglais, POS, les fréquences relatives, l'évaluation du mappage, la profondeur absolue dans WordNet, un certain nombre d'écarts entre le Synset et le sommet de la hiérarchie WordNet et les sources contenants la paire.

Les mots arabes dans les ressources bilingues doit être normalisée et lemmatisée [**DIA04**], [**HAB05**], mais les voyelles et les signes diacritiques doivent être maintenus. Les racines arabes n'ont pas de voyelles.

Extension : Après le prétraitement, l'ensemble des mots marqués arabe/anglais des paires Synset deviennent une entrée à l'étape de validation manuelle. Nous procéderons par blocs d'unités connexes (ensembles de Synsets WordNet connexes, par exemple les chaînes d'hyponymie et l'ensemble des mots arabes connexes (C'est à dire, les mots ayant la même racine) au lieu des unités individuelles (Synsets, sens, mots). [**BLA06**]

Finalement, AWN sera complété par l'ajout d'une terminologie et des entités nommé1, pour combler les lacunes de sa structure qui couvre certain domaine spécifique.

8.4. L'interface Utilisateur

Outre la recherche et la navigation simple et facile sur l'ensemble de la base de données pour les utilisateurs finaux, les lexicographes ont besoin aussi d'une interface d'édition. Une variété de composants hérités sont disponibles, chacun d'eux avec ses avantages relatifs.

Dans [BLA06], William BLACK et Sabri ELKATEB et al, ont choisi d'adapter celle décrite dans Black et Elkateb (2004), car elle peut prendre en charge l'écriture arabe. Toutefois, cette méthode a présenté un tout autre modèle de données, dans lequel les mots arabes étaient directement liés aux écarts représentant les Synsets de WordNet. Elle a également été structurée pour supporter la navigation et la recherche dans un espace de Synset entièrement en anglais et par un simple mappage des mot-Synset pour introduire l'arabe.

Cette nouvelle interface a tenté de mettre les deux langues sur un même pied d'égalité et effectivement être indifférente à la direction d'alignement entre les structures

1 - **Entités nommées** : Sa reconnaissance est une sous-tâche de l'activité d'extraction d'information dans des corpus documentaires. Elle consiste à rechercher des objets textuels (c'est-à-dire un mot, ou un groupe de mot) catégorisables dans des classes telles que noms de personnes, noms d'organisations ou d'entreprises, noms de lieux, quantités, distances, valeurs, dates, etc.
[**http://fr.wikipedia.org/wiki/Entit%C3%A9s_nomm%C3%A9es**]

conceptuelles des deux langues. Par ailleurs, l'éditeur de l'interface communique avec le serveur de base de données en utilisant le protocole SOAP1 (Simple Object Access Protocol). Il s'agit de permettre aux multiples lexicographes de différents sites de maintenir une base de données commune.

9. Conclusion

WordNet est sans doute le précurseur et la référence en matière de base lexicales sémantiques et informatiques devenue peu à peu à caractère ontologique, une description sommaire de ce dernier à était faite et constitue donc un exemple concret de notre présentation des ontologies.

Nous avons tenté à travers ce chapitre, de donner un aperçu global de ce qu'est l'ontologie WordNet. Cette ressource lexicale nous a surpris tant dans sa structure que dans le travail et les efforts investis pour la réaliser. Ce chapitre est un avant plan de ce qu'on prévoit à étudier comme approche à la réalisation d'un WordNet arabe. Le chapitre suivant donne un état de l'art des différentes méthodes de l'apprentissage ontologique.

1 - SOAP : un protocole de communication basé sur XML pour permettre aux applications de s'échanger des informations via HTTP. [**http://igm.univ-mlv.fr/~dr/XPOSE2003/axis_seng/soap.html**]

Chapitre 4

Etat de l'art
Apprentissage ontologique
Extraction des connaissances à partir des textes

Introduction

Dans le cadre de notre recherche qui puise des sources textuelles « arabes », on peut se poser la question pourquoi les textes et pas autres choses ? Cette question trouve sa justification dans la définition même d'une ontologie citée par Grüber 1993, quand il fait intervenir la notion de consensus et de la conceptualisation partagée. En effet les textes sont très riches en connaissances et accumulent un vocabulaire partagé entre une grande communauté d'un domaine.

Notre problème est donc d'acquérir, à partir d'un texte, un ensemble de connaissances utiles pour la construction d'une ontologie. Il est question donc, de s'informer à partir de la langue écrite pour acquérir de la connaissance, dès lors que l'on veut y accéder à l'aide d'outils informatiques.

Dans cette partie, on présente un état de l'art d'extraction de connaissances à partir de textes. En ingénierie ontologique cet axe de recherche s'appelle « l'apprentissage ontologique » ou communément parlant « Ontology learning ». Ce chapitre va être sectionné en deux parties :

Dans une première partie, nous allons présenter un spectre illustrant aux utilisateurs différentes systèmes existants (liste non exhaustive) dans ce domaine ainsi que les moyens qui leurs permettrons de les comparer. Pour cela une présentation et un recensement sur toutes les techniques, stratégie, disciplines, approches et backgrounds agissants dans ce contexte seront étudiées. [**SHA02-a**] présente une classification en dimensions selon des critères pour faciliter les interactions entre ce domaine et les autres disciplines.

La seconde partie, est réservée pour la description d'un processus consensuelle de l'apprentissage ontologique (Ontology learning), ainsi qu'une classification des approches les plus importantes.

Partie I

Comparaison entre différents systèmes & approches

Une cinquantaine (50) de systèmes d'extraction de connaissances (d'apprentissage ontologique à partir des textes) issus des travaux récents de laboratoires, de conférences et de revues publiés, sont exploités par [SHA02-a] et choisi, parmi eux sept, systèmes les plus distingués pour ensuite relever leur différences dans un cadre de comparaison.

1. Les systèmes d'apprentissages ontologiques

L'apprentissage ontologique se réfère à l'extraction des éléments ontologiques (connaissances conceptuelles) à partir des textes et construit ensuite une ontologie avec ces éléments. La construction manuelle des ontologies est une tâche lourde et assez coûteuse en temps, chers, biaisé en fonction de leur développeur, non-flexible aux changements et spécifiques seulement aux objectifs tracés. L'automatisation de la construction d'ontologies élimine non seulement les coûts, mais aussi, il en résulte une meilleure ontologie correspondante à son application.

Beaucoup de systèmes, utilisant l'approche semi-automatiques d'apprentissage ontologique, ont attiré notre attention en vue de la préparation de notre état de l'art. Par exemple : Adaptiva, SOAT, OntoLearn, TextStorm, ASIUM, HASTI, DODDLE II, SVETLAN, SYNDICATE, TEXT-TO-ONTO, WEBGroup de systèmes →KB. Mais nous n'avons retenue que sept systèmes, de base, pour ce cadre de comparaison, les autres ne sont qu'une image des 7 systèmes modèles de notre études. Ainsi les systèmes retenues sont :
ASIUM, HASTI, DODDLE II, SVETLAN, SYNDICATE, TEXT-TO-ONTO, WEB→KB

Noms des Systèmes	Références	Caractéristiques
ASIUM	(Faure, et al., 1998; Faure & Poibeau, 2000)	Cadre d'apprentissage des verbes et les connaissances taxonomiques, basée sur l'analyse statistique de l'analyse syntaxique de textes en français. « **A**cquisition of **S**emant**I**c knowldge **U**sing **M**achine learning method »
DODDLE II	(Yamaguchi, 2001)	Outil de traitement pour apprendre les relations taxonomiques et non taxonomiques en utilisant de méthodes statistiques (analyse de co-occurrence), et l'exploitation des dictionnaires numérique (WordNet) et des textes spécifiques à un domaine.
HASTI	(Shamsfard 2003; Shamsfard & Barforoush, 2000; 2002a;b)	Apprentissage des mots, des concepts, des relations et des axiomes, dans les deux modes incrémental et non-incrémental, à partir d'un petit noyau (ou apprentissage à partir de zéro), en utilisant une approche hybride symbolique, les combinaisons logiques, basée sur la linguistique, basé sur des patrons, et des méthodes heuristiques.
SVETLAN'	(Chalendar & Grau, 2000)	Permet d'acquérir automatiquement des classes de noms par domaines sémantico-pragmatiques à partir de textes. Il regroupe des mots jouant le même rôle syntaxique par rapport à un même verbe, où seuls les mots les plus pertinents pour décrire le domaine sont retenus.
SYNDIKATE	(Hahn & Schnattinger, 1998; Hahn & Romacker, 2001; Hahn & Marko, 2002)	« SYNthesis of Distributed Knowledge Acquired from Texts » L'apprentissage progressif de mots, de concepts et de relations, est basé sur la compréhension du texte ou de la phrase, en utilisant deux sources, linguistique et conceptuelle « de qualité» des différentes formes d'éléments.
TEXT-TO-ONTO	(Maedche & Staab, 2000a; b; 2001)	Apprentissage des concepts et des relations à partir de données non structurées, semi-structurées et structurées, en utilisant une méthode multi-stratégie d'une combinaison de règles d'association, une analyse formelle de concepts et de Clustering.
WEB→KB	(Craven et al., 1998; 2000)	Combinaison de statistiques (bayésien) et de méthodes logiques (règles d'apprentissage FOL) pour apprendre les instances et l'extraction des règles à partir de documents du Web
Adaptiva	C. Brewster, F. Ciravegna, and Y. Wilks, 2002	Basée sur des modèles linguistiques et d'apprentissage automatique, mais avec un peu plus d'itérative et d'approche coopérative.
SOAT	T. Yamaguchi, 2000	Système hautement automatisé, apparemment très efficace. Quatre différentes relations entre les concepts sont extraites; catégorie, synonyme, d'attributs et d'événements. L'inconvénient, une préparation très lourde.
OntoLearn	A. Maedche et al. 2000,2001,2002, 2003	L'architecture OntoLearn se compose de trois phases principales : 1. Extraction terminologique 2. Interprétation Sémantique 3. Création d'une vue spécialisé de WordNet
TextStorm	A. Oliveira et al, 2001	Le système TextStorm analyse et étiquette un fichier texte, en utilisant WordNet, puis extraits des prédicats binaires à partir du corpus de texte. Les prédicats symbolisent une relation entre deux termes, extraite d'une phrase.

Tableau 6 : Systèmes proposés et sélection du cadre de l'étude de comparaison

2. les six dimensions de comparaison

Un *framework* de comparaison est proposé par Shamsfard [SHA02-a], montrant ainsi les points qui font la différence entre une méthode et une autre. Ce cadre de comparaison réunit les caractéristiques et les techniques de plusieurs approches. (Voir figure 33)

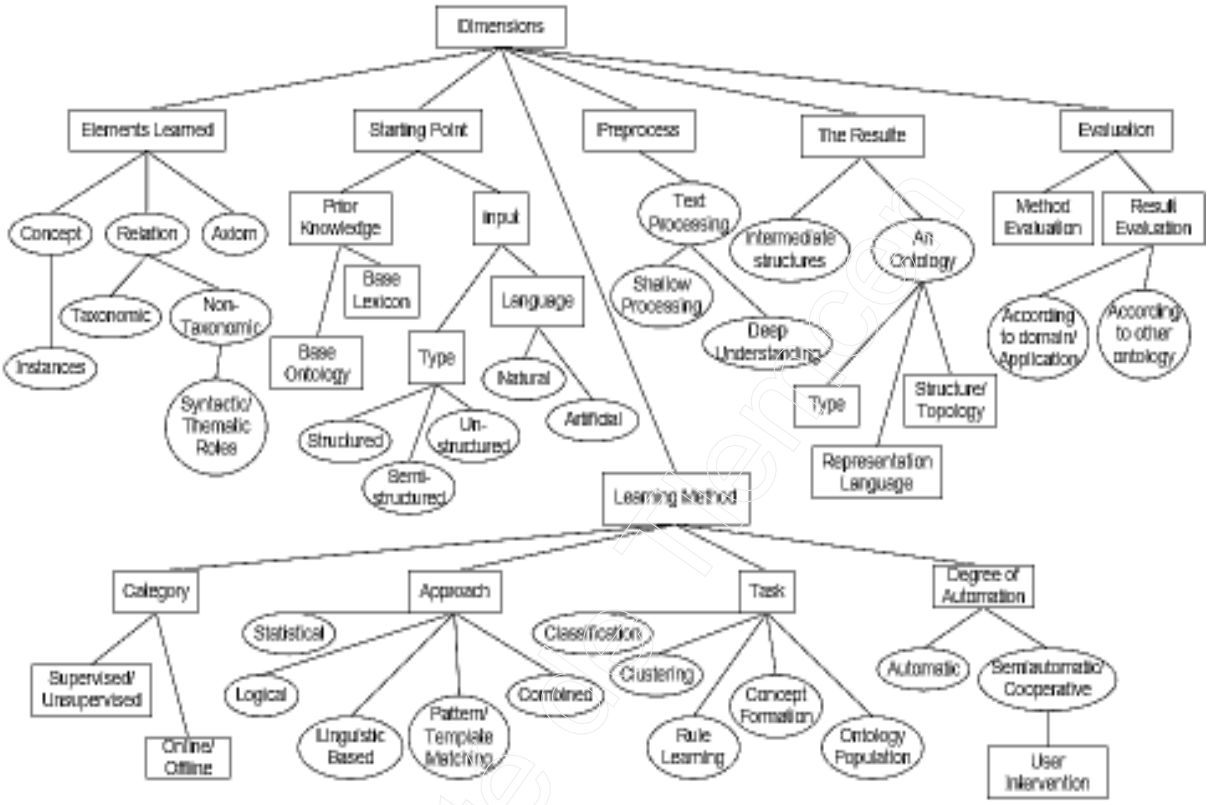


Fig. 33 : La taxonomie des six dimensions de comparaison

1. Les éléments à apprendre : Concepts, relations, axiomes, règles, instance, etc.).
2. Les sources d'apprentissages : Point de départ (textes, documents structurés, documents semi structurés, bases de données, dictionnaires, base de connaissances,…)
3. Le prétraitement : Traitement linguistique tels que la compréhension profonde ou peu profonde de traitement de texte.
4. La méthode d'apprentissage comprend :
   - Les catégories d'apprentissage : Supervisé, non supervisé {on line[1], off line[2]}
   - Les approches d'apprentissage : Statistique, symbolique, logique, basé sur la linguistique, *pattern matching*, méthodes hybrides,…

1 - L'apprentissage << on-line >> : Les exemples sont présentés les uns après les autres au fur et à mesure de leur disponibilité.
2 - L'apprentissage << off-line >> : toutes les données sont dans une base d'exemples d'apprentissage qui sont traités simultanément.

- Les tâches d'apprentissages : classification, clustering, apprentissage des règles, former des concepts, peuplement d'ontologie)
- Le degré d'automatisation : manuel, semi-automatique, coopératif, automatique.
- Le résultat : ontologie, structures intermédiaires, etc.
- Les méthodes d'évaluation : évaluation de la méthode d'apprentissage, évaluation de l'ontologie résultante.

2.1. Les éléments à apprendre

Les mots sont les principaux éléments lexicaux à apprendre. Mais comme principaux éléments ontologiques ce sont les concepts, relations et axiomes.

2.1.1. Les termes

Bien que la majorité des méthodes utilisent des sources lexicales existantes (**Text-To-Onto** [MAE00-a]; **DODDLE II** [YAM01], [KIE00], [BOR97]). D'autres systèmes soutirent par eux-mêmes la connaissance lexicale relative aux termes, comme le cas pour **SyndiKate** [HAH01] et **HASTI** [SHA02-b].

- **SyndiKate :** Utilise une hiérarchie des différentes classes des mots pour être capable de prédire la catégorie syntaxique du mot entrée, et par la suite déduire toutes les informations grammaticales qui en découlent.
- **HASTI :** Traite la phase morphologique et les catégories syntaxiques des mots avant de passer à la phase sémantique.

2.1.2. Les concepts

Un concept peut être [COR00] :

- Une définition d'un objet abstrait ou concret, élémentaire ou composé, réel ou virtuel.
- Une description d'une tâche,
- Une description d'une fonction,
- Une description d'une action,
- Une description d'une stratégie,
- D'un processus de raisonnement, etc.

Une ontologie est représentée sous forme d'une taxonomie avec les nœuds comme concepts. Ces derniers peuvent être prélevés à partir des sources d'entrées ou bien crées au cours d'un processus de raffinement via d'autres concepts.

2.1.3. Les instances

On peut trouver des systèmes qui se limitent simplement par l'enrichissement des classes de l'ontologie, cette technique est nommée « ***peuplement d'ontologie*** ». Dans cette catégorie on a le système : **WEB→KB**. [CRA00] et [SUR00]

2.1.4. Les relations entre concepts

Les relations se manifestent en deux classes : Taxonomiques et non taxonomiques.

- *Relations Taxonomiques :*

  Les ontologies sont organisées autour d'une taxonomie qui utilise les relations généralisations/spécialisations et engendre les deux type d'héritages : simple et multiple. La relation d'hyponymie « is-a » est la relation de base pour la hiérarchie. Citons des exemples de ces systèmes : **SyndiKate** [HAH01] et **HASTI** [SHA02-**b**], **DODDLE II** [YAM01], [TOD00], [AGI00], [SUR00], [HEY01], [CAR99], [DEI01], [SUN02] et [SPO02].

- *Relations Non taxonomique* :

  Ce sont toutes les relations qui excluent la relation « is-a ». On peut donc citer :

  La méronymie – la synonymie – l'antonymie – attribute-of – la possession – la causalité, ou autres. Plusieurs systèmes raisonnent avec ces types de relations :
  HASTI [SHA02-b], **Texte-to-Onto** [MAE00a], [AGI00] et [GAM02]

2.1.5. Les axiomes

Les axiomes sont utilisés pour modéliser les phrases toujours vraies. Ils sont très utiles afin de formaliser les contraintes contenues dans une ontologie, la vérification de son exactitude ou de déduire de nouvelles informations [FAR96]. Peu de système utilise l'apprentissage de part sa complexité, néanmoins le système **HASTI** apprend les axiomes dans des situations limitée, il transforme les axiomes explicites décrits à l'aide des phrases conditionnelles et quantifiées du langage naturel en des axiomes exprimés à l'aide de KIF (Knowdlge Interchange Format). Des travaux sont en cours pour étendre **HASTI** afin qu'il soit capable d'apprendre les axiomes implicites.

2.1.6. Les Méta-connaissances

Les méta-connaissances sont des connaissances ontologiques primitives qu'un système essaie d'acquérir (règles pour extraire des instances, modèles de connaissances, etc.),

pour essayer par la suite de l'exploiter dans l'extraction des connaissances ontologiques. Finkelstein et Morin [FIN99] proposent une approche pour apprendre des patrons lexico-syntaxiques pour extraire des connaissances à partir des textes. Par contre **WEB→KB** [CRA00] apprend des règles pour extraire des instances à partir des textes.

2.2. Les sources d'apprentissages

La question posée dans cette dimension est « *A partir de quoi l'ontologie va t'elle apprendre ?* ». La plupart des approches soutiennent l'idée d'acquisition à partir des connaissances déjà présentes (afin de les réutiliser) ou bien d'enrichir par de nouveaux éléments, à partir d'autres sources d'entrées (documents, Web,…). La qualité, et la quantité de la connaissance déjà existante et qui va être réutilisé, sa structure, son type, et le langage de la deuxième source d'entrée diffèrent d'un système à un autre.

2.2.1. Les sources réutilisables (Ontologie de base)

Les connaissances de base essentielles varient selon le type et le volume dans les différentes approches. Les connaissances préalables peuvent être présentées en linguistique (lexical, grammatical, modèles, etc.) ou sous forme de ressources ontologiques (l'ontologie de base). Beaucoup de projet utilise une base de connaissance lexicale (Lexicon) pour traiter des textes comme dans [KIE00], ou à des Ontologies comme Wordnet ou EuroWordNet dans **Text-To-Onto** [MAE00-b], **SyndiKate** [HAH01] et **DODDLE II** [YAM01], [WAG00], [AGI00], [TER01]. Le volume de ces sources diffère d'une approche à une autre. Le système **HASTI** [SHA02-b], démarre le processus à partir d'un noyau presque vide, dans [BRE01] à partir d'une esquisse d'ontologie ou d'un petit ensemble de mots représentant les concepts de haut niveaux [HWA99] ou bien encore d'une ontologie générique telle que **CYC** dans [LEN90].

2.2.2. Les entrées

Les sources d'entrée varient selon le type et la langue.

a. Type :

<u>*Données structurés*</u> :

- *Kashyap* extrait les connaissances à partir des schémas de base données. [KAS99]
- *Suryanto* le fait à partir d'une base de connaissances (*database schemata*). [SUR00]
- *William*, par contre, à traves une ontologie existante. [WIL00]

Mais les approches qui réutilisent WordNet couvrent la littérature.

Données Semi-structurés

C'est parce que le web est immensément riche en source d'informations, que plein de concepteurs se sont hâtés vers les documents HTML, XML et DTDs (*Documents Type Definition*) : par exemple **WEB→KB** [CRA00] et [KAV02]. Les dictionnaires aussi sont considérés comme sources d'entrées semi structuré.

Données Non structurés

Trop complexe, ce type de sources, pour extraire de la connaissance : elle peut être du texte en langage naturel comme le projet **HASTI** [SHA02-a], **SVETLAN** [HAH01] et [HEY01] ou bien on épuise à partir des textes du Web comme **Text-To-Onto** [MAE00-b] et [TOD00].

b. Langage :

Les sources d'entrées peuvent être des textes en langages naturels comme l'anglais dans **DODDLE II** [YAM01], [WAG00], [TER01], l'allemand dans **SyndiKate** [HAH01] et [HEY01], le Français dans **ASIUM** [FAU98], **SVETLAN** [CHA00] et [TOD00], le persan dans **HASTI** [SHA02-a], et aussi dans d'autres langages artificiels XML dans **Text-To-Onto** [MAE00-b] ou **RDF** dans [DEI01].

2.3. Le Prétraitement

La question posée dans ce contexte est : « Quels sont les outils à utiliser pour transformer ces entrées en une structure exploitable ? ».

Dans la catégorie des entrées textuelles, le premier traitement a fortiori est le traitement linguistique. De plus, La compréhension profonde des textes ralentisse le processus de construction de l'ontologie, mais elle permet de fournir des relations spécifiques entre les concepts, alors que les techniques peu profondes pourraient fournir des connaissances génériques sur les concepts [AGI00]. Notons que beaucoup de systèmes préfèrent les techniques du *Shallow text processing* qui engendre des techniques telles que le *tokeninzing*[1] ***P**art **O**f **S**peech tagging*[2] *(PoS)* et les analyses syntaxiques. Le système **Text-To-Onto** [MAE00-b] utilise SMES (Saarbrücken Message Extraction System) pour traiter les textes

1 - Tokeninzing : Il s'agit du processus permettant de marquer les différentes sections d'une chaîne de caractères. En effet, un ordinateur n'est pas capable seul de déterminer quels sont les mots d'une phrase ; il n'y voit qu'une chaîne de caractères. Un processus de tokenization consisterait donc à séparer ces mots, selon les espaces. [http://fr.wikipedia.org/wiki/Analyse_lexicale].

2 - En linguistique, l'**étiquetage grammatical** (*POS tagging* : *part-of-speech tagging* en anglais) est le processus qui consiste à associer aux mots d'un texte leur fonction grammaticale, grâce à leur définition et leur contexte (c'est-à-dire leur relation avec les mots adjacents dans un terme, une phrase ou un paragraphe).

allemands, **ASIUM** [FAU98] utilise *Sylex*[1] pour les textes Français, **SynDiKATe** utilise compréhension profonde pour extraire des connaissances ontologiques du texte, *InfoSleut*[2] [HWA99] fait appel un simple marqueur *Part Of speech (PoS) tagger* pour parfaire une analyse syntaxique peu profonde. Par contre HASTI [SHA02-a] utilise le système Petex qui est un traitement de texte Persan. Quand aux approches qui manipulent les Databases et les bases de connaissances, ont recours à la discipline du DATA Mining.

2.4. Les méthodes d'apprentissages

On se pose la question suivante : « Quels sont les méthodes d'extractions de connaissances ? » Comme réponses directe, on peut dire qu'il existe plusieurs méthodes selon les approches les plus simples (statistiques) aux plus complexes (logiques), comme elles peuvent être supervisées ou non supervisées. Beaucoup de systèmes diffèrent de part leurs approches méthodologiques ou par leurs tâches de réalisations. On peut dire alors que chaque approche apprend en réalisant une tâche bien précise, comme la classification, ou le *clustering*.

2.4.1. Approches d'apprentissage

Les approches de l'apprentissage ontologique peuvent être statistiques ou symboliques (basé sur la logique, la linguistique, et celles qui utilisent les patrons, chacune de ces dernières peut être combinée avec des techniques heuristiques). Les approches hybrides ne sont pas excluent, il donne un profit maximum de chacune des deux premières méthodes citées ci-dessus.

a. L'approche statistique

L'analyse statistique est appliquée sur les données extraites à partir des entrées. **Web→KB** [CRA00] utilise une analyse statistique nommée « *Bag-of-Words* » pour classer les pages web. Wagner, [WAG00] exploite une modification de l'algorithme de Li & Abe (1996) pour l'acquisition des concepts préférés dans la phase de sélection et localise le niveau

1 - **Sylex** : Un outil permettant l'affichage multilingue de mots, de phrases ou de sous phrase, avec leur contexte dans des textes déjà traduits. Cet outil doit permettre aux traducteurs de rechercher des exemples de traductions ainsi qu'aux réviseurs de vérifier la traduction d'un mot, d'une phrase ou d'une tournure de phrase dans son contexte. **[**http://www.issco.unige.ch/en/research/projects/sylex/intro.html**]**

2 - **InfoSleuth** : est un système multi-agents qui peut être configuré pour exécuter différentes activités dans le cadre de la gestion d'information dans un environnement distribué ainsi qu'un système pour la recherche coopérative d'informations dans des bases de données distribuées.
http://www.limsi.fr/~jps/enseignement/examsma/2005/2.applications/parties/Rechercheinfo.htm

de généralisation appropriée dans l'ontologie. Tandis que **Text-To-Onto** [MAE00-b], Heyer [HEY01] et **DODDLE II** [YAM01] utilise l'analyse statistique de cooccurrence des données pour apprendre des relations conceptuelles à partir de textes. [BIK99] utilise les chaîne de Markov cachées (HMM, Hidden Markov Chain) pour localiser et étiqueter les noms et les entités numériques. Notons que les approches statistiques peuvent opérer sur des mots isolés ou sur des mots dans leurs contextes.

On appelle le modèle basé sur les mots isolés : Model *bag-words* ou *unigram*. Il ignore la séquence dans laquelle le mot apparaît. Les méthodes qui traitent les mots indépendamment de leurs séquences fond appel aux règles bayésiennes, elles sont appelées *naïve bayes* comme l'approche **Web→KB** présenté par Craven et al. 2000 dans [CRA00]. Ce système classifie les documents web par la méthode naïve bayes modifiée, en construisant un modèle probabiliste pour chaque classe de document web, pour classifier ensuite chaque nouvelle page web dans la classe la plus probable à contenir les mots qui décrivent celle-ci.

La seconde classe des méthodes s'intéresse aux mots dans leurs séquences. D'une autre façon, l'identité sémantique d'un mot se reflète dans sa distribution dans des contextes différents, de sorte que le sens d'un mot est représenté en termes de mots qui lui sont co-occurrents et la fréquence des cooccurrences, c'est l'idée adoptée par ***Maedche*** [BIK99]. Quand la fréquence de deux mots ou plusieurs est élevée lors d'une construction bien définie alors celle-ci est appelée collocation. L'apprentissage par co-occurrence et collocation sont plus usités dans les méthodes statistiques.

b. L'approche logique

Plusieurs méthodes logiques sont utilisées pour extraire des connaissances à partir des entrées. Citons parmi ces approches : **ILP** (Inductive Logique Programming), le clustering basé sur FOL (First Order Logic) et l'apprentissage propositionnel basé sur la logique ; tous utilisent la déduction ou l'induction et présentent le résultat sous forme de propositions logiques de premier ordre ou d'ordre supérieur. **HASTI** [SHA02-**a**] profite de la déduction logique et des règles d'inférences pour produire de nouvelles connaissances à partir d'autres connaissances déjà présentes. Par contre le système **Web→KB** [CRA00] et [BOW00] sont basés sur l'induction des hypothèses à partir des observations (exemples) et assemble de nouvelles connaissances à partir des expériences. **ILP** (Inductive Logique Programming) se positionne au croisement de la programmation logique et l'apprentissage inductif. **FOL** est le système de ILP le plus réussi et le plus apprécié, il est repris par quelques systèmes d'apprentissage ontologique, comme **Web→KB**.

c. Les approches linguistiques

Ces approches sont beaucoup plus usitées dans la construction des ontologies à partir des textes. Parmi ces méthodes linguistiques, citons à titre d'exemple, l'analyse syntaxique d'**ASIUM** [FAU98], l'analyse morpho-syntaxique dans [ASS97], le modèle lexico-analyse syntaxique de [FIN99], le traitement sémantique de **HASTI** et la compréhension du texte utilisées par **SynDiKATe**. Toutes ces méthodes sont exploitées dans le but d'extraire des connaissances essentielles pour construire des ontologies à partir de textes en langage naturel.

Prenons un détour pour voir la méthode utilisée par ***Assadi et al***, dans [ASS97][1], il effectue une analyse morpho-syntaxique partielle pour extraire "des termes candidats" à partir de textes techniques. Ensuite l'ingénieur de connaissances, assisté par un outil de classification automatique de *clustering*, construit les champs conceptuels du domaine. Le résultat de cette analyse est un graphe constitué de syntagmes nominaux (Phrases nominales). Tout terme complexe ou composé est à son tour décomposé en deux parties : la tête et l'expansion, tous deux liées aux termes candidats complexes dans le graphe terminologique. Le graphe sera ensuite utilisé par l'analyseur conceptuel pour construire l'arbre de l'ontologie.

Quand à **ASIUM** [FAU98][2], acquiert des connaissances sémantiques à partir de textes techniques analysés syntaxiquement par le parseur *Sylex* qui fourni comme résultat des cadres syntaxique. Les adjectifs et les « mots vides » sont retirés et il n'est retenu que les « tête » des prépositions et les compléments. Un clustering conceptuel sur mots « tête » qui apparaissent dans des régularités syntaxiques. Les mots « têtes » qui apparaissent avec les <*verb*> ((<*preposition*>/<*function*>) <*headword*>) sont regroupés dans un même concept.

Exemple : **ASIUM**

« Amine voyage en bateau »

<Voyager> <Sujet> <Amine>
<en> <bateau>

Et on peut avoir :

<Voyager> <Sujet> <Bedro>
<en> <train>

<Voyager> <Sujet> <Fethi>
<en> <Voiture>

Dans l'exemple précédent **ASIUM** va créer deux *clusters* de base pour le verbe « Voyager » :

1. (Voiture, Train) est associé au verbe : « voyager » + « en » et
2. (Voiture, avion) est associé au verbe : « Conduire » + « Objet ».

SynDiKATe [HAH01], utilise les techniques de compréhension de textes pour extraire de la connaissance. Comme résultat de l'analyse syntaxique, un graphe de dépendance avec comme

1 - http://www.ldc.upenn.edu/acl/P/P97/P97-1066.pdf
2 - http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.21.2735&rank=1

nœud les mots et ses arcs les relations conceptuelles en mappant les mots avec les concepts qui leurs sont équivalent dans la base de connaissance (utilisé comme background).

HASTI [SHA02-a], exploite à la fois l'analyse morpho-syntaxique et sémantique des textes en entrée pour extraire des connaissances lexicales et ontologiques. L'analyse morpho-syntaxique prédit les caractéristiques des mots inconnus et crée des structures de phrase, qui indiquent les rôles thématiques dans la phrase. L'analyse sémantique complète les emplacements vides dans la structure de phrase et amène le processus d'extraction de connaissance conceptuelle à utiliser des modèles sémantiques.

d. Les approches basées sur les Patrons (Pattern matching approaches)

Largement utilisé dans le domaine d'extraction d'information et sont également hérités du domaine d'apprentissage ontologique. Dans ces méthodes une recherche sera effectuée sur les entrées (en général du texte) des mots clés, des patrons qui indiquent certaines relations (par exemple, *hyponymie*). On distingue une variété de patrons (*templates*): syntaxique, sémantique, à but général ou spécifique pour extraire les différents éléments d'une ontologie.

A ce titre, on cite un travail très émergeant, sur le *Pattern matching*, celui réalisé par Hearst (1992) [HER92], qui présente quelques patrons lexico-syntaxiques sous la forme d'expressions régulières pour extraire les relations d'*hyponymie* / d'*hyperonymie* à partir de textes, parmi ces patrons on dicte :

NPSuchas {*NP*,}* (*and* | *or*) *NP*

NP {, *NP*}* {,} (*or* | *and*) *NP*

NP {,} *including* {, *NP*}* (*or* | *and*) *NP*

Ainsi, grâce au troisième parton cité par exemple, les relations d'hyponymie suivante ont était détecté :

```
All common-law countries, including
Canada and England ...
```
$\Longrightarrow$ *hyponym("Canada", "common-law country"),*
hyponym("England", "common-law country")

HASTI [SHA02-a], est un autre système qui utilise des patrons lexico-syntaxique et sémantiques (templates) pour extraire à partir de textes, les relations taxonomiques et non taxinomiques comme hyponymie, méronymie, les rôles thématiques, valeurs des attributs («has-prop» relation) et d'autres relations et axiomes. Un exemple de son modèle lexico-syntaxique est le modèle de l'exception pour extraire les hyponymies :

{*all* | *every*} NP_0 *except* NP_1 {*(and* | ,) *NPi*}*… (*i>1*), *implies* (*sub-class NPi NP0*) (*i* ≥*1*)

Un autre travail est fait par **Sundblad** (2002) [SUN02] dans lequel certains patrons linguistiques sont utilisés pour extraire les relations d'hyponymie et méronymie à partir des questions d'un corpus telle que :

Who is/was X?
What is the location of X?
What is/was the X of Y?
How many X are in/on Y?

Heyer et al, (2001) [HEY01] a proposé deux patrons pour extraire les prénoms et les relations à partir des phrases. Les patrons peuvent être de nature générale, tels que ceux proposées par *Hearst*, **HASTI** *et* **Sundblad** ou spécifiques à un domaine d'application tels que ceux utilisés par *Assadi* (1999) pour extraire des connaissances à partir des textes de planification du réseau électrique. Mais d'autre part, les patrons peuvent être définie manuellement [SUN02], [GAM02] ou peut être extrait (semi) automatique comme dans **Promethee** [FIN99], **AutoSlog**-*TG* [RIL96] et **Crystal** [SOD95].

e. Les approches heuristiques

Outre les approches vues précédemment, l'approche heuristique peut être aussi utilisée. En d'autres termes les méthodes heuristiques ne sont pas indépendantes, elles sont plutôt utilisées pour appuyer et compléter les autres approches. On peut citer quelques exemples dans cet axe de recherche, l'approche **Texte-To-Onto** ; **HASTI** ; **Infosleuth** [HWA99] **et [**GAM02**].**

f. Stratégie d'apprentissage Multiples

La plupart des systèmes, qui apprennent plus d'un type d'éléments d'ontologie, combinent différentes approches. Elles appliquent de multiples stratégies d'apprentissage pour apprendre les différentes composantes de l'ontologie. Elles utilisent différents algorithmes d'apprentissage tels que **Texte-To-Onto** qui utilise des règles d'association et une analyse formelle de concepts ainsi que des techniques de clustering, **WEB → KB** combinant règle d'apprentissage de FOL avec les règles d'apprentissage bayésien, **HASTI** quand à lui, appliquant une combinaison de méthodes logiques, linguistiques basé sur des patrons heuristique et *A.Termier* et al, dans [TER01] combine le clustering statistiques et le clustering sémantique des mots et des documents.

2.4.2. Les tâches d'apprentissage

Les méthodes d'apprentissage peuvent être classées en fonction de la tâche qu'ils accomplissent. La tâche effectuée dans [SUR00] est la classification, dans **HASTI**

[SHA02-a], et **ASIUM** [FAU98] nous retrouvons le clustering, par contre dans Texte-To-Onto [MAE00-b] et dans [RIC92] l'apprentissage des règles et dans **Web→KB** [CRA00] et une analyse conceptuelle formelle dans [BRE01].

La tâche d'apprentissage peut être utilisée pour extraire des connaissances d'entrée ou pour affiner une ontologie. Ci-dessous nous allons parcourir la tâche de clustering, l'une des tâches les plus employés appliqués dans l'apprentissage des ontologies.

- ***Le clustering conceptuel***

Les méthodes de clustering (voir [BIS00]) se distinguent par quatre facteurs adoptés par *Maedeche* [MAE02] : mode du clustering, direction du clustering, mesure de similarité, et la stratégie de traitement.

→ *Le mode de clustering* **:**

Online Vs Offline : Le clustering Online effectue un regroupement au fur et à mesure on parle d'un clustering incrémentale. Par contre Offline l'effectue périodiquement.

Hiérarchique Vs Non Hiérarchique : dans le premier les clusters obtenus sont reliés par des relations hiérarchiques, mais dans le second ces relations sont absentes

Simples Vs multiple : En « Multi-clustering chaque concept peuvent être regroupé en plusieurs clusters ou en d'autres termes, dans le graphe orienté généré par le clustering, chaque nœud peut avoir de nombreux parents et/ou de nombreux enfants.

→ *La direction du clustering :*

Le clustering hiérarchique peut être fait dans l'une des directions suivante : du haut en bas, du bas en haut, ou bien Middle-out (une combinaison des deux).

→ *Les mesures de similarité :*

Les algorithmes de clustering utilisent les mesures de similarité pour calculer la distance sémantique de deux clusters (classes). Dans la littérature, deux grands types de similarité ont été abordés [EAG96] :

-« Similarité Sémantique» (appelée aussi similarité paradigmatique or substitutionnelle)

-« Semantic relatedness » (appelée aussi similarité syntagmatique)

→ *Stratégie de traitement :*

Pour calculer la similarité entre deux clusters, nous pourrons utiliser la stratégie de lien unique « signle link » dans laquelle la similitude entre deux clusters est

la similarité entre deux objets les plus proches en eux. Une autre stratégie est celle de la liaison complète « complete link » dans lequel la similarité de deux clusters est la similarité de leurs deux membres les plus dissimilaires. La troisième stratégie est la similarité moyenne du cluster « Average group » dans laquelle la similarité est la similarité moyenne entre les membres.

2.4.3. Le degré d'automatisation

La phase d'acquisition de connaissances peut être manuelle, semi-automatique ou tout simplement automatique. Comme notre problématique souligne une construction semi-automatique, nous allons donc nous pencher sur les approches d'un certain degré d'automatisation.

HASTI [SHA02-a] et [WAG00] utilise des outils d'acquisition automatique, tandis que **Text-To-Onto** [TOD00] préfère des outils semi-automatique. Mais il existe aussi des systèmes utilisant des méthodes coopératives comme **HASTI**, **ASIUM**.

Dans les systèmes semi-automatiques et de coopération, le rôle des utilisateurs varie selon la méthode adoptée. Il peut proposer une ontologie initiale, et de valider ou modifier les différentes versions proposées par le système [BRE01] ou de sélectionner des patrons dans les relations entre classes [SUR00] ou bien de contrôler les niveaux de généralité et étiqueter nouveaux concepts tel que **ASIUM**, et confirmer les décisions du système comme dans **HASTI**.

2.5. Les Résultats

Cette dimension est concernée par le résultat du processus d'apprentissage et répond à la question : « que va-ton construire et quelles sont ses fonctions ? »

On assiste à des systèmes qui construisent effectivement des ontologies, tandis que d'autres ne font qu'aider et guider l'utilisateur, un expert ou un autre système pour produire une ontologie. Autrement parlant, il existe des systèmes d'apprentissages ontologiques autonomes et d'autres qui sont simplement des modules effectuant une tâche pour aboutir à un ensemble de données intermédiaires qui sera utilisé pour construire l'ontologie. **DODDLE II** [YAM01], **SVETLAN** [CHA00] et [MOI00] sont classés dans la seconde catégorie car la structures initiales pour construire l'ontologie est déjà existante.

Le résultat peut être classé selon trois critères : le type de l'ontologie, sa structure et enfin le langage de représentation de l'ontologie.

2.6. L'évaluation

Il existe jusqu'à maintenant deux approches pour évaluer les méthodes d'apprentissages. L'évaluation des méthodes d'apprentissages : Cette tâche n'est pas assez simple et nous dirons aussi que cette tâche est non triviale, c'est pour cela qu'elle est moins prise en compte dans la littérature, car elle vise à mesurer la justesse des techniques d'apprentissage.

L'évaluation des ontologies résultantes : Ainsi, c'est la méthode la plus courante pour évaluer un système d'apprentissage ontologique. Elle consiste à évaluer (partiellement) leurs ontologies résultantes avec l'une des méthodes suivantes :

- Méthode cité dans [MAE01-b], comparer plusieurs ontologie pour un domaine.
- Méthode de comparaison d'ontologie selon les applications dans lesquelles elles sont utilisées.

En effet, plusieurs approches et systèmes proposent leur propre environnement de tests et d'évaluations, en fonction de leurs applications et le domaine choisi. La majorité des systèmes sont évalués par le calcul du *Recall* et *Precision.* Le Recall est calculé en divisant le nombre de concepts extraits valides acquis sur le nombre total des concepts existants dans d'échantillons d'entrée. Quand à la *Précision* c'est le résultat de la division du nombre de concepts extraits valides sur le nombre total des concepts extraits. Mais ces calculs ne donnent pas une vision réelle de comparaison entre les différentes approches car chacune d'elles peut utiliser des entrées de domaines différents.

Parti II

Apprentissage Ontologique

Techniques et Approches

1. Introduction

Ontologie Learning : Pouvons nous se poser la question légitime si la roue n'est pas réinventée, par la question suivante : « Est-ce que ***l'apprentissage ontologique*** n'est pas simplement une réédition des notions et des techniques déjà existantes sous un nouveau nom ? ».

La réponse est assurément « Non ». Bien que les objectifs d'acquisition[1] de connaissances et de l'apprentissage ontologique[2] (à partir du texte) sont certainement comparables. Les recherches sur les ontologies sont devenues de plus en plus répandues dans la communauté informatique. Les ontologies sont utilisées dans de nombreux domaines tels que le web sémantique, les moteurs de recherche, le traitement du langage naturel, l'ingénierie des connaissances, l'extraction et la recherche d'information, les systèmes multi-agents, le e-commerce, la modélisation qualitative des systèmes physiques, la conception de base de données, les sciences de l'information géographique et les bibliothèques numériques.

2. Classification des sources d'apprentissage

Dans la plupart des cas, il existe déjà des sources de connaissances différentes qui peuvent être incorporés dans un processus d'ingénierie ontologique. Ces sources d'information peuvent être des documents, des bases de données, des taxonomies, des sites web, des applications et d'autres choses.

La question est de savoir comment extraire les connaissances incorporées dans ces sources automatiquement, ou du moins semi-automatiques, et la reformuler dans une ontologie. Alexandre Maedche [MAE03], présente une classification des différentes approches, du domaine d'apprentissage ontologique, selon le type d'entrées : « sources d'apprentissage ».

---

1 - L'essentiel de cette technique c'est qu'elle permet l'acquisition des connaissances explicite, implicitement contenue dans les données (textuelles).

2 - Dans l'apprentissage ontologique, il existe toutefois, un certain nombre d'aspects nouveaux et innovateurs permettent de le distinguer parmi beaucoup de travaux antérieurs d'acquisition des connaissances.

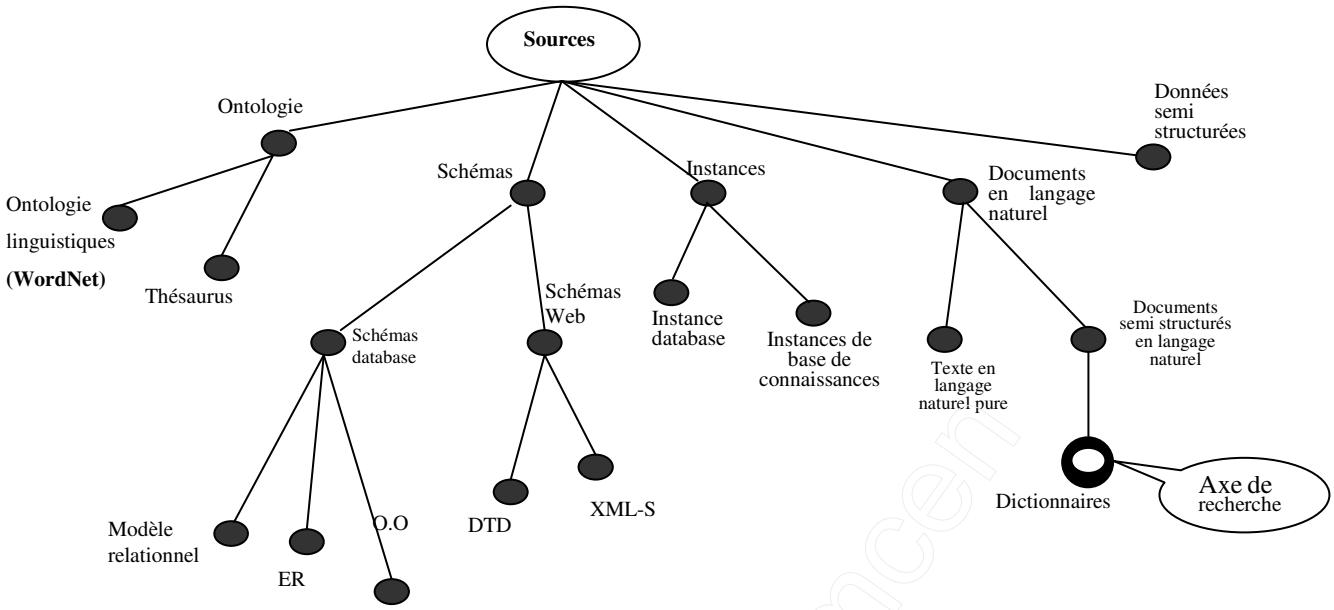


Fig. 34. Classification de *Maedche* : Sources d'apprentissages

La vision de Maedche était que chaque type de source faisait intervenir des traitements et des techniques de transformation différentes qui dictaient dès lors des réutilisations spécifiques. Nous distinguons les approches d'apprentissage à partir de textes :

- de dictionnaires [HER92], [JAN99]
- de bases de connaissances [SUR01],
- de schémas semi structurés [DEI01], [DOA00], [PAP02].
- et de schémas relationnels [JOH94], [KAS99], [RUN02].

Plusieurs techniques sont mises en jeu dans l'apprentissage d'ontologies comme les patrons lexico-syntaxiques, l'extraction basée sur les règles d'association, l'extraction basée sur le *clustering*, l'extraction basée sur le calcul des fréquences et l'extraction basée sur des techniques hybrides.

3. Un Processus d'apprentissage consensuel

D'un point de vue l'ensemble des méthodes énoncées par [RIN04], on peut distinguer six étapes suivantes dans un processus d'apprentissage d'ontologies à partir de textes (qui sont d'une certaine façon ou d'une autre, commun à la plupart des méthodes publiées) :

- Collection, sélection et prétraitement d'un corpus (textes) approprié (outils TAL).
- Découvrez les ensemble des mots (candidats-termes) et expressions équivalentes.
- Validation de l'ensemble (établir des concepts) avec l'aide d'un expert du domaine.

- Découvrir des ensembles de relations sémantiques en concepts.
- Validation des relations et extension des définitions des concepts à l'aide d'un expert du domaine.
- Créer une représentation formelle.

Il ne faut pas croire, que seulement les termes, les concepts et les relations entres eux qui sont importantes, mais aussi le sens des « *gloss* » et la formalisation (axiomes) des concepts ou des relations. Comment mener à bien ces étapes ? Une multitude de réponses peuvent être données. De nombreuses méthodes nécessitent l'intervention humaine avant que le déroulement réel du processus (étiquetage des candidats-termes - apprentissage supervisé, compilation/adaptation d'un dictionnaire sémantique ou des règles de grammaire d'un domaine,...) [RIN04].

Les méthodes non supervisées n'ont pas besoin d'étape préliminaire - cependant, ils ne donnent pas d'assez bon résultats, et le corpus peut empêcher l'utilisation de certaines techniques : par exemple, méthodes d'apprentissage automatique nécessitent un corpus suffisamment large - donc, certains auteurs utilisent l'Internet comme une source supplémentaire. Certaines méthodes nécessitent un prétraitement d'un corpus (par exemple, l'ajout de balises ou étiquette de position, l'identification de la terminaison d'une phrase, ...) indépendant de la langue. Encore une fois, il existe diverses manières d'exécuter ces tâches. Ainsi, de nombreux outils d'ingénierie linguistique ne peuvent être misent en faveur.

4. Méthodes d'extraction des termes (lexicaux)

4.1. Extraction des futurs concepts

L'extraction des termes (futur concept) est une opération pré-requise pour tout apprentissage d'ontologie à partir des textes. Elle implique des niveaux avancés de traitements linguistiques. Les concepts ne sont en général qu'un ensemble de termes. Les termes sont des mots ou suite de mots susceptible d'être retenus comme des entrées (terme, concept) dans une ontologie. Tous les nouveaux travaux convergent vers l'extraction de cette entité. On distingue les méthodes linguistiques basées sur des règles syntaxiques, les méthodes statistiques basées sur les fréquences de séquences et les méthodes hybrides.

Plusieurs modèles sont issues de ces 3 approches. Par exemple la méthode du dictionnaire qui s'appuie sur une ressource externe qui retienne les mots et expressions figées voir semi-figées susceptibles d'être rencontrées dans un texte du domaine, ils sont les plus

utilisées dans l'identification des concepts. La méthode des cooccurrences permet de créer un lexique par la répétition des formes présentes dans un texte. La méthode des segments répétés se base sur la détection de chaînes constituées de fraction fréquentes dans le même texte. La méthode des bornes travaille avec des délimiteurs. [TUR01]

4.2. Outils d'extraction

Les méthodes n'agissent pas directement sur les corpus bruts (textes) mais utilisent un « *shallow text processing* » basé sur des études de traitement des textes peu profonde (TAL), et d'analyses syntaxiques ou tout autres traitement fournissant une sortie normalisée et exploitable par des algorithmes d'apprentissage automatiques. Ces outils empruntés au TAL, sont conçus avec plusieurs éléments chacun d'eux est dédié à une tâche bien précise :

- Tockenizer : Extrait toutes les unités lexicales d'une phrase ou d'un texte.
- Lemmatiseur : PoS tagger pour identifier la classe d'une unité : Nom, Verbe,…
- Name Entity : Reconnaisseur d'entité et décider si l'entité est une personne, un matériel, une date, un horaire, un nom de société, etc.

4.2.1. Méthodes statistiques

Une méthode très répandu dans la recherche d'information (IR) est le calcul de la fréquence d'occurrence d'un terme dans un corpus ou dans un texte. Mais très vite, d'autres techniques émergent et prouvent leurs efficacités, comme la méthode issue de la recherche d'information et basée sur la mesure Tfidf « *Frequency Term Inverted Document Frequency*». [MAE03] :

- *Term Frequency* Tf (t, d): fréquence d'occurrence du terme « t » dans le document « d » ∈ D (corpus, ensemble de document).

- *Documents frequency* df (***t***) : le nombre des documents dans le corpus D dans lesquels apparaît le terme.

- Inverse *Documents frequency* idf(***t***) : idf(t) = log (|D| / df(t)), où |D| : le nombre total de documents dans un corpus D. Un mot qui apparaît dans un peu de documents possède une grande valeur au calcul de la mesure idf(*t*), à l'inverse de celle qui a une valeur haute de tf*idf est reconnue comme un terme candidat et pertinent pour le document. Alors tfidf du terme ***t*** pour un document d est :

- tfidf (t,d) = tf(t,d) * log (|D| / df(t)).

- Corpus Frequency cf(***t***) : est le nombre d'occurrence du terme « t » dans tous les documents du corpus D. C'est clair que df(***t***) ≤ cf(***t***) et ∑ tf(***t,d***) = cf(***t***).

4.2.2. Méthodes à base de dictionnaires (notre axe de recherche)

Il existe des approches qui préfèrent des ressources issues des dictionnaires comme un outil d'amorce pour repérer les termes pertinents ou acquérir directement des termes contenus dans ces dictionnaires qui constituent une mine très riche d'information lexicale et sémantique (au cas où ils existent). Il offre une stabilité pour un bon amorçage du processus d'extraction.

Un souci majeur pour une exploitation facile se situe dans leur transformation en des représentations facilement exploitable par des machines. Kiez, dans [KIE00], a présenté des travaux pour la construction d'ontologie de domaine (assurance) ainsi que Maedche et Staab dans [MAE03] pour la télécommunication.

4.3. Extraction de relation

Plusieurs ressources lexicales sont utilisées pour relever les relations sémantiques entre les concepts, on cite alors : les dictionnaires, les ontologies (existantes), les patrons syntaxiques, la notion de collocations de termes ou bien la combinaison de toutes ces ressources.

A titre d'exemple, dans les partons lexico-syntaxiques (hérités du TAL), on trouve les relations sujet-verbes, verbes-objet, ou le groupement des termes selon leurs cooccurrences avec le verbe qui permettra d'acquérir par la suite des relations sémantiques.

4.4. Relations taxonomiques :

Deux grandes approches émergent dans l'apprentissage ou l'acquisition des taxonomies [MAE03] :

- Approches moyennant le *clustering* : Basé sur les hypothèses distributionnelles, ce sont des approches statistiques (groupement des termes et calcul de similarité,…).
- Approches utilisant les patrons lexico-syntaxiques : se sont des approches symboliques pour détecter les relations d'hyponymie proposé dans [HEA92].

→ ***Clustering et les relations***

Dans la famille des méthodes de regroupement non supervisées, on distingue les méthodes agglomératives (plus proche voisin, distance maximum...) qui regroupent des clusters existants selon des mesures de similarité et des méthodes de divisions (bisection k-means).

[CIM04-b] expose un aperçu de plusieurs approches : Il commence avec les premiers travaux liés au *clustering,* citant tout d'abord les travaux de Hindle [HIN90], où les noms sont

regroupés selon leurs apparitions comme sujets ou objet de verbes similaires. Quand à Pereira [PER93], il présente une approche du « *Top-down clustering* » pour bâtir une taxonomie non étiquetée de noms (Les relations de la taxonomie non étiquetée). Par contre l'approche ittérative « bottom up of clustering » a été présenté dans [FAU98], privilégiant ainsi la fréquence des mots apparaissant dans un même contexte. Cette méthode nécessite un suivi manuelle (méthode supervisée), par conséquent elle est n'est pas privilégiée par rapport aux méthodes (semi) automatiques. Dans [BIS00], Bisson et al, fournit un outil complet assistant le concepteur dans le domaine de construction d'ontologie, en utilisant une comparaison des distances de similarités (distances sémantiques) afin d'arriver à un clustering « bottom up ». Des études assez récente dans [CIM04-a], Viz utilise une *FCA* (*Formal Concept Analysis*), analyse des concepts formelle pour grouper les concepts et d'en extraire une hiérarchie à partir des textes.

→ ***Patrons lexico-syntaxiques et les relations***

Les patrons lexico-syntaxiques fournissent une relation entre des concepts d'un domaine. Ces relations ne sont repérées que lorsque les concepts appartiennent à la même phrase. Deux axes supplémentaires se sont développés :

- Dans la littérature linguistique, des patrons relatifs aux relations hiérarchiques (hyperonymie, définition, méronymie – partie de –) ou de synonymie, ont été capitalisés avec l'espoir de pouvoir les réutiliser sur tout type de textes. L'état de l'art montre que ces patrons sont plus ou moins adéquats et doivent toujours être ajustés.

- Dans les recherches de l'extraction d'information, de nouveaux patrons sont redéfinis pour repérer des relations spécifiques au domaine étudié.

En 1992, Hearst a proposé une approche pour extraire des relations d'hyponymies à partir d'une encyclopédie scolaire « Grolier », cette méthode utilise des patrons lexico-syntaxiques manuellement capturés à partir d'un corpus. [CHA99] donne une approche pour apprendre la relation « Part of », mais ceux [VEL01] manipule des techniques heuristiques. [MOR98] développe Prométhée pour palier à la lourdeur de la méthode Hearst (confection manuelle des patrons). C'est un outil d'apprentissage automatique pour l'extraction des patrons lexico-syntaxiques relatifs à la spécification conceptuelle des relations.

Conclusion

Dans ce chapitre, nous avons fait un passage horizontal sur les différentes techniques, approches et outils de base utilisées dans la création d'une ontologie, en générale. Le point de rencontre commun à tous les systèmes étudiés est la réutilisabilité et le partage de l'ontologie.

L'extraction de connaissances ou communément parlant « apprentissage d'ontologies » a pour but la construction semi-automatique d'ontologie. Les méthodes de construction d'ontologies à partir des documents semi structuré favorisent souvent l'étude du texte, proprement dit, que ce soit selon une approche statistique, symbolique ou linguistique.

Le dernier chapitre va surtout mettre en lumière l'approche de la solution adoptée à la construction d'une ontologie lexicale en prenons l'ontologie WorNet comme modèle de travail, et en utilisant comme source d'entrée pour l'apprentissage, les données d'un dictionnaire arabe « Al ghannye ».

Chapitre 5
Approche Adoptée
Conception et Implémentation

1. Introduction

1.1. Ontologie lexicale

Les ontologies lexicales peuvent être considérées, aussi bien, comme un lexique ou comme une ontologie [GRA04] et sont significativement différents des ontologies classiques [GRU93]. Ils ne sont pas basés sur un domaine spécifique mais ils sont destinés à fournir des connaissances structurées sur les questions lexicales (mots) d'une langue en les reliant par leur sens [Wan07]. En outre, l'objectif principal d'une ontologie lexicale est de rassembler des informations lexicales et sémantiques, au lieu de stocker la connaissance de même sens [Wan07].

Princeton WordNet [FEL98] est la ressource lexico-sémantique la plus représentative pour l'anglais et aussi le modèle le plus accepté d'une ontologie lexicale. Toutefois, la création d'un WordNet, ainsi que la création de la plupart des ontologies, est typiquement manuel et implique beaucoup d'efforts de l'homme. Certains auteurs [GER08] propose la traduction Princeton WordNet à des WordNets dans d'autres langues, mais si cela pouvait convenir à plusieurs applications, un problème se pose parce que des langues différentes représentent différents milieux socio-culturels, ne perçoivent pas exactement la même partie du lexique et, même si elles avèrent être communes, plusieurs concepts sont lexicalisés différemment [GRA04].

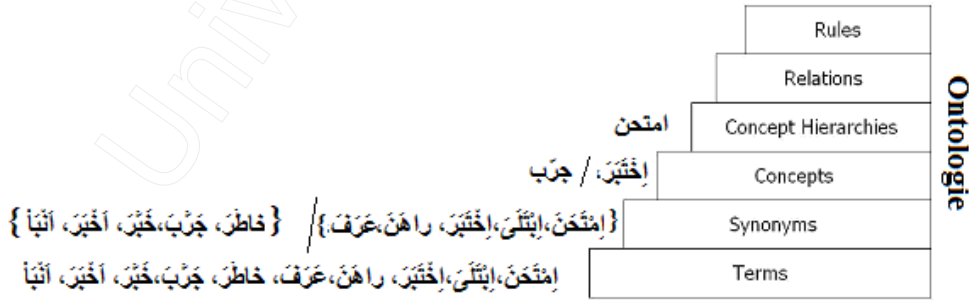


Fig. 35 : Hiérarchie visée par notre approche d'ontologie lexicale

1.2. Objectif

L'objectif global est de construire une ontologie lexicale pour la langue arabe moderne adoptant WordNet comme modèle de représentation de la sémantique lexicale. Dans

le cadre de ce mémoire, notre travail s'est limité à la découverte des concepts lexicaux ou en d'autre terme, les synsets arabe.

La construction d'une ontologie lexicale à partir de textes arabes devrait présenter les concepts relatifs à la langue arabe moderne ainsi que les différentes relations sémantiques entre ces concepts. Mais devant l'indisponibilité des outils d'étiquetage morphosyntaxique de corpus pour la langue arabe (Il n'existe actuellement que des analyseurs morphologiques) ou de corpus arabes étiquetés en libre téléchargement, on s'est penché vers la découverte automatique des Synsets en s'appuyant sur un dictionnaire arabe (**معجم الغني**)[1] en limitant le traitement rien que sur les verbes arabes dans une première perspective. La solution proposée nous permettra d'extraire des relations de synonymies ou de quasi synonymies (relation de base pour la construction des concepts) en analysant morphologiquement un verbe arabe en entrée.

2. Conception de l'approche

Notre solution est basée sur l'approche adoptée par Oliviera [OLI09] utilisée pour la construction semi automatique d'une ontologie lexicale (WordNet portugais) qui utilise dans sa première partie, une méthode s'appuyant sur la représentation en graphe des synonymes découvert dan un dictionnaire monolingue ainsi qu'une méthode de clustering (regroupement) exploitant ce graphe et qui va permettre la construction automatique des groupes de synonymes ou de quasi synonymes, chaque groupe ainsi construit sera considéré comme un Synset.

Si on se réfère à notre cadre de comparaison des systèmes de bases {chapitre 4}, nous nous situerons dans une approche similaire au système DOODLE II (2001).

Il faut noter que l'approche proposée, c'est-à-dire la construction d'ontologie lexicale « à partir de zéro» pour la langue arabe, reste un travail fastidieux et nous espérons ouvrir des portes pour approfondir cet axe de recherche.

L'approche utilisée permet la construction d'un lexicon sémantique (ou lexique sémantique) de **large couverture** constitués des verbes arabes et organisé selon la relation de la synonymie, cette structure ainsi organisé est la base pour aboutir à une ontologie lexicale. Ce travail repose sur la transformation de la connaissance lexicale basée sur les verbes arabes en connaissances basées sur les synset c'est à dire une connaissance de l'ordre conceptuel ou sémantique, cette opération est souvent appelé ontologisation.

1 - http://lexicons.sakhr.com/

Dans cette section nous décrirons les concepts qui décrivent les critères d'utilisation et l'exploitation du dit dictionnaire pour l'extraction des synonymes, que nous considérons les éléments de base pour la constitution des Synsets. Nous présenterons aussi une méthode mathématique d'analyse de graphe afin de repérer les clusters à l'intérieur du graphe analysé, appelée « processus de clustering de Markov. Cette procédure de regroupement appliqué sur un réseau de synonymie extrait depuis le dictionnaire, nous permettra d'obtenir les Synsets.

2.1 Hypothèse de base

Un dictionnaire monolingue de la langue arabe, est apprêté afin d'éclaircir la langue pour un lecteur généralement parlant l'arabe. Ce dictionnaire est un tout, constitué d'un lexique général ou spécifique à un domaine où chaque mot (entrée du dictionnaire) est décrit par une définition lexicographique.

Le dictionnaire est donc un document semi-structuré et l'exploitation de cette structure par un algorithme approprié, nous permettra de déduire des propriétés sémantiques sur le vocabulaire de la langue que décrit ce dictionnaire.

L'approche adoptée par [OLI09] est basée sur l'hypothèse que la méthode mathématique/statistique appliquée sur le graphe du dictionnaire fournit un ensemble de groupes de synonymes dont les éléments de chaque groupe sont considérés assez proche et peuvent être considéré comme formant un Synset.

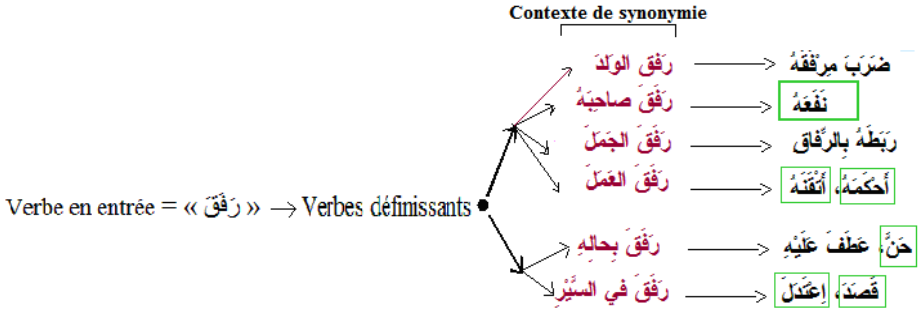
L'hypothèse selon laquelle repose notre l'approche pour la construction automatique des Synsets peut être exprimée de la façon suivante :

HYPOTHESE :

« Le regroupement en clusters à partir du graphe de synonymes d'un dictionnaire fournit un groupe de synonymes assez proches qui peut être considéré comme Synset »

Ainsi, l'algorithme d'extraction de la synonymie est basé sur un modèle exploitant la structure définitionnelle du dictionnaire source et l'analyse se fait en respectant l'hypothèse définie pour détecter et d'en extraire les synonymes. Cette hypothèse de base va nous permettre l'exploitation de la structure naturelle du Verbe-défini ↔ Verbe-définissant issus du dictionnaire afin de tirer le maximum des synonymes correspondants à une entrée.

Exemple : 2 entrées pour le verbe « رَفَقَ »



Il reste donc à définir le modèle mathématique qui va refléter les liens entre les entrées et les mots définissants et aussi le procédé de calcul (algorithme) permettant de *regrouper les définissants*.

L'hypothèse citée ci-dessus couvre toutes les catégories grammaticales d'une langue. Cependant, dans la solution proposée, cette hypothèse a été limitée aux verbes de la langue arabe (issus du dictionnaire de travail), constituants ainsi une catégorie grammaticale la plus polysémique de notre langue. Notons aussi que nous n'avons pas pris en compte tous les verbes définissants d'une entrée verbale du dictionnaire. Nous nous sommes limités aux définitions « directes » c'est-à-dire que les définitions avec des contextes ont été écartées et seule la définition « propre » a été prise en considération. Dans l'exemple précédent, pour l'entrée verbale : « رَفَقَ » on en a retenue seulement les définissants : « نَفَعَهُ », « أَحْكَمَهُ», « أَتْقَنَهُ », « حَنَّ », « قَصَدَ », « اِعْتَدَلَ »

Nous allons par la suite, expliquer comment l'approche exploite le dictionnaire et sa structure en graphe pour détecter les groupes de synonymes (Synset).

2.2. Dictionnaire, graphe des synonymes et clustering

2.2.1. Dictionnaire source

L'ensemble de dictionnaires monolingues arabes sont devenu par le temps des références dans la linguistique arabe. Excepté le critère le type du vocabulaire défini ainsi que la rigueur et la richesse des informations lexicales contenues dans le dictionnaire, un autre critère important déterminera le choix de notre dictionnaire source.

Les anciens dictionnaires arabes tels que « Lissan Al'arab » présentent la propriété d'avoir des informations lexicales (formes dérivées, genre, nombre…etc.), des définitions et des exemple(s) d'utilisation(s). Ils sont dans la plus part des cas, toutes décrites dans un seul paragraphe non segmenté ou structuré (absence de ponctuation et d'autres symboles séparateurs). Segmenter la définition vers les différents sens et localiser les

différentes composantes telles que les informations lexicales, les exemples et les citations devient difficile à réaliser, une méthodologie d'extraction d'informations spécifique doit être réalisé pour segmenter de telles définitions. للعلامة ابن المنظور–لسان العرب

@**أبأ**: قال الشيخ أَبو محمد بن بَرِّي رحِمه اللّه: الأَبَاءة لأَجَمَةِ القَصَبِ، والجمعُ أَباءٌ. قال وربما ذُكر هذا الحرف في المعتلّ من الصِّحاح وإِن الهمزةَ أَصلها ياءٌ. قال: وليس ذلك بمذهب سِيبَوَيهِ بل يحمِلها على ظاهرها حتى يقومَ دليلٌ أَنها من الواو أَو من الياءِ نحو: الرِّداء لأَنه من الرَّدْية، والكِساء لأَنه من الكُسْوة، واللّه أَعلم.
@**أتأ**: حكى أَبو علي، في التَّذكرة، عن ابن حبيب: أَتْأَةُ أُمُّ قَيْس بن ضِرار قاتل المقدام، وهي من بَكر وائل. قال: وهو من باب أَجأ(1) (1 قوله قال «وهو من باب الخ» كذا بالنسخ والذي في شرح القاموس وأنشد ياقوت في أجأ لجرير.) . قال

Fragment de texte du dictionnaire Lissan Al'arab

De ce fait, notre choix s'est finalement dirigé vers l'utilisation d'un dictionnaire monolingue arabe contemporain qui pour des raisons de clarté et simplicité de consultation par l'utilisateur, adopte une présentation structuré de la définition lexicographique. El-raid, EL-mounjid et EL-Moujiz sont des exemples de tels dictionnaire qui sont devenus des dictionnaires référence. Vu que les dictionnaires contemporains monolingues arabes cités ci-dessus ne sont pas disponibles sous format exploitable par la machine, on a préconisé l'utilisation de l'un des lexicons monolingues arabes mis en disposition en ligne par la société Sakhr[1], on y trouve donc deux dictionnaires contemporains assurant nos besoins : "El-Ghannye (الغني)", "Al Wassit (الوسيط)", chaque entrée du dictionnaire est décrite dans un fichier Html propre.

Le texte structuré utilisé dans la définition d'une entrée dans le dictionnaire choisi "El-Ghannye (الغني)", nous permettra de segmenter le texte de la définition de chacun des sens et ainsi d'extraire la définition proprement dite (sans exemple d'utilisation et sans contexte). Notre document de base peut être représenté par un graphe orienté :

$G_D = \{E, D_E\}$ où :

- E est l'ensemble des entrées verbales et l'ensemble des verbes définissants de G_D.
- V est l'ensemble des couples des verbes (d_1, d_2) tels que d_2 apparait dans la définition de d_1.

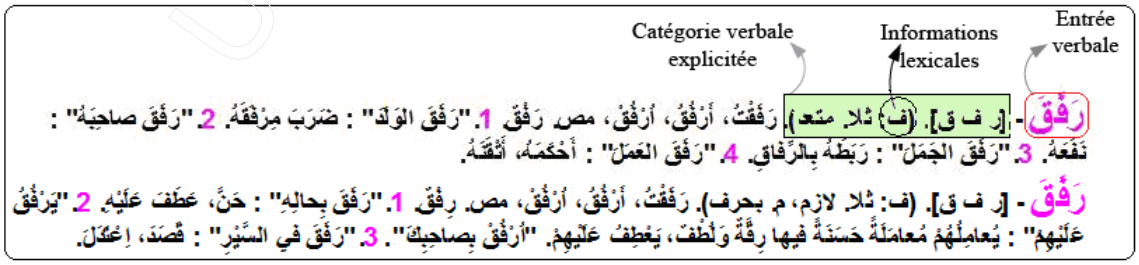


Fig. 36. Structure du dictionnaire de l'approche « El-Ghannye »

1 - http://lexicons.sakhr.com/

Dans notre étude, seules les entrées verbales du dictionnaire nous intéressent. Cette indication ainsi que d'autres informations supplémentaires sur le verbe, se trouve dans l'information grammaticale associé à l'entrée verbale. Le texte de la définition du sens d'une entrée verbale est structuré de la manière suivante :

- Entrée-verbale : [.Informations grammaticales.]
  - Sens 1- [« Contexte » :] PV [« exemple d'utilisation »]*
  . . .
  - Sens n- [« Contexte » :] Phrase-V [« exemple d'utilisation »]*

→ « Contexte » est une phrase qui contient le verbe défini, son rôle est de faire apparaitre pour un sens donné, des contraintes sémantiques sur les éléments de la phrase qui accompagnent le verbe et peut concerner donc le sujet, les prépositions ou le complément d'objet de la phrase. Ainsi, la phrase du contexte peut préciser par exemple le type du sujet ou du complément : être vivant, animé/non-animé, machine, outil, personnes ou objets particuliers…etc.

→ « Exemple d'utilisation » : est une phrase qui illustre une utilisation du verbe défini.

→ Phrase-V : représente une phrase verbale (peut inclure des lettres de conjonctions) ou seulement un verbe.

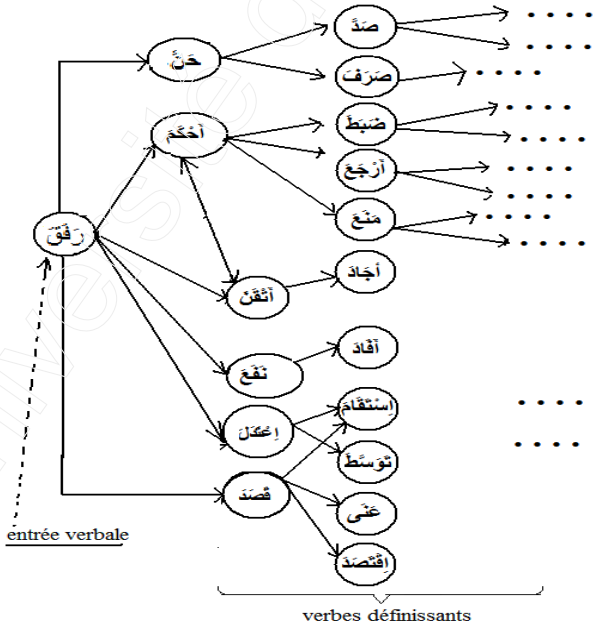


Fig. 37. Sous graphe de G_D : Relation entre un verbe entrée et ses verbes définissants

Notons qu'on peut trouver un verbe qui ne soit pas une entrée verbale d'un dictionnaire, donc toutes les entrées verbales ne correspondent pas toutes aux sommets du graphe G_D. Cependant, un algorithme a été mis au point pour extraire un certain nombre de verbes (d_2) qui apparaissent dans la définition de (d_1), selon un patron lexico-syntaxique.

2.2.2. Patrons lexico-syntaxiques

Dans l'état de l'art (voir chapitre 4), le système HASTI, 2002 [SHA02-a], utilisait des patrons lexico-syntaxique et sémantiques pour extraire, à partir de textes, les relations taxonomiques et non taxonomiques comme hyponymie, méronymie. Comme pour l'approche adoptée par [OLI09], nous nous sommes inclinés devant cette optique pour sélectionner les verbes définissants qui vont former notre graphe G_S (entrées verbales – synonymes) sous graphe de G_D. Ainsi un patron morphologique a été considéré pour une première approche, mais d'autres patrons sont envisageables pour une suite future de l'étude.

La relation de synonymie est la relation lexicale de base et la recherche des synonymes constitue la première étape nécessaire pour la construction des Synsets. L'utilisation d'un dictionnaire monolingue et des définitions lexicographiques courtes induit l'utilisation des patrons lexico-syntaxiques afin d'extraire les différentes relations lexicales et sémantique. Ainsi dans [OLI09] une liste de patrons (pattern) appliqué sur l'ensemble du dictionnaire ont permis d'extraire les synonymes.

Dans notre cas d'étude, pour la langue arabe et en utilisant le dictionnaire « el-ghannye » comme source lexicale, nous avons établi un ensemble de patrons lexicaux permettant d'extraire le plus grand nombre de synonymes à partir des définitions dans le dictionnaire. Prenons l'exemple de l'entrée verbale « **رَفَقَ** » on en a retenue rien que les verbes définissants « نَفَعَهُ », « أَحْكَمَهُ», « أَتْقَنَهُ », « حَنَّ », « قَصَدَ », « اِعْتَدَلَ» car nous prenons l'hypothèse qu' :

> **« Un verbe définissant est un synonyme proche s'il est le seul verbe de la phrase de la définition ».**

Ainsi en considérant V_E une entrée verbale du dictionnaire, le premier patron lexicale (ou morphologique) a était définit :

> **[[[[... كلمة4 ،] كلمة3 ،] كلمة2 ،] كلمة1** : V_E

كلمة1 , كلمة2, … sont les mots définissants de l'entrée verbale V_E.

Exemple : Soit le verbe en entrée : « صَفَقَ », analysant ses définitions :

Le tableau suivant donne les verbes non lemmatisés synonymes détectés grâce à l'application du patron lexical définit ci-dessus :

Contexte définition	Définitions
صَفَقَ البَابَ خَلْفَهُ	رَدَّهُ بِعُنْفٍ وَسُمِعَ صَوْتُهُ، أَغْلَقَهُ
صَفَقَتِ الرِّيحُ الأَشْجَارَ	حَرَّكَتْهَا
صَفَقَ البَيْعَ	ضَرَبَ يَدَهُ عَلَى يَدِ الآخَرِ إِعْلاَناً بِالْمُوَافَقَةِ عَلَى البَيْعِ
صَفَقَهُ مِنْ بَلَدٍ إِلَى بَلَدٍ	أَخْرَجَهُ ذُلاًّ وَقَهْراً
صَفَقَ القَوْمَ عَنْ أَمْرِهِمْ	صَرَفَهُمْ، رَدَّهُمْ
صَفَقَ الطَّائِرُ جَنَاحَيْهِ وَبِهِمَا	حَرَّكَهُمَا
صَفَقَ البابَ	رَدَّهُ
صَفَقَ العُودَ	طَرَبَ أَوْتَارَهُ
صَفَقَ القَدَحَ	مَلأَهُ

Tableau 7 : Définissants du verbe « صَفَقَ »

Remarque Importante

Notons aussi, qu'on peut constater des phrases (longues et courtes) ou une série de mots et des phrases, pour définir des entrées verbales. Alors nos patrons vont agir dans un 1er temps sur les mots seuls, exemple (voir tableau 07) :

Le verbe en entrée du dictionnaire : « صَفَقَ », possède plusieurs définitions (lignes du tableau 07) chaque ligne, à son tours, peut contenir plusieurs définitions séparées par des virgules « , », alors notre algorithme de recherche va parcourir les lignes une à une et relève toutes les définitions se trouvant à l'intérieure de chaque lignes afin d'augmenter le nombre de synonyme.

Ainsi, si nous devons calculer le nombre de verbes définissants pour l'entrée verbale « صَفَقَ », on notera, alors selon les huit (08) définitions prises par rapport à leurs contextes, on comptera sept (07) au total. Appliquons cette remarque sur les définitions du tableau 07, et prenant aussi en compte le 1er patron morphosyntaxique, de ce fait on retiendra les verbes des mots définissants encadrés dans le tableau 07 ci-dessous, soit alors : « أَغْلَقَهُ », « حَرَّكَتْهَا », « صَرَفَهُمْ», « رَدَّهُمْ », « حَرَّكَهُمَا », « رَدَّهُ », « مَلَأَهُ»

Marquons aussi, que le patron ne prend pas en compte les verbes des mots définissants qui sont constituées d'une phrase longue. Par exemple dans la définition du verbe en entrée, on a : « أَخْرَجَهُ ذُلاًّ وَقَهْراً » (ligne 4 du tableau 07), ainsi le verbe définissant de la lexie « أَخْرَجَهُ » est ignoré. Une procédure est mise au point pour extraire d'un mot son verbe afin de le mémoriser comme synonyme du verbe en entrée. Cette dernière prend en considération toutes les flexions du mot extrait du dictionnaire, elle a l'aptitude d'ignorer les verbes définissants en doubles par exemple : « حَرَّكَتْهَا » et « حَرَّكَهُمَا » ou bien « رَدَّهُمْ» et «رَدَّهُ».

- Exploitation d'autres patrons lexico-syntaxiques :

Nous exploitons d'autres patrons, afin d'enrichir le graphe des synonymes et pour couvrir une large gamme de verbes définissants, le tableau suivant résume ces patrons :

Numéro du patron	Forme du patron
2	VE : كلمة ـــ فِيهِ/فِيهَا espace 1 seul Mot
3	VE : كلمة ـــ عَنْهُ/عَنْهَا espace 1 seul Mot
4	VE : كلمة ـــ عَلَيْهِ/عَلَيْهَا espace 1 seul Mot
5	VE : كلمة ـــ إِلَيْهِ/إِلَيْهَا/إِيَّاهُ espace 1 seul Mot
6	VE : كلمة ـــ بِهِ/بِهَا/بِهِمْ espace 1 seul Mot
7	VE : كلمة ـــ مِنْهُ/مِنْهَا espace 1 seul Mot

Tableau 8 : Patrons morphologiques

Exemple : Soit le verbe en entrée : « صَفَحَ », ses définitions, selon notre dictionnaire (معجم الغني), sont représentés par le tableau ci-dessous :

Contexte	Définition
صَفَحَ عَنْهُ	أَعْرَضَ عَنْهُ
صَفَحَ عَنْ ذُنُوبِهِ	عَفَا عَنْهُ
صَفَحَ أَوْرَاقَ الكِتَابِ	قَلَّبَهَا، تَصَفَّحَهَا، أَيْ عَرَضَهَا وَرَقَةً وَرَقَةً
صَفَحَ فِي الأَمْرِ	نَظَرَ فِيهِ
صَفَحَ جَارَهُ عَنْ حَاجَتِهِ	رَدَّهُ
صَفَحَ الثَّوْبَ	جَعَلَهُ عَرِيضاً
صَفَحَهُ بِالسَّيْفِ	ضَرَبَهُ بِعَرْضِهِ لاَ بِحَدِّهِ
صَفَحَ في الأَمْرِ	نَظَرَ

Tableau 9 : Définissants du verbe « صَفَحَ »

Selon le premier patron nous retenons les mots définissants suivants : « رَدَّهُ », « نَظَرَ », « قَلَّبَهَا », « تَصَفَحَهَا » mais si on s'arrête à ce niveau beaucoup de définitions seront écartées et une grande partie des synonymes seront ignorés. C'est dans ce contexte que viennent les autres patrons (le 2ème et le 3ème patron) pour enrichir notre graphe de synonymie.

En appliquant les autres patrons on aura, en plus des quatre verbes issus des quatre mots trouvés, les verbes définissants suivants : « أَعْرَضَ », « عَفَا ».

L'exemple, suivant, nous éclaircira sans doute, sur l'utilisation de ces patrons. Soit alors l'entrée verbale « صَهَرَ », en consultant le dictionnaire nous dressons sa table des définissants suivante :

Contexte	Définition
صَهَرَ الشَّحْمَ بِالنَّارِ	أَذَابَهُ
صَهَرَ الْحَدِيدَ	أَذَابَهُ
صَهَرَهُ الْحَرُّ	أَحْرَقَهُ، اِشْتَدَّ عَلَيْهِ
صَهَرَ جِسْمَهُ بِالدُّهْنِ	دَهَنَهُ
صَهَرَهُ إِلَيْهِ	قَرَّبَهُ إِلَيْهِ، أَدْنَاهُ
صَهَرَ الْخُبْزَ	خَلَطَهُ بِالصُّهَارَةِ، أَيِ بِالشَّحْمِ

Tableau 10 : Définissants du verbe « صَهَرَ »

En analysant les définitions selon le contexte nous pourrons tirer les verbes définissants suivants selon :

- 1er Patron : « أَذَابَ », « أَحْرَقَ », « دَهَنَ », « أَدْنَى »,
- 2ème Patron : Aucun
- 3ème Patron : Aucun
- 4ème Patron : « اِشْتَدَّ »,
- 5ème Patron : « قَرَّبَ»,

Ainsi les verbes synonymes, ressortissant de ces mots, sont en nombre de six. Un cinquième exemple pour catégoriser les patrons 6 et 7 : les entrées verbales « هَادَ » et « هَابَ », en recherchant les définitions de ces verbes on obtient le tableau 11 suivant :

Contexte	Définition
هَادَ الْوَلَدَ	حَرَّكَهُ، أَخَافَهُ، أَفْزَعَهُ، أَزْعَجَهُ
هَادَ السَّائِلَ	صَرَفَهُ، صَاحَ بِهِ، زَجَرَهُ

Contexte	Définition
هَابَ مُعَلِّمَهُ مِنَ الدَّرْسِ الأَوَّلِ	أَجَلَّهُ وَعَظَّمَهُ
لاَ يَهَابُ أَحَداً	لاَيَحْذَرُ، لاَيَتَّقِي، لاَيَخَافُ أَحَداً
كَانَ يَهَابُ ظِلَّهُ	يَخَافُ مِنْهُ، كِنَايَةً عَنِ الْجُبْنِ
هَابَ الرَّاعِي بِغَنَمِهِ	صَاحَ بِهَا لِتَقِفَ

Tableau 11 : Définissants des verbes « هَادَ » et « هَابَ »

Remarquons que là aussi, si nous devons appliquer le premier patron, nous retiendrons les quatre mots définissants l'entrée verbale « هَاذَ » de la première ligne, puis les deux autres de la deuxième ligne - « زَجَرَهُ » et « صَرَفَهُ » - ensuite vient le 6ème patron pour en ajouter « صَاحَ ». Quand à la seconde entrée « هَابَ », seulement la phrase courte correspondant au 7ème patron : « يخَافُ مِنْهُ » est retenue, pour extraire le verbe définissant «خَافَ». Bien que, dans la deuxième ligne contient des mots simple, mais notre algorithme les ignorent du fait que « لاَمُ الجزم » succède la lexie candidate, suivi de l'un des caractères : {أ ن ي ت}.

Remarque : une extension de notre étude pourra augmenter le nombre de relations, en exploitant d'autres relations autre que la synonymie, dont **l'antonymie et la Troponymie**.

Exemple :

Contexte	Définition
هَاثَ الشَّيْءُ	تَحَرَّكَ
هَاثَ لَهُ	أَعْطَاهُ شَيْئاً قَلِيلاً يَسِيراً
هَاثَ الْقَوْمُ	دَخَلَ بَعْضُهُمْ فِي بَعْضٍ عِنْدَ الْخُصُومَةِ
هَاثَ فِي الأَمْرِ	عَاثَ، أَفْسَدَ فِيهِ بِعُنْفٍ
هَاثَ التُّرَابَ بِرِجْلِهِ	نَبَشَهُ

Tableau 12 : Définissants du verbe « هَاثَ »

Nous remarquons dans la quatrième ligne de définition de l'entrée verbale en question : **« هَاثَ »,** et plus exactement la phrase définissante : « **أَفْسَدَ فِيهِ بِعُنْفٍ** » peut nous décrire une « manière de », du verbe « **أَفْسَدَ** » et ceci peut conclure à considérer que : **« هَاثَ »** est Troponyme de « **أَفْسَدَ** ».

2.2.3. Sous graphe de synonymies

La plupart des méthodes proposées pour extraire les relations à partir du texte ont des triplets à base de termes en sortie. Un tel **triplet, [Terme1, Terme2, Relation],** indique qu'un sens possible de terme1 est lié à un sens possible de terme2 par le biais d'une relation. Bien qu'il soit possible de créer un réseau lexical à partir de ce dernier, ce type de réseaux est souvent peu pratique pour les applications informatiques, telles que celles qui traitent de l'inférence. Par exemple, en appliquant une règle simple transitive,

$$A\ Synonym\_of\ B\ \wedge\ B\ Synonym\_of\ C\ \Rightarrow A\ Synonym\_Of\ C$$

Sur un ensemble de triplets à base de termes, cela peut mener à de sérieuses contradictions, cela se produit parce que le langage naturel est ambigu, surtout quand on traite une large couverture de connaissances. [OLI09].

Comme déjà mentionné plus haut, notre algorithme recherche les synonyme directs d'une entrée verbale du dictionnaire « **معجم الغني** ». Ensuite, les verbes définissants deviennent des entrées et à leurs tours on recherche chacun de ces verbes définissants, et ainsi de suite… en prenant une seul fois le verbe trouvé afin de pouvoir s'arrêter dans les recherches. Une fois terminé, un sous graphe est tracé et une matrice d'adjacence est établit.

Prenons un simple exemple pour illuminer notre compréhension, soit l'entrée verbale « **صَفْصَفَ** ». Le tableau 13 de définition est le suivant :

Contexte	Définition
صَفْصَفَ الرَّجُلُ	صَارَ فِي الفَلاَةِ مُنْفَرِداً
صَفْصَفَ الطَّائِرُ	زَقْزَقَ

Tableau 13 : Définissants du verbe « صَفْصَفَ »

Nous remarquons qu'ici, il y a un seul mot définissant selon nos patrons : « **زَقْزَقَ** » et que ce même verbe, en recherchant ces définissants on obtient le tableau 14 de définition :

Contexte	Définition
زَقْزَقَ الطَّائِرُ	تَفَرَّدَ بِصَوْتِهِ
زَقْزَقَ الطَّائِرُ صِغارَهُ	زَقَّ، أَطْعَمَهُمْ
زَقْزَقَ الطِّفْلَ	جَعَلَهُ يَرْقُصُ

Tableau 14 : Définissants du verbe « زَقْزَقَ »

Les verbes extraits sont « **زَقَّ** », « **أَطْعَمَ** ». Nous continuons à rechercher les synonymes des deux verbes précédemment trouvée. Le premier verbe « **زَقَّ** », (voir tableau 15), n'a pas de définissant selon nos patrons :

Contexte	Définition
زَقَّ الطَّائِرُ صِغارَهُ	أَطْعَمَهُمْ بِمِنْقارِهِ، بِفَمِهِ
زَقَّ الذَّبِيحَةَ	سَلَخَها مِن قِبَلِ رَأْسِها إِلَى رِجْلِها

Tableau 15 : Définissants du verbe « زَقَّ »

Par contre, le second verbe « **أَطْعَمَ** », en possède « **قَدَّمَ** », « **قَاتَ** », « **غَذَّى** », « **طَعَّمَ** » et « **رَزَقَ** » (voir tableau 16) :

Contexte	Définition
أَطْعَمَهُ خُبْزاً أَوْ بَيْضاً	قَدَّمَ لَهُ
يُطْعِمُ الجِياعَ	يَقوتُهُمْ، يُغَذِّيهِمْ، يُقَدِّمُ لَهُمْ طَعاماً
أَطْعَمَ الغُصْنَ بِغُصْنٍ آخَرَ مِنْ غَيْرِ شَجَرَتِهِ	طَعَّمَهُ، وَصَلَهُ بِهِ لِيَتَكَوَّنَ مِنَ الغُصْنَيْنِ غُصْنٌ ثالِثٌ يُعْطي ثَمَراً آخَرَ
أَطْعَمَ الأَكْلُ	صارَ لَهُ طَعْمٌ
أَطْعَمَ الشَّجَرُ	أَدْرَكَ ثَمَرُهُ وَطَابَ
أَطْعَمَهُ اللَّهُ	رَزَقَهُ

Tableau 15 : Définissants du verbe « أَطْعَمَ »

et ainsi de suite jusqu'à épuisement des définitions de chaque verbe trouvé. De ce fait, on obtient un sous graphe orienté dont le sommet est l'entrée verbale et les arcs représentent la relation de synonymie. Voilà une partie du sous graphe engendré par l'entrée verbale « صَفْصَفَ» :

Notons par Gs le graphe global de synonymie qui est un sous graphe de G_D dont on grade seulement la relation de synonymie entre les nœuds. Marquons aussi, que dans le graphe de synonymie global Gs (issu du graphe G_D par l'application des patrons), il existe un certain nombre de sous graphe de synonymie.

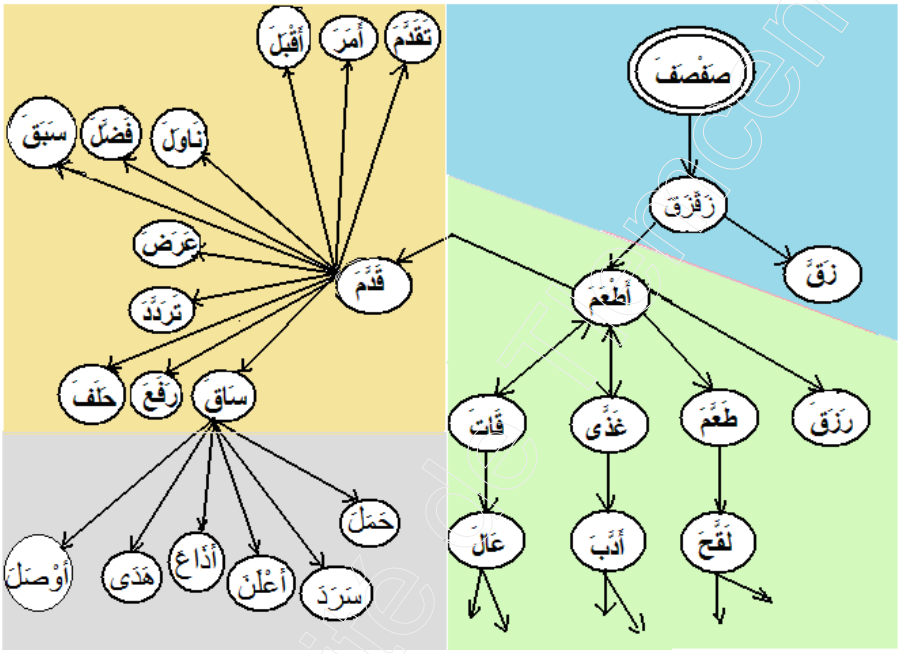
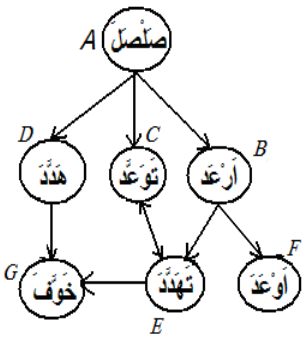


Fig. 38 : Graphe G_S du Verbe « صَفْصَفَ »

Chaque entrée verbale du dictionnaire est en relation avec les verbes synonymes présents dans sa définition, l'ensemble des couplets [entrée verbale – Verbe synonymie] représentant cette relation, et qu'on a obtenue par l'application des patrons morpho syntaxique définie sur l'intégralité du dictionnaire, vont former le graphe global de la synonymie du dictionnaire « El ghannye ».

2.2.4. Sous graphe de synonymie et matrice d'adjacence



La définition d'un graphe comme relation permet sa représentation formelle et mathématique par une matrice. Le coefficient $m_{i,j}$ désigne le nombre d'arcs d'origine « i » et d'extrémité « j ».

Un graphe orienté à n sommets peut être représenté par une matrice n × n notée encore M telle que :

M [i, j] = 1 {s'il existe un arc de « i » vers « j » et 0 sinon}.

Exemple : Pour le réseau de synonymie suivant de l'entrée verbale « صَلْصَلَ » :

On peut tracer sa matrice d'adjacence M_{7x7} comme suit :

Commençons d'abord par tracer le vecteur (Vs) des verbes entrant dans la construction du graphe de synonymie en prenant l'ensemble des verbes (une seule fois) représentant les lignes, éventuellement les colonnes de notre matrice :

Vs =

A	*B*	*C*	*D*	*E*	*F*	*G*
صَلْصَلَ	أَرْعَدَ	تَوَعَّدَ	هَدَّدَ	تَهَدَّدَ	أَوْعَدَ	خَوَّفَ

$$M_{7x7} = \begin{bmatrix} & A & B & C & D & E & F & G \\ A & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ B & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ C & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ D & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ E & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ F & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ G & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Une fois la matrice d'adjacence extraite, l'étape déterminante de construction des concepts entre en application – Le clustering –. Cette étape a été initialement définie après avoir constaté des éléments séparés dans le réseau de synonymie extrait à partir du dictionnaire, que l'on suppose, avait une structure en clusters apparemment convenable pour identifier des synsets. Ceci a été aussi remarqué par [GFE05] qui a utilisé l'algorithme de clustering de Markov (MCL) [DON00] pour trouver des clusters dans un réseau de synonymie.

Par conséquent, puisque MCL avait déjà été appliquée à des problèmes similaire que le nôtre ([GFE05], [DOR06]), il semble convenir à notre objectif – Il serait non seulement possible d'organiser un réseau basé sur les termes d'un dictionnaire, mais aussi, si le réseau obtenu, provient de plusieurs ressources, le regroupement aurait homogénéisé la représentation de synonymie.

Notons que dan un graphe, une simple transitivité telle que x synonyme y et y synonyme z alors z synonyme de x, s'appelle la ***fermeture transitive***. Cette dernière nous permettra d'élargir notre sous graphe de synonymie en graphe couvrant ainsi notre graphe G_D et d'appliquer ensuite l'algorithme de regroupement MCL.

2.2.5. Markov Cluster aLgorithm (MCL) & Clustering

Le graphe global de synonymie d'un dictionnaire monolingue est généralement constitué de sous-graphes séparés c'est-à-dire, non connectés. L'algorithme de clustering qu'on présentera n'opère que sur la matrice d'adjacence du graphe de synonymie, il est alors très utile d'appliquer cet algorithme à chaque sous graphe séparée, ce qui permet de réduire la taille de la matrice d'adjacence et d'éviter de lancer des calculs matriciels sur une seule matrice trop large qui peut alourdir considérablement le temps de calcul et nécessitant du coup un traitement parallèle non nécessaire.

MCL trouve les groupes en simulant des chemins aléatoires au sein d'un graphe en comptant alternativement les chemins aléatoires de longueur plus importante, et en augmentant les probabilités des chemins d'intra-cluster. MCL peut être décrit brièvement en cinq étapes (plus de détails dans la section qui suit) :

(i) Prendre la matrice d'adjacence A du graphe,

(ii) Normaliser chaque colonne de A à 1 afin d'obtenir une matrice S stochastique,

(iii) Calculer S^2,

(iv) Prendre « e » puissance de chaque élément de S^2 puis normaliser chaque colonne à A,

(v) Revenir à (ii) jusqu'à ce que le MCL converge vers une matrice idempotente - étapes (ii) et (iii).

Étant donné que MCL est un puissant algorithme de clustering, il attribue chaque terme un seul cluster en éliminant ainsi toutes ambiguïtés. Pour faire face à cela, Gfeller et al. (2005) proposent une extension de MCL pour trouver des nœuds instables dans un graphe, qui les dénotent souvent par « mots ambigus ». Ceci est fait en ajoutant aléatoirement du bruit stochastique λ pour les entrées non nulles de la matrice d'adjacence puis en exécutant MCL avec du bruit à plusieurs reprises. En observant les groupes obtenus à chaque exécution, une nouvelle matrice peut être remplie, qui sera basée sur la probabilité que chaque paire de mots appartiennent à un même cluster.

Nous suivons la même procédure mais avec de légères différences. Tout d'abord, nous avons observé que, pour le réseau utilisé, les groupes obtenus sont plus proche des résultats souhaités si : $-0.5 < \lambda < 0.5$. En plus, dans la première étape du MCL, nous utilisons la fréquence-pondérée de la matrices d'adjacence F, où chaque élément Fij correspond au nombre d'instances de synonymie existantes entre i et j.

Bien qu'en utilisant un seule dictionnaire, chaque instance de synonymie sera extraite tout au plus deux fois (a synonyme de b et b synonyme de a). Si par contre, plusieurs ressources sont utilisées, MCL permettra de renforcer la probabilité que deux mots apparaissant fréquemment comme synonymes sont regroupés dans un même cluster.

- ***Clustering :*** Par conséquent la phase de clustering comporte les étapes suivantes :

(i) Diviser le réseau d'origine en sous-réseaux, de sorte qu'il n'y aura pas de chemin entre deux éléments dans les différents sous-réseaux, et calculer la fréquence pondérée de la matrice d'adjacence F de chaque sous-réseau ;

(ii) Ajouter du bruit stochastique à chaque entrée de F, $Fij = Fij + Fij * \beta$;

(iii) Exécuter MCL, avec **r** = 1,6 sur F pendant 30 itérations ;

(iv) L'utilisation du Hard-clustering obtenues par chacune des 30 exécutions permet de créer une nouvelle matrice P des probabilités de chaque paire de mots en F appartenant au même cluster ;

(v) Créer des clusters basés sur la nouvelle matrice P avec un seuil donné $\alpha = 0.2$, Si $Pij>\alpha$, i et j appartiennent au même cluster ; afin de nettoyer les résultats, éliminer :

(a) Les groupes assez grands, B, s'il y a un groupe de clusters C = C1, C2, … Cn tel que $B = C1 \cup C2 \cup \ldots Cn$;

(b) Les groupes complètement inclus dans les autres groupes.

En appliquant cette procédure au réseau de la figure 38 ci-dessus, on obtient les quatre clusters ainsi représentés sur la figure.

2.2.6. Regroupement (clustering) en synsets

Une fois le regroupement réalisé par l'algorithme MCL, nous allons chercher comment extraire les Synsets à partir des clusters obtenus.

Notre dictionnaire D, et un réseau sémantique N, basé sur les termes où chacune de ses arêtes dénote une relation sémantique R, localisant la signification des termes entre deux nœuds. En utilisant D et N, cette étape va tenter de mapper d'un triplet basé sur les termes vers un triplet basé sur les synsets. En d'autres termes, d'attribuer à chaque terme, « a » et « b », d'un triplet (a R b) $\in$ N, un synset approprié. Le résultat est une base de connaissances organisé comme un Wordnet. Afin d'assigner un terme « a » à un synset A, « b » est fixé et tous les synsets contenant a, $Sa \subset D$, sont collectées.

Si « a » n'est pas dans le Dictionnaire D, il est affecté à un nouvel synset A = (a). Sinon tous les synset $Sa_i \in Sa$, avec n_{ai} est le nombre de termes [$t \in Sa_i$], déterminés par le triplet à base de termes (t R b).

Alors $Pa_i = \frac{n_{ai}}{Sa_i}$ est calculé.

Note :

Si R est une relation transitive, la procédure peut profiter d'un niveau de transitivité pour l'appliquer au réseau : x R y ^ y R z → x R z. Cependant, puisque les relations concernent seulement les termes, certains triplets obtenus pourraient être incorrectes même s'ils peuvent être utilisés pour aider à la sélection d'un synset convenable, ils ne devraient pas être mappés aux mêmes synsets.
Finalement, tous les synsets avec la plus haute valeur Pa_i sont ajoutés au cluster *C*, ainsi :

- si |*C*| = 1 alors le terme « a » est attribué à l'unique synset dans C,
- si |*C*| > 1 alors soit *C'* : l'ensemble des éléments de *C* avec la plus haute valeur de n_a et, si |*C'*| = 1, alors le terme « a » est attribué au synset dans *C'*, sauf si $Pa_i \langle 4$.
- Si elle n'est pas possible d'affecter un synset au terme « a », il demeurera non affecté. Le terme « b », quant à lui, est attribué à un synset, en utilisant cette même procédure, mais en lui fixant le terme « a ».

Ainsi les clusters trouvés seront désignés par leurs indices dans le tableau T. Si V est le vecteur des verbes du sous graphe (il faut respecter l'ordre des verbes), on reconstruit les clusters trouvés en remplaçant les indices stockés dans i par leur Nom correspondants à partir de V (**V[t[i][j]]**)

2.2.7. Détails sur l'algorithme MCL

Entrées

M[N,N] avec des 0 et des 1.

Fixer le paramètre de force (Power) : **e** = 2 (par exemple).

Fixer le paramètre d'inflation **r** : 1.6 (par exemple).

Sortie

S[N,N] une matrice constitué de probabilités (score entre 0 et 1) :

- Quelques éléments non nuls et une majorité de 0
- Sur chaque ligne les éléments non nuls constituent un cluster

Début

1) Normaliser les scores de chaque colonne la matrice M[N,N] => M devient matrice stochastique stocker le résultat dans la matrice S :

$S(i,j) = M(i,j) \Big/ \sum_{K=1}^{K=n} M(k,j)$ (Diviser le score M[i,j] par le total des score de la colonne **j**).

2) Faire la multiplication matricielle suivante (calculer S^2), si e=3 faire $S3=S^3=S*S*S$.

S2=S(i,j)*S(i,j). Stocker le résultat dans S2.

3) Chaque élément de S2(i,j) de S2, le remplacer par S2(i,j) $^{1.6}$ (si r=1.6). (élever à la puissance **r** chaque élément de la matrice) puis **normaliser** S2 comme dans l'étape 1.

4) Comparer S et S2 : Si convergence (égalité) alors arrêté. Sinon faire S=S2 et aller à 2). (Refaire les calculs)

5) Interpréter le résultat après convergence => Construire les clusters à partir de la matrice calculée.

Fin.

Détection des clusters à partir de la matrice

Analyser chaque ligne -> pour chaque ligne non nulle :

➔ Mettre les éléments non nuls de la ligne dans le même cluster.

L'exemple ci-dessous, présente une matrice de clustering 12x12, il y'a seulement quatre ligne non nul, il en résulte quarte clusters :

$$M_{mcl}^{\infty} = \begin{pmatrix} 1.000 & -- & -- & -- & -- & 1.000 & 1.000 & -- & -- & 1.000 & -- & -- \\ -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\ -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\ -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\ -- & 1.000 & 1.000 & -- & 1.000 & -- & -- & -- & -- & -- & -- & -- \\ -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\ -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\ -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\ -- & -- & -- & 0.500 & -- & -- & -- & 0.500 & 0.500 & -- & 0.500 & 0.500 \\ -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\ -- & -- & -- & 0.500 & -- & -- & -- & 0.500 & 0.500 & -- & 0.500 & 0.500 \\ -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \end{pmatrix}$$

1. Le cluster {1, 6, 7, 10}
2. Le cluster {2, 3, 5}
3. Le cluster {4, 8, 9, 11, 12}
4. Le cluster {4, 8, 9, 11, 12} déjà extrait.

3. Implémentation

Dans cette section, nous allons présenter la plate de forme de développement et l'ensemble des ressources intégré que nous avons utilisé pour la réalisation de la solution proposé.

3.1 Plateforme de développement

Environnement de développement : L'application a était réalisée avec le IDE Visual Studio 6.0 de Microsoft et était écrite en langage Visual Basic 6. Le Système de gestion de base de données – SGBD – de notre base de donnés lexicale a était réalisé avec le serveur de base de donnée Microsoft SQL Server 2008 Express Edition. La plate forme est donc exclusivement Microsoft, ce choix est justifié de part la possession de la licence VB6, la gratuité de certaines éditions – SQL server Express 2008 (mappage possible en XML), la facilité et l'intuitive de l'utilisation des langages et des interfaces proposés.

3.2 Ressources utilisés

3.2.1. Outils linguistiques.

Nous n'avons pas utilisé d'outils linguistiques ou autres (outils non disponibles gratuitement), mais nous avons confectionné notre petit outil morphologique que nous avons appelé « Extracteur ». Il est intégré dans la solution proposé, ce dernier a permit grâce aux patrons morphologiques d'extraire des verbes à partir des termes définissants de notre dictionnaire.

3.2.2. Ressources lexicales.

Le dictionnaire est disponible en consultation en ligne depuis le site de la société Sakhr[1]. Le dictionnaire analysé « Al ghannye » est constitué de prés de 30.000 fichiers Html, la définition de chaque entrée est décrite dans un fichier Html propre. Notons que la transformation de ces fichiers Html en base de données SQL Server a été réalisée par une équipe de sidi Bel-Abbes dans le projet « KalimNet ».

Le choix s'est porté sur ce dictionnaire parce qu'il adopte une structure claire dans la description des entrées ce qui nous a simplifié l'opération d'extraction des différentes entités qui composent la définition. Une autre faculté, dans l'utilisation du dictionnaire « Al ghannye », est l'information que décrit ces entrées : on peut discerner entre les verbes et les autres entrées, ceci nous a permit d'extraire avec certitude toutes les entrées verbales. L'absence de cette information nous aurait obligés de passer par une analyse morphologique non exclu d'ambigüité pour la déduction des entrées verbales.

1 - http://lexicons.sakhr.com/

3.3. Architecture générale de l'application

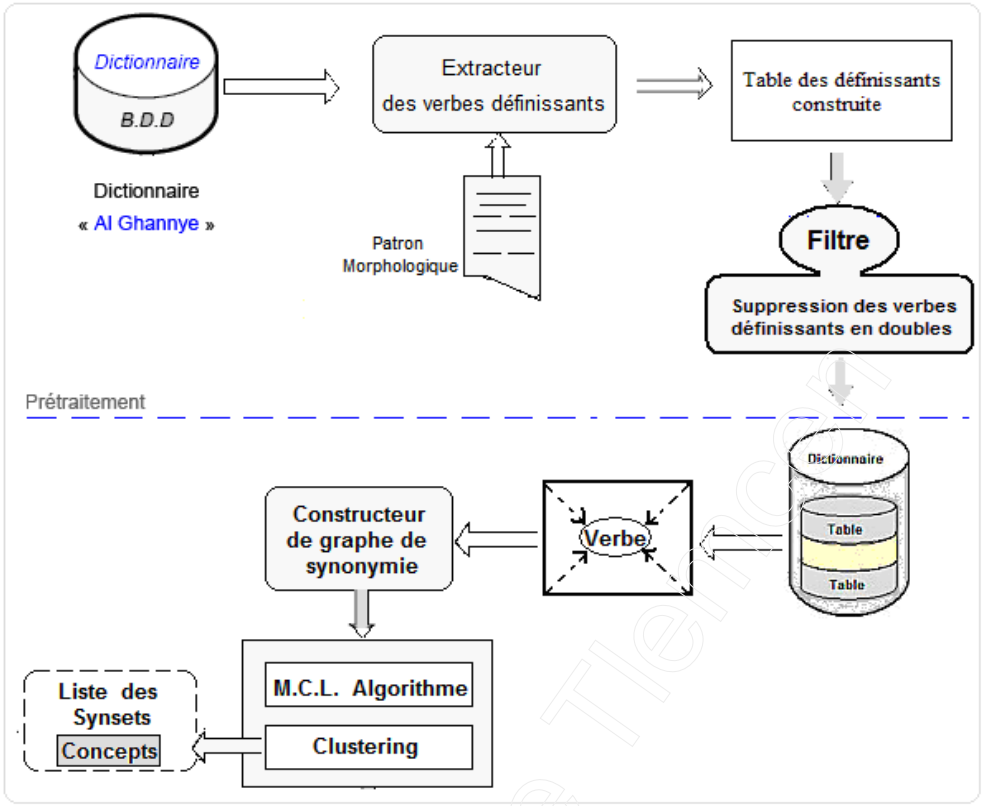


Fig. 39 : Architecture de la solution

La figure 39 montre les différentes étapes de traitement et l'interaction des ressources dans le procédé du calcul. Ainsi un extracteur de verbes définissants basée sur l'utilisation des patrons morphologiques présentés précédemment est appliqué au dictionnaire d'entrée pour obtenir la table « dbo.Tverbe ». Cette étape est immédiatement suivie d'une opération de nettoyage qui consiste à la suppression des verbes définissants en doubles dans la table « dbo.Tverbe ». D'autres procédures sont appliquées pour nettoyer et préparer la table à la phase de traitement. La base de données ainsi récupérée contient la table des verbes sans doubles appelée « dbo.VecteurSansDouble » (figure 40).

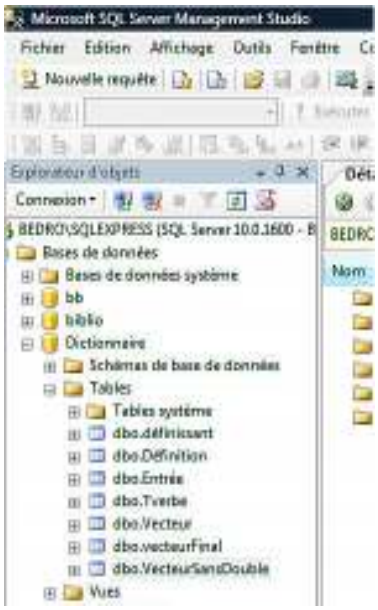


Fig. 40 Bases de données et tables utilisées

Soulignons que cette phase préparatoire est très importante pour la collecte de l'ensemble des verbes définissants que l'on considère la ressource de base pour notre travail. Après le peuplement complet de notre base lexicale effectué lors de la première phase du

traitement, la construction de graphe de synonymie d'une entrée verbale nécessaire pour l'algorithme de clustering de Markov, devient possible et ceci, grâce aux liens (Entrée verbale, Verbes définissants) décrits par la table « dbo.vecteurFinal ».

Voici quelques propriétés statistiques sur la base de données lexicales obtenue après la phase de prétraitement :

- Table Entrée
  - 27.799 entrés (mots).
  - 10.003 entrés verbales (verbes). 34,73%
- Table des définissants obtenue « VecteurSansDouble » : 21.597 définitions au total réparties comme suit :

Entrée Verbale	Définissants marqués comme entrée verbale	Définissants **Non** marqués comme entrée verbale	Total des verbes
6603	14561	433	21597
30.57%	67.42%	2.01%	100%

Tableau 16 : Distribution des entrée verbales dans le dictionnaire «Al ghannye »

3.4 Interfaces de recherches

L'interface du logiciel de recherche se compose de trois sous interfaces :

La première permet d'initier la recherche des synonymes d'une entrée verbale du dictionnaire (ou choisie dans une liste ou saisie par clavier) et la construction du graphe.

La deuxième est utilisé pour montrer l'étape de nettoyage et la dernière montre les différentes étapes de calcul effectué dans l'algorithme MCL et les clusters obtenus de la recherche demandée. Notons qu'une fois que notre matrice d'adjacence établie, l'algorithme de Markov – MCL – entre dans son application, cette étape de calcul est suivie par une procédure de clustering afin de déduire les groupes de synonymes proches qu'on considère chacun comme formant un synset.

3.4.1. Interfaces de la phase de prétraitement

La figure 41, illustre l'entrée de notre application, c'est-à-dire la première étape du prétraitement qui va engendrer tous les traitements sous-adjacents. Cette étape consiste à préparer la table des entrées verbales avec leurs définissants. Chaque définissant, à son tour, est marqué lorsqu'il est identifié comme une entrée verbale. Toute la procédure

prend un certain temps (≈ 1 journée) de calcul : un parcours des 21597 verbes de notre base de données est nécessaire.

Fig. 41 : Interface de prétraitement : Etape 1

Une deuxième étape de prétraitement est exécutée afin de nettoyer la table « dbo.Tverbe », générée de l'étape précédente, des définissants en doubles et d'identifier (par la clé de la table) chaque définissants comme entrée verbale ou non.

La troisième étape de préparation consiste à rechercher l'identifiant d'un verbe définissant et le classer. Cette partie est essentielle pour construire la table « VecteurSansDouble », laquelle repose tout notre traitement.

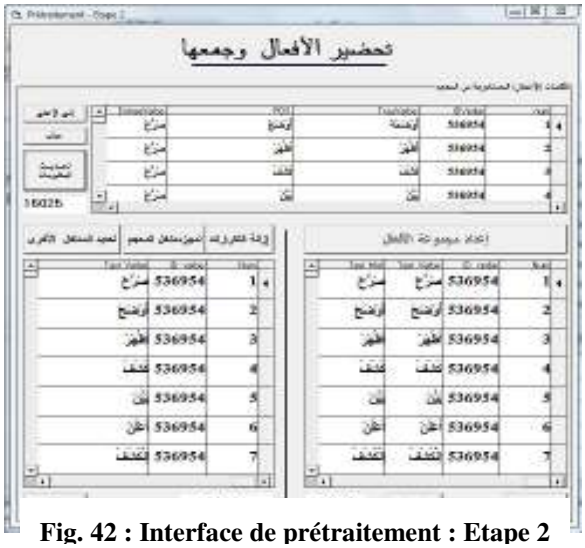


Fig. 42 : Interface de prétraitement : Etape 2

3.4.2. Interfaces de la phase de traitement

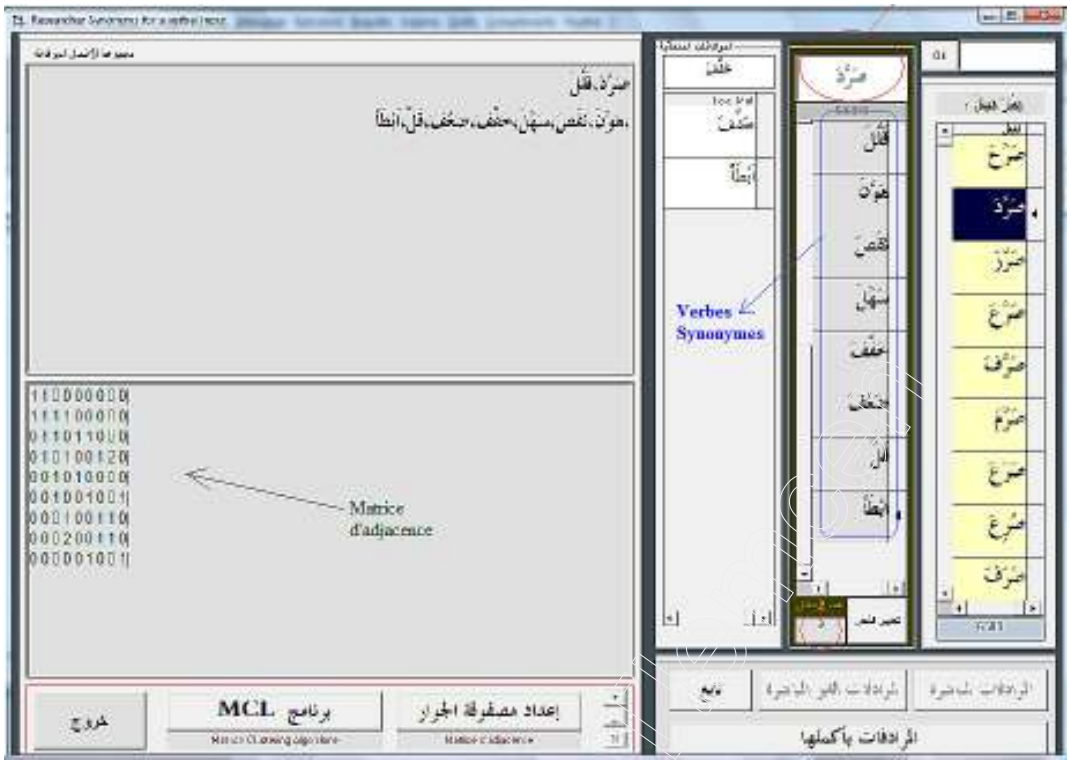


Fig. 43 : Interface de traitement : Etape 1

La construction du graphe de synonymie G_D et l'agencement du vecteur globale « VecteurSansDouble » comprenant toutes les entrées verbales ainsi que leurs synonymes, demandent beaucoup de temps de calcul (plus de 4 heures pour une sa préparation – Non Stop). Nous avons jugé utile de stocker ce dernier vecteur dans une table de notre dictionnaire.

A la fin dans cette première partie, un second vecteur est constitué, regroupant l'ensemble des verbes définissants d'une entrée verbale choisie. L'exemple de la figure 43 de l'entrée verbale : « **صَرَّدَ** » donne le vecteur final suivant :

صَرَّدَ	قَلَّلَ	هَوَّنَ	نَقَّصَ	سَهَّلَ	خَفَّفَ	ضَعُفَ	قَلَّ	أَبْطَأَ	نَدَرَ	تَضاءَلَ	تَأَخَّرَ	سَقَطَ	جَرَّبَ	…

Ce même vecteur est pris horizontalement et verticalement pour ensuite construire la matrice d'adjacence, l'objet de travail de l'étape 2 de notre traitement. Dans cette deuxième étape l'algorithme de clustering de Markov entre en application pour générer les groupes de verbes qui présentent un maximum de rapprochement selon la relation de la synonymie. Nous avons mis au point un programme MCLnxn (pour n verbes définissants) pour mettre en transparence cette étape de programmation :

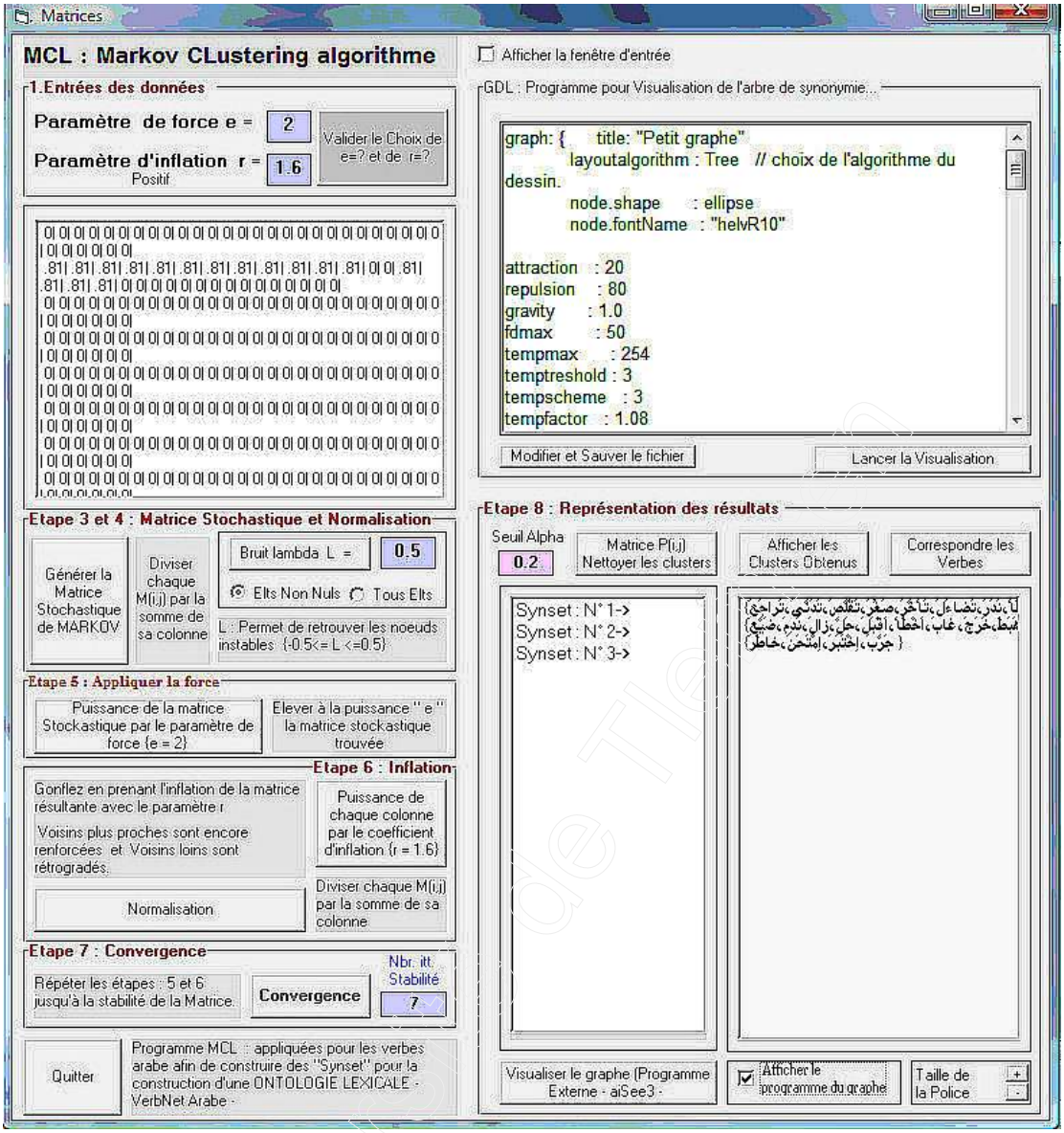


Fig. 44 : Interface de traitement : Etape 2

Une fois les synsets établis pour l'entrée verbale choisie, une description textuelle utilisant la grammaire GDL (*Graph Description Language*) a été générée. Cette production de grammaire permet d'une part de garder une trace de la matrice d'adjacence construite pour une entrée verbale et d'autre part la visualisation du sous graphe de synonymie grâce à des outils spécialisés pour la manipulation et la représentation des descriptions GDL comme le logiciel aiSee (licence offerte par le constructeur via email) qu'on a utilisé.

Le GDL est une description textuelle, utilisant une syntaxe simple et intuitive qu'on peut considérer comme une retranscription, selon la grammaire définie, de la matrice d'adjacence du graphe concerné (d'autres attributs peuvent être insérés pour personnaliser le comportement et l'affichage du graphe par le visuel).

Etant donnée que la construction d'un sous graphe de voisinage (ou sa matrice d'adjacence), de certaines entrées verbales, nécessite un temps de calcul fort considérable, il était plus raisonnable d'opter pour le stockage immédiat dans la base lexicale du sous graphe obtenue. Ceci nous a permit d'éviter de lancer le calcul nécessaire à la construction du sous graphe de synonymie à chaque fois qu'on recherche les synsets pour un entrée verbale.

A titre d'exemple, la figure 45 montre l'obtention de quatre (04) cluster ainsi quatre Synsets – par l'algorithme MCL, chaque groupe de verbe appartenant au même cluster a une couleur différente afin de mieux discerner les différents clusters du graphe.

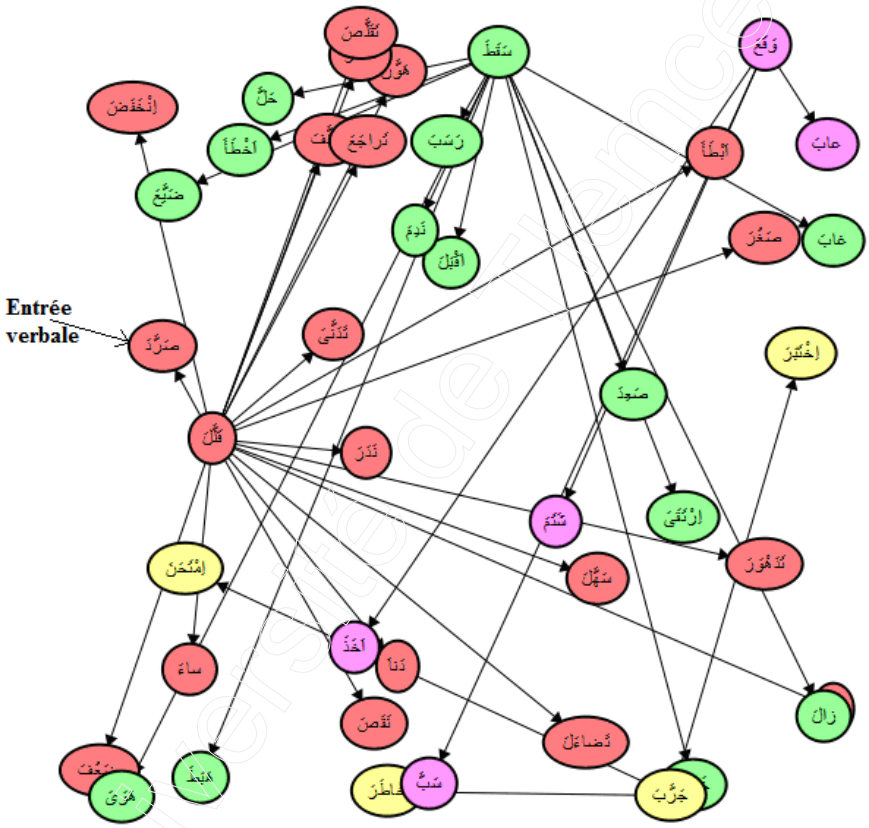


Fig. 45 : GDL du graphe de synonymie d'une partie de l'entrée verbale « صَرَّدَ »

4. Résultat & évaluation

Nous présenterons dans cette section les résultats obtenues sur un échantillon de verbes sélectionnés. Notons que l'approche proposée bien qu'entièrement automatique suppose la validation des synsets calculé par un expert de la langue, nous concevons donc à l'intégrer dans une solution de construction de synset semi automatique.

Nous avons opté pour une évaluation entièrement manuelle des synsets de l'échantillon choisi, l'expert s'est donc chargé d'évaluer selon son estimation la précision de chaque synset et les erreurs présents.

Notre évaluation s'est porté sur un échantillon composé de deux verbes ; « حَلَّلَ » et « اِمْتَحَنَ ». L'évaluation de l'expert de cet échantillon s'est basée sur deux critères : le nombre de sens véhiculés dans un synset calculé et le nombre de verbes impertinent par synset.

Un synset par définition étant un ensemble de synonyme proche véhiculant un seul sens (Matrice lexicale), le nombre correcte de sens véhiculé d'un synset (cluster) est égale à un, un nombre de sens supérieur dans ce cas signifie alors un clustering moins précis.

L'expert s'est chargé dans son évaluation d'un synset calculé de diviser ce dernier en deux sous synset ou plus si le nombre de sens véhiculé dépasse un seul sens.

Les verbes impertinents désignés par l'expert représentent des verbes qui ne doivent pas figurer dans un synset calculé et qui ne peuvent appartenir à aucun des sous synset proposés par l'expert.

De ce fait on explique les différentes colonnes du tableau en partant de droite vers la gauche :

- → Colonne 2 : Synset (cluster) calculé par le logiciel.
- → Colonne 3 : Nombre de sens évoqué par le synset calculé.
- → Colonne 4 : Nombre de verbes non pertinent dans le synset, même si on fragmente ce synsets en plusieurs d'autre synset correcte.
- → Colonne 5 : Note de l'expert pour un synset basé sur les valeurs de la colonne2 et colonne3.

Pour le protocole de test, nous avons commencé par utiliser les valeurs usuelles de l'approche originelle [OLI09] en occurrence, un r = 1.6 (paramètre inflation) et e=2 (paramètre de puissance), les trois tableaux suivants présentent les résultats obtenus avec ces paramètres ainsi que l'évaluation de l'expert de ces résultats :

a. Evaluation des résultats de l'échantillon avec un r=1.6

Taux moyen de synsets	Nombres de verbes impertinents	Différents Sens évoqué du Synset	Synsets calculés A partir d'une entrée verbale	Nbr.
100%	**0**	1. Analyser et expliquer	**1.** حَلَّلَ، زَوَّجَ، شَرَحَ، فَسَّرَ، بَيَّنَ، شَرَّحَ، أَبانَ	01
20%	**3**	1. Dissocier pour réparer 2. Echanger 3. Faire apparaître 4. Eclaircir	**1.** فَكَّ، حَلَّ، فَصَلَ، عَدَّلَ، سَوَّى، أَشْرَعَ، **2.** اِسْتَبْدَلَ، غَيَّرَ، حَوَّلَ، نَقَلَ، أَحالَ، أَفْشَى **3.** أَظْهَرَ، أَوْضَحَ، أَبْرَزَ، أَخْرَجَ، أَصْدَرَ، أَذاعَ، أَحْكَمَ، نَثَرَ، **4.** اِتَّضَحَ، بانَ، ظَهَرَ، أَرْشَدَ، نَشَرَ، أَقامَ، أَنْضَجَ، زَكَّى **5.** ، قَبِلَ، ~~سَحَبَ~~، أَوْرَقَ	02
65%	**0**	1. Viser 2. Entreprendre	**1.** سَدَّدَ، صَوَّبَ، فَتَحَ، وَجَّهَ، أَصْلَحَ، عَلَّمَ **2.** شَرَعَ، قَدَّمَ، اِتَّخَذَ،	03
30%	**2**	1. Disperser 2. Verser 3. Embellir	**1.** فَضَّ، فَرَّقَ، شَتَّتَ، قَطَعَ، قَسَّمَ، وَزَّعَ، **2.** سَكَبَ، ذَرَفَ، أَفاضَ، صَبَّ، سالَ، أَسالَ، اِنْحَدَرَ، سَرَّحَ **3.** حَسَّنَ، زَيَّنَ ، رَقَّ، **4.** ~~خَوَّنَ~~، ~~اِشْتاقَ~~	04
100%	1	1. Achever 2. Vouloir	1. قَضَى، أَفْنَى، أَهْلَكَ، نالَ، اِسْتَغْرَقَ،أَنْهَى، قَتَلَ، وَفَّى، أَدَّى، أَنْفَذَ، أَبْلَغَ، أَوْصَى، 2. أَرادَ، أَخَذَ ، أَمَرَ،	05
100%	0	1. Humidifier	1. رَطَّبَ، بَلَّ، أَرْطَبَ، أَلانَ، نَعَّمَ	06
100%	0	1. Faire en sorte que ...	1. جَعَلَ، اِعْتَمَدَ، خَلَقَ، صَنَعَ، وَضَعَ، أَلْقَى، قَصَدَ عَيَّنَ، رَمَى،	07
80%	1	1. Casser et Broyer	1. كَسَّرَ، حَطَّمَ، هَشَّمَ، كَسَرَ، فَتَّتَ، دَقَّ، 2. ~~حانَ~~	08
100%	**0**	1. Etre rempli …	**1.** أَتَمَّ، اِمْتَلَأَ، اِكْتَمَلَ	09
90%	1	1. Commencer	**2.** بَدَأَ، اِنْطَلَقَ، أَنْشَأَ، اِنْتَقَلَ، حَدَثَ، حَصَلَ، ~~اِقْتَحَمَ~~	10
100%	**0**	1. Plonger dans …	1. خاضَ، أَسْرَعَ، خَلَطَ، حَرَّكَ	11
100%	**0**	1. Introduire	1. مَهَّدَ، مَهَدَ، بَسَطَ، سَهَّلَ	12

Tableau 17 : Evaluation verbe « حَلَّل » avec r = 1.6

Taux moyen de synsets	Nombres de verbes impertinents	Différents Sens évoqué du sysnset	Synsets calculés A parit d'une entrée verbale	Nbr.
50%	**0**	1 Examiner, 2 – s'expérimenter 3 - informer	**1.** {اِمْتَحَنَ،اِبْتَلَى،اِخْتَبَرَ، راهَنَ،عَرَفَ} **2.** {خاطَرَ، جَرَّبَ،خَبَّرَ } **3.** {اَخْبَرَ، اَنْبَأَ}	01
40%	**1**	1 - Patienter صبر، 2 - se faire tout petit 3 – tourner le dos à	**1.** {صَبَرَ، تَجَلَّدَ، تَحَمَّلَ، اِحْتَمَلَ، حَبَسَ، اَمْسَكَ } تَصَبَّرَ، اَطاقَ، اَغْضَى، سَكَتَ} **2.** اِرْتَحَلَ، شَكَرَ، قَبَضَ، كَفَّ، اِمْتَنَعَ، صَمَتَ } ~~اَظْلَمَ~~	02
100%	**0**	1 – maitriser	**1.** ضَبَطَ، اَتْقَنَ، اَحْكَمَ، كَبَحَ}	03
90%	**1**	1- Venir	**3.** قَبِلَ، اَتَى، حَلَّ، هَبَّ، ~~**4.** رَقَّعَ~~	04
100%	**0**	1 – pardonner	**1.** صَفَحَ، قَلَّبَ، تَصَفَّحَ، رَدَّ، نَظَرَ	05

Tableau 18 : Evaluation verbe « اِمْتَحَنَ » avec r = 1.6

Taux moyen de synsets	Nombres de verbes impertinents	Différents Sens évoqué du sysnset	Synsets calculés A parit d'une entrée verbale	Nbr.
100%	0	1	{أَفَقَ، دَبَغَ }	01
50%	0	1 2 3 4	{عالَجَ، داوَى، عايَنَ} { زاوَلَ، مارَسَ، خَدَمَ، سَعَى} { صَبَّ، حاوَلَ} { اِشْتاقَ، رَقَّ}	02

Tableau 19 : Evaluation verbe « أَفَقَ» avec r=1.6

b. Statistiques sur le résultat avec r =1.6

- Nombre moyen des sens évoqué par synset : 18 + 4 + 5 /12 + 5 + 2 = 1,42 sens
- Pourcentage des synsets avec un seul sens 8+3+1 /12+5+2 = 63,15 % des synsets :
- Pourcentage de synsets avec deux sens = 2/ 12+5+2 = 10,56 % (28,56% des synset à sens non unique).
- Pourcentage de synset ayant plus de deux sens = (2+2+1) / (17+2)= 26,31% (71,24% des synset à sens non unique).

- Nombre moyen des verbes impertinents par synset = 9/12+5+2 = 0,64 verbes
- Taux des synset avec 0 verbe impertinent = 6+3+2/12+5+2= 57,9% des synset
- Taux des synset avec un seul verbe impertinent = 5/12+5+2= 29,41% des synset
- Taux des synset avec deux verbes impertinents ou plus = 2/12+5+2 = 11,76 %

Sur cet échantillon évalué nous remarquons qu'une majorité des synsets calculés sont à sens unique et ne comporte aucun verbe impertinent ou juste un seul.

Nous jugeons donc ces résultat satisfaisant du moment ou un nombre de verbes impertinents inferieur à deux (zéro ou un) reste très acceptable et non négativement significatif sur la précision du résultat du clustering.

Toutefois le tableau montre un clustering parfois trop imprécis ou beaucoup trop de sens totalement distinct sont réunis ; ensemble dans le même cluster ou synset (Synset N°2 du Tableau 17 par exemple). Ce type de synset retourné par le programme reçoit d'ailleurs les pires notes d'évaluation de la part de l'expert. Ce problème peut correspondre à un problème de granularité du clustering adopté, cette granularité parfois pas assez petite peut parvenir du choix de la valeur du paramètre r (inflation) de l'algorithme MCL. Le paramètre d'inflation r étant généralement choisie dans un intervalle entre 1,6 et 2 [OLI09], c'est la valeur 1,6 qui a était suggéré dans [OLI09] après plusieurs tests puisque c'est avec cette valeur que le clustering le plus satisfaisant a était obtenue mais cette valeur qu'on a appliqué est apparemment non adéquate dans notre cas d'étude.

Nous avons donc relancé notre algorithme de clustering sur l'échantillon de deux verbes en augmentant la valeur du paramètre d'inflation afin d'obtenir une granularité plus fine dans les résultats du clustering (Nombre d'éléments par cluster réduit).
Le tableau ci-dessus représente les nouveaux clusters obtenus avec une valeur du paramètre r égale à 2.2.

b. Résultats de l'échantillon avec une inflation : r=2.2

Synsets avec **r =1.6**	Synsets avec **r = 2.2**
{اِمْتَحَنَ، اِبْتَلَى، اِخْتَبَرَ، راهَنَ، عَرَفَ}	1. {اِمْتَحَنَ، اِبْتَلَى، اِخْتَبَرَ، عَرَفَ، جَرَّبَ، خاطَرَ، راهَنَ}
{خاطَرَ، جَرَّبَ، خَبَّرَ}	
{اَخْبَرَ، اَنْبَأَ}	2. {خَبَّرَ، اَخْبَرَ، اَنْبَأَ}
{صَبَرَ، تَجَلَّدَ، تَحَمَّلَ، اِحْتَمَلَ، حَبَسَ، اَمْسَكَ}	3. {صَبَرَ، تَجَلَّدَ، حَبَسَ، تَصَبَّرَ، اَطاقَ}
{تَصَبَّرَ، اَطاقَ، اَغْضَى، سَكَتَ}	4. {حْتَمَلَ، تَحَمَّلَ، اَطاقَ، اَغْضَى، اِرْتَحَلَ، شَكَرَ}
{اِرْتَحَلَ، شَكَرَ، قَبَضَ، كَفَّ، اِمْتَنَعَ، صَمَتَ}	5. {بَطَ، اَتْقَنَ، اَحْكَمَ، كَبَحَ}
~~اَظْلَمَ~~	6. {مْسَكَ، قَبَضَ، كَفَّ، اِمْتَنَعَ، سَكَتَ، صَمَتَ}
{ضَبَطَ، اَتْقَنَ، اَحْكَمَ، كَبَحَ}	7. {قَبِلَ، اَتَى، حَلَّ، هَبَّ، رَقَّعَ}
{قَبِلَ، اَتَى، حَلَّ، هَبَّ}	8. {صَفَحَ، قَلَّبَ، تَصَفَّحَ، رَدَّ، نَظَرَ}
~~رَقَّعَ~~	
{صَفَحَ، قَلَّبَ، تَصَفَّحَ، رَدَّ، نَظَرَ}	

1. {أَفَقَ، دَبَغَ}	1. {أَفَقَ، دَبَغَ}
1. {عالَجَ، داوَى، عايَنَ}	2. {عالَجَ، داوَى، زاوَلَ، مارَسَ، خَدَمَ، حاوَلَ، عايَنَ، سَعَى}
2. {زاوَلَ، مارَسَ، خَدَمَ، سَعَى}	3. {صَبَّ، اِشْتاقَ، رَقَّ}
3. {صَبَّ، حاوَلَ}	
4. {اِشْتاقَ، رَقَّ}	

Nous remarquons que le nombre de Clusters/Synsets passe de Cinq à huit pour le graphe de « **IMTAHANA** » et est presque égale aux neufs sous synsets corrects proposés par l'expert, ce qui veut dire qu'on terme de nombre de synsets retournés les résultats sont plus précis.

En ce qui concerne la précision des huit nouveaux synsets obtenus on remarque deux choses :

- Les synsets/clusters totalement correcte ou presque (synsets 6 ,7 et 8) ont était redécouvert et n'ont pas changé.
- Les synsets/clusters à grosse granularité obtenus précédemment ont était fractionné en des synsets/cluster plus fin et qui se rapproche beaucoup plus des sous synsets correcte proposés par l'expert.

Pour le verbe « أَفَقَ » les même remarque peuvent être réitéré à savoir un nombre de clusters plus proche de la répartition en sous synset correct et à granularité plus fine que l'expérience précédente et nous avons obtenus plus de synset/clusters totalement correcte ou presque. Ce changement de valeur du paramètre r (de 1,6 à 2,2) nous suggère, d'après cet échantillon, une granularité beaucoup plus fine et que les synsets/clusters correspondent beaucoup plus à ce qu'il devrait être par rapport à l'expérience précédente avec le paramètre r à 1.6.

D'après notre expérimentation et les différents résultats obtenus, on peut dire que l'augmentation du paramètre r à la valeur 2.2 est un choix judicieux qui nous a permis de retrouver une granularité plus adéquate et qui se rapproche le plus des solutions correctes.
En ce qui concerne les verbes impertinents, il est à rappeler que compte tenu de la lourdeur des calculs et du temps nécessaire, le sous graphe de synonymie a était à chaque fois tronqué et ne représente pour le premier verbe 'hallala' par exemple que le dixième du total du nombre des éléments (100 élément considérés sur 1000 environs). Si les sous graphes de synonymie ont était pris dans leur intégralité, ils se pourrait que certains verbes impertinents peuvent quitter des clusters actuels pour rejoindre un nouveau cluster, dans l'impossibilité de vérifier en pratique cette hypothèse, cette dernière reste toujours plausible et ne pourra être vérifié qu'une fois l'expérimentations sur les sous graphes complets effectuée.

En ce qui concerne la précision des clusters/synsets obtenus, ils sont en respect avec la sémantique lexicale arabe. Les résultats obtenus sur l'échantillon sélectionnés, bien que réduit, montrent que les synsets/clusters proposés, restent parfaitement exploitable dans une solution de construction de synsets semi automatique, qui nécessite l'intervention de l'expert pour l'étape de la validation.

On note cependant que si la granularité est plus adéquate avec une inflation à 2.2, des verbes sont à la mauvaise place (dans le mauvais synset), en d'autre terme, il faudra échanger un verbes ou plus entre deux synsets pour obtenir des synsets totalement correct. Nous pensons dans ce cas que les structures de groupe découvertes par l'algorithme MCL de façon très efficace d'un point de vue mathématique sont parfois sémantiquement incorrectes. Mais il faut rappeler que dans un contexte d'exploitation semi automatique les clusters automatiquement découverts sont d'une grande aide et permettraient de soutenir efficacement l'expert dans sa tâche de création des concepts d'une ontologie lexicale.

Conclusion Générale

Actuellement, la recherche d'informations sur le web se fait sur la base d'une correspondance entre des mots clés (métadonnées) stockés et les mots clés formulée par les utilisateurs. Le processus de recherche de demain est plus "***intelligent***" car capable de prendre en compte le **sens** des mots clés, plutôt que de les considérer comme de simple chaîne de caractères dépourvus de toute signification. C'est l'un des objectifs du web sémantique. C'est là que la notion ***d'ontologie*** intervient, en organisant sous forme de graphe un ensemble de concepts (les mots-clés) par des relations sémantiques (ex : est-synonyme-de, est-une-sorte-de, est-analogue-à, etc.). C'est une façon technique de simuler de la connaissance et prendre l'initiative d'augmenter les résultats de recherches sur un sujet fortement connexe.

La construction entièrement manuelle d'une ontologie est une tache rude, complexe et qui surtout nécessite beaucoup de temps et de ressources. Le recours à des méthodes automatiques ou semi automatiques est devenu indispensable, toutefois le recours aux experts pour la validation des résultats au cours de ce processus de la création permet d'aboutir à une ontologie plus accomplie et plus précise.

Princeton WordNet étant notre ontologie modèle pour la représentation des concepts et leurs relations. La découverte automatique des synsets était notre objectif dans le travail réalisé (la découverte des relations entre synsets nécessite plus de temps et de moyen et dépasse le cadre de ce magister). Pour cela on a utilisé un dictionnaire monolingue de la langue arabe pour en extraire un ensemble des synonymes sur lequel on a appliqué l'algorithme MCL (Markov Clustering aLgorithme) pour la détection des synsets dans cet ensemble de synonyme.

Concernant les résultats et les pourcentages obtenue par notre approche appliqué sur un échantillon de verbes choisie étaient très encourageants. Ainsi on peut dire que le Clustering s'est avéré être une bonne alternative pour créer les concepts (synsets) de notre ontologie lexicale à partir du réseau de synonymie d'un dictionnaire.

Par ailleurs, comme améliorations à ce qui a était déjà accompli et comme travail futur, nous prévoyons de compléter la solution proposé par la détection des relations entre les synsets découverts à partir des relations entre termes qu'on peut détecter directement dans le dictionnaire grâce à des patrons morpho syntaxiques spécifiques. L'utilisation d'autres ressources lexicales qu'un dictionnaire est un autre moyen d'améliorer l'approche testée, les corpus textuels arabes annotés ou pas nous permettrons d'expérimenter d'autres techniques pour la détection des relations lexicales et sémantiques, il est à noter que pour exploiter les corpus textuels non étiquetés, l'acquisition d'un étiqueteur morpho syntaxique est indispensable.

Notre but est d'acquérir les expériences récentes menées pour les différentes langues et de les investir dans une conception des ressources linguistiques et terminologique en langue arabe.

Le travail réalisé nous a permis de découvrir l'importance d'acquisition (semi) automatique des ressources ontologiques et spécialement pour la langue arabe.

Nous espérons finalement par le modeste travail réalisé, apporter une contribution significative au projet de création d'une ontologie pour la langue arabe et qu'il permettra avec les travaux futurs, d'aboutir à la création concrète d'une ontologie lexicale.

Bibliographie

[ASS97] Assadi, H., Knowledge Acquisition from Texts: Using an Automatic Clustering Method Based on Noun-Modifier Relationship, Proceedings of 35th Annual Meeting of the Association for Computational Linguistics (ACL'97), Madrid, Spain, 1997.
[AGI00] Agirre, E., Ansa, O., Hovy, E., and Martínez, D., Enriching very large ontologies using the WWW, Proceedings of the ECAI 2000 Workshop on Ontology Learning (OL'2000), 2000.

[AUS00] Aussenac-Gilles, N., Biébow, B. & Szulman, S. Revisiting, *Ontology Design: a methodology based on corpus analysis*. Proceedings of the 12th *International Conference on Knowledge Engineering and Knowledge Management: EKAW'2000*, R. Dieng & O. Corby (eds), LNAI 1937, Springer, pp. 172-188, 2000.

[BAC00] B. Bachimont, "Engagement sémantique et engagement ontologique : conception et réalisation d'ontologies en ingénierie des connaissances". Eyrolles, 2000.

[BAC03] B. Bachimont, J. Charlet & R. Troncy. "Ontologies pour le Web Sémantique". ction spécifique 32 CNRS / STIC Web sémantique Rapport final. 2003

[BAN-03] Kornél Robert BANGHA, *La place des connaissances lexicales face aux connaissances du monde dans le processus d'interprétation des énoncés*, thèse de doctorat en linguistique et intelligence artificielle, Université de Montréal, Août 2003.

[BAZ05] Mustapha BAZIZ, Indexation conceptuelle guidée par ontologie pour la recherche d'information, thèses de doctorat de l'université Paul Sabatier spécialité informatique, 2005

[BIE99] BIEBOW M. & SZULMAN S., TERMINAE : a method and a tool to build of a domain ontology, in Proceedings of the 11th European Knowledge Acquisition Workshop (EKAW'99), Springer, 1999.

[BIK99] Bikel, D. A., Schwartz, R., and Weischedel, R., An Algorithm that Learns What's in a Name, Machine Learning, 34, 211-231, 1999.

[BIS00] Bisson G., Nedellec C., and Canamero, D., Designing Clustering Methods for Ontology Building- The Mo'K Workbench, Proceedings of the ECAI 2000 Workshop on Ontology Learning (OL'2000), 2000.

[BLA98] M. Blazquez, M. Fernandez, J. Garcia-Pinar. & A. Gomez-Perez, "Building Ontologies at the Knowledge Level using the Ontology Design Environment", in Proceedings of the Banff Workshop on Knwoledge Acquisition for Knwoledge-based Systems, 1998.

[BLA04] W. J. Black, & Elkateb, S.: « prototype English-Arabic Dictionary Based on WordNet », Proceedings of 2nd Global WordNet Conference, GWC2004, Czech Republic: 67-74

[BLA06] William J. Black, Sabri Elkateb, Christiane Fellbaum, Musa Alkhalifa, Adam Pease, Horacio Rodríguez, Piek Vossen (2006) « Introducing the Arabic WordNet project

Proceedings of the 3rd Global Wordnet Conference », Jeju Island, Korea, January, 2006. URL: http://www.lsi.upc.edu/~nlp/papers/fellbaum-alkhalifa-206.pdf

[BOR97] Borgo, S., Guarino, N., Masolo, C., and Vetere, G., Using a large linguistic ontology for internet based retrieval of object-oriented components, Proceedings of conference on Software Engineering and Knowledge Engineering, IL, USA, 1997.

[BOU-02] Caroline BOUSQUET-VERNHETTES, *Compréhension robuste de la parole spontanée dans le dialogue oral homme-machine – Décodage conceptuel stochastique*, thèse de doctorat en informatique, Université de Toulouse III, Septembre 2002.

[BRE01] Brewster, C., Ciravegna F., and Wilks, Y., Knowledge Acquisition for nowledge Management, Proceedings of the IJCAI 2001 Workshop on Ontology Learning (OL'2001) (Position Paper), USA, 2001.

[BRE02] C. Brewster, F. Ciravegna, and Y. Wilks. User-centred Ontology Learning for Knowledge Management. In Proceedings 7th International Workshop on Applications of Natural Language to Information Systems, Stockholm, 2002.

[BRI04] Laurent Brisson. Mesures d'intérêt subjectif et représentation des connaissances. Rapport technique, Laboratoire I3S, Université Sophia Antipolis, Nice (France), Octobre 2004.

[CAR99] Caraballo, S. A., Automatic Construction of Hypernym Labeled Noun Hierarchy from Text, Proceedings of ACL'99, 1999.

[CHA01] Charlet J., L'ingénierie des connaissances : une science, un enseignement ?, in Actes des journées francophones d'Ingénierie des Connaissances (IC'2001), Presse Universitaire Grenobloise, pages 233-252, 2001.

[CHA99] E.Charniac, M. Berland, Finding parting very large corpora, in In Proceedings of the 37th annuel meeting of the ACL, pp57-64, 1999.

[CHR09] Christine Froidevaux, LRI – Equipe Bioinformatique, CNRS UMR 862, Université Paris Sud, « Ontologies ».

[CIM04-b] P. Cimano, A. Pivk, L.Shmidh-Thieme, S. Staab, Learning Taxonomic relation from heterogeneous Evidence, ECAI-2004 Workshop on ontology Learning and population.

[COL98] COLLINS A. M. & and Quillian M. R., 1969. « Retrieval Time From Semantic Memory ». Journal of Verbal Behavior and Verbal Learning 8 : 240-247.

[COR00] O. Corcho, and A. Gómez-Pérez, Evaluating Knowledge Representation and Reasoning Capabilities of Ontology Specification Languages, Proceedings of the ECAI 2000 Workshop on Applications of Ontologies and Problem-Solving Methods, Berlin, 2000.

[CRA00] Craven, M., DiPasquo, D., Freitag, D., McCallum, A., Mitchell, T., Nigam, K., Slattery, S., Learning to construct knowledge bases from the World Wide Web, Artificial Intelligence, 118: 69-113, 2000.

[DAR-02] Darwish, D. *Building a shallow Arabic Morphological Analyzer in one day*. Proceedings of the ACL-02 workshop on Computational approaches to semitic languages, Philadelphia, Pennsylvania.

[DEI01] Deiteil, A., Faron-Zucker, C., Dieng, R., Learning ontologies from RDF annotations, Proceedings of the IJCAI 2001 Workshop on Ontology Learning (OL'2001), Seattle, USA, 2001

[DIA04] M. Diab, « The Feasibility of Bootstrapping an Arabic WordNet leveraging Parallel Corpora and an English WordNet». Proceedings of the Arabic language technologies and resources, nemlar, cairo 2004.

[DIB04] R. Dib Thèse : Utilisation des ontologies contextuelles pour le partage sémantique entre les systèmes d'information dans l'entreprise. Soutenue le 20 décembre 2004, Pour l'obtention du grade doctorat.

[DIE01] DIENG R., CORBY O., GANDON F., GIBOIN A., GOLEBIOWSKA J., MATTA N. & RIBIÈRE M., Méthodes et outils pour la gestion des connaissances : une approche pluridisciplinaire du knowledge management (2ième édition), Dunod Edition Informatiques Séries Systèmes d'Information, 2001.

[DOA00] Doan, A., P. Domingos, and A. Levy (2000). Learning Source Descriptions for Data Integration. Proceedings of the Third International Workshop on the Web and Databases.

[DOG00] S. M. van Dongen. 2000. Graph Clustering by Flow Simulation. Ph.D. thesis, University of Utrecht.

[DOR06] Beate Dorow. 2006. A Graph Model for Words and their Meanings. Ph.D. thesis, Institut fur Maschinelle Sprachverarbeitung der Universitat Stuttgart.
[EAG96] Eagles homepage at http://www.ilc.pi.cnr.it/EAGLES96/rep2, 1996.

[FAL01] G. Falquet & C.L. Mottaz-Jiang, "Navigation hypertexte dans une ontologie multi-points de vue", NîmesTIC' . 2001.

[FAR96] Farquhar, A., Fikes, R., Rice, J. The Ontolingua Server: A Tool for Collaborative Ontology Construction, Proceedings of KAW96, Banff, Canada, 1996.

[FAR05] D. Farreres, « Creation of wide-coverage domain-independent ontologies». PhD thesis, Univertitat Politècnicade Catalunya, 2005

[FAU98] Faure, D., Nedellec C., and Rouveirol, C., Acquisition of Semantic Knowledge using Machine Learning Methods: The System ASIUM, Technical Report number ICS-TR-88-16, Laboratoire de Recherche en Informatique, Inference and Learning Group, Universite Paris Sud, 1998.

[FEL98] Christiane, Fellbaum, Wordnet – An Electronic Lexical Database, The MIT Press, Cambridge, Mass. ed. 1998

[FEL06] Christiane Fellbaum, Musa Alkhalifa, William J. Black, Sabri Elkateb, Adam Pease, Horacio Rodríguez, Piek Vossen (2006) « Building a WordNet for Arabic » Proceedings of the the 5th Conference on Language Resources and Evaluation LREC2006,May,

[FER99] Fernandez-Lopez, M., Overview Of Methodologies For Building Ontologies. Proceedings of the IJCAI'99 Workshop on Ontologies and Problem-Solving Methods, Stockholm (Suède), pp. 4/1,4/13, 1999

[FIK00] R. Fikes, D. L. McGuinness, J. Rice and S. Wilder. (2000). An Environment for Merging and Testing Large Ontologies. Principles of Knowledge Representation and Reasoning: Proceedings of the Seventh International Conference (KR2000). A. G. Cohn, F. Giunchiglia and B. Selman, editors. San Francisco, CA, Morgan Kaufmann Publishers.

[FIN99] Finkelstein-Landau, M., Morin, E., Extracting Semantic Relationships between Terms: Supervised vs. Unsupervised Methods, Proceedings of International workshop on Ontological Engineering on the Global Information Infrastructure,71-80, Dagstuhl-Castle, Germany, 1999.

[FUR02] F. Furst, "L'ingénierie ontologique". Rapport de recherche N°02-07. 2002.

[GAE02] Lortal GAËLLE. État de l'art ontologies et intégration/fusion d'ontologies, 2002.

[GAM02] Gamallo, P., Gonzalez, M., Agustini, A., Lopes, G., and de Lima, V. S., Mapping Syntactic Dependencies onto Semantic Relations, Proceedings of the ECAI 2002 Workshop on Machine Learning and Natural Language Processing for Ontology Engineering (OLT'2002), Lyon, France, 2002.

[GAN02] F. Gandon. Ontology engineering: a survey and a return on experience. Technical Report RR-4396, INRIA, 03 2002

[GER08] Gerard de Melo and Gerhard Weikum. 2008. On the utility of automatically generated wordnets. In Proc. 4th Global WordNet Conf. (GWC), pages 147–161, Szeged, Hungary. University of Szeged.

[GFE05] Gfeller David, Jean-C´edric Chappelier, and Paulo De Los Rios. 2005. Synonym Dictionary Improvement through Markov Clustering and Clustering Stability. In Proc. of International Symposium on Applied Stochastic Models and Data Analysis (ASMDA), pages 106–113.

[GOM95] Asunción GÓMEZ-PÉREZ, Natalia JURISTO et Juan PAZOS. Evaluation and assessment of the knowledge sharing technology. In Towards very large knowledge bases, pages 289–296. IOS Press, 1995.

[GOM97] Mariano FERNANDEZ, Asuncion GOMEZ-PEREZ et Natalia JURISTO. Methontology: from ontological art towards ontological engineering. In Proceedings of the AAAI97, Spring Symposium Series on Ontological Engineering, pages 33–40, Stanford, USA, March 1997.

[GOM99] A. Gomez-Pérez, "Développements récents en matière de conception, de maintenance et d'utilisation des ontologies". Actes du colloque de Nantes, pp 9-20. 1999.

[GOM99b] Asuncion Gomez-Perez, "Ontological Engineering: a State of the Art", Expert Update. British Computer Society. Vol. 2. nº 3. pp. 33 – 43 (1999).

[GOM04] Gómez-Pérez, A., Fernández-López, M., and Corcho, O. Ontological Engineering with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web. Springer, 2004.

[GRA04] Graeme Hirst. 2004. Ontology and the lexicon. In Steffen Staab and Rudi Studer, editors, Handbook on Ontologies, International Handbooks on Information Systems, pages 209–230. Springer.

[GRU91] T.R.Gruber, The Role of Common Ontology in Achieving Sharable, Reusable Knowledge Bases, In J.A.Allen, R.Fikes, and E.Sandewall (Eds), Principles of Knowledge Representation and Reasoning, Proceedings of the Second International Conference, Cambridge, MA, 1991, Morgan Kaufmann, pp. 601-602, 1991.

[GRU93] Thomas Gruber, "A translation Approach to Portable Ontology Specifications", Knowledge Acquisition, 5(2), pp. 199 – 220.

[GRU95a] GRUNINGER M. & FOX M. S., Methodology for the design and evaluation of ontologies, in Proceedings of the Workshop on Basic Ontological Issues on Knowledge Sharing, IJCAI'95, 1995.

[GRU95b] T.R. Gruber, "Toward principles for the design of ontologies used for knowledge sharing". International Journal of Human Computer Studies. 1995.

[Gua97] N. Guarino, "Understanding, building and using ontologies". International J. Human-Computer Studies, pp 293-310. 1997.

[GUA98] Guarino, N., «Formal Ontology and Information Systems», Formal Ontology in Information Systems. IOS Press, 1998.

[HAB05] N. Habash, and O. Rambow, « Arabic Tokenization, Part-of-Speech Tagging and Morphological Disambiguation in One Fell Swoop ». In Proceedings of the 43rd Meeting of the Association for Computational Linguistics (ACL), 2005.

[HAH98] Hahn U., and Schnattinger, K., Towards Text Knowledge Engineering, Proceedings of 15th National Conference on Artificial Intelligence (AAAI'98), 524-531, Madison, Wisconsin, 1998.

[HAH01] Hahn U., Romacker, The SYNDIKATE Text Knowledge Base Generator, Proceedings of the 1st International Conference on Human Language Technology Research, San Diego, USA, 2001.

[HAH02] Hahn U., and Marko, K. G., Ontology and Lexicon Evolution by Text Understanding, Proceedings of the ECAI 2002 Workshop on Machine Learning and Natural Language Processing for Ontology Engineering (OLT'2002), Lyon, France, 2002.

[HER92] Hearst, M.A., Automatic Acquisition of Hyponyms from Large Text Corpora, Proceedings of the 14th International Conference on Computational Linguistics, Nantes, France, July, 1992.

[HEY01] Heyer G., Läuter, M., Quasthoff, U., Wittig, T., Wolff, C., Learning Relations using Collocations, Proceedings of the IJCAI 2001 Workshop on Ontology Learning (OL'2001), Seattle, USA, 2001.

[HIN90] D. Hindle. Noun classification from predicate-argument structures. In Proceedings of the Annual Meeting of the Association for Computational Linguistics, pages 268–275, 1990.

[HWA99] Hwang, C.H., Incompletely and imprecisely speaking: Using dynamic ontologies for representing and retrieving information. Proceedings of the 6th International Workshop on Knowledge Representation meets Databases (KRDB'99), Linköping, Sweden, July, 1999.

[JAN99] Jannink, J. (1999). Thesaurus Entry Extraction from an On-line Dictionary. In Proceedings of Fusion '99, Sunnyvale CA.

[JOH94] Johannesson, P. (1994). A Method for Transforming Relational Schemas into Conceptual Schemas. In 10th International Conference on Data Engineering, Ed. M. Rusinkiewicz, Houston, 115 - 122.

[JPM-00] Jean-Marie Pierrel, *Ingénierie des langues*, HERMES Science publication, 2000.

[KAS99] Kashyap, V., Design and Creation of Ontologies for Environmental Information Retrieval. Proceedings of the 12th Workshop on Knowledge Acquisition, Modeling and Management, Alberta, Canada, October, 1999.

[KAS00] KASSEL G., ABEL M., BARRY C., BOULITREAU P., IRASTORZA C.&PERPETTE S., Construction et exploitation d'une ontologie pour la gestion des connaissances d'une équipe de recherche, in Actes des journées francophones d'Ingénierie des Connaissances (IC'2000), 2000.

[KHO-01] Shereen Khoja. 2001. *APT: Arabic Part-of-speech Tagger*. Proc. of the Student Workshop at NAACL 2001.

[KHO-03] Shereen Khoja, Roger Garside and Gerry Knowles. *An Arabic Tagset for the Morphosyntactic Tagging of Arabic*. In Wilson,A, Rayson, P, McEnery, T (Ed.) A rainbow of corpora: Corups linguistics and the langage of the world. Lincom-Europa, Munich, pp.57-92.

[KIE00] Kietz, J. U., Maedche, A., Volz, R., A method for Semi-Automatic Ontology acquisition from a corporate Intranet, Proceedings of the EKAW 2000 workshop on Ontologies and Texts, France, 2000.

[LEC02] M.Leclere, F. Trichet & F. Furst, "Operationalising domain ontologies : towards an ontological level for the SG family", in Foundations and Applications of Conceptual Structure, contributions to the International Conference on Conceptual Structures. 2002.

[LEN90] Lenat D. B., and Guha, R. V., Building Large Knowledge Based Systems, Representation and inference in the Cyc Project, Readings, (MA: Addison Wesley, 1990).

[MAE00-a] Alexander Maedche , Steffen Staab, Mining Ontologies from Text, in Proceeding EKAW '00 Springer lecture note in artificial intelligence (LNAI-1937), 2000.

[MAE00-b] A. Maedche, and Said. Staab, Semi-Automatic Engineering of ontology from the text. Proceedings of the 12th International Conference on Software Engineering and Knowledge Engineering (SEKE'2000), Chicago, 2000.

[MAE00-c] A. Maedche and S. Staab. The TEXT-TO-ONTO Ontology Learning Environment. Software Demonstration at ICCS-2000 - Eight International Conference on conceptual Structures August 14-18, 2000, Darmstadt, Germany, available at: http://www.aifb.uni-karlsruhe.de/˜sst/ Research/Publications/Publications.htm, August 2000. (Réf 53)

[MAE01-a] Maedche, A., and Staab, S., (a), Ontology learning for the Semantic Web, IEEE journal on Intelligent Systems, Vol. 16, No. 2, 72-79, 2001

[MAE01-b] A. Maedche and R. Volz. The ontology Extraction & Maintenance Framework Text-To-Onto. In ICDM'01: The 2001 IEEE International Conferenceon Data Mining Workshop on Integrating Data Mining and Knowledge Management, November 2001. (Réf56)

[MAE02] MAEDCHE A., MOTIK B., SILVA N. & VOLZ R., MAFRA : a mapping framework for distributed ontologies, in Proceedings of the International Conference EKAW'2002, Springer LNAI 2473, pages 235-250, 2002.

[MAE02] Maedche, A., and Staab, S., Measuring Similarity between Ontologies, Proceedings of EKAW'02, Spain, 2002.

[MAE03] A. Maedche and S. Staab. Ontology Learning. In S. Staab and Studer R., editors, Handbook on Ontologies in Information Systems. Springer, 2003. (Réf 55)

[MEL07] Naçima MELLAL, Réalisation de l'interopérabilité sémantique des systèmes, basée sur les ontologies et les flux d'information, THÈSE de Docteur De L'université De Savoie, 2007. pp 25-30.

[MIL93] George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine Miller; Introduction to WordNet: An On-line Lexical Database (Revised August 1993)

[MIZ97] R. MIZOGUCHI & M. IKEDA, Towards ontolgy engineering, in Proceedings of the Joint Pacific AsianConference on Expert Systems, 1997.

[MIZ03] Riichiro Mizuguchi, "Tutorial on ontological engineering - Part 1: Introduction to Ontological Engineering", New Generation Computing, OhmSha&Springer, Vol.21, No.4 (2003), pp.365 – 384.

[MOR99] Morin E (1999) Automatic acquisition of semantic relations between terms from technical corpora.Proc. Of the Fifth Int. Congress on Terminology and Knowledge Engineering (TKE-99), TermNet-Verlag, Vienna

[NED05] C. Nédellec1 and A. Nazarenko2, "Ontology and Information Extraction : A Necessary Symbiosis", in Ontology Learning from Text: Methods, Evaluation and

Applications, edited by P. Buitelaar, P. Cimiano and B. Magnini, IOS Press Publication, July 2005.

[NOB00] NOBÉCOURT J. & BIÉBOW B., MDOS: a modelling language to build a formal ontology en either Description Logics or Conceptual Graphs, in Proceedings of the 12th International Conference on Knowledge Engineering and Management (EKAW'2000), Springer-Verlag LNAI 1937, pages 57-64, 2000.

[NOY02] NOY N & MUSEN M. A., Evaluating ontology-mapping tools: requirements and experience, in Proceedings of the Workshop on Evaluation of Ontology Tools (EON'2002) at EKAW'2002, 2002.

[OLI01] A. Oliveira, F. Camara Pereira, and A. Cardoso. Automatic Reading and Learning from Text. In Proceedings of the International Symposium on ArtificialIntelligence, ISAI 2001, 2001. (Réf 73)

[OLI09] Oliveira Hugo Gonçalo, Paulo Gomes : Towards the Automatic Creation of a Wordnet from a Term-based Lexical Networkn. Proceedings of the 2010 Workshop on Graph-based Methods for Natural Language Processing, ACL 2010, pages 10–18,

[ONT06] Stanford University, Ontolingua Home Page:

[OSC05] Oscar Corcho, "A layered declarative approach to ontology translation with knowledge", Frontiers in Articificial pp. 1-2

[PAP02] Papatheodrou, C., A. Vassiliou and B. Simon (2002). Discovery of Ontologies for Learning Resources Using Wordbased Clustering, EDMEDIA 2002, Copyright by AACE, Reprinted, Denver, USA.

[PER93] PEREIRA F., N TISHBY, L LEE, "Distributional Clustering of English Words", dans les actes de 30th conference of the Association for Computational Linguistics (ACL), pp. 183-190, 1993

[PSY03] Valéry Psyché, Olavo Mendes, Jacqueline Bourdeau, "Apport de l'ingénierie ontologique aux environnements de formations à distance", revue sticef.org, Volume 10, 2003.

[REN07] A. RENOUF. Modélisation de la formulation d'applications de traitement d'images. PhD thesis, Université de Caen, September 2007

[RIC92] Richards, B.L. & Mooney, R.J., Learning relations by pathfinding, Proceedings of AAAI-92, San Jose, CA, pp. 50–55, 1992.

[RIL96] Riloff, E., Automatically Generating Extraction Patterns from Untagged Text. Proceedings of the 13th Conference on Artificial Intelligence, 1044-1049, AAAI Press/ MIT Press, 1996.

[RIN04] M.L. Reinberger, P. Spyns, Discovering knowledge in texts for the learning of DOGMA-inspired ontologies. In Proceedings of the Workshop Ontology learning and population ECAI 2004, August 2004, Valencia, Spain, (pp-19-24)

[ROT96] T. R. Rothenfluh, J.H. Gennari. 1996. Reusable ontologies, knowledge-acquisition tools, and performance systems: PROTEGE-II solutions to Sisyphus-2. International Journal of Human-Computer Studies 44: 303-332.

[RUN02] Rubin D.L., M. Hewett, D.E Oliver., T.E Klein, and R.B Altman (2002). Automatic data acquisition into ontologies from pharmacogenetics relational data sources using declarative object definitions and XML. In: Proceedings of the Pacific Symposium on Biology, Lihue, HI.

[SAN99] M. Sanderson and B. Croft. Deriving concept hierarchies from text. In Research and Development in Information Retrieval, pages 206–213. 1999.

[SCH88] SCHANK et ABELSON. Scripts, Plans, Goals and Understanding. Kaufmann, San Mateo, CA, 1988

[SHA02-a] M. Mehrnoush Shamsfard+ and Ahmad Abdollahzadeh Barforoush, « The State of the Art in Ontology Learning: A Framework for Comparison », Intelligent Systems Laboratory. Published in: Journal « The Knowledge Engineering» Review, Volume 18 Issue 4, December 2003.

[SHA02-b] Shamsfard M., and Barforoush, A. A., (a) An Introduction to HASTI: An Ontology Learning System,iiiiiiij Proceedings of 6th Conference on Artificial Intelligence and Soft Computing (ASC'2002), Banff, Canada, June, 2002.

[SOD95] Soderland, S., Fisher, D., Aseltine, J., Lehnert, W., Issues in Inductive Learning of Domain-Specific Text Extraction Rules, Proceedings of the IJCAI 95 Workshop on Approaches to Learning for Natural Language Processing, 1995.

[SOW84] John SOWA, Conceptual structures: information processing in mind and machine, Addison-Wesley, 1984.

[SOW92] John SOWA. Conceptual graphs summary 1992, pp. 3–51

[SOW00] John SOWA. Knowledge Representation: Logical, Philosophical, and Computational Foundations. Brooks/Cole, Aout 2000.

[SPO02] Sporleder, C., A Galois Lattice based Approach to Lexical Inheritance Hierarchy Learning, Proceedings of the ECAI 2002 Workshop on Machine Learning and Natural Language Processing for Ontology Engineering (OLT'2002), Lyon, France, 2002.

[SUN02] Sundblad, H., Automatic Acquisition of Hyponyms and Meronyms from Question Corpora, Proceedings of the ECAI 2002 Workshop on Machine Learning and Natural Language Processing for Ontology Engineering (OLT'2002), Lyon, France, 2002.

[SUR00] Suryanto H., Compton, P., Learning classification taxonomies from a classification knowledge based system, Proceedings of the ECAI 2000 Workshop on Ontology Learning (OL'2000), 2000.

[SUR01] Suryanto, H. and P. Compton (2001). Discovery of Ontologies from Knowledge Bases. Proceedings of the First International Conference on Knowledge Capture, The Association for Computing Machinery,New York, USA, pp171-178.

[TER01] Termier, A., Rousset, M. C., Sebag, M., Combining Statistics and Semantics for Word and Document Clustering, Proceedings of the IJCAI 2001 Workshop on Ontology Learning (OL'2001), Seattle, USA, 2001

[TIM04] Buckwalter, T. *Arabic morphology Analysis*. URL: http://www.qamus.org/morphology.htm

[TOD00] Todirascu, A., de Beuvron F., Galea, D., Rousselot, F., Using Description Logics for Ontology Extraction, Proceedings of the ECAI 2000 Workshop on Ontology Learning (OL'2000), 2000.

[TON07] Tony DUJARDIN, De l'apport des ontologies pour la conception de systèmes multi-agents ouverts. Mémoire de Master Recherche Informatique mention Intelligence Artificielle et Extraction de Connaissances, Université de Lyon 2007

[TUF04]: D. Tufis, "Special Issue on the BalkaNet project",Romanian Journal of Information Science and Technology, Vol. 7, nos 1-2. Ed. 2004.

[TUR01] Nicolas Turenne, Etat de l'art de la classification automatique pour l'acquisition de connaissances à partir de textes, UMR INRA-INAPG – Biométrie et Intelligence Artificielle (BIA), Technical Report, INRA, 2001

[USC95] M. Uschold & M. King, "Towards a Methodology for Bbuilding Ontologies", in Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing, IJCAI'95. 1995.

[Usc96] Mike USCHOLD et Michael GRÜNINGER, "Ontologies : Principles, methods and applications". Knowledge Engineering Review, 11:93-136 / 11(2):93-155, 1996

[USC98] M. Uschold, Knowledge level modelling: Concepts and terminology. Knowledge Engineering Review, 1998, 13(1).

[VEL01] Velardi P., Missikoff M., and Fabriani P. (2001). Using Text Processing Techniques to Automatically enrich a Domain Ontology. ACM conference on Formal Ontologies in Information Systems (FOIS 2001), Maine, USA (2001)

[VOS98] P. Vossen, EuroWordNet : A Multilingual Database with Lexical Semantic Networks, Kluwer Academic, ed. 1998

[WAG00] Wagner A., Enriching a Lexical semantic Net with selectional Preferences by Means of Statistical Corpus Analysis, Proceedings of the ECAI 2000 Workshop on Ontology Learning (OL'2000), Berlin, Germany, 2000.

[WAN07] Tonio Wandmacher, Ekaterina Ovchinnikova, Ulf Krumnack, and Henrik Dittmann. 2007. Extraction, evaluation and integration of lexical-semantic relations for the

automated construction of a lexical ontology. In Third Australasian Ontology Workshop (AOW), volume 85 of CRPIT, pages 61–69, Gold Coast, Australia. ACS.

[WAT86] Waterman, D. A. A Guide to Expert Systems. Addison-Wesley. Boston, Masachusets (USA) 1986.

[WEL01] WELTY C. & GUARINO N., Supporting ontological analysis of taxonomic relationships, Data et Knowledge Engineering (39), pages 51-74, 2001.

[WID04] D.Widdows, Geometry and Meaning, CSLI, Stanford University, 2004.

[WIL00] Williams, A. B., Tsatsoulis, C., An Instance-based Approach for Identifying Candidate Ontology Relations within a Multi-Agent System, Proceedings of the ECAI 2000 Workshop on Ontology Learning (OL'2000), Berlin, Germany, 2000.

[WRI98] J. Wright and M. Guinness . Conceptual Modeling for Configuration: A Description Logicbased Approach. Artificial Intelligence for Engineering Design, Analysis, and Manufacturing – special issue on Configuration, 1998.

[WU02] S.-H. Wu and W.-L. Hsu. SOAT: A Semi-Automatic Domain Ontology Acquisition Tool from Chinese Corpus. In COLING 2002: The 17th International Conference on Computational Linguistics: Project Notes, Taipei, Taiwan, 2002.

[YAM01] T. Yamaguchi, Acquiring Conceptual Relations from domain-Specific Texts, Proceedings of the IJCAI 2001 Workshop on Ontology Learning (OL'2001), Seattle, USA, 2001.

[ZWE99] ZWEIGENBAUM P., Encoder l'information médicale : des terminologies aux systèmes de représentation des connaissances, in Innovation stratégique en information de santé (ISIS) (2-3), pages 27-47, 1999.

ملخّص :

إنّ نتيجة البحوث لبناء أنطولوجيا انطلاقا من النصوص كوّنت إطاراً توافقيا، وعبر مراحل أربع. مبادرة بتحليل لساني للمدوّنة ثمّ تشكيل المادّة اللغوية، استخراج المفاهيم وأخيرا عملية الهندسة الأنطولوجية.

وهدفت بهذه الدراسة إلى تقديم منهجية نصف ميكانيكية للأنطولوجيا، وذلك استنادا إلى نصوص عربية. والجدير بالإشارة أنّ الصعوبات والآفاق المتّصلة بهذا الإجراء قد تمّ تحديدها، مع العلم أنّ الأهمّية الكبرى من هذه الدراسة هو مراقبتها الفعلية من قبل مختصّ في هذا المجال. ولقد انحصر تدخّلنا في هذا المشروع حول أنطولوجيا معجمية، آخذين نصب أعيننا نموذج الأنطولوجيا وورد نيتWordNet. ففي "المدخل" وضعت الأفعال العربي التي استقيناها من قاموس وحدوي اللغة ومعاصر كان بالنسبة لنا قاعدة المعطيات المعجمية المفضّلة. كما كان الفعل الذي يعدّ محور الجملة، الأساس المباشر في إنشاء المفاهيم مع اتّخاذ السانسيست synsets نموذجا لتمثيل المعنى.

ولقد سمح لنا خوارزم ماركوف لتجميع البيانات المولّدة من طرف مختلف المحدّدات المستخلصة عن طريقة التعدية، باكتشاف الأفعال المتشابهة، مع تحديد مجموعة المترادفات الخاصّة بكلّ مدخل فعلي.

الكلمات المفتاحية : الأنطولوجية، النصوص العربية، أنطولوجيا معجمية ، خوارزم ماركوف .

Résumé

Un Framework consensuel de construction d'ontologie à partir de textes en quatre étapes à été le fruit de toutes les études de l'ingénierie d'ontologies : la constitution d'un corpus de documents, une analyse linguistique du corpus, une conceptualisation et finalement l'opérationnalisation de l'ontologie.

Le but de cette étude est de présenter une méthodologie de création semi automatique d'ontologie à partir des textes arabes. Les contraintes et enjeux de ce processus sont identifiés, en soulignant l'importance de considérer cette tâche comme un processus supervisé par un expert du domaine. Notre intervention dans ce projet s'est focalisée sur une ontologie lexicale, en prenant comme modèle l'ontologie WordNet et comme source d'entrée, « les verbes arabes » d'un dictionnaire monolingue contemporain sous forme d'une base de données lexicale. Le verbe, pivot de la phrase, est notre objectif pour la création des concepts en s'appropriant les synsets comme modèle de représentation du sens.

L'algorithme de Markov pour le clustering du graphe, généré par des différents définissants obtenue par une fermeture transitive, nous a permis de détecter les verbes semblables et d'identifier ainsi pour une entrée verbale donnée l'ensemble de ces synonymes.

Mots Clés : Ontologie, construction d'ontologie, Arabs Texts, TALN, Markov clustering.

Summary

A consensual Framework for the construction of ontology starting from texts in four steps has been the fruit of all studies of the engineering of ontologies: the constitution of a corpus of documents, a linguistic analysis of the corpus, a conceptualization and finally the operationalization of the ontology.

The goal of this study is to present a methodology of semi automatic creation of ontology starting from the Arab texts. The constraints and stakes of this process are identified, by stressing the importance to regard this task as a process supervised by an expert of the field. Our intervention in this project was focused on a lexical ontology, by taking as model WordNet ontology, and as source of entrance "the Arab verbs" of a contemporary monolingual dictionary in the form of a lexical database. The verb, pivot of the sentence, is our objective for the creation of the concepts by adapting the synsets like model of representation of the sense.

The algorithm of Markov for the clustering of the graph generated by different defining obtained by a transitive closing, enabled us to detect the similar verbs; and thus to identify for each verbal entry given the whole of these synonyms.

Keywords **: Mots Clés : Ontologie, construction d'ontologie, textes, arabe, TALN, Markov clustering.**