

# 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].

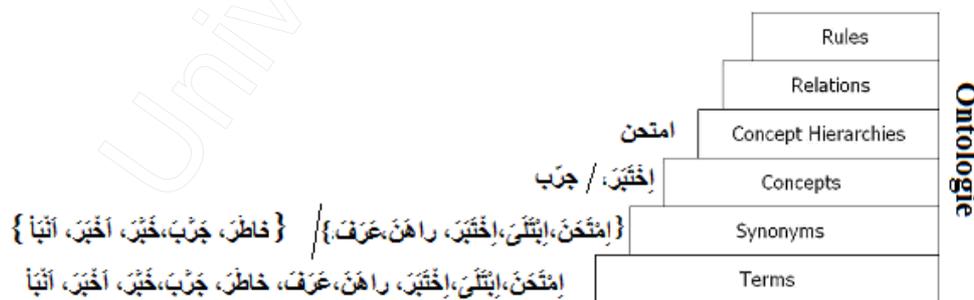


Fig. 35 : Hiérarchie visée par notre approche d'ontologie lexicale

##### 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)<sup>1</sup> 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 dans 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 décrivons 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.

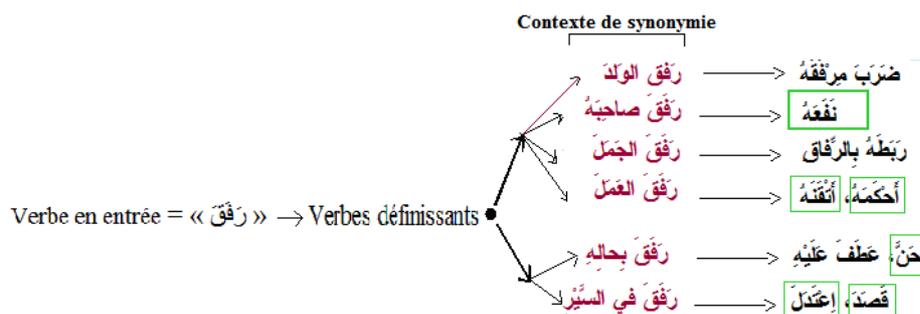
L'hypothèse selon laquelle repose notre l'approche pour la construction automatique des Synsets peut être exprimée de la façon suivante :

### 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 », « أَتَقَنَهُ », « حَنَّ », « عَطَفَ عَلَيْهِ », « عَتَدَلَ », « قَصَدَ ».

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 أبو محمد بن برّي رحمه الله: الأباة لأجمّة القصب، والجمع أباء. قال وربما ذكر هذا الحرف في المعتلّ من الصّحاح وإنّ الهمزة أصلها ياء. قال: وليس ذلك بمذهب سيبويه بل يحملها على ظاهرها حتى يقوم دليل أنّها من الواو أو من الياء نحو: الرّداء لأنه من الرّديّة، والكساء لأنه من الكسوة، والله أعلم.  
@أتأ: حكى أبو علي، في التذكرة، عن ابن حبيب: أتأه أمّ قيس بن ضرار قاتل المقدم، وهي من بكر وائل. قال: وهو من باب أجا(1) (1 قوله قال «وهو من باب الخ» كذا بالنسخ والذي في شرح القاموس وأنشد ياقوت في أجا 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 structurée de la définition lexicographique. El-raïd, 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<sup>1</sup>, 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  $d_1$ .

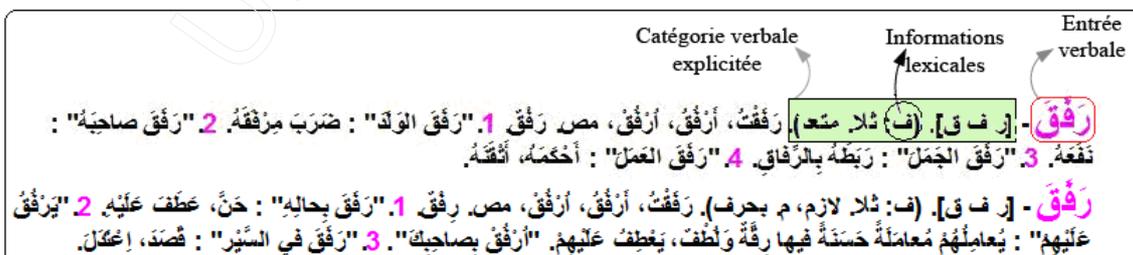


Fig. 36. Structure du dictionnaire de l'approche « El-Ghannye »

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.

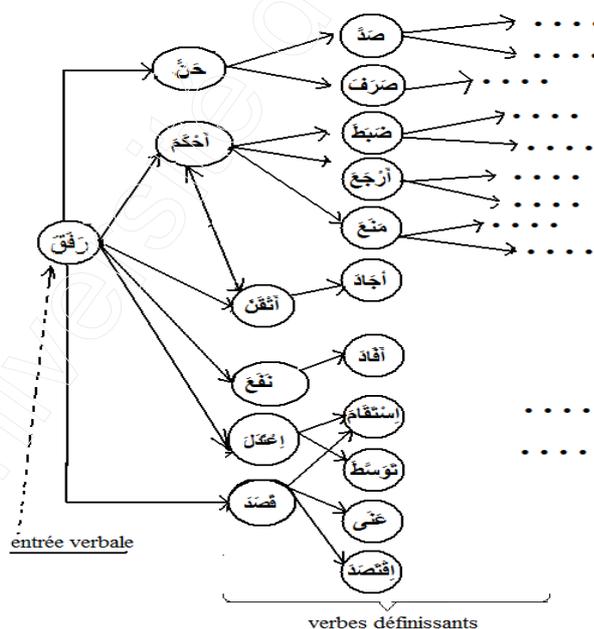


Fig. 37. Sous graphe de  $G_D$  : Relation entre un verbe entrée et ses verbes définissants

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 », « أَنْقَنَهُ », « حَنَّ », « قَصَدَ », « إَعْتَدَلَ » 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 été défini :

$V_E$  : كلمة<sub>1</sub> , كلمة<sub>2</sub> , كلمة<sub>3</sub> , كلمة<sub>4</sub> ...

كلمة<sub>1</sub>, كلمة<sub>2</sub>, ... sont les mots définissants de l'entrée verbale  $V_E$ .

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

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

Contexte définition	Définitions
صَفَقَ الْبَابَ خَلْفَهُ	رَدَّهُ بَعْنَفٍ وَسَمِعَ صَوْتَهُ، أَغْلَقَهُ
صَفَقَتِ الرِّيحُ الْأَشْجَارَ	حَرَكَتَهَا
صَفَقَ الْبَيْعَ	ضَرَبَ يَدَهُ عَلَى يَدِ الْآخَرِ إِعْلَانًا بِالْمُوَافَقَةِ عَلَى الْبَيْعِ
صَفَقَهُ مِنْ بَلَدٍ إِلَى بَلَدٍ	أَخْرَجَهُ ذُلًّا وَقَهْرًا
صَفَقَ الْقَوْمَ عَنْ أَمْرِهِمْ	صَرَفَهُمْ، رَدَّهُمْ
صَفَقَ الطَّائِرُ جَنَاحَيْهِ وَبِهِمَا	حَرَكَهُمَا
صَفَقَ الْبَابَ	رَدَّهُ
صَفَقَ الْعُودَ	طَرَبَ أَوْ تَارَهُ
صَفَقَ الْقَدْحَ	مَلَأَهُ

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 1<sup>er</sup> 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 1<sup>er</sup> patron morphosyntaxique, de ce fait on retiendra les verbes des mots définissants encadrés dans le tableau 07 ci-dessous, soit alors : « أَغْلَقَهُ », « حَرَكَتَهَا », « مَلَأَهُ », « رَدَّهُ », « حَرَكَهُمَا », « رَدَّهُمْ », « صَرَفَهُمْ »

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 ذُلًّا وَقَهْرًا » (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 : « حَرَكَتَهَا » et « حَرَكَهُمَا » 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	$\boxed{\text{فِيهِ/فِيهَا}} \boxed{\text{كَلِمَةً}} : V_E$ <p style="text-align: center;">↑                    ↑ espace            1 seul Mot</p>
3	$\boxed{\text{عَنْهَا/عَنْهُ}} \boxed{\text{كَلِمَةً}} : V_E$ <p style="text-align: center;">↑                    ↑ espace            1 seul Mot</p>
4	$\boxed{\text{عَلَيْهِ/عَلَيْهَا}} \boxed{\text{كَلِمَةً}} : V_E$ <p style="text-align: center;">↑                    ↑ espace            1 seul Mot</p>
5	$\boxed{\text{إِلَيْهِ/إِلَيْهَا/إِيَّاهُ}} \boxed{\text{كَلِمَةً}} : V_E$ <p style="text-align: center;">↑                    ↑ espace            1 seul Mot</p>
6	$\boxed{\text{بِهِ/بِهَا/بِهِمْ}} \boxed{\text{كَلِمَةً}} : V_E$ <p style="text-align: center;">↑                    ↑ espace            1 seul Mot</p>
7	$\boxed{\text{مِنْهُ/مِنْهَا}} \boxed{\text{كَلِمَةً}} : V_E$ <p style="text-align: center;">↑                    ↑ espace            1 seul Mot</p>

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 الْكِتَابِ	قَلَّبَهَا، تَصَفَّحَهَا، أَيْ عَرَضَهَا وَرَقَةً وَرَقَةً
صَفَحَ فِي الْأَمْرِ	نَظَرَ فِيهِ
صَفَحَ جَارَهُ عَنْ حَاجَتِهِ	رَدَّهُ
صَفَحَ التَّوْبَ	جَعَلَهُ عَرِيضاً
صَفَحَهُ بِالسَّيْفِ	ضَرَبَهُ بِعَرَضِهِ لَا بِحَدِّهِ
صَفَحَ فِي الْأَمْرِ	نَظَرَ

Tableau 9 : Définissants du verbe « صَفَحَ »

Selon le premier patron nous retenons les mots définissants suivants : « رَدَّهُ », « نَظَرَ », « قَلَّبَهَا », « تَصَفَّحَهَا » 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 2<sup>ème</sup> et le 3<sup>ème</sup> 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، اِسْتَدَّ عَلَيْهِ
صَهَرَ جِسْمَهُ بِالذَّهْنِ	دَهَنَهُ
صَهَرَهُ إِلَيْهِ	قَرَّبَهُ إِلَيْهِ، أَذْنَاهُ
صَهَرَ الْخَبِزَ	خَلَطَهُ بِالصُّهَارَةِ، أَيِ الشَّحْمِ

Tableau 10 : Définissants du verbe « صَهَرَ »

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

- 1<sup>er</sup> Patron : « أَذَابَ », « أَحْرَقَ », « دَهَنَ », « أَدْنَى »,
- 2<sup>ème</sup> Patron : Aucun
- 3<sup>ème</sup> Patron : Aucun
- 4<sup>ème</sup> Patron : « اِسْتَدَّ »,
- 5<sup>ème</sup> 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 مَوْلَاهُ مِنَ الدَّرْسِ الْأَوَّلِ	أَجَلَهُ وَعَظَّمَهُ
لَا يَهَابُ أَحَدًا	لَا يَحْذَرُ، لَا يَتَّقِي، لَا يَخَافُ أَحَدًا
كَانَ يَهَابُ ظِلَّهُ	يَخَافُ مِنْهُ، كِنَايَةً عَنِ الْجُبْنِ
هَابَ الرَّاعِي بِقَتْمِهِ	صَاحَ بِهَا لِتَتَيْفَ

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 « هَادَ » de la première ligne, puis les deux autres de la deuxième ligne - « زَجَرَهُ » et « صَرَفَهُ » - ensuite vient le 6ème patron pour en ajouter « صَاَحَ ». Quand à la seconde entrée « هَابَ », seulement la phrase courte correspondant au 7ème patron : « بِخَافٍ 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 « لَامُ الْجَزْمِ » 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
هَاتَ الشَّيْءُ	تَحَرَّكَ
هَاتَ لَهُ	أَعْطَاهُ شَيْئًا قَلِيلًا يَسِيرًا
هَاتَ الْقَوْمُ	دَخَلَ بَعْضُهُمْ فِي بَعْضٍ عِنْدَ الْخُصُومَةِ
هَاتَ فِي الْأَمْرِ	عَاتَ، أَفْسَدَ فِيهِ بَعْثًا
هَاتَ التُّرَابَ بِرِجْلِهِ	نَبَشَهُ

Tableau 12 : Définissants du verbe « هَاتَ »

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

$$A \text{ Synonym\_of } B \wedge B \text{ Synonym\_of } C \Rightarrow A \text{ Synonym\_Of } C$$

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 synonymes 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 établi.

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

Contexte	Définition
صَفَّصَفَ الرَّجُلُ	صَارَ فِي الْفَلَاةِ مُنْقَرِداً
صَفَّصَفَ الطَّائِرُ	زَفَزَقَ

Tableau 13 : Définissants du verbe « صَفَّصَفَ »

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

Contexte	Définition
زَفَزَقَ الطَّائِرُ	تَفَرَّدَ بِصَوْتِهِ
زَفَزَقَ الطَّائِرُ صِغَارَهُ	زَقَى، أَطْعَمَهُمْ
زَفَزَقَ الطِّفْلَ	جَعَلَهُ يَرْفُصُ

Tableau 14 : Définissants du verbe « زَفَزَقَ »

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

Contexte	Définition
زَقَى الطَّائِرُ صِغَارَهُ	أَطْعَمَهُمْ بِمِنْقَارِهِ، بِقِيهِ
زَقَى الدَّبِيحَةَ	سَلَخَهَا مِنْ قِبَلِ رَأْسِهَا إِلَى رِجْلِهَا

Tableau 15 : Définissants du verbe « زَقَى »

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

Contexte	Définition
أَطْعَمَهُ خَبِيراً أَوْ بَيْضاً	قَدَّمَ لَهُ
يُطْعِمُ الْجِياعَ	يَقُوْنُهُمْ، يَغْدِيهِمْ، يَقْدِمُ لَهُمْ طَعاماً
أَطْعَمَ الْفُصْنَ بَقْضِ أَخْرَ مِنْ غَيْرِ شَجَرِيهِ	طَعَّمَهُ، وَصَلَهُ بِهِ لِيَتَّكُونَ مِنَ الْفُصْنَيْنِ غَصْنَ نالِثَ يُعْطِي ثَمراً أَخْرَ
أَطْعَمَ الأَكْلُ	صارَ لَهُ طَعْمٌ
أَطْعَمَ الشَّجَرَ	أَدْرَكَ ثَمْرَهُ وَطابَ
أَطْعَمَهُ اللهُ	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 « صَفِّفَ » :

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

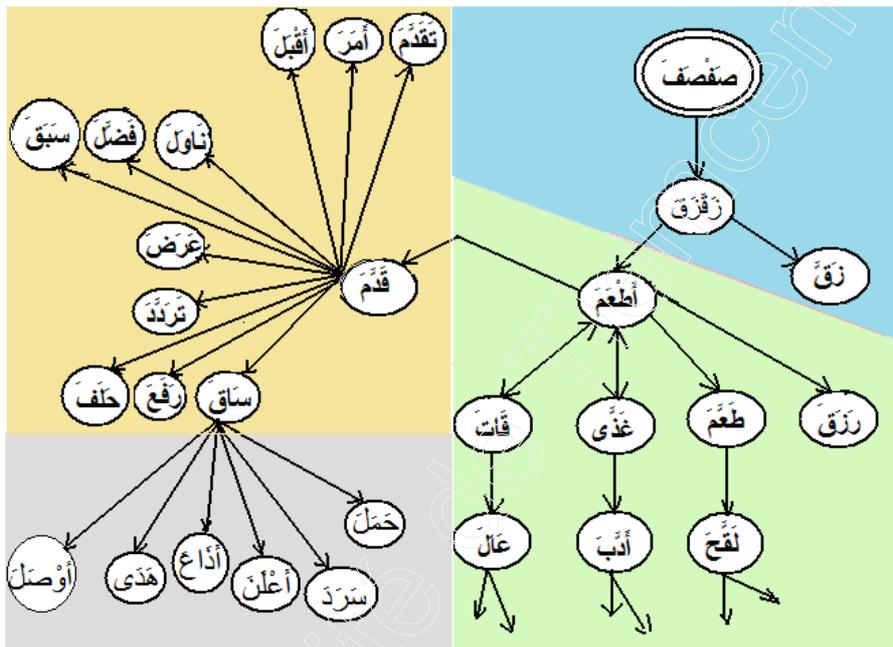


Fig. 38 : Graphe  $G_s$  du Verbe « صَفِّفَ »

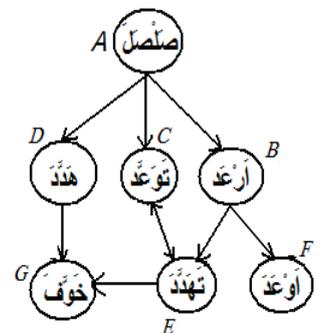
Chaque entrée verbale du dictionnaire est en relation avec les verbes synonymes présents dans sa définition, l'ensemble des couplets [entrée verbale – Verbe synonymie] représentant cette relation, et qu'on a obtenue par l'application des patrons morpho syntaxique définie sur l'intégralité du dictionnaire, 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  $n \times n$  notée encore M telle que :

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



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

On peut tracer sa matrice d'adjacence  $M_{7 \times 7}$  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 :

$$V_s =$$

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

$$M_{7 \times 7} =$$

	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

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 dans 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  $\lambda$  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  $F_{ij}$  correspond au nombre d'instances de synonymie existantes entre  $i$  et  $j$ .

Bien qu'en utilisant un seul 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  $F$ ,  $F_{ij} = F_{ij} + F_{ij} * \beta$  ;
  - (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  $P_{ij} > \alpha$ ,  $i$  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  $C = C_1, C_2, \dots, C_n$  tel que  $B = C_1 \cup C_2 \cup \dots \cup C_n$  ;
    - (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  $a$ ,  $S_a \subset D$ , 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).

$$\text{Alors } Pa_i = \frac{n_{ai}}{Sa_i} \text{ est calculé.}$$

Note :

Si R est une relation transitive, la procédure peut profiter d'un niveau de transitivité pour l'appliquer au réseau :  $x R y \wedge y R z \rightarrow x R z$ . 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  $Pa_i < 4$ .
- 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  $M[N,N] \Rightarrow M$  devient matrice stochastique stocker le résultat dans la matrice S :

$$S(i, j) = \frac{M(i, j)}{\sum_{k=1}^{K=n} M(k, j)} \quad (\text{Diviser le score } M[i,j] \text{ par le total des score de la colonne } \mathbf{j}).$$

2) Faire la multiplication matricielle suivante (calculer  $S^2$ ), si  $e=3$  faire  $S^3=S^3=S*S*S$ .

$$S^2 = S(i,j) * S(i,j). \quad \text{Stocker le résultat dans } S^2.$$

- 3) Chaque élément de  $S^2(i,j)$  de  $S^2$ , le remplacer par  $S^2(i,j)^{1.6}$  (si  $r=1.6$ ). (élever à la puissance  $r$  chaque élément de la matrice) puis **normaliser**  $S^2$  comme dans l'étape 1.
- 4) Comparer  $S$  et  $S^2$  : Si convergence (égalité) alors arrêté. Sinon faire  $S=S^2$  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 quatre clusters :

$$\begin{pmatrix}
 1.000 & -- & -- & -- & -- & 1.000 & 1.000 & -- & -- & 1.000 & -- & -- \\
 -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\
 -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\
 -- & 1.000 & 1.000 & -- & 1.000 & -- & -- & -- & -- & -- & -- & -- \\
 -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\
 -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\
 -- & -- & -- & 0.500 & -- & -- & -- & 0.500 & 0.500 & -- & 0.500 & 0.500 \\
 -- & -- & -- & 0.500 & -- & -- & -- & 0.500 & 0.500 & -- & 0.500 & 0.500 \\
 -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\
 -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & --
 \end{pmatrix}$$

$M_{mcl}^{ns}$

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 été 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 données lexicale a été réalisé avec le serveur de base de données Microsoft SQL Server 2008 Express Edition. La plateforme 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 proposée, ce dernier a permis 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<sup>1</sup>. Le dictionnaire analysé « Al ghannye » est constitué de près 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 permis 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'ambiguïté pour la déduction des entrées verbales.

---

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

### 3.3. Architecture générale de l'application

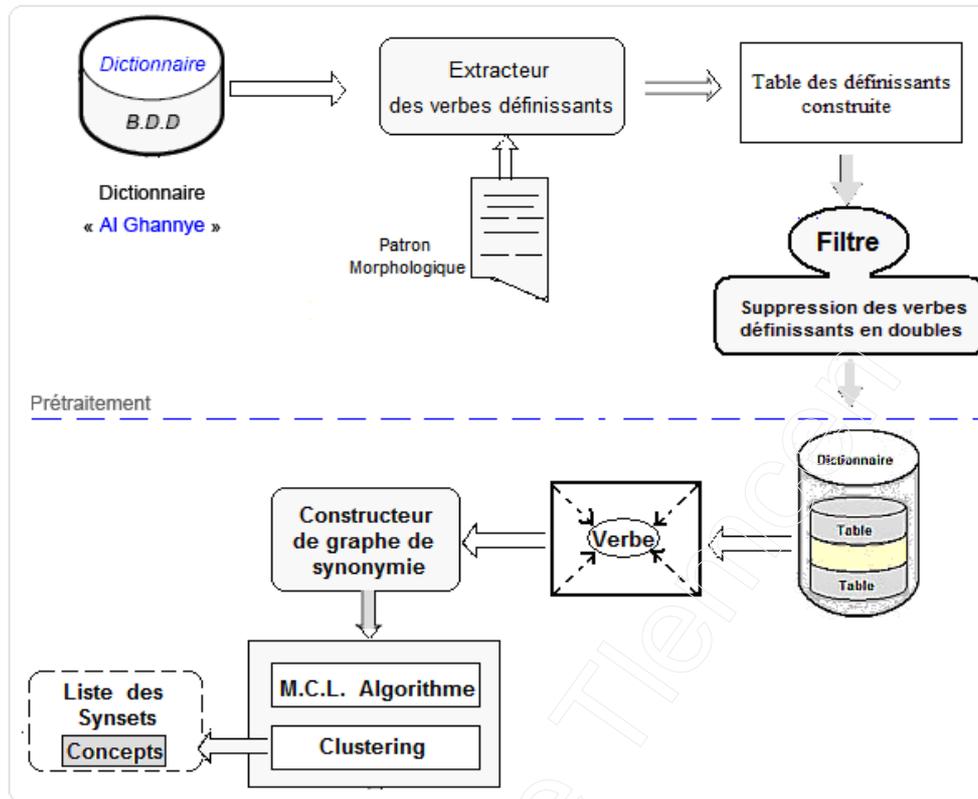


Fig. 39 : Architecture de la solution

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).

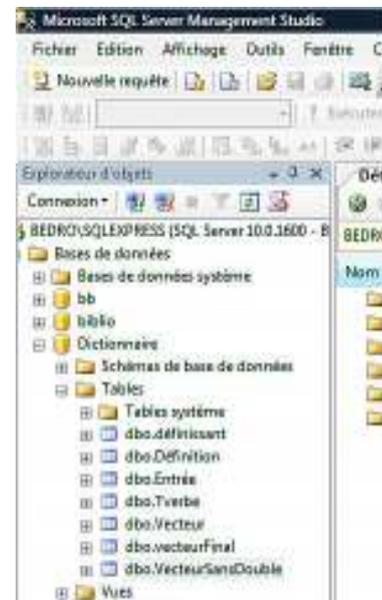


Fig. 40 Bases de données et tables utilisées

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 <b>Non</b> 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 ( $\approx 1$  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

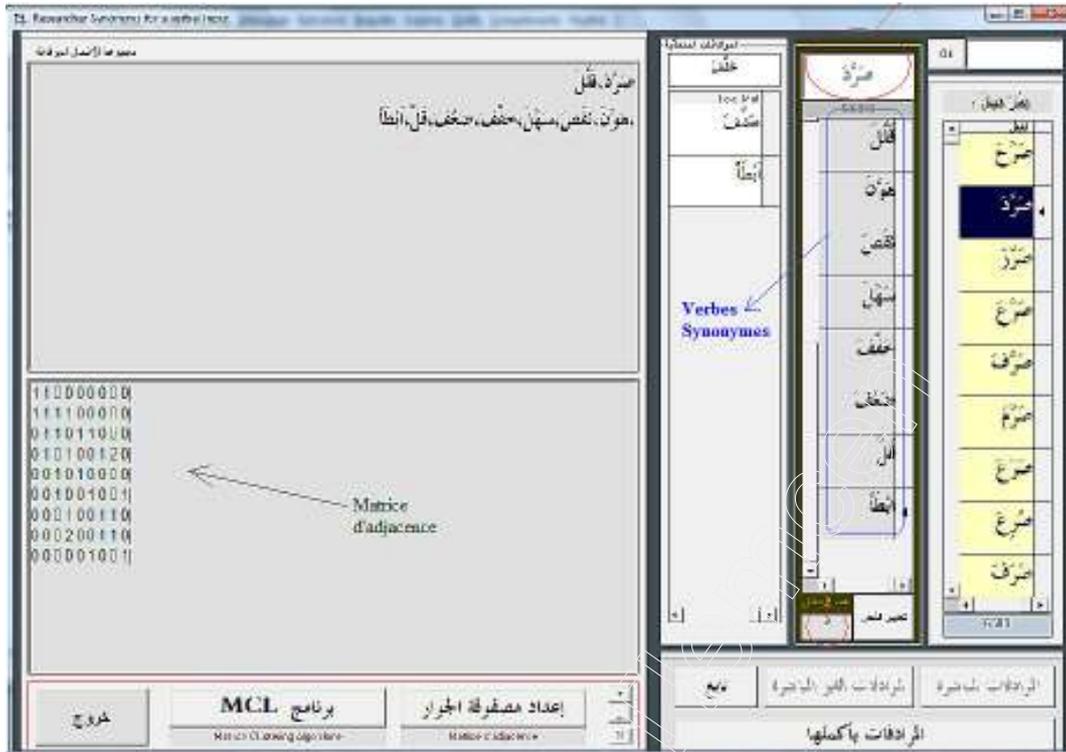


Fig. 43 : Interface de traitement : Etape 1

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 : « صَرَدَ » donne le vecteur final suivant :

صَرَدَ	قَلَّ	هَوَّنَ	نَقَصَنَ	سَهَّلَ	خَفَّفَ	ضَعُفَ	قَلَّ	أَبْطَأَ	نَدَّرَ	تَضَاعَلَ	تَأَخَّرَ	سَقَطَ	جَرَبَ	...
--------	-------	---------	----------	---------	---------	--------	-------	----------	---------	-----------	-----------	--------	--------	-----

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 MCL $n \times n$  (pour n verbes définissants) pour mettre en transparence cette étape de programmation :

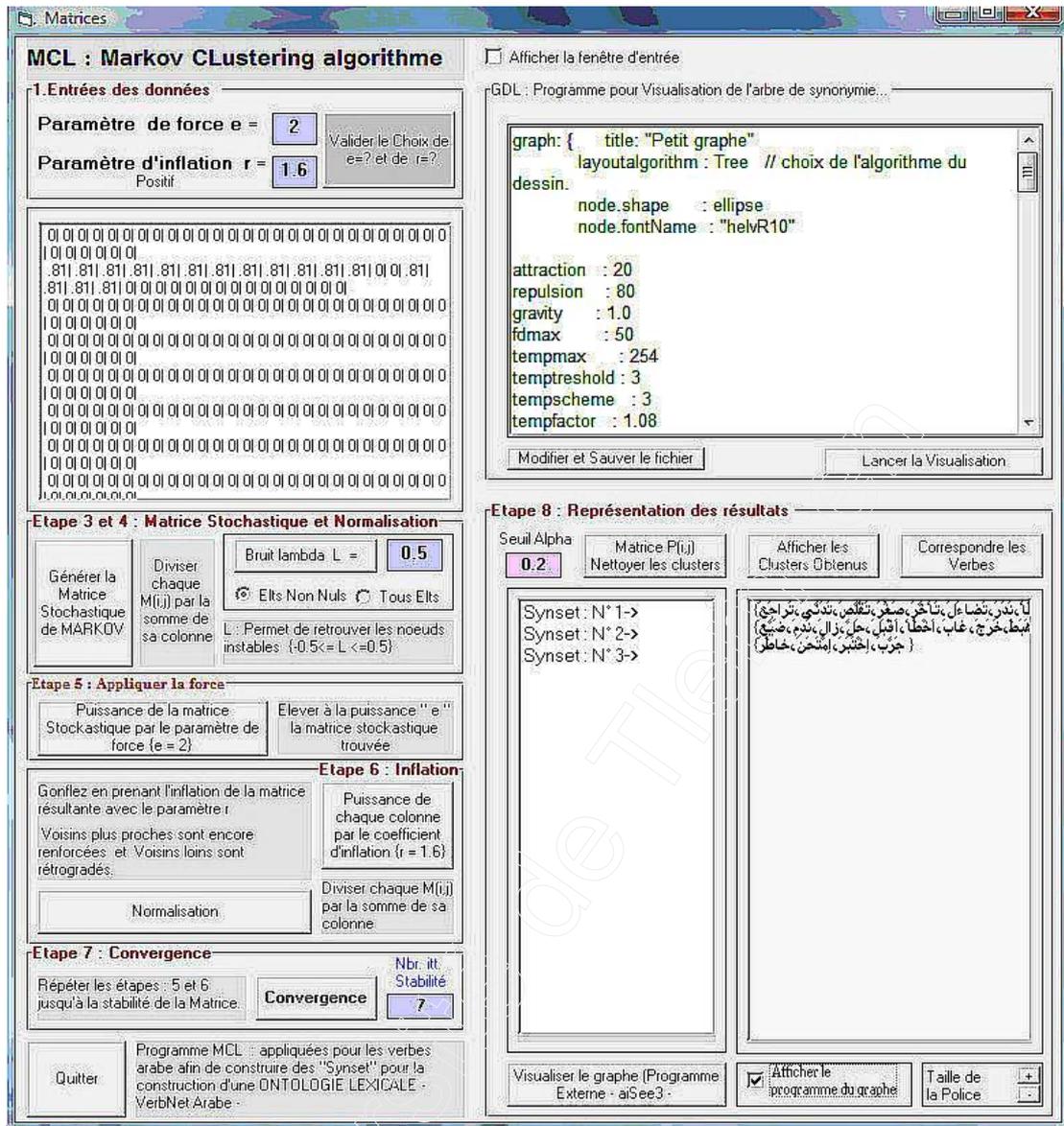


Fig. 44 : Interface de traitement : Etape 2

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 permis 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.

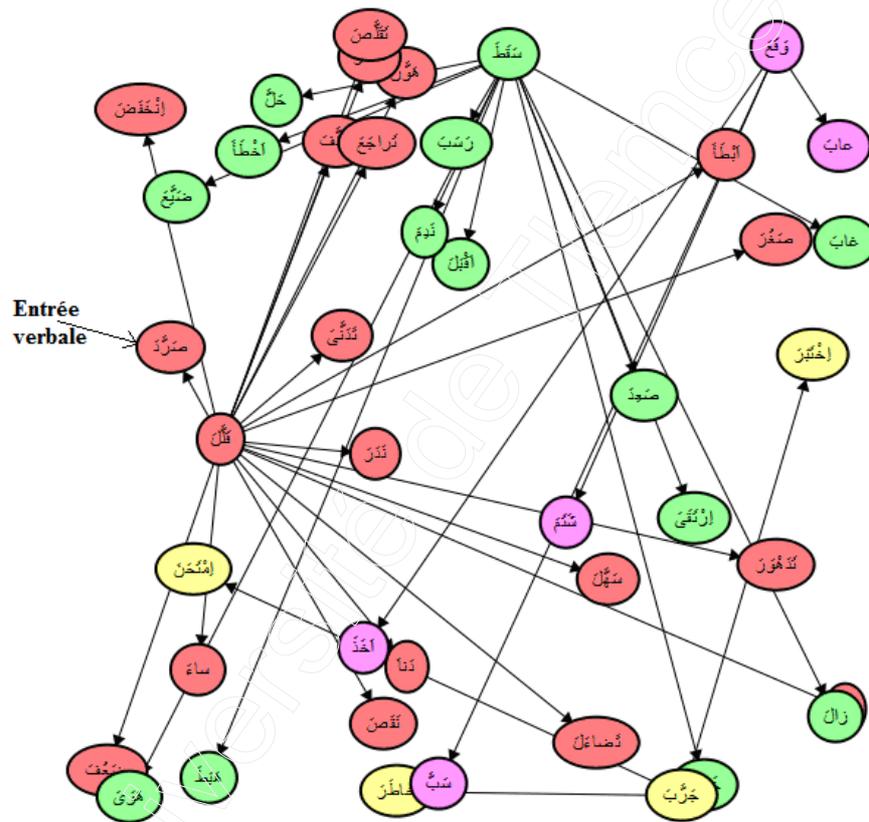


Fig. 45 : GDL du graphe de synonymie d'une partie de l'entrée verbale « صَرَدَ »

#### 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	Nb r.
100%	0	1. Analyser et expliquer	1. حَلَّلَ، زَوَّجَ، شَرَّحَ، فَسَّرَ، بَيَّنَّ، شَرَّحَ، أَبَانَ	01
20%	3	1. Dissocier pour réparer 2. Echanger 3. Faire apparaître 4. Eclaircir	1. فَكَّ، حَلَّ، فَصَّلَ، عَدَّلَ، سَوَّى، أَشْرَعَ، 2. اسْتَبَدَّلَ، غَيَّرَ، حَوَّلَ، نَقَلَ، أَحَالَ، أَفْشَى 3. أَظْهَرَ، أَوْضَحَ، أَبْرَزَ، أَخْرَجَ، أَمْدَرَ، أَدَاعَ، أَحْكَمَ، نَشَرَ، 4. أَنْصَحَ، بَانَ، ظَهَرَ، أَرشَدَ، نَشَرَ، أَقَامَ، أَنْصَحَ، زَكَّى 5. قِيلَ، سَحَبَ، أَوْرَقَ	02
65%	0	1. Viser 2. Entreprendre	1. سَدَّدَ، صَوَّبَ، فَتَحَ، وَجَّهَ، أَصْلَحَ، عَلَّمَ 2. شَرَعَ، قَدَّمَ، اتَّخَذَ،	03
30%	2	1. Disperser 2. Verser 3. Embellir	1. فَضَّ، فَرَّقَ، شَتَّتَ، قَطَعَ، قَسَمَ، وَزَّعَ، 2. سَكَّبَ، ذَرَفَ، أَفَاضَ، صَبَّ، سَالَ، أَسَالَ، إِنْحَدَرَ، سَرَّحَ 3. حَسَّنَ، زَيَّنَ، رَقَّ، 4. خَوَّنَ، إِشْتَبَقَ	04
100%	1	1. Achever 2. Vouloir	1. قَضَى، أَفْنَى، أَهْلَكَ، نَالَ، اسْتَعْرَقَ، أَنْهَى، قَتَلَ، وَفَى، أَدَّى، أَنْقَدَ، أَبْلَغَ، أَوْصَى، 2. أَرَادَ، أَخَذَ، أَمَرَ،	05
100%	0	1. Humidifier	1. رَطَّبَ، بَلَّ، أَرْطَبَ، أَلَانَ، نَعَّمَ	06
100%	0	1. Faire en sorte que ...	1. جَعَلَ، اعْتَمَدَ، خَلَقَ، صَنَعَ، وَضَعَ، أَلْقَى، قَصَدَ عَيْنَ، رَمَى،	07
80%	1	1. Casser et Broyer	1. كَسَّرَ، حَطَّمَ، هَشَّمَ، كَسَرَ، قَتَّتَ، ذَقَّ، 2. حَانَ	08
100%	0	1. Etre rempli ...	1. أَتَمَّ، امْتَلَأَ، اكْتَمَلَ	09
90%	1	1. Commencer	2. بَدَأَ، انْطَلَقَ، أَنْشَأَ، انْتَقَلَ، حَدَّثَ، حَصَلَ، انْفَحَمَ	10
100%	0	1. Plonger dans ...	1. خَاضَ، أَسْرَعَ، خَلَطَ، حَرَّكَ	11
100%	0	1. Introduire	1. مَهَّدَ، مَهَّدَ، بَسَطَ، سَهَّلَ	12

Tableau 17 : Evaluation verbe « حَلَّلَ » avec r = 1.6

Taux moyen de synsets	Nombres de verbes impertinents	Différents Sens évoqué du sysnet	Synsets calculés A parit d'une entrée verbale	Nbr.
50%	0	1 Examiner, 2 - s'expérimenter 3 - informer	.1 {إِمْتَحَنَ، ابْتَلَى، اخْتَبَرَ، رَاهَنَ، عَرَفَ} .2 {حَاطَرَ، جَرَّبَ، خَبَّرَ} .3 {أَخْبَرَ، أَنْبَأَ}	01
40%	1	1 - Patienter، صبر، 2 - se faire tout petit 3 – tourner le dos à	.1 {صَبَرَ، تَجَلَّدَ، تَحَمَّلَ، اِحْتَمَلَ، حَبَسَ، أَمْسَكَ} تَصَبَّرَ، أَطَاقَ، أَغْضَى، سَكَتَ .2 {إِرْتَحَلَ، شَكَرَ، قَبَضَ، كَفَّ، اِمْتَنَعَ، صَمَتَ} أَظْلَمَ	02
100%	0	1 – maîtriser	.1 {صَبَطَ، أَتَقَنَ، أَحْكَمَ، كَبَحَ}	03
90%	1	1- Venir	.3 قَبِلَ، أَتَى، حَلَّ، هَبَّ، .4 رَفَعَ	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 sysnet	Synsets calculés A parit d'une entrée verbale	Nbr.
100%	0	1	{أَفَقَ، دَبَّغَ}	01
50%	0	1 2 3 4	{عَالَجَ، دَاوَى، عَايَنَ} {زَاوَلَ، مَارَسَ، خَدَمَ، سَعَى} {صَبَّ، حَاوَلَ} {إِشْتَاقَ، 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 résultats satisfaisants du moment où un nombre de verbes impertinents inférieur 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 où beaucoup trop de sens totalement distincts 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 provenir du choix de la valeur du paramètre  $r$  (inflation) de l'algorithme MCL. Le paramètre d'inflation  $r$  étant généralement choisi dans un intervalle entre 1,6 et 2 [OLI09], c'est la valeur 1,6 qui a été suggérée dans [OLI09] après plusieurs tests puisque c'est avec cette valeur que le clustering le plus satisfaisant a été obtenu mais cette valeur qu'on a appliquée 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 = 2.2	Synsets avec r =1.6
1. { اَمْتَحَنَ، اِبْتَلَى، اِخْتَبَرَ، عَرَفَ، جَرَّبَ، حَاطَرَ، رَاهَنَ }	{ اَمْتَحَنَ، اِبْتَلَى، اِخْتَبَرَ، رَاهَنَ، عَرَفَ }
{ }	{ حَاطَرَ، جَرَّبَ، خَبَّرَ }
2. { خَبَّرَ، اَخْبَرَ، اَنْبَأَ }	{ اَخْبَرَ، اَنْبَأَ }
3. { صَبَرَ، تَجَلَّدَ، حَبَسَ، تَصَبَّرَ، اَطَاقَ }	{ صَبَرَ، تَجَلَّدَ، تَحَمَّلَ، اِحْتَمَلَ، حَبَسَ، اَمْسَكَ }
4. { حَتَمَلَ، تَحَمَّلَ، اَطَاقَ، اَغْضَى، اِرْتَحَلَ، شَكَرَ }	{ تَصَبَّرَ، اَطَاقَ، اَغْضَى، سَكَتَ }
5. { بَطَّ، اَثَقَنَ، اَحْكَمَ، كَبَحَ }	{ اِرْتَحَلَ، شَكَرَ، قَبَضَ، كَفَّ، اِمْتَنَعَ، صَمَتَ }
6. { مَسَكَ، قَبَضَ، كَفَّ، اِمْتَنَعَ، سَكَتَ، صَمَتَ }	اَطْلَمَ
7. { قَبَلَ، اَتَى، حَلَّ، هَبَّ، رَقَعَ }	{ ضَبَطَ، اَثَقَنَ، اَحْكَمَ، كَبَحَ }
8. { صَفَحَ، قَلَّبَ، تَصَفَّحَ، رَدَّ، نَظَرَ }	{ قَبَلَ، اَتَى، حَلَّ، هَبَّ }
	رَفَعَ
	{ صَفَحَ، قَلَّبَ، تَصَفَّحَ، رَدَّ، نَظَرَ }

1. { اَفَقَ، دَبَغَ }	1. { اَفَقَ، دَبَغَ }
2. { عَالَجَ، دَاوَى، زَاوَلَ، مَارَسَ، خَدَمَ، حَاوَلَ، عَايَنَ، سَعَى }	1. { عَالَجَ، دَاوَى، عَايَنَ }
3. { صَبَّ، اِشْتَاقَ، رَقَّ }	2. { زَاوَلَ، مَارَسَ، خَدَمَ، سَعَى }
	3. { صَبَّ، حَاوَلَ }
	4. { اِشْتَاقَ، 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 été redécouvert et n'ont pas changé.
- Les synsets/clusters à grosse granularité obtenus précédemment ont été fractionné en des synsets/cluster plus fin et qui se rapproche beaucoup plus des sous synsets correcte proposés par l'expert.

Pour le verbe « أَفَقَّ » 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 été 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 été 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-sortede, 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 tâche 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 été déjà accompli et comme travail futur, nous prévoyons de compléter la solution proposée 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 permettront 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.