Constitution d'une ressource sémantique arabe à partir d'un corpus multilingue aligné

Authoul Abdulhay

UNIVERSITÉ DE GRENOBLE

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Informatique et Sciences du Langage**

Arrêté ministériel : 7 août 2006

Présentée par

Authoul Abdul Hay

Thèse dirigée par **Olivier Kraif**

codirigée par **Francis Grossmann**

préparée au sein du **Laboratoire** de linguistique et didactique des langues étrangères et maternelles

dans **l'École Doctorale** Langues Littératures et Sciences humaines

Constitution d'une ressource sémantique arabe à partir de corpus multilingues alignés

Thèse soutenue publiquement le **23 novembre 2012**, devant le jury composé de :

M. Mathieu LAFOURCADE
MCF, HDR, laboratoire LIRMM, Montpellier, Rapporteur.
M. Jean-Louis Duchet
Professeur émérite, MSHS, Poitiers, Rapporteur et président du jury.
M. SEMMAR Nasredine
Chercheur, Laboratoire LVIC, Gif-sur-Yvette, Examinateur.
M. Francis Grossmann
Professeur, Laboratoire LIDILEM, Grenoble, Co-directeur de thèse.
M. Olivier Kraif
Maître de conference, Laboratoire LIDILEM, Grenoble, Directeur de thèse.

Université Joseph Fourier / Université Pierre Mendès France / Université Stendhal / Université de Savoie / Grenoble INP

REMERCIEMENTS

Je tiens à remercier à toutes les personnes grâce auxquelles j'ai pu mener à bien cette thèse :

Je tiens tout d'abord à remercier mon directeur de thèse, Olivier Kraif, pour m'avoir guidée tout au long de ces 8 années, une aventure qui a commencé dès la maîtrise et qui s'est poursuivie jusqu'à la fin de cette thèse. Je lui remercie pour la qualité de son encadrement, pour sa présence continuelle. Il a toujours su me guider dans mon travail de recherche et m'a apporté une aide inestimable dans l'élaboration de mon mémoire et de ma thèse. Je le remercie énormément pour sa patience, sa relecture, ses remarques et ses multiples corrections scientifiques et linguistiques…

Je remercie également mon co-directeur de thèse, monsieur Francis Grossmann, pour son assistance, ses conseils précieux, et son soutien indéfectible.

Mes respects et ma gratitude vont également aux membres de jury qui m'ont fait l'honneur de juger ce travail. Aux rapporteurs de cette thèse, Messieurs Mathieu Lafourcade et Jean-Louis Duchet dont les remarques m'ont permis d'améliorer le contenu de ce document. A Monsieur Nasredine Semmar, l'examinateur, pour avoir accepté de participer au jury.

Je tiens ainsi à exprimer ma grande reconnaissance à mes proches pour leur soutien :

Je suis infiniment reconnaissante à mes parents pour leur amour, leur patience, leur confiance, leur soutien sans faille et à tout ce qu'ils ont pu m'apporter pour franchir les obstacles les plus difficiles.

Mes sincères remerciements à mes sœurs, Arwa pour ses bons repas, pour ces belles discussions "détaillées", et surtout pour sa tendresse. Enas, je te remercie également "bien-sûr" pour ton soutien financier, pour nos grandes balades, pour me partager tous les beaux et mauvais moments de ma vie et me partager l'obsession de la découverte des nouveaux restos…

Mon mari, Loai, qui a changé positivement ma vie, qui a toujours cru en moi, même dans les moments plus difficiles, qui a supporté mes humeurs au gré de cette thèse et qui a du faire beaucoup des sacrifices et des concessions.

Je tiens à remercier à mon cœur, mon neveu Basil, qui m'a donné toujours l'envie de continuer et de ne jamais abandonner mes rêves, qui m'a lassé voir le bon coté de la vie, qui m'a fait vivre des beaux moments inoubliables et qui m'a toujours surpris par son niveau en philosophie.

Mes remerciements sont aussi adressés à mes amis :

Shefa Abdulhay pour m'avoir toujours soutenue dans mes choix, pour nos discussions sans fin, parce qu'elle regarde dans la même direction que moi et bien plus encore, pour être mon ombre et simplement pour le fait qu'elle sera toujours là pour moi.

Chircu Catalina pour tous les moments salutaires de détente, pour tous les sourires que je lui dois et toute l'aide qu'elle m'a apportée.

Wafa Alkhateeb, ma chère amie, à qui je souhaite la réalisation de ses projets et à qui je remercie pour tous les moments de joie que j'ai partagés avec elle.

Ali Kobeissi qui m'a appris à connaitre la réalité de la vie, je lui remercie pour tous les conseils (utiles ou non) qui m'ont encouragé à achever mon rêve, pour tous les beaux moments partagés et pour toutes les heures qu'il a consacrées à m'expliquer les bases de l'informatique.

Enfin j'adresse mes remerciements à Kamel Amirou "le grand joueur de guitare", Abeer Albsool, Nadia Halazoun pour leur gentillesse et pour tout ce qu'ils m'ont offert tout au long de cette thèse.

Résumé : Cette thèse vise à la mise en œuvre et à l'évaluation de techniques d'extraction de relations sémantiques à partir d'un corpus multilingue aligné. Ces relations seront extraites par transitivité de l'équivalence traductionnelle, deux lexèmes possédant les mêmes équivalents dans une langue cible étant susceptibles de partager un même sens. D'abord, nos observations porteront sur la comparaison sémantique d'équivalents traductionnels dans des corpus multilingues alignés. A partir des équivalences, nous tâcherons d'extraire des "cliques", ou sous-graphes maximaux complets connexes, dont toutes les unités sont en interrelation, du fait d'une probable intersection sémantique. Ces cliques présentent l'intérêt de renseigner à la fois sur la synonymie et la polysémie des unités, et d'apporter une forme de désambiguïsation sémantique. Elles seront créées à partir de l'extraction automatique de correspondances lexicales, basée sur l'observation des occurrences et cooccurrences en corpus. Le recours à des techniques de lemmatisation sera envisagé. Ensuite nous tâcherons de relier ces cliques avec un lexique sémantique (de type Wordnet) afin d'évaluer la possibilité de récupérer pour les unités arabes des relations sémantiques définies pour des unités en anglais ou en français. Ces relations permettraient de construire automatiquement un réseau utile pour certaines applications de traitement de la langue arabe, comme les moteurs de question-réponse, la traduction automatique, les systèmes d'alignement, la recherche d'information, etc.

Mots-clés : Corpus multilingue aligné, désambigüisation sémantique, cliques, ressource sémantique arabe.

Abstract: This study aims at the implementation and evaluation of techniques for extracting semantic relations from *a multilingual aligned corpus*. Firstly, our observations will focus on the semantic comparison of translational equivalents in multilingual aligned corpus. From these equivalences, we will try to extract "*cliques*", which ara maximum complete related sub-graphs, where all units are interrelated because of a probable semantic intersection. These cliques have the advantage of giving information on both the synonymy and polysemy of units, and providing a form of semantic disambiguation. Secondly, we attempt to link these cliques with a semantic lexicon (like WordNet) in order to assess the possibility of recovering, for the Arabic units, a semantic relationships already defined for English, French or Spanish units. These relations would automatically build a semantic resource which would be useful for different applications of NLP, such as Question Answering systems, machine translation, alignment systems, Information Retrieval…etc.

Keywords: Multilingual aligned corpus, semantic disambiguation, cliques, Arabic semantic resource.

Table des matières

Listes des figures

Liste des tableaux

Convention d'écriture

– En cas d'ambiguïté, les unités tirées de notre corpus seront préfixées par deux lettres ISO indiquant leur langue (en : anglais, fr : français, ar : arabe, es : espagnol) : p. ex. *en-book*.

– Les unités lexicales constituant les cliques multilingues seront écrites en italique et préfxées par leur code de langue et de catégories : p. ex. l'unité lexical *fr-Verb-écrire.*

– Les exemples arabes seront écrits en caractères arabes, suivis de leur translittération Buckwalter et de leur traduction en français : ex. (ar-كتب) (trans.ktb/trad. écrire).

– Les sens, les gloses et les ILI-RECORDS seront écrites entre barres obliques / /. p. ex. (fr-Verb-viser en-Verb-target ar-Verb-yhdf es-Verb-apuntar) -> /cible/

– Les types de relations et les valeurs sémantiques seront écrits entre guillemets simples droits ' ' : p. ex. 'partie_de'.

– Les signes de segmentation seront écrits entre parenthèses ( ) : p. ex. (!?;).

– Les noms de fichiers seront écrits entre parenthèses ( ) : p. ex, (script.pl).

– Les cliques seront écrites entre parenthèses ( ) : p. ex. la clique (fr-N-économie, en-N-saving, it-N-risparmio, de- N-Einsparung).

– Les clusters seront écrits entre accolades {} :ex. le cluster :{(fr-N-économie fr-N-épargne de-N-Einsparung en-N-saving it-N-risparmio) (fr-N-économie fr-N-épargne de-N-Einsparung en-N-saving es-N-ahorro) (fr-N-économie fr-N-épargne fr-N-gain en-N-saving es-N-ahorro)}.

Introduction

Pourquoi faire des recherches sur les réseaux sémantiques ?

Les réseaux sémantiques ont été très utilisés dans le domaine de l'intelligence artificielle, et notamment pour le TAL, depuis les années 1960. Ils ont été mis en oeuvre dans différents contextes tels que l'extraction de connaissances, la recherche d'information ou le résumé automatique. Les réseaux sémantiques ont été appliqués très tôt pour répondre au problème de la représentation du sens des mots.

Ces réseaux montrent une bonne adaptation à la représentation du langage naturel, et la capacité de modéliser toute forme de connaissances que l'on peut représenter dans un système symbolique (Hendrix, 1979). Ainsi, ils ont pu être employés efficacement dans quelques applications visant la désambigüisation sémantique, par exemple pour la recherche d'un équivalent traductionnel correct en traduction automatique (Marrafa et al., 2007), ou la désambiguïsation des termes d'une requête en recherche d'information (Baziz et al., 2003).

La création de réseaux sémantiques multilingues pourrait s'avérer très utile, notamment dans ces deux domaines - traduction automatique et recherche d'information interlingue - dans la perspective prometteuse mais encore assez utopique de ce qu'on appelle le Web sémantique.

Motivations

La création d'un réseau sémantique en fonction d'objectifs spécifiques est une opération complexe et coûteuse à mettre en œuvre. C'est pour cela qu'il devient primordial, pour une langue donnée, de bénéficier de réseaux sémantiques génériques déjà développés, afin de rattraper l'écart technologique en termes de contenus, de services et d'usages entre les langues et les cultures du monde sur les réseaux d'information.

En fait, lorsqu'on tente d'établir une ressource sémantique pour une langue quelconque à partir de zéro, comme c'est le cas pour l'arabe, on peut profiter d'un réseau sémantique préexistant comme WordNet pour l'anglais (Miller, 1990) qui a été utilisé largement dans ce but (Resnik, 1995, Carroll et McCarthy 2000, Hawkins et Nettleton, 2000). Il est clair que pour un large éventail d'applications dans ce contexte d'élaboration d'une ressource sémantique, WordNet est devenu un standard *de facto*, malgré certaines limites et

imperfections qu'on peut lui reprocher, tels que la circularité, les erreurs, les incohérences, et l'inadéquation de son organisation des sens à d'autres langues que l'anglais (Mallak, 2011).

Objectif et hypothèse de travail

Ce travail vise à étudier la possibilité de constituer une ressource sémantique pour la langue arabe en bénéficiant de réseaux sémantiques de type WordNet préexistants, notamment les réseaux d'EuroWordNet[1]. Notons que les synsets, dans l'architecture de WordNet, représentent l'intersection sémantique d'un ensemble d'unités, et constituent l'identification implicite d'une acception (sens), en organisant les unités selon deux propriétés : synonymie et polysémie. La synonymie dans un wordnet monolingue est basée sur l'équivalence (souvent partielle) entre les unités regroupées dans synset.

La Figure 1 donne les différents synsets de WordNet pour le nom anglais *situation* : *(situation, state of affairs) (situation, position) (situation) (site, situation) (position, post, berth, office, spot, billet, place, situation).* Chacun de ces synsets correspond à une certaine acception (*sense*), explicité par une glose, mais surtout caractérisé par un ensemble d'unités synonymes susceptibles de partager cette acception.

[1] Cf. http://www.illc.uva.nl/EuroWordNet/

Figure 1 : Exemple de synsets de WordNet pour le nom *en-situation*[2]

Ainsi la mise en évidence de l'équivalence de certaines unités, autour d'un sens donné, conduit également à une prise en compte du fait polysémique, par le fait qu'une même unité est susceptible d'intervenir dans différents synsets. La figure 2 montre la structuration interne des données d'EuroWordNet, qui constitue une extension multilingue de WordNet : un certain sens (WORD_MEANING), correspondant à une partie du discours (PART_OF_SPEECH), est lié à une ou plusieurs formes (LITERAL) pour lesquelles le sens correspond à un numéro d'acception (SENSE). Des relations internes à la langue (INTERNAL_LINKS), correspondant à différents types de relations sémantiques (RELATION) permettent de pointer d'autres sens (TARGET_CONCEPT) qui correspondent à une acception précise d'une forme donnée (LITERAL/SENSE). Par ailleurs des liens d'équivalence traductionnelle (EQ_LINKS) permettent de pointer des sens de Princeton WordNet repérés par leur numéro d'index (WORDNET_OFFSET).

[2]Cf.http://wordnetweb.princeton.edu/perl/webwn/webwn?s=situation&sub=Search+WordNet&o2=&o0=1&o8=1&o1=1&o7=&o5=&o9=&o6=&o3=&o4=&h=, consulté en juin 2012

Gloss: "an arrangement of similar objects in a row or in tiers; "he operated a bank of switches"	**Gloss: a supply or stock held in reserve esp for future emergency use; "the Red Cross has a blood bank for emergencies"**
0 @32739@ WORD_MEANING 1 PART_OF_SPEECH "n" 1 VARIANTS 2 LITERAL "bank" 3 SENSE 4 3 EXTERNAL_INFO 4 SOURCE_ID 1 5 TEXT_KEY "05364891-n" 1 INTERNAL_LINKS 2 RELATION "has_hyperonym" 3 TARGET_CONCEPT 4 PART_OF_SPEECH "n" 4 LITERAL "array" 5 SENSE 4 1 EQ_LINKS 2 EQ_RELATION "eq_synonym" 3 TARGET_ILI 4 PART_OF_SPEECH "n" 4 WORDNET_OFFSET 5364891	0 @8227@ WORD_MEANING 1 PART_OF_SPEECH "n" 1 VARIANTS 2 LITERAL "bank" 3 SENSE 9 3 EXTERNAL_INFO 4 SOURCE_ID 1 5 TEXT_KEY "08204599-n" 1 INTERNAL_LINKS 2 RELATION "has_hyperonym" 3 TARGET_CONCEPT 4 PART_OF_SPEECH "n" 4 LITERAL "backlog" 5 SENSE 1 2 RELATION "has_hyponym" 3 TARGET_CONCEPT 4 PART_OF_SPEECH "n" 4 LITERAL "soil bank" 5 SENSE 1 1 EQ_LINKS 2 EQ_RELATION "eq_synonym" 3 TARGET_ILI 4 PART_OF_SPEECH "n" 4 WORDNET_OFFSET 8204599

Figure 2 : L'unité polysémique *bank* s'insérant dans deux synsets liés à deux sens différents

La synonymie et la polysémie sont donc les éléments structurants de ce type de réseau, de manière conjointe et inséparable. Ceci est lié à la non-congruence entre le niveau des lexèmes et le niveau, plus fin, des sens (un lexème pouvant avoir plusieurs sens, et plusieurs lexèmes pouvant partager le même sens).

Or nous faisons l'hypothèse que ce type de structuration du sens peut être déduite des relations d'équivalence traductionnelle observées sur des corpus de textes traduits (que nous nommerons désormais corpus parallèles). En effet, nous pensons qu'une approche multilingue basée sur des corpus parallèles permet de donner des renseignements utiles tant sur le plan de la polysémie (un lexème possédant des équivalents différents étant susceptible d'avoir différentes acceptions) que sur celui de la synonymie (deux lexèmes possédant les mêmes équivalents dans une langue cible étant susceptibles de partager un même sens).

La traduction, par le réseau de relations qu'elle constitue, peut peut-être jouer le rôle de révélateur par rapport à la structuration interne des sens d'une langue, vue sous l'angle de cette dialectique entre synonymie et polysémie. En outre, nous pensons que l'utilisation de plus de deux langues permet de renforcer des hypothèses concordantes issues de sources d'information différentes : une unité très polysémique aura sans doute de nombreux équivalents dans différentes langues cibles. Et si deux unités partagent un même sens, elles partageront sans doute des équivalents dans leurs traductions vers plusieurs langues.

Ainsi une seule langue cible n'est pas forcément suffisante pour porter un éclairage sur la variabilité sémantique d'une unité : il peut arriver qu'une unité polysémique puisse être traduite dans une langue cible par une unité équivalente présentant le même type de polysémie. Mais il est peu vraisemblable que cette même structuration se retrouve à l'identique dans plusieurs langues cibles. Par exemple, le nom *terme* en français présente la même ambiguïté que l'anglais *term* : il peut (entre autres) prendre les sens de /mot/ ou de /fin, échéance/. On trouve la même ambiguïté dans d'autres langues romanes, comme l'espagnol ou l'italien. Mais les équivalents allemands *Begriff* ou *Abschluss*, qui correspondent à ces sens, ne présentent pas cette ambiguïté.

Chaque langue organise et distribue différemment la répartition des sens à travers son lexique, et ses différences d'organisation démultiplient, lorsque plusieurs langues sont en jeu, ce rôle de révélateur joué par le passage à la traduction. Si cette hypothèse est valide, l'utilisation d'un corpus parallèle mettant en jeu plus de deux langues devrait donner des indices sémantiques encore plus fins : c'est le pari que nous avons fait en constituant un corpus parallèle français, anglais, espagnol et arabe.

Par ailleurs, dans la perspective d'extraire des unités de sens qui puissent être rapprochées des synsets de WordNet, nous pensons que le lexème n'est pas une entrée consistante pour l'organisation des sens, notamment pour l'enregistrement des relations d'*équivalence traductionnelle* en détachant les unités de leurs contextes d'occurrence. Nous proposons plutôt de nous appuyer sur *des cliques* de lexèmes, c'est-à-dire des ensembles de lexèmes qui partagent tous, pris deux à deux, un certain contenu sémantique. En effet, nous croyons que ces cliques permettent d'organiser le lexique en fonction des sens, à un niveau de granularité plus fin que celui des lexèmes, qui demeurent très ambigus hors contexte.

En extrayant de telles cliques à partir de notre corpus parallèle, nous espérons trouver une organisation des unités suffisamment cohérente pour apporter des informations fiables sur les deux propriétés qui nous intéressent, à savoir la synonymie et polysémie. Nous tenterons notamment de vérifier que les cliques extraites à partir des ensembles d'unités liées par des relations d'équivalences automatiquement extraites, grâce à des techniques d'alignement de corpus, sont apparentées aux synsets des réseaux sémantiques multilingues tels qu'EuroWordNet. De plus, ces cliques, par leur structuration fondamentalement multilingue, sont peut-être moins ancrées dans les particularités du découpage sémantique d'une langue donnée, et pourraient former de meilleurs candidats, pour un ajustement mutuel de différents wordnets, que les synsets de WordNet.

On peut s'attendre à ce que l'utilisation de plusieurs langues dans ces cliques permette de désambiguïser les lexèmes polysémiques : comme nous l'avons déjà évoqué, il est peu probable qu'une même polysémie se retrouve dans de nombreuses langues différentes.

Une telle ressource peut donc avoir des applications directes pour la désambigüisation en traduction automatique, dans le contexte spécifique d'une traduction en langue tierce connaissant déjà d'autres traductions outre le texte original (nous pensons aux traductions des textes de l'Union européenne, qui mettent en jeu jusqu'à 23 langues différentes).

Une fois ces cliques multilingues obtenues, nous tenterons de les associer aux synsets existants d'EuroWordNet. Les retombées seraient multiples :

- d'une part, cela permettrait d'établir un lien entre des lexèmes arabes et ces synsets. A partir de cette association, on pourrait envisager de projeter le réseau, avec toute les relations qu'il comporte (non seulement synonymie et polysémie, mais aussi hyper/hyponymie[3], antonymie[4], méronymie, etc.) vers l'arabe. Bien que problématique (rien ne permet d'affirmer à priori qu'un WordNet arabe doit être congruent à WordNet), cette possibilité permettrait d'amorcer la construction d'un nouveau réseau, et de récupérer automatiquement un grand nombre d'informations de nature sémantique.

- d'autre part, cela permettrait de mettre au point une méthode pour l'enrichissement automatique d'un réseau de type EuroWordNet, et consolidant des liens interlingues

3 Hyponymie : Relation d'inclusion, ou de subsomption, entre les désignations de deux lexèmes. (Ex. cheval est hyponyme de mammifère).
4 Antonymie : Relation entre deux lexèmes ayant un sens opposés ou contraires.

existants (qui seront nommés plus loin ILI-RECORDS, pour Inter Lingual Index), voire en ajoutant des nouveaux.

Ainsi, nous espérons que l'analogie entre nos cliques et les synsets de wordnets déjà créés, nous permettra de dégager une méthode pour amorcer automatiquement la construction d'une ressource sémantique arabe. A terme, une telle ressource serait utile pour de nombreuses applications du traitement de la langue arabe, comme la recherche d'information, la traduction automatique, les moteurs de question-réponse, la veille informationnelle, l'analyse d'opinion, etc.

Principales étapes de notre travail et plan de la thèse

Le contenu de cette thèse est structuré en quatre chapitres :

Le chapitre introductif est consacré à une étude générale des réseaux sémantiques, nous montrons leur utilité, les principes de leur structuration et leurs limites.

Dans le chapitre 2, nous introduisons la notion de clique, sur les plans théoriques et formels. Ensuite nous effectuons une expérimentation préliminaire en nous appuyant sur des ressources dictionnairiques, afin d'étudier la faisabilité de la méthode, avant même de l'appliquer sur des corpus parallèles. Enfin, nous y détaillons les étapes de constitution de notre corpus parallèle, en décrivant le fonctionnement des outils que nous avons mis en œuvre, tels que l'étiqueteur morphosyntaxique et le système d'alignement choisi.

Le chapitre 3 est consacré à la méthode d'extraction des cliques multilingues en quatre langues (fr-en-es-ar) autour d'une unité donnée et à partir du corpus aligné. Cette méthode s'appuie sur l'extraction automatique des équivalents traductionnels, en se basant sur l'observation des occurrences et cooccurrences en corpus aligné. Cette approche permet ainsi de traiter les lexèmes polysémiques, très fréquents dans notre corpus. Nous expliquons notamment le principe de la clusterisation des cliques voisines, rendue nécessaire du fait de la faible taille du corpus. Nous y faisons une première évaluation, très sommaire, de la qualité des cliques obtenues. Nous nous attendons à ce que les unités des cliques obtenues soient en interrelation du fait d'une probable intersection sémantique (des sens voisins ou connexes).

Dans le chapitre 4, nous développons notre méthode de rattachement des cliques à EWN. Nous y faisons une évaluation plus détaillée des résultats et des différents types de cas de figure rencontrés, notamment en vue de projeter les informations d'EWN vers la langue arabe. Nous évaluons la possibilité de récupérer pour les unités arabes des relations sémantiques déjà déclarées pour des unités en anglais, français et espagnol, dans leurs réseaux respectifs. Ces relations, une fois projetées sur le lexique arabe, permettraient de construire automatiquement un réseau utile pour certaines applications de traitement de la langue arabe.

Nous terminons par nos conclusions sur la validité de nos hypothèses, la généralité de la méthode et ses limites. Dans cette partie conclusive, nous tenterons de dégager des perspectives en vue du développement de cette approche, et identifierons de nouvelles pistes de recherche sur la question de la constitution de ressources lexicales à partir d'un corpus parallèle.

Chapitre I : Les réseaux sémantiques (utilité, structuration et limites)

I.1 Présentation générale

I.1.1 Définition et utilité

Un *réseau sémantique* est un outil permettant de représenter les entités du monde en les structurant selon des relations spécifiques inspirées de l'organisation mémorielle ou des raisonnements humains. Ces entités sont en interrelation en fonction de critères sémantiques particuliers.

Les réseaux sémantiques sont les premiers schémas de représentation des connaissances. L'approche sémantique de ces réseaux essaye de clarifier les relations linguistiques, afin d'organiser et d'afficher le contenu des unités au sein d'une base de connaissances.

D'un point de vue formel, ce type de réseau correspond à une structure de graphe qui encode les connaissances *taxonomiques* d'objets (concepts ou lexèmes) ainsi que leurs propriétés. C'est un graphe de nœuds interconnectés par des arcs (Sowa, 1991). La théorie des graphes rend l'utilisation des réseaux relativement souple, car elle permet de formaliser de nombreuses opérations et de les implémenter informatiquement. Ainsi la structure des réseaux permet de rendre opératoire certaines propriétés, telles que la proximité sémantique des connaissances.

Ces réseaux entretiennent des liens étroits avec la logique formelle, car les graphes utilisés pour la représentation des réseaux peuvent avoir une interprétation logique.

Généralement, les différentes familles de réseaux sémantiques peuvent différer sur un certain nombre de points :

- *" Les questions philosophiques du sens*
- *Les méthodes de représentation de tous les quantificateurs et opérateurs logiques*
- *Les techniques de manipulation de réseaux et d'exécution des inférences, et*
- *Les conventions stylistiques pour représenter les nœuds et arcs et leurs labels avec des mots ou autres symboles. "* (Madani, 1994).

Ces différences seront clarifiées plus loin dans la partie (2.3), mais quelles que soient ces différences, le but initial de réseaux sémantiques reste similaire, qui est la représentation et la compréhension d'un ensemble de connaissances attachées à des unités du langage naturel.

Voici une brève typologie des réseaux sémantiques les plus couramment utilisés d'après (Sowa, 2002):

- Réseaux de définition (*Definitional network*) qui utilisent les deux relations hiérarchiques 'partie_de' (relation de méronymie) & 'est_un' (relation d'hyponymie) (domaine des thésaurus).
- Réseaux de représentation de faits (*Assertional networks*) qui sont construits pour encoder des propositions.
- Réseaux de raisonnement (*Implicational networks*) qui peuvent être utilisés pour représenter des relations causales ou des inférences.
- Réseaux exécutables (*Executable networks*) qui utilisent un schéma d'exécution pour rechercher un certain nombre de motifs ou d'associations.
- Réseaux d'apprentissage (*Learning networks*) qui présentent la possibilité de rajouter ou de supprimer des nœuds en modifiant les valeurs numériques (appelés poids) associées aux nœuds et aux arcs.
- Réseaux hybrides (*Hybrid Networks*) qui utilisent plusieurs des techniques précédemment évoquées.

Les réseaux sémantiques ont été très utilisés dans le domaine de l'intelligence artificielle, et notamment pour le TAL, depuis les années 1960. Ils sont utilisés dans différents contextes tels que l'extraction de connaissances, la recherche d'information, le résumé automatique, la désambigüisation sémantique, etc. Les réseaux sémantiques ont été appliqués très tôt au problème de la représentation du sens des mots.

Ces réseaux montrent une bonne adaptation à la représentation du langage naturel, et la capacité de modéliser toute forme de connaissances que l'on peut représenter dans un système symbolique (Hendrix ,1979). Ainsi, ils ont pu s'appliquer efficacement dans quelques applications visant à la désambigüisation sémantique *(ex. en Traduction automatique pour la recherche de termes voisins, dans la recherche d'information, pour les systèmes de question-réponse, etc.)*.

En revanche, nous verrons dans la suite que la création d'un réseau sémantique en fonction d'objectifs spécifiques est une opération complexe et coûteuse à mettre en œuvre. C'est pour cela qu'il devient primordial, pour une langue donnée, de bénéficier de réseaux sémantiques génériques déjà développés, afin de rattraper l'écart technologique en termes de contenus, de services et d'usages entre les langues et les cultures du monde sur les réseaux d'information. Par ailleurs, la

création de réseaux sémantiques multilingues pourrait s'avérer très utile, notamment dans le domaine de traduction automatique et de la recherche d'information interlingue - dans la perspective prometteuse mais encore assez utopique de ce qu'on appelle le Web sémantique.

Par exemple, un réseau sémantique comme *EuroWordNet*, qui est devenu le plus important projet de développement des *wordnets multilingues*, maximise la compatibilité entre les *wordnets* à créer de différentes langues, et se concentre sur un codage manuel des concepts les plus importants et les plus compliqués. Ceci peut servir par exemple, dans les applications de recherche d'information multilingue (*Multilingual Information Retrieval*), soit à enrichir la requête (remplacement de ses termes par les nœuds correspondant et utilisation des relations de *synonymie* ou d'*hyperonymie*) soit à calculer une *distance conceptuelle* entre la requête et les documents à sérier, afin d'élargir le résultat à de nouveaux documents.

Par conséquent, il devient très motivant d'exploiter les ressources existantes afin de développer de nouveaux réseaux sémantiques, pour des langues qui sont moins dotées en terme d'outils et de ressources, c.à.d. pour lesquelles la question de la disponibilité des ressources pour le TAL reste, encore aujourd'hui, une question cruciale.

Par ailleurs, la diffusion de masses toujours plus importantes de données et de connaissances, rend la tâche d'organisation et de structuration des contenus indispensable - d'où l'effort important en direction du *Web sémantique*.

Nous trouverons dans la suite un exposé plus détaillé sur les réseaux sémantiques. Nous souhaitons donner une vision générale afin de trouver la modélisation adéquate permettant l'élaboration de notre ressource sémantique arabe.

I.2 Typologie des réseaux sémantiques

I.2.1 La sémantique de la représentation

La construction d'un réseau s'appuie sur une définition sémantique de ce qu'il représente. Il s'agit d'expliciter les *primitives structurelles* qui déterminent comment on construit le réseau, et quelle interprétation on donne à ses éléments. Ce sont ces primitives qui organisent les éléments et définissent les opérations qui permettent de manipuler la base de connaissance.

C'est de ces éléments qu'il convient par conséquent de partir pour construire un tel réseau, en leur donnant la possibilité de se développer des manières les plus diverses. La proximité sémantique des *nœuds* du réseau repose sur la définition de ces primitives structurelles.

Généralement, un réseau sémantique peut représenter des objets individuels, des classes d'objets, et des relations entre objets ou classes. Les objets sont représentés par des nœuds et les relations sont représentées par des pointeurs.

En fait, la nature des nœuds et des pointeurs (nous les nommerons désormais des *liens*) dans les réseaux est variable d'un auteur à l'autre. L'étude de la sémantique de la représentation a été l'objet d'un certain nombre de recherches, dont on cite ici quelques unes des plus notables :

- Chez (Woods, 1975*)*, les nœuds peuvent correspondre à des concepts ou à des propositions, d'où la mise en œuvre de différents types de liens :

  1- *"Les liens assertionnels entre concepts, utilisés pour décrire les propriétés des concepts (par exemple le lien de dépendance d'un sous-concept à son ou ses sur-concepts)"* (Godbert, 1991*)*

  2- "*Les liens structurels ou définitionnels (par exemple les liens casuels utilisés pour décrire un nœud propositionnel)."* (Godbert, 1991)

- (Brachman, 1977) a étudié la nature des concepts et leur sémantique, et distingue différentes interprétations de liens 'sorte_de' ou 'est_un' utilisés dans les réseaux. Il présente les caractéristiques d'un réseau idéal et il classifie les réseaux en différents niveaux en fonction de la nature des liens qui y sont définis (implémentation, logique, épistémologique, conceptuel et linguistique) (Brachman, 1979).

  Il propose un modèle baptisé KL-ONE et le range dans le niveau *épistémologique*. Ce niveau permet à un nouveau type de formalisme de réseau d'être spécifié. Ce formalisme permet précisément des opérations telles que l'individualisation des descriptions, la structure de concept interne en termes de rôle et d'interrelations entre eux, et l'héritage structuré. Etant donné que KL-ONE utilise des nœuds simples de type *concept* et de type *rôle*, la base de connaissances reste légère, puisque ces nœuds simples ne s'accroissent pas rapidement lors de l'enregistrement de données.

On constate que les nœuds et les relations peuvent être définis de multiples façons :

1. **Nœuds** :
    a. Nœuds représentant des classes d'objets (des propriétés, des attributs, des catégories taxonomiques, des prédicats, des événements, des propositions, des ensembles, des concepts d'entités, etc.)
    b. Nœuds représentant les objets individuels d'un domaine.

On distingue les nœuds représentant une classe de ceux qui sont des individus, en ajoutant des liens variés pour distinguer ces différents aspects.

Le *concept*, unité de base de nombreux réseaux sémantiques, peut en fait recouvrir des catégories, des classes d'objets ou bien quelque chose de plus abstrait comme des notions ou des idées, et n'acquiert tout son sens que par les relations qui le lient aux autres concepts.

Les concepts sont, par nature, intensionnellement distincts : deux concepts, équivalents en extension, doivent cependant apparaître sur des sommets distincts du réseau : "*on asserte de ces deux entités intensionnelles qu'elles dénotent un seul et même objet (extensionnel) "(Woods, 1975).* Nous reviendrons sur cela plus en détails prochainement.

2. **Liens** (autrement dit arcs ou pointeurs). Ils sont munis d'une étiquette indiquant le type de relation sémantique orientée entre deux nœuds :
    a. Liens de sous-ensemble (liens 'est-un' ou 'sorte_de': spécifiant qu'un concept est plus général ou plus spécifique qu'un autre. Cette relation, aussi appelée *relation de subsomption,* permet d'organiser les connaissances en hiérarchies de catégories auxquelles on peut attacher des mécanismes d'héritage. Cette relation lie un nœud plus général, dit *hyperonyme*, à un nœud plus spécialisé dit *hyponyme.*
    b. Liens d'appartenance d'un individu à une classe. Les liens de méronymie (ou encore liens 'partie_de') sont parfois assimilés à ce type de lien.

    a. Liens fonctionnels : permettant d'associer un objet à l'action dans laquelle cet objet est impliqué.

Les liens peuvent avoir des fonctions différentes de type:

- Logique (*ex. SNEPS : Semantic Network Processing System (Shapiro, 2000, Shapiro et Kandefer, 2005, Shapiro, 1996, Rapaport, 2000) :* C'est un projet ancien dédié initialement à l'étude de la structure des réseaux sémantiques pour le raisonnement). Ces liens définissent l'appartenance à une catégorie, par exemple "si l'objet possède telle propriété du concept poisson, alors cet objet appartient à la catégorie poisson".
- Conceptuelle *(ex. lien de sous-ensemble 'sorte_de') :* Ces liens sont responsables de l'enchaînement de mots et qui sont reliées à un ou plusieurs concepts.
- Casuelle *(ex. Les liens entre les nœuds propositionnels de (Norman et Rumelhart, 75)* : Ces liens relient entre des actions et leurs participants *(ex. Agent, Objet…etc.).*

Les liens fréquemment utilisés dans les réseaux sémantiques de type taxinomique sont hiérarchiques, associant un générique (*hyperonyme*) à plusieurs spécifiques (*hyponymes)* : (*ex. un animal est un type d'être vivant, un mammifère est un type d'animal, etc.*) Cette relation d'hyperonymie est assimilée souvent à la relation 'est_un'. D'autres liens sont souvent employés, comme les liens de *synonymie* (unités de même sens), de *méronymie* (unités liées par une relation 'partie-de'), *etc.* mais on en trouve également d'autres qui décrivent des configurations particulières et plus spécialisées (ex. 'sert à', 'fait en', etc.).

Les réseaux sémantiques sont en général basés sur les propriétés suivantes :

• ***Transitivité des catégories :*** La transitivité de certaines relations est supportée, ce qui autorise l'héritage des propriétés d'une entité à celles qu'elle subsume. Par conséquent, il s'en suit une économie de mémoire de stockage pour la représentation de plusieurs relations.

Exemple: p222 est une imprimante et aussi une machine.

• ***Héritage des propriétés*** de la super-catégorie à la sous-catégorie. On pourrait représenter exhaustivement les diverses propriétés des concepts présents dans le réseau, mais, plutôt que d'indiquer les mêmes propriétés pour tous les concepts spécifiques, ce qui aboutit à des répétitions superflues et surcharge la mémoire inutilement, on attachera ces propriétés au niveau des concepts parents, ce qui permet de retrouver les propriétés des descendants par simple héritage des propriétés le long des liens 'sorte_de'.

Exemple: Les imprimantes au laser utilisent également une prise de courant au mur pour leur énergie

Nous présentons dans la suite quelques exemples de types de liens de réseaux sémantiques avec diverses utilisations proposées, mais nous nous contentons de citer quelques relations parmi les plus couramment utilisées :

- Le lien 'sorte_de' et ses diverses utilisations :
    - Hiérarchie : c'est le lien fondamental qui correspond à la typologie des connaissances, il indique la dépendance d'un concept à un sur-concept, l'appartenance d'un élément à une classe (*ex. voiture-Titine*) et l'inclusion entre sous-classes (*ex.véhicule-voiture*) - dont on déduit un héritage des propriétés. Ce qui correspond à la structuration d'*Hyperonymie/Hyponymie*. Ce lien alors lie un élément hyponyme à son hyperonyme (*ex. chat →sorte_de→animal*)
    - Particularisation : c'est le lien inverse du lien hiérarchique qui indique qu'un concept donné peut se particulariser en un concept plus spécifique (*ex. un chien particulier est le caniche*) ou il indique quelquefois (Casagrande et Halle, 1967*)* qu'un sujet ou un objet est distinctif pour un verbe ou une propriété donnée : Par exemple c'est le lien qui existerait entre briller et soleil (*ex. Le soleil est un objet particulier qui brille*).
    - Equivalence : dans un réseau sémantique monolingue, ce lien exprime des relations de synonymie. Ce lien peut être utile pour mettre en évidence des *polysémies* potentielles : si A est synonyme de B et si A est synonyme de C alors que B n'est pas synonyme de C, c'est que probablement A possède deux sens qui devraient être différenciés par deux nœuds du réseau.
    - Contraste : ce lien exprime des relations de différence et d'opposition entre deux éléments. C'est le contraire du lien 'équivalence'. (*Ex. le contraire ordinaire : court vs long).*
- Le lien 'partie_de' : parfois appelée 'partie-tout' ou holonymie / méronymie ; cette relation exprime la manière de composer des éléments complexes en fonction d'autres éléments plus simples. TOUT dispose des propriétés qui ne sont pas obligatoirement transmises à ses parties comme dans la relation d'hyponymie *(ex. Dans le corps humain, la tête et les jambes font partie du corps mais elles ne disposent pas des mêmes propriétés).*
- Le lien 'scénario' : Représente une relation entre des situations diverses liées par une certaine contingence, avec ou sans notion de causalité (Madani, 1994) *(ex. croissant - petit déjeuner).*

- Le lien 'succession' : Qui représente la constitution d'une relation d'ordre entre certaines connaissances définies comme des listes ordonnées *(ex. les jours de la semaine*).
- Le lien 'fonction' : Indique l'action usuelle qui est impliquée par l'objet considéré (Madani, 1994) (*ex. pelle et creuser*)

Si on se place au plan lexical, pour un réseau dont les nœuds représenteraient des lexèmes, ces relations correspondent aux relations sémantiques traditionnelles :

- Hyper/hyponymie : Un hyperonyme est un lexème subsumant toute une classe de lexèmes plus spécifiques. Si on nomme, avec Rastier (1989), taxème une classe minimale où les sémèmes sont interdéfinis, l'hyperonyme correspond aux sèmes génériques communs à toutes les unités du taxème, et les cohyponymes s'interdéfinissent en s'opposant à l'intérieur de cette classe. Par exemple, (le taxème /blanc/, en parlant de *linge de maison*, met en opposition les unités *torchons*, *serviettes*, *draps*, *nappes*, etc).
- Antonymie : relation lexicale entre lexèmes s'opposant au sein d'un taxème binaire, l'opposition pouvant indiquer la contrariété, l'inversion, la réciprocité, la complémentarité, etc. (*Ex. jour vs nuit, homme vs femme, sortir vs entrer, mais aussi droite vs gauche).*
- Synonymie : deux lexèmes sont synonymes s'ils sont interchangeables dans certains contextes linguistiques, sans altération de l'interprétation globale de l'énoncé. On sait qu'il n'existe pas de synonymie parfaite, néanmoins il est fréquent que des lexèmes différents puissent désigner des référents identiques. (*Ex. chef vs directeur vs tête).*
- Polysémie : généralement, la polysémie indique la capacité d'un lexème à pouvoir prendre des sens différents (Haton, 2006 : 36*),* autrement dit, à rentrer dans des taxèmes différents. ex. (*souris* s'oppose à *rat* et *mulot* d'une part, mais aussi à *clavier* et *écran* dans le champ de l'informatique).
- Méronymie : Il s'agit de la relation de 'partie-tout'. Les parties ne pouvant s'interdéfinir de façon différentielle comme des cohyponymes au sein d'une même classe, on sort ici de la perspective structurale. La méronymie désigne plus une propriété référentielle ou conceptuelle que sémantique, néanmoins le méronyme subsume d'une certaine manière les éléments qui le composent, ce qui se retrouve aussi dans les propriétés distributionnelles de la langue, par métonymie (un *deux-roues* pour un *vélo*). Dans une définition logique, on dira que X est un méronyme de Y si la proposition "Un X est une partie d'un Y" est vraie.

Notons que les composants initiaux de toutes les relations précédentes sont : *Le mot*, *le sens* et certainement *le concept*. Ces composants sont eux-mêmes des composants de triangle sémiotique. Donc, il convient de clarifier ces définitions sémantiques afin de comprendre mieux ces types de relations:

Triangle sémiotique

On constate que les terminologies sont variables, même si les objets sont souvent analogues. La nature de leurs composants, comme déjà mentionnée, diffère suivant la problématique pour laquelle les réseaux sont créés. Pour saisir le niveau de représentation mis en œuvre dans un réseau sémantique, il faut effectuer quelques distinctions terminologiques en se référant *au triangle sémiotique*, qui représente la conception traditionnelle du rapport entre monde en langage, depuis Aristote jusqu'à Carnap, en passant par la scolastique ou la grammaire de Port Royal.

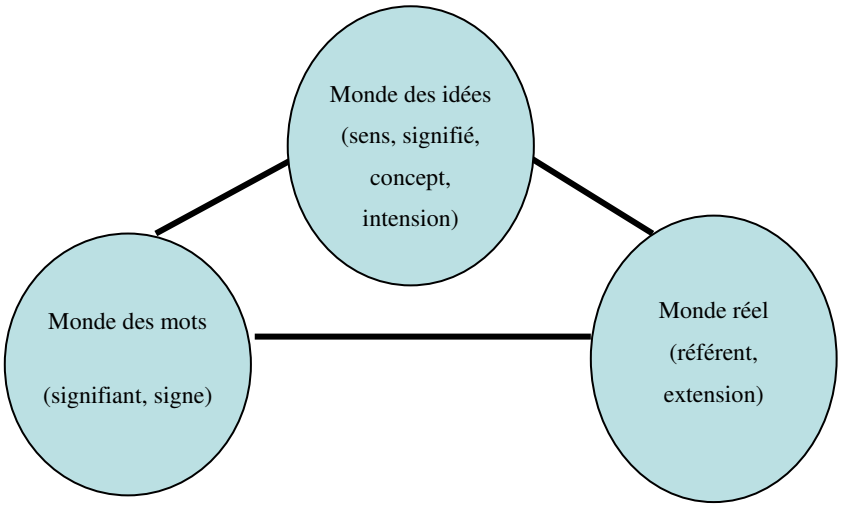

Figure 3: Triangle sémiotique

Le mot

On doit distinguer entre le *mot* pris en tant qu'unité graphique, qui peut éventuellement servir d'étiquette pour désigner des choses ou des concepts, et le *lexème* pris comme unité du système. La plupart du temps, le mot n'est pris que comme étiquette conventionnelle, ou symbole, d'une relation extralinguistique.

Très souvent les réseaux sémantiques utilisent des mots pour étiqueter les nœuds, mais ceux-ci représentent en fait des objets ou concepts, reliés entre eux en fonction de critères logico-

sémantiques particuliers. Il est sans doute abusif de considérer qu'il s'agit là du *signifiant* associé au mot, car dans une perspective structurale, le signifiant est indissociable du signifié, et il forme la *valeur* du signe non par ses relations avec les référents extralinguistiques, mais par ses relations entre les signes eux-mêmes.

Le sens

Le *sens* ou l'*intension* (voir Figure 3) représentent donc des objets psychologiques ou logiques. D'un point de vue linguistique, c'est la composition d'une valeur différentielle par rapport aux autres signes et d'une valeur référentielle qui se rapporte au *signifié (F. de Saussure)* : cette valeur n'est donc pas extrinsèque au signe et au signifiant, et ne peut être autonomisée dans la sphère des idées, comme le suggère la triade.

Il est donc important de déterminer, pour un réseau sémantique donné, si les liens et les nœuds sont interprétés de manière autonome (i.e. en quelque sorte indépendamment d'une langue donnée), ou s'ils représentent des unités et des relations linguistiques, comme dans un dictionnaire. Dans un dictionnaire, le sens d'un mot est explicité par une définition, qui détaille ses acceptions et liste éventuellement ses emplois. Par contre, dans la plupart des réseaux sémantiques, le sens est construit par les relations entre concepts, et de fait, le sens attaché à un nœud dépend de sa position dans le réseau - sans nécessairement s'attacher à la structuration sémantique d'une langue en particulier.

Par ailleurs, à travers la relation de désignation, le sens peut s'interpréter de manière extensionnelle (*p.ex. le sens de "fruit" a pour signification 'pomme', 'orange',…)* et référentielle (en relation avec l'interprétation dans un contexte précis de communication).

Le concept

Le *concept,* généralement, est une représentation (psychologique, mentale, logique, idéelle, peu importe ici) d'ordre extra-linguistique. Dans la triade, il est assimilé au sens. Pour le sémanticien, le concept est un objet extra-linguistique, de même que les objets du monde sur lesquels portent les connaissances (appelés *classes* aussi). Le concept peut représenter un objet du monde réel (*ex. un comprimé de médicament*), une classe d'objets, ou encore une notion abstraite (*ex. la quantité*) ou une idée.

Voici quelques exemples de la manière dont cette notion est employée par différents auteurs :

- (Woods, 1975*)* identifie des concepts distincts qui ne peuvent pas être exprimé par la logique du 1er ordre, ni à travers des classes ; il les appelle *concepts intensionnels* (*ex. 'étoile du matin' et la 'planète Vénus')*.
- (Touretzky, 1986) identifie la notion de concept avec la notion de prédicat dans la logique du 1er ordre. C'est la notion d'abstraction, qui désigne une collection de propriétés partagées par les membres d'un ensemble.
- (Kayser, 1988*)* propose de séparer la notion de concept de celle de prédicat, et de représenter les concepts par une famille ouverte d'entités du système.
- (Haton et al., 1991) explique que la connaissance d'un concept ne nécessite pas la connaissance des conditions nécessaires et suffisantes d'appartenance d'un individu à la catégorie de ce concept, ni la connaissance de tous ses sous-types.

En terminologie, il est communément admis que le concept se compose de trois parties :

1. Le *terme* est l'expression linguistique utilisée couramment pour y faire référence. C'est la matérialisation linguistique du concept.
2. Une *notion* contient la sémantique du concept qui est définie à l'aide de propriétés, d'attributs, de règles et de contraintes (tous formant l'intention du concept).
3. Un *ensemble d'objets* auxquels le concept fait référence, autrement dit, de ces instances, ces instances formant l'extension du concept.

Cette structuration terminologique du sens, qui reprend la structuration triadique, impose ses propres contraintes qui dépassent la simple structuration des signifiés en langue. On voit bien que le concept n'est pas assimilable au signifié, le signifié étant un objet structuré au niveau linguistique, tandis que le concept est construit sur un plan extralinguistique (*ex. au niveau d'une discipline scientifique, d'une pratique technologique, etc.*). Le signe linguistique (la paire signifié/signifiant) unit non une chose et un nom, mais une idée (la substance du contenu) et une image acoustique. Le signifié ne peut donc rentrer dans la triade, comme on le fait naïvement.

Quand on parle de réseau sémantique, il faudrait donc parler, la plupart du temps, de réseau conceptuel. Un concept, dans un tel réseau, est la résultante d'un ensemble de nœuds fortement liés et activés simultanément. D'après (Sidhom, 2006) *"le sens d'un concept se réduit à sa position relative par rapport aux autres concepts, il ne prend donc un sens que par rapport à un réseau*

sémantique modélisant les connaissances générales du système". Il se définit comme une unité de base qui acquiert son sens par les relations qui le lient avec d'autres concepts.

Mais les concepts utilisés dans les réseaux ne sont pas forcément de même type. Généralement, Il existe deux types:

- Les concepts simples ou primitifs : *"Un concept simple est un concept dont on connaît un ou plusieurs sur-concepts, mais que l'on ne sait pas définir par des conditions suffisantes, permettant de décider si un individu lui appartient ou non. "* (Madani, 1994). En fait, chaque concept simple a une propriété différente de celle des autres éléments du réseau. Cette propriété ne peut pas être entièrement définie à partir d'autres éléments du réseau. Ces primitives forment en quelque sorte l'axiomatique de l'ensemble.
- Les concepts définis : " *ils peuvent être entièrement définis à partir des propriétés et de concepts existant déjà dans le réseau "*. (Madani, 1994). Ils modélisent des objets complexes, éventuellement dénotés par des expressions composées (*ex. fleur jeune*).

Bref, pour interpréter un réseau sémantique, il est important de déterminer au préalable la nature de ses primitives structurelles, nœuds et liens, en fonction de l'objectif visé. Mais il ne faut pas oublier que la détermination de la structure du réseau est aussi essentielle.

I.2.2 Structuration des réseaux sémantiques

Sur le plan de la formalisation, la structure des réseaux sémantiques peut prendre plusieurs formes dont nous ne citons que les plus connues :

- *Graphe* : C'est la structure la plus générale de réseau, pris comme ensemble de nœuds reliés par des liens. Pour réaliser un encodage de connaissances taxonomiques concernant des objets ainsi que leurs propriétés, le graphe doit être orienté (i.e. les relations sont directionnelles) et acyclique (c'est-à-dire ne contient pas de séquence circulaire, en suivant les relations) - cette dernière possibilité garantissant la possibilité de hiérarchiser les éléments (un élément ne pouvant être à la fois super-ordonné et subordonné à un autre élément). Un graphe orienté est un couple $G = (S, A)$ où S est un ensemble de nœuds (ou sommets) et A est une partie de S x S, l'ensemble des liens.

- *Treillis* : c'est une structure particulière de graphe, constituant un ensemble partiellement ordonné (si l'ensemble est totalement ordonné on parle alors de chaîne) dans lequel tout couple d'éléments a une borne inférieure (c'est-à-dire un moins spécifique descendant commun) et une borne supérieure (c'est-à-dire un moins générique ancêtre commun).

  Le treillis peut-être perçu comme une hiérarchie de concepts. Chaque concept est une paire composée d'une extension représentant un sous-ensemble d'instances et d'une intension représentant les propriétés communes aux instances. Un treillis complet est un treillis pour lequel tout sous-ensemble de nœuds possède une borne supérieure et une borne inférieure unique. Un exemple de treillis est présenté dans la Figure 4.

  L'ordre d'un treillis permet de définir un héritage entre les concepts. Ainsi chaque concept hérite des objets des concepts qui sont entre lui et ses sous-concepts et des attributs qui sont entre lui et ses super-concepts. C'est pour cela qu'un treillis est dit être une double hiérarchie de concepts.

  Certains structures sont dérivées, telles que les treillis de Galois qui ont été introduits dans (Birkhoff, 1940*)* et par (Barbut et Monjardet, 1970), treillis d'héritage (Godin, 1995), etc.

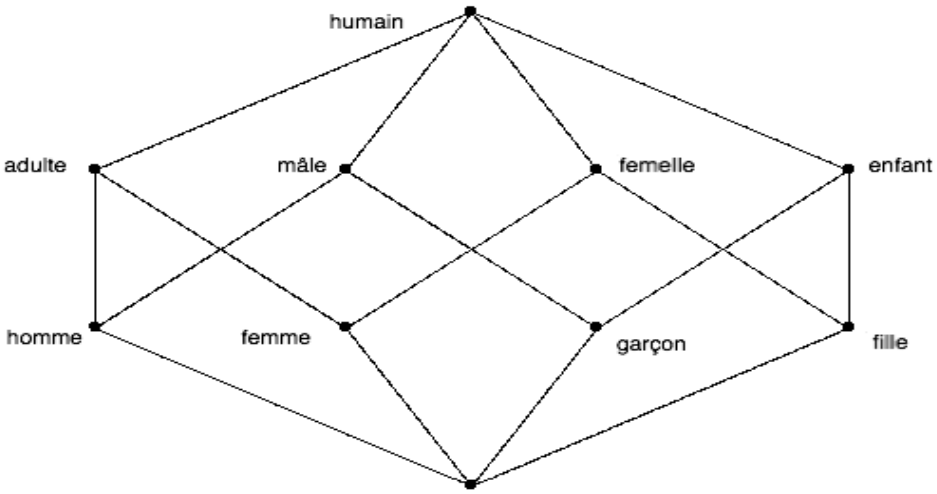

Figure 4 : Exemple d'un treillis

- *Arbre* : L'arbre est un cas particulier de graphe. Dans un arbre, chaque nœud est relié à un superordonné unique (le genre), c'est-à-dire qu'un concept ne peut avoir qu'un et un seul concept parent. La signification d'un nœud se détermine en fonction de ses plus proches voisins. Les plus proches voisins sont d'une part le concept parent et d'autre part les concepts frères. Il faut donc déterminer la signification d'un nœud en fonction de son parent et de ses frères.

I.2.3 Formalismes de représentation de connaissances sous forme de réseaux sémantiques

Un des éléments importants de la théorie des réseaux est le type de formalisme mise en œuvre (Elisabeth Godbert, 1991). Il existe plusieurs formalismes de représentation des connaissances sous forme de réseaux sémantiques. Ces formalismes sont proches au niveau de la signification, mais diffèrent au niveau de leur utilisation.

Depuis les recherches de Quillian (Quillian 1968), divers types de formalismes de représentation des connaissances ont été développés, tels que les ontologies, les thésaurus, les taxonomies et les graphes conceptuels. Nous proposons ci-dessous de caractériser dans les grandes lignes ces différents formalismes, tout en gardant à l'esprit qu'il n'y pas de frontière stricte entre ces notions.

I.2.3.1 Ontologie

Une ontologie est un réseau sémantique portant sur un ensemble de concepts décrivant (ou tentant de décrire) complètement un domaine. Elle contient généralement des relations taxonomiques (hiérarchisation des concepts) et sémantiques afin de relier les concepts les uns aux autres.

Dans son acception philosophique, ce terme lui-même date du XVIIe. Il désigne alors une discipline en philosophie initiée par Aristote, en référence à la science de l'être en tant qu'être.

Une première définition d'ontologie liée à la *conceptualisation* est introduite par (Gruber, 1993). Il définit une ontologie comme une spécification partagée d'une conceptualisation. (Guarino & Welty, 2000) ont clarifié cette définition de conceptualisation : Il s'agit d'une part de distinguer les individus et les propriétés qui peuplent un domaine, et, d'autre part, de préciser quelles sont les propriétés essentielles garantissant l'unicité, l'identité.

Les ontologies sont utilisées en Ingénierie des Connaissances (IC) et en Intelligence Artificielle (IA) pour rassembler les concepts qui sont considérés comme des briques élémentaires permettant d'exprimer les connaissances dont on dispose dans un domaine.

Contrairement aux définitions précédentes (Bachimont, 2000) définit une ontologie comme la signature fonctionnelle et relationnelle d'un langage formel de représentation avec une sémantique associée.

En outre, une ontologie d'après (Cahrlet, 2002 :P44) *"[…] implique ou comprend une certaine vue du monde par rapport à un domaine donné. Cette vue est souvent conçue comme un ensemble de concepts – e.g. entités, attributs, processus –, leurs définitions et leurs interrelations. On appelle cela une conceptualisation.* "

[. . .]

" *Une ontologie peut prendre différentes formes mais elle inclura nécessairement un vocabulaire de termes et une spécification de leur signification. "*

En d'autres termes, une ontologie est une structure où les concepts et les relations sont organisés hiérarchiquement pour admettre une relation de subsomption et des relations sémantiques au sein d'un domaine donné.

Une ontologie contient donc des liens d'*hyperonymie*, d'*hyponymie* et des relations d'association entre les différents concepts de l'ontologie qui sont en général définis avec une plus grande précision.

Les concepts d'une ontologie sont indépendants du plan linguistique, c'est pourquoi dans des ontologies multilingues où les concepts sont reliés à un ou plusieurs termes dans plusieurs langues, une des langues concernées pourrait manquer de termes adéquats pour désigner un concept. Par exemple, en anglais, selon (Cruse, 1986) il n'y a pas de terme générique pour désigner l'ensemble des couverts.

En outre, l'organisation conceptuelle d'une ontologie demande de se doter d'une théorie soit sur la connaissance, soit sur le monde afin de pouvoir organiser les concepts les plus abstraits qui vont structurer le reste de l'ontologie. Cette organisation s'appelle parfois ontologie supérieure (*upper ontology)*[5]*,* et est utilisée afin de classifier sémantiquement les concepts de base.

Parmi les réseaux sémantiques, très 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 "* (Sowa, 1992).

[5] Cf. http://suo.ieee.org/

I.2.3.2 Graphes conceptuels

Le modèle des graphes conceptuels est un modèle de représentation de connaissances du type réseaux sémantiques qui a donné lieu à un certain nombre de travaux depuis son introduction par John F. Sowa en 1984. L'une des particularités de ce modèle est de permettre de représenter des connaissances sous forme graphique.

Les graphes conceptuels sont d'après John F. Sowa un système de logique basé sur les graphes existentiels de Charles Sanders Peirce et les réseaux sémantiques de l'intelligence artificielle. Ils représentent le sens dans une forme logique, lisible et manipulable informatiquement.

En outre, ces graphes s'utilisent comme un langage intermédiaire pour interpréter les formalismes orientés objet et le langage naturel.

Un graphe conceptuel est défini comme étant un multi-graphe fini, bipartite, non-orienté connexe et étiqueté ; les deux classes de sommets sont étiquetées respectivement par des noms de *concepts* et des noms de *relations conceptuelles*. Les concepts sont formés d'un type de concept et un référent (instanciation du type de concept). En revanche, les relations se composent d'un type de relations. Un exemple de graphe conceptuel est donné dans la Figure 5 suivante :

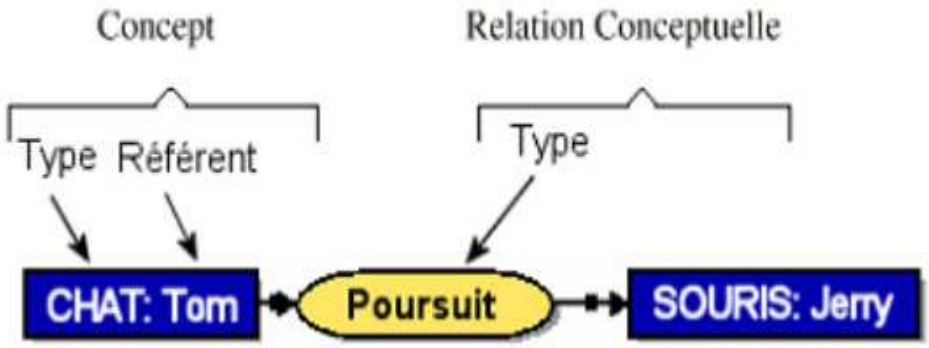

Figure 5 : Exemple de graphe conceptuel[6].

Ces graphes étant constitués par un ensemble de nœuds reliés par des liens, la connaissance concernant une entité représentée dans le réseau lui est directement attachée ; le parcours des liens qui partent d'un nœud permet de retrouver toute la connaissance attachée à ce nœud.

[6] Cf. http://archimede.bibl.ulaval.ca/archimede/fichiers/25348/ch02.html

Ces graphes reposent sur la logique du premier ordre. Leur intérêt réside dans leur non-ambiguïté et leur facilité d'utilisation. C'est pour cette raison qu'on les utilise souvent dans l'acquisition des connaissances, la recherche d'informations et le raisonnement sur la connaissance conceptuelle (Mellal, 2007).

Cette représentation graphique des connaissances facilite aux utilisateurs la compréhension et la manipulation des données, en mettant en évidence les méthodes de récupération et d'exploitation de l'information stockée, d'une façon beaucoup plus simple que celle d'une représentation sous forme de formules logiques. La théorie de graphes utilisée dans les réseaux développés rend l'utilisation de ces réseaux relativement facile.

I.2.3.3 Thésaurus

Les thésaurus sont principalement utilisés pour la gestion, l'indexation et la classification de documents - et peuvent servir de support, également, à l'analyse de contenu. En ce sens, ils constituent un véritable *langage documentaire*. Un thésaurus a pour fonction d'éviter de regrouper de signifiés différents sous une même forme signifiante, et d'éviter la dispersion de l'information sous des termes plus ou moins proches sémantiquement.

Conçu dés la fin des années 1950, une définition répandue est celle de la norme internationale ISO 2788 (1986 : 2) : " *vocabulaire d'un langage d'indexation contrôlé organisé formellement de façon à expliciter les relations a priori entre les notions (par exemple relation générique-spécifique) ".*

Il faut noter que le vocabulaire dans les thésaurus est artificiellement contrôlé en fonction des besoins et ressources du service d'information. La signification de ses vocables est arbitrairement fixée et relativement figée, comme dans toute langue artificielle.

Pour représenter les concepts, les thésaurus contiennent des unités lexicales appelées *descripteurs* et *non-descripteurs* (ou bien termes préférentiels et termes rejetés) et, pour indiquer leurs relations, un ensemble de notations normées au niveau international (ISO 2788 ; Iyer, 1995).

Les descripteurs peuvent apparaître dans différents contextes d'un même thesaurus, mais les concepts associés sont définis de façon unique dans un *réseau sémantique*.

En effet, le thésaurus est un répertoire contenant une liste de mots fonctionnels, un dictionnaire des synonymes, des paraphrases, ainsi qu'une hiérarchie des termes (ou descripteurs). Ces descripteurs

sont organisés de manière conceptuelle pour faciliter la description d'un domaine et harmoniser la communication et le traitement de l'information. Lorsque ces relations ne sont que de simples hiérarchies, on parle de *classification*.

L'élaboration de thésaurus nécessite la collecte et le choix des termes, le choix de leur forme (singulier ou pluriel, termes composés ou pas), le choix de l'arrangement (thématique ou à facettes), l'établissement des relations (d'équivalence, hiérarchiques et associatives), le choix des formats de présentation (alphabétique, systématique, graphique) et la compilation.

Les thésaurus fournissent également des informations sur les relations entre les mots, particulièrement la relation de synonymie (*ex. le Roget's International Thesaurus*).

Mais la constitution manuelle d'un thésaurus est un processus lent et coûteux (acquisition de terminologie à partir de gros corpus). En outre, il permet un travail uniquement sur la terminologie, il n'est pas adapté pour évaluer des proximités sémantiques entre deux mots, deux notions ou deux textes.

I.2.3.4 Taxonomie

Une taxonomie est une forme d'ontologie dont la grammaire n'a pas été formalisée. En d'autres termes les taxinomies et les thésaurus peuvent être considérés comme des ontologies dont les relations seraient restées implicites car *évidentes* pour le lecteur (mais pas pour l'ordinateur).

Le terme *taxonomie* fut introduit, sous cette orthographe, par un botaniste genevois, Augustin Pyrame de Candolle, pour définir sa théorie des classifications. L'orthographe fut corrigée en *taxinomie* par Émile Littré mais la forme concurrente reste pourtant très répandue.

En botanique ou en zoologie, toutes les classifications se présentent sous la forme d'un arbre, allant de la racine incluant les différents règnes, jusqu'aux différents embranchements, classes, ordres, familles, genres, espèces, sous-espèces, variétés, puis formes. Chaque nœud de l'arbre définit un taxon, qui regroupe tous les sous-taxons qu'engendre le nœud.

Une taxonomie est donc un type simple d'ontologie où toutes les relations sont de type 'sorte_de', sans formalisation spécifique des propriétés.

En conclusion de ce tour d'horizon, une comparaison de ces différentes catégories de réseaux sémantiques peut être faite en s'appuyant sur les types de relations gérées :

- La relation hiérarchique de sous-ensemble 'sorte_de' ou relation d'hyponymie est commune à tous ces formalismes, elle correspond au classement humain des objets par inclusion de *classes*.
- La relation d'appartenance 'partie_de' ou relation de méronymie est commune à tous ces formalismes sauf que dans le cas de la taxinomie et du thésaurus, elle est similaire à la relation hiérarchique précédente.
- La relation 'est_un' : Dans le cas des thésaurus et taxinomies, elle est similaire à la relation hiérarchique 'sorte-de', mais pas dans les cas des ontologies, où elle exprime l'appartenance d'un individu ou sous- catégorie à une catégorie.
- La synonymie est gérée plus spécifiquement par les thésaurus.
- Les relations associatives des thésaurus sont plus détaillées (en relations étiquetées) dans les ontologies.

I.2.4 Des réseaux de concepts aux réseaux de sens

La notion de réseau a été utilisée en premier par le psychologue Quillian en 1966 (Quillian, 1966). Son but était d'élaborer un réseau (le système TLC : Teachable language comprehender) reliant des concepts d'une manière hiérarchique taxinomique (super-classe, sous-classe), afin de pouvoir comparer la part objective du sens de ces concepts. Il s'est appuyé sur des expériences psychologiques permettant de retracer le raisonnement de la pensée humaine (Quillian, 1985). Ses expériences portent sur le lexique anglais. A chaque mot anglais est associé un concept. Ces concepts sont représentés par des nœuds reliés entre eux par des liens associatifs de divers types. La proximité des nœuds chez Quillian représente la distance sémantique entre les concepts, mesurée en fonction d'un algorithme précis.

Cependant, ce système de Quillian (système TLC) n'était pas très efficace. En fait, l'un des problèmes de ce système est que les propriétés sont seulement stockées avec le concept le plus général. Quillian suppose que tous les items liés à un concept doivent forcément partager toutes ses propriétés. Mais cette hypothèse n'est pas applicable dans tous les cas et cela peut poser un problème lors de l'exploitation des données: Par exemple, l'un des propriétés de la plupart d'oiseaux est la capacité de voler mais (l'autruche), qui est un oiseau, n'a pas cette propriété. Ainsi, on ne

trouve pas de distinction entre les propriétés d'instances et les propriétés des classes. En outre, les liens utilisés dans ce système sont des liens d'association généraux et non pas des vrais liens sémantiques structuraux aidant à gérer l'héritage de liens multiples.

Plusieurs modèles de réseaux, développés par la suite, ont apportè de nouveaux éléments au domaine de la représentation des connaissances. Par exemple, (Schank, 1973) tente de modéliser la structure sémantique du langage naturel, il considère, dans son projet *Margie*, que le facteur essentiel pour le traitement du langage naturel est la compréhension sémantique profonde d'un énoncé (sous la forme d'entités conceptuelles), et non pas la syntaxe qu'il voit comme un facteur secondaire. C'est pourquoi il élabore des relations conceptuelles entre objets et actions, dont ces dernières sont décrites par la combinaison de onze éléments primitifs, afin de modéliser des actions primitives et non primitives. Ces relations sont présentées sous la forme de graphes de *dépendance conceptuelle*.

(Fahlman, 1979), de son côté, introduit dans ces travaux le concept de *frame*, une structure de données représentant une connaissance type qui est formé de *slots*, nœuds-rôles, qui permettent de structurer la description des concepts, sous la forme de liste d'attributs. Il introduit des relations d'exception, qui sont annulables, et des relations universelles, qui expriment des relations toujours correctes. Ex. Si le sujet de chanter est par exemple déclaré dans la catégorie des animés, la mise en correspondance du nœud de premier niveau *('sorte-de' 'objet-inanimé')* avec le groupe nominal *(ce livre)* dans la phrase *: Ce livre chante les louanges de la vie campagnarde.,* provoque l'insertion d'un lien d'exception, et l'annulation d'un lien annulable.

Mais l'élaboration de ces premiers systèmes, mentionnés ci-dessus, avait été marquée par une démarche empirique, et par l'absence de méthodologie précisant les principes généraux à la base de la construction de ces formalismes. En effet, l'ensemble de primitives sur lesquelles s'appuient ces réseaux est définie dans ces systèmes d'une façon assez vague, et elle devient trop imprécise lorsqu'il s'agit de représenter un domaine complexe.

D'autres travaux, par la suite, ont cherché à tirer parti de dictionnaires électroniques pour la représentation sémantique comme ceux de (Bruce & Guthrie, 1992), (Rigau et al.1997) , (Richardson, 1997), (Hindle, 1990, Grefenstette, 1994) et (Veronis & Ide, 1990, Warnesson, 1992*)*. Mais ces derniers se basent sur des dictionnaires classiques lacunaires, qui souffrent du manque de relations conceptuelles, car la mention d'un seul synonyme ne permet pas en général de caractériser

un sens précis. De ce fait, la construction d'un réseau sémantique à partir de ce type de ressources monolingues est très complexe et coûteuse en temps.

Face à ces problèmes, une autre piste a été explorée. Certains lexicographes ont commencé à combiner des dictionnaires et des thésaurus : *WordNet* est un exemple (Fellbaum, 1998; Miller et al, 1990) de ce type de démarche. Ce type de lexique sémantique ne contient pas seulement des gloses pour les sens des termes qui y sont décrits, mais aussi des listes de *synonyme*s, ainsi que des liens vers des *antonyme*s, *hyponyme*s…, etc.

I.2.4.1 Lexiques sémantiques

A la différence d'un dictionnaire classique, WordNet est organisé selon des sens lexicaux : le principe organisateur est onomasiologique, des sens vers les mots, et non sémasiologique des mots vers les sens. Il traite de sens lexicaux, c'est-à-dire de sens présents dans le lexique de la langue, et non de concepts supposés indépendants de celle-ci. Les lexèmes partageant un même sens sont regroupés dans des ensembles nommés synsets. D'une certaine manière, cette intersection sémantique entre plusieurs lexèmes d'un même synset constitue une caractérisation linguistique du sens, complémentaire des gloses qui en sont données.

Le WordNet de Princeton, du fait de son succès, a inspiré toute une série d'initiatives, et est devenu un paradigme pour toute une famille de réseaux, que par commodité nous appellerons des wordnets.

Toutefois, les informations dans WordNet sont limitées et lacunaires, du fait de l'insuffisance des informations des dictionnaires (classiques et électroniques) et thésaurus utilisés. Ce problème est inhérent à toutes les ressources lexicales : le lexique est par nature ouvert, et ne peut être épuisé par une ressource intrinsèquement fermée. Nous reviendrons sur ce problème un peu plus loin.

I.2.4.2 Relations multilingues entre lexiques sémantiques

Les techniques actuelles de développement des réseaux de la famille wordnet se répartissent en deux types d'approche :

-L'approche *par fusion*, consistant à construire un wordnet à partir de plusieurs ressources monolingues. Certains problèmes se posent :

- Cette approche est souvent totalement ou partiellement manuelle, ce qui la rend très coûteuse.

La structure des relations et des synsets d'une langue A pourra être complètement différente de celle d'un wordnet pour une autre langue B. Cette spécialisation peut constituer un avantage, car la ressource ainsi créée permettra de décrire plus finement les structurations sémantiques d'une langue particulière, mais elle pose problème lorsqu'il s'agit d'établir des passerelles entre des wordnets de langues différentes.

-L'approche *par extension* (Vossen, 1996), consistant à traduire les informations d'un wordnet d'une langue source vers une langue cible, en calquant la structuration du wordnet déjà construit. Le wordnet source est, la plupart du temps, le Princeton WordNet, considéré comme la référence la plus complète. Cette approche a montré beaucoup d'avantages :

- Facilité de mise en œuvre et faible coût.
- Gain de temps permettant de viser une large couverture.
- Structure compatible avec Wordnet, ce qui facilite les comparaisons.
- Réutilisation des ressources liées à Wordnet (SUMO, domaines de Wordnet etc.), voire des outils informatiques déjà développés (pour la recherche, l'édition, la visualisation) dans le cas d'une implémentation identique.
- Possibilité d'utiliser WordNet comme pivot dans la comparaison avec d'autres langues que l'anglais.

Mais la transposition d'un wordnet à un autre pose également certains problèmes. Un des critiques les plus fréquentes sur les travaux de transposition utilisant PWN ou EWN est que parfois le sens des unités de ces deux wordnets sont si proches que la distinction est difficile à faire (Gonzalo et al, 2000) ce qui produit certaines lacunes lexicales lorsqu'on les relier à un nouveau wordnet p.ex. (*Synset ENG2009740423n de PWN contient performer et performing artist).* Cependant, il n'y a pas d'équivalent français décrivant les acteurs, chanteurs et d'autres artistes collectivement (Sagot et Fišer, 2008).

Par ailleurs, il n'y a pas d'équivalence totale d'une langue à l'autre et certains champs lexicaux peuvent être structurés différemment (Mounin, 1963) à cause de l'interaction différente des cultures. P.ex. (*en anglais, on différencie souvent la viande de l'animal - ce qu'on ne fait pas en français veal/calf, pork/pig , mutton/sheep, chicken/hen, beef/cow*). Un autre exemple concernant les lacunes

lexicales et l'indisponibilité des équivalents : p.ex. En arabe on trouve des lexèmes qui n'ont pas d'équivalents français ou anglais (voir Tableau 1):

Lexèmes arabes	Translitération Buckwalter	Définition
عيد الفطر	Eyd AlfTr	Un événement socioreligieux dans lequel les musulmans célèbrent la fin de leur jeûne pendant le mois de Ramadan
اضحية	>DHyp	Le mouton abattu comme un sacrifice le jour de l'Aïd
سحور	sHwr	Un repas léger avant de commencer une nouvelle journée de Ramadan
زكاة	zkAp	L'aumône annuelle obligatoire (2,5%) des économies d'un musulman quand un montant ou des biens excèdent

Tableau 1 : Exemple de lacunes lexicales (Elkateb et al, 2006)

Étant donné que la relation entre les sens de différentes langues n'est pas toujours une relation de correspondance exacte, on peut considérer trois types de relations entre deux langues L1 et L2 :

1- Relation nulle : aucune équivalence, même partielle n'existe. Par exemple le substantif italien *astemio*, porte un sens ('personne qui ne boit pas d'alcool') qui n'a pas de support lexical en français (d'où la nécessité de le traduire par une périphrase).
2- Relation partielle : une ou plusieurs unités de la langue L2 portent des sens qui recouvrent partiellement un sens de L1, sans toutefois qu'il y ait correspondance exacte.
3- Relation pleine : il existe une ou plusieurs unités en L2 qui portent le même sens lexical qu'un sens de L1.

Dans le cadre de l'approche par extension, (Sagot et Fišer, 2008*)* proposent une méthode intéressante faisant intervenir des ressources multilingues (des corpus alignés) afin de construire un réseau sémantique de type *wordnet* pour le français (WOLF). Ils utilisent des ressources librement disponibles, tels que: Wikipedia, le thésaurus EUROVOC20 et un corpus parallèle CCR-Acquis19 de cinq langues alignées. Leur but est de traiter tous les types de lexèmes, y compris polysémiques.

Ces lexèmes sont ensuite désambigüisés sémantiquement à l'aide des différents *wordnets* de langues concernées. Ils se basent sur l'idée suivante :

> *"Les différents sens des mots ambigus dans une langue donnée donnent souvent lieu à des traductions différentes dans une autre langue. À l'inverse, nous supposons que si deux mots ou plus sont traduits par le même mot dans une autre langue, ils partagent souvent un élément de sens. En outre, ces phénomènes sont renforcés par l'utilisation de plus de deux langues, d'où l'intérêt d'une approche par alignement multilingue. "* (Sagot et Fišer 2008 : P3)

Pour l'extraction des synsets pour des lexèmes monosémiques, ils utilisent Wikipedia et EUROVOC puisque un lexique bilingue est suffisant, car aucune désambiguïsation n'est nécessaire. À cette fin, ils ont créé un lexique bilingue anglais-français avec 314 713 entrées.

Pour l'extraction des synsets pour des lexèmes polysémiques, ils utilisent un corpus parallèle afin de créer un lexique multilingue pour l'anglais, français, roumain, tchèque et bulgare. Ils ont, ensuite, comparé les entrées de chaque langue, dans le lexique multilingue, avec le correspondant WordNet dans BalkaNet.

Les identifiants (ID) des synsets ont ensuite été prises à partir des WNs et si toutes les langues, à l'exception de la langue française, partagent le même ID pour la même entrée lexicale multilingue, le même ID sera attribué au lexème français.

Pour l'évaluation, ils ont comparé WOLF avec le WordNet français dans EWN. En ce qui concerne les noms, ils ont obtenu 80,4% de précision et 74,5% de rappel. Par rapport aux verbes, ils ont obtenu 63,2% de précision et 52,5% de rappel. On voit que le découpage des sens et l'organisation lexicale varient d'une langue à l'autre. Les variations polysémiques des lexèmes sont rarement parallèles d'une langue à l'autre, et les acceptions voisines s'éloignent dans des directions différentes dans chacune de langues. D'où la nécessité d'organiser les lexiques sémantiques en fonction des sens, à un niveau de granularité plus fin que celui des mots, comme dans l'architecture de WordNet.

Dans une perspective voisine, quoique plus empirique, (Ploux et Ji, 2003) ont montré que le mot n'est pas l'unité adaptée pour l'établissement des équivalences traductionnelles.

En effet, (Ploux et Ji, 2003) ont utilisé une base lexicale de synonymes afin d'organiser automatiquement le sens des mots dans un espace multidimensionnel. Ils créent des graphes de synonymes qui leur permettent d'identifier des *cliques,* représentant selon eux des unités de granularité du sens plus fine que le mot. Ces cliques sont *"une intersection de plusieurs aires associées à un ensemble de synonymes tous synonymes entre eux*" (Ploux et Ji, 2003). Les unités ainsi construites sont censées faciliter la désambigüisation sémantique et l'établissement d'équivalence traductionnelles, *"parce qu'elles représentent des grains plus fins, qu'elles sont moins sensibles aux découpages d'une langue donnée et qu'elles forment de meilleurs candidats pour un ajustement mutuel"* (Ploux et Ji, 2003).

Notons que dans un réseau sémantique monolingue, la synonymie est basée sur l'équivalence (souvent partielle) entre deux sens. La relation de synonymie, en tant que principe d'équivalence, peut être utilisée pour délimiter les nœuds au sein d'un réseau sémantique monolingue : Si A est synonyme de B et si A est synonyme de C alors que B n'est pas synonyme de C, c'est que probablement A possède deux sens qui devraient être différenciés par deux nœuds du réseau. (Levrat et Sabah, 1990 : 93) Ainsi la mise en évidence de l'équivalence de certaines unités, conduit également à une prise en compte du fait polysémique. La synonymie et la polysémie sont donc des phénomènes conjoints et inséparables, liés au fait de la non-congruence entre le niveau des lexèmes et le niveau, plus fin, des sens (un lexème pouvant avoir plusieurs sens, et plusieurs lexèmes pouvant partager le même sens).

Le modèle sémantique de (Ploux et Ji, 2003*)* a aussi été appliqué à deux langues. Les cliques de la langue cible sont traduites en langue source en se basant sur trois bases lexicales : une base de synonymes de la langue source, une base de traduction entre langue source et langue cible, et une base de synonymes de la langue cible. Ensuite, le résultat est projeté sur une carte sémantique afin de mettre en évidence les appariements entre les valeurs sémantiques de la langue source et celles de la langue cible.

Cet ensemble de cliques multilingues permet de proposer des termes candidats pour la traduction dans la langue cible. Mais la liste des synonymes préparée manuellement pour chaque unité est indispensable pour la mise en œuvre de ce modèle.

Dans notre projet d'élaboration d'une ressource sémantique arabe, nous nous inspirons du principe des cliques proposé par (Ploux et Ji, 2003). Nous visons également à résoudre le problème du

recouvrement des mots polysémiques avec leurs équivalents multilingues, *mais d'une manière totalement automatique*. Nous envisageons de récupérer les relations sémantiques lexicales de types *équivalences traductionnelles* en construisant des cliques de mots quadrilingues dont toutes les unités sont en interrelation, du fait d'une probable intersection sémantique.

Nous choisissons de travailler sur des cliques et non pas des mots, comme dans le travail de (Sagot et Fišer 2008) parce qu'on suppose que les cliques permettent d'organiser les lexiques sémantiques en fonction des sens, à un niveau de granularité plus fin que celui des mots, de plus, ces cliques sont moins sensibles aux découpages d'une langue donnée et elles forment de meilleurs candidats pour un ajustement mutuel.

Ces cliques présentent l'intérêt de renseigner à la fois sur la synonymie et la polysémie des unités, et d'apporter une forme de désambiguïsation sémantique. En effet, l'utilisation de plusieurs langues permet de désambiguïser les lexèmes polysémiques dans le cas de corpus alignés. Il est en effet peu probable qu'une même polysémie se retrouve dans de nombreuses langues différentes.

Nous supposons qu'en reliant ces cliques, apportant une forme de désambiguïsation sémantique avec un lexique sémantique de type wordnet, nous pourrons récupérer pour les unités arabes, des relations sémantiques définies pour des unités d'autres langues.

Nous espérons qu'une comparaison positive entre nos cliques et les synsets des wordnets, déjà créés, nous permettrait de construire automatiquement une ressource sémantique arabe utile pour certaines applications de traitement de la langue arabe, comme les moteurs de question-réponse, la traduction automatique, les systèmes d'alignement, la recherche d'information, etc.

Mais à présent examinons plus en détail de quelle manière sont organisés le Princeton WordNet et les réseaux dérivés.

I.3 Les réseaux de type " Wordnet "

I.3.1 Présentation générale

Depuis les années 1990, de nombreuses bases de connaissances ont abouti à la construction de lexiques sémantiques en TAL, afin de répondre aux besoins en ressources sémantiques lexicales élaborées.

Une des ressources les plus complètes et les plus répandues est *WordNet* (noté WN, parfois appelé Princeton WordNet pour la distinguer des autres wordnets). Il s'agit d'une grande base de données lexicale (*computational lexicons*) pour l'anglais, développée par le psychologue George Miller (Miller et al, 1990) et ses collègues à l'université de Princeton. Cette ressource, d'abord utilisée dans le cadre d'expériences de psychologie, a été élaborée dans le but de répertorier, classifier et mettre en relation les sens et les lexèmes de la langue anglaise.

" The most ambitious feature of WordNet, however, is its attempt to organize lexical information in terms of word meanings, rather than word forms.

[…]

In order to reduce ambiguity, therefore, ''word form'' will be used here to refer to the physical utterance or inscription and ''word meaning'' to refer to the lexicalized concept that a form can be used to express. Then the starting point for lexical semantics can be said to be the mapping between forms and meanings" …. (Miller 1993 :P.4)

Aujourd'hui WN est devenu une des ressources les plus utilisées dans les applications de compréhension et d'interprétation automatique, les moteurs de question-réponse, le résumé automatique, et tout spécialement pour les tâches désambiguïsation sémantique. C'est devenu un standard de fait, et de nombreux dérivés ont été construits pour d'autres langues, avec la même structure, pour une meilleure interopérabilité. Quoique ce soit un choix discutable, du fait des spécificités de la langue anglaise, il a été utilisé comme une ressource pivot pour de nombreux couples de langues.

La version actuelle est la 3.0 réalisée en 2005, et contient quatre catégories de lexèmes : noms, verbes, adverbes, et adjectifs. La composante sémantique atomique est nommée le "*synsets*" : "

Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept." [7]

I.3.1.1 Les synsets

WN est construit manuellement et structuré autour d'unités de base, les synsets, correspondant à des concepts définis, d'après les auteurs, sur le plan *cognitif*. Chaque synset dénote donc un "concept" différent d'un mot précis, décrit par une courte définition appelée *gloss*.

Il ne s'agit donc pas, dans la définition, de sens lexical stricto sensu. Pourtant, dans la mesure où chaque synset est explicité par un ensemble de lexèmes groupés autour d'un même sens (*synset* venant de *synonym set*), nous pensons que ces "concepts" possèdent, par construction, un certain ancrage linguistique dans la langue anglaise. De nombreux auteurs ont remis en discussion l'universalité supposée des concepts inscrits dans WordNet, à commencer par le choix d'ancrer ces concepts aux 4 catégories dites fondamentales (nom, verbes, adjectifs et adverbes). Rastier (2004) note que cette conception est héritée des approches *référentielles* du *cognitivisme orthodoxe*, elle-même issue des conceptions ontologiques de la philosophie du langage : *"Les grandes ontologies unilingues* (*projet WordNet développé à Princeton*) *ou multilingues* (*Eurowordnet*) *restent partagées en secteurs différents, pour les noms, les verbes et les adjectifs, les adverbes, sans aucune autre justification que des préjugés ontologiques – coûteux pour la pensée comme pour le contribuable – sur la référence supposée de ces parties du discours."* (Ontologies, Revue d'Intelligence artificielle, 2004, vol. 18, n°1, p. 15-40). Or ces catégories, ont le sait, n'ont rien d'universel en ce qui concerne les systèmes linguistiques : les grammaires classiques de l'arabe, par exemple, distinguent seulement 3 catégories principales - le nom, le verbe et la particule.

Ne reconnaissant pas l'universalité de WordNet et tant qu'ontologie, nous préférons parler, en ce qui nous concerne, de *lexique sémantique* et préférons utiliser le terme de *sens* plutôt que celui de *concept* pour la définition des synsets. Bien que limité et ancré dans une certaine tradition linguistique et logique, l'intérêt de WordNet est pour nous de nature opératoire, pour des applications de TAL, et tient à sa double organisation entre synsets et lexèmes, qui permet de rendre compte de façon très souple, on l'a vu, de l'organisation de la synonymie et de la polysémie.

---

[7] Cf. http://wordnet.princeton.edu/, consulté le 7 mars 2012

I.3.1.2 Les relations

Les synsets sont reliés entre eux par des relations conceptuelles-sémantiques et lexicales (Turenne, 2000).

Les relations établies entre les concepts dans WordNet sont les suivantes :

- Relation d'hyperonymie-hyponymie 'IS-A' :

Ex. arbre IS-A plante.

- Relation de méronymie-holonomie 'HAS-A' ou 'HAS-PART' :

Ex. voiture HAS-PART roues.

- Relation d'antonymie (VS.) :

Ex. joli VS. laid

- Relation d'implication, définie seulement pour les verbes ' ENTAILS'.

Ex. boiter ENTAILS marcher

- Relation de dérivation morphologique, définie seulement pour les adjectifs, adverbes et noms 'PERTAINS TO'.

Ex . parental PERTAINS TO parent

WN structure hiérarchiquement les mots et les relie par des relations d'antonymie et de méronymie. Ces relations permettent donc d'aboutir à des concepts plus abstraits de plus haut niveau que les mots et leurs sens.

Exemple : Canari a les hyperonymes suivants : oiseau, animal, organisme, objet, entité.

WN, du fait de son architecture, permet d'articuler de façon économe ces deux propriétés fondamentales des langues que sont la synonymie et la polysémie :

- **Synonymie :** La synonymie signifie qu'un même sens est exprimé par différents lexèmes. Même si on admet qu'il n'existe pas en langue de synonyme parfait, cette propriété peut être

conçue sous une forme plus restreinte, comme désignant l'intersection sémantique partielle entre deux lexèmes, qu'on considère comme étant potentiellement substituables dans certains contextes sans altération de la dénotation (en faisant abstraction des valeurs de connotation, de registre, etc.). Cette relation symétrique est à la base de la construction des synsets :

Exemple:

{beat, hit, strike}

{car, motorcar, auto, automobile}

- **Polysémie :** Elle signifie qu'un même lexème peut prendre plusieurs sens. On constate que les lexèmes les plus fréquents sont aussi les plus polysémiques. Dans WN, la polysémie d'un lexème se manifeste simplement par l'appartenance à différents synsets.

Exemple :

{table, tabular_array}

{table, piece_of_furniture}

{table, mesa}

{table, postpone}

WN divise les adjectifs en deux catégories :

- Les **adjectifs descriptifs** forment la plus grande catégorie. Leur fonction logique consiste à assigner à un certain attribut de nom une valeur (*ex. Nom : paquet → Attribut : poids → valeur : adj.-lourd*). Les adjectifs descriptifs sont structurés par les relations d'antonymie et de similarité.
- Les **adjectifs relationnels** forment une classe beaucoup plus petite. Les adjectifs relationnels sont reliés par dérivation à un nom *ex. (electrical*).

Enfin, les verbes dans WN sont regroupés dans trois classes : Actions, Evénements et Etats.

Le succès de WordNet a conduit à l'émergence de plusieurs projets visant la construction de wordnets pour d'autres langues que l'anglais (Hamp & Feldweg, 1997), (Artale et al. 1997). En effet

la demande de wordnets multilingues se fait de plus en plus pressante pour de nombreuses langues, afin de rattraper l'écart technologique en termes de contenus, de services (*ex. traduction automatique, recherche d'information, etc.*) et d'usages sur les réseaux d'information.

I.3.2 Wordnets pour d'autres langues que l'arabe

I.3.2.1 EuroWordNet (EWN)

EuroWordNet est un ensemble de wordnets développés conjointement dans le cadre du projet de la Commission Européenne dans le but de développer une base de données pour plusieurs langues européennes, initialement le néerlandais, l'italien, l'espagnol, l'allemand, le français, le tchèque et l'estonien. Ces wordnets sont structurés de la même manière dont Princeton wordnet est structuré, en ce qui concerne les synsets et relations sémantiques. Pour chaque langue a été développé un wordnet autonome, qui représente un système de langue-interne unique de lexicalisations, mais ils sont tous liés entre eux par des enregistrements nommés ILI-RECORDS (ILI pour Inter-Lingual Index), faisant référence à des synsets de WordNet 1.5, qui a joué le rôle de pivot. A partir de ces ILI-RECORDS il est possible d'extraire des correspondances dans plusieurs langues, pour n'importe quel couple de l'ensemble considéré. L'accès à certains échantillons d'EWN est disponible gratuitement, tandis que l'accès à la totalité de la base de données n'est pas libre.

EWN emprunte à WordNet sa structure de base, fondée sur les synsets et les relations hiérarchiques. De nouvelles relations ont été ajoutées ultérieurement, comme exemple la relation entre un verbe et son sujet ou les actants sémantiques des verbes. (*Ex. étudiant est relié au verbe enseigner par la relation 'ROLE_PATIENT'*).

Des extensions de EuroWordNet ont été développées par la suite pour de nombreuses autres langues européenne (suédois, norvégien, danois, grec, portugais, basque, catalan, roumain, lithuanien, russe, bulgare et slovène). Globalement EWN contient environ 50 000 synsets qui sont en corrélation avec les 20 000 mots les plus fréquents (seulement des noms et des verbes dans la première étape du développement) dans chaque langue.

L'architecture multilingue d'EWN se compose des éléments suivants :

- *Wordnet monolingue* : Chaque langue a son wordnet individuel avec des relations internes qui reflètent des propriétés spécifiques pour cette langue. Cependant, chaque wordnet

monolingue est structuré par un ensemble commun de 1024 concepts de base (concepts qui sont relativement haut dans les hiérarchies sémantiques et qui ont plusieurs relations avec d'autres concepts). Ceux-ci ont été vérifiés manuellement pour être adaptés à tous les wordnets monolingues. C'est cette superstructure conceptuelle qui est censée garantir l'adéquation et la compatibilité entre wordnets, réduisant les disparités dans la hiérarchie.

- *Inter-Lingual-Index (ILI-RECORDS)* : Les ILI correspondent à un *sur-ensemble* de tous les sens existant dans les wordnets monolingues. Ils ont d'abord été utilisés comme une collection de pointeurs vers les synsets de WN 1.5, et ont été enrichis avec l'ajout de nouveaux sens, rendu nécessaire par la croissance du réseau et la prise en compte de nouvelles langues. Toutes les relations interlinguistiques sont construites sur ces ILI-RECORDS.
- *Relations interlinguistiques* : Les wordnets sont reliés aux ILI-RECORDS via différents types de relations d'équivalence interlinguistique :
  - Synonymie interlinguistique: (Ex. it-*anatra* EQ-NEAR-SYNONYME en-*duck)*
  - Hyperonymie interlinguistique : (Ex. nl-*hoofd* (tête humaine) EQ-HAS-HYPERONYM en-*head)*.
  - Hyponymie interlinguistique : (Ex. es-*dedo* (le doigt ou l'orteil) EQ-HAS-HYPONYM en-*finger*, es-*dedo* EQ-HAS-HYPONYM en-*toe)*.

Les relations interlinguistiques complexes (hyperonymes et hyponymes) indiquent de nouvelles relations potentielles entre ILI-RECORDS. Après chaque étape de construction, toutes les relations complexes sont rassemblées et comparées à travers des langues, et les nouvelles relations entre ILI-RECORDS sont ajoutées si elles sont appropriées. Ces relations facilitent l'extraction des correspondances interlinguistiques.

- *Ontologie de concepts supérieurs* (top-ontology): Il s'agit d'une hiérarchie de 63 concepts indépendants des langues reflétant des relations d'opposition jugées universelles (*ex. object vs substance voir Figure 6*). Cette ontologie est reliée aux concepts de base via les ILI-RECORDS. Elle s'utilise pour clustériser, comparer et échanger des concepts à travers les langues.

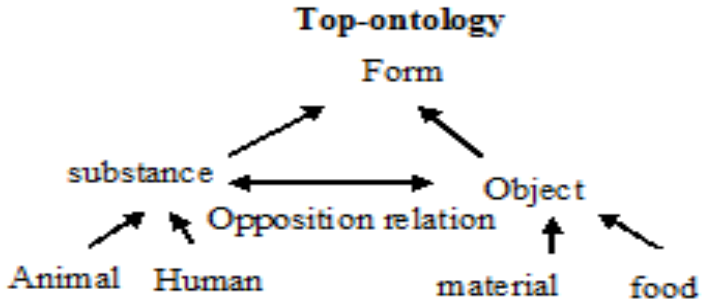

Figure 6 : Exemple d'une relation d'opposition dans EWN

- *Hiérarchie d'étiquettes de domaine* : Cette hiérarchie est également liée avec les ILI-RECORDS, et ainsi héritée par chaque WN monolingue (ex. *Sport, Politique, Médecine, Economie…etc.*)

I.3.2.2 BalkaNet

Le but de BalkaNet est de construire une base de données lexicale multilingue contenant des wordnets pour les langues du Centre et de l'Est d'Europe (tchèque, roumain, grec, turc, bulgare et serbe).

Il s'agit d'un prolongement de la base de données d'EWN. La première liste de sens de base de BalkaNet contenait 1 310 synsets, tirés d'EWN. Chaque wordnet monolingue est structuré selon les principes déjà utilisés dans EWN, et les relations d'équivalence entre les synsets des différentes langues sont établies via les ILI-RECORDS de EWN. Comme nous l'avons déjà mentionné, les ILI-RECORDS constituent une collection non structurée de sens dont le but est d'établir des relations entre les différentes langues. Chaque wordnet de BalkaNet a d'abord été construit séparément, puis reliés aux entrées des ILI-RECORDS.

Afin de garder la compatibilité avec EWN, l'ontologie de concepts supérieurs a également été utilisée, parallèlement aux ILI-RECORDS. Mais du fait de grandes différences de sens avec les ILI-RECORDS existants, les équivalents sémantiques entre wordnets ne sont pas toujours reliés aux mêmes ILI-RECORDS, bien qu'ils restent dans le même domaine. P.ex. Les deux équivalents (*es-universidad et nl-universiteit*) ne partagent pas le même ILI-RECORDS à cause de la granularité sémantique très fine dans PWN (voir Figure 7).

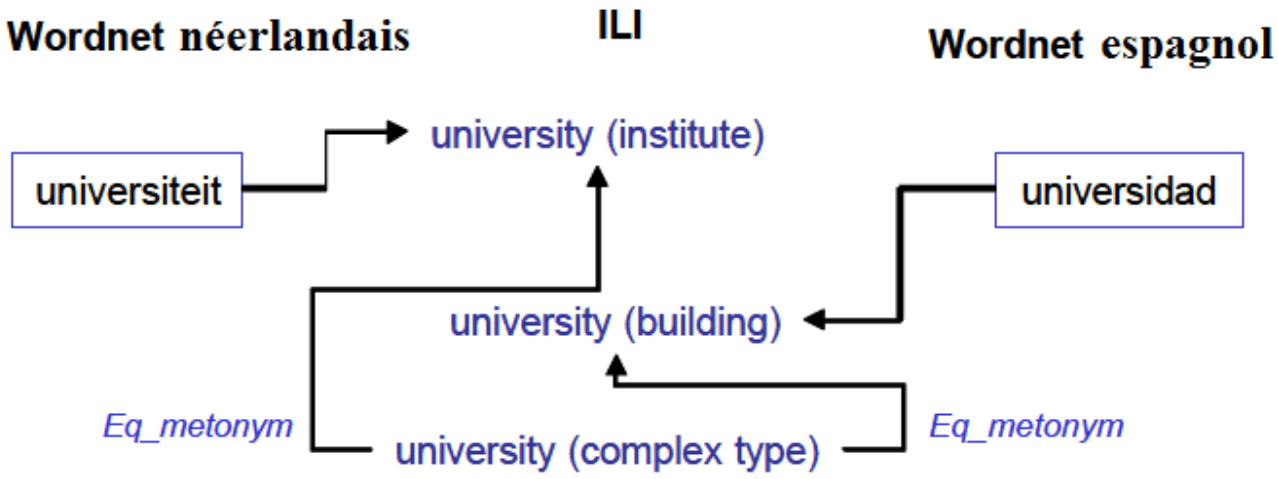

Figure 7 : Equivalents reliés aux différents ILI-RECORDS[8]

Différents processus ont été mis en œuvre pour les langues de BalkaNet. Par exemple, pour l'élaboration du wordnet bulgare, les auteurs ont utilisé un dictionnaire anglais-bulgare et bulgare-anglais. Les candidats synsets ont été construits automatiquement par traduction en bulgare de chaque mot des synsets anglais, en utilisant le dictionnaire anglais-bulgare. Les synsets ont ensuite été filtrés en sélectionnant, grâce au dictionnaire bulgare-anglais, les meilleurs candidats parmi les différentes traductions possibles pour un mot source anglais. Cette façon de projeter les synsets anglais vers le bulgare a obtenu, d'après les auteurs (Koeva et al. 2004), de bons résultats au niveau des noms. Cette méthode, qui s'appuie sur une correspondance exacte entre synsets bulgare et synsets anglais, nous semble discutable car elle présuppose que les synsets anglais sont des concepts à vocation universels, ce qu'ils ne sont pas.

Pour l'élaboration du wordnet tchèque, les auteurs ont utilisé un programme qui analyse des entrées de dictionnaires et qui sélectionne les définitions contenant des hyperonymes. Par ailleurs, un programme de traduction capable d'établir un dictionnaire bilingue (tchèque-anglais-tchèque) a été créé, pour relier les entrées tchèques avec leurs équivalents en Princeton WN.

Les concepts de base de BalkaNet ont d'abord été choisis selon des critères spécifiques pour chaque langue. Dans un second temps, ces concepts multilingues ont été comparés afin de choisir les concepts communs ou les plus proches. Si deux équivalents proches sont trouvés dans deux langues, ils sont sélectionnés pour former la liste de concept de base. Sinon, on recherche un équivalent plus

[8] cf. http://dare.ubvu.vu.nl/bitstream/1871/10550/1/VossenMulti1.pdf

proche en suivant une relation d'hyper/hyponymie. Actuellement, les concepts de base de BalkaNet constituent environ 5 000 concepts.

Enfin, des relations spécifiques ont été ajoutées, du type cadre valenciel (*valency frames*), du fait de la richesse morphologique de certaines langues.

> *Exemple de relations : role_agent, involved_agent→ drive→driver*
>
> *Exemple de valency frames ajoutés au synsets de verbes :*
>
> *"start to operate; start to function; of an enterprise such as a business, school, play, opera, etc.& 2ndOrderEntity 41 BoundedEvent Cause Dynamic SituationType Social Usage" =(who1|what2)/(Person, Institution) - open - what4/pat/(school, business)*

I.3.3 Stratégies pour la construction d'un wordnet multilingue

Des expériences intéressantes dans le domaine de la construction d'un WN multilingue ont été faites pendant ces dernières années, en se basant sur des ressources lexicales préexistantes comme WN ex. *(Hawkins et Nettleton, 2000).*

Ces derniers expériences utilisent l'architecture hiérarchique de WN, principalement les différents sens définis ainsi que les relations sémantiques établies entre les nœuds, pour choisir celui des sens qui, en fonction de la structure de WN, se rapproche le plus du sens de lexème de la langue cible. Ces WNs multilingues élaborés ont grandement amélioré l'efficacité des applications s'appuyant sur la désambigüisation sémantique (*ex. la traduction automatique, la recherche d'information…etc.).*

Différentes stratégies sont utilisées pour créer un nouveau wordnet relié avec WN, afin d'obtenir un réseau multilingue. Comme nous l'avons exposé précédemment (section 2.4), nous distinguons essentiellement deux approches:

- l'approche *par fusion*, consistant à créer d'abord un wordnet indépendant, puis à relier Les synsets de la langue A avec les synsets de la langue B, en annotant les relations d'équivalence afin d'avoir un wordnet parallèle. Ces relations d'équivalence peuvent être calculées en générant des traductions pour les synsets afin de les apparier (ex. Wordnet tchèque (Pala and SMR·Z, 2004)). Cette approche est souvent utilisée pour les langues mieux dotées en termes d'outils et de ressources (*high density language*) (*ex. DanNet project (Pedersen and Nimb 2008)).*

- l'approche *par extension*, consistant à créer les synsets de la langue B en traduisant ceux de la langue A. Les relations lexicales sémantiques de la langue B sont alors projetées à partir des relations correspondantes en langue A, ce qui permet ensuite d'hériter de la structure du wordnet déjà construit. Moins coûteuse, cette approche est plutôt utilisée pour les langues moins dotées en termes d'outils et de ressources (*Low-density language*) (*ex. wordnet suédois et russe*).

Cette dernière approche présente des avantages certains sur le plan pratique, mais elle se base sur l'hypothèse d'une forte similarité avec l'organisation sémantique des sens de l'anglais, hypothèse linguistiquement très discutable.

Avant de montrer les limitations de ces wordnets et d'évaluer les méthodes suivies pour leur construction, arrêtons-nous sur les tentatives spécifiques concernant l'arabe.

I.3.4 La langue arabe

Un problème majeur au sein de la communauté des utilisateurs de wordnets est la disponibilité des wordnets développés. Actuellement, seuls quelques-uns d'entre eux sont librement disponibles.

De nombreux efforts ont été consacrés au traitement automatique de la langue arabe, mais les différents problèmes posés par cette langue et leur spécificités graphiques et morphologiques ont ralenti le processus du développement des outils dans ce domaine.

Dans la suite, nous présenterons certaines propriétés morphologiques et syntaxiques de la langue arabe.

I.3.4.1 Particularité de la langue arabe

La langue arabe est la langue officielle de centaines des millions de personnes dans vingt pays du Moyen Orient et du nord de l'Afrique, et c'est la langue religieuse de tous les musulmans de diverses ethnies dans le monde entier. L'arabe est une langue sémitique qui diffère de langues indo-européennes syntaxiquement, morphologiquement et sémantiquement. Sur le plan de l'écrit, on trouve dans la langue arabe des voyelles diacritiques qui sont écrites au dessus ou au dessous une consonne pour lui donner le son et le sens désiré. Mais les arabophones utilisent rarement ces voyelles dans leur écriture, puisque pour eux le contexte suffit à leur faire comprendre le sens exact

du mot. Cela peut cependant aboutir à de nombreuses ambiguïtés lors du traitement automatique de la langue.

En fait, dans la langue arabe on utilise différents types de voyelles qui déterminent la prononciation des lettres:

1. Voyelles brèves : ces voyelles sont notées avec des signes diacritiques qui permettent de savoir comment une lettre est prononcée dans un mot donné. L'écriture de ces voyelles est facultative et concerne peu de textes (Coran et textes didactiques), ce qui cause beaucoup d'ambiguïté au niveau de la prononciation et la compréhension de textes courants.

Ces voyelles ne s'écrivent pas dans le corps du mot, mais ils sont ajoutés sur ou sous la consonne à laquelle ils se réfèrent. La graisse FATHA et le KASRA sont représentés par un tiret mis respectivement au dessus et au dessous de la consonne à laquelle ils se réfèrent. Le DAMMA ressemble à un petit WAW (و) et il est écrit au-dessus de sa consonne.

Les voyelles courtes en arabe sont :

Signe	Translittération	Nom arabe	Nom de voyelle
◌َ	A	فَتْحة	FATHA
◌ِ	I	كَسْرة	KASRA
◌ُ	U	ضَمّة	DAMMA

Tableau 2 : Voyelles courtes arabes

2. Voyelles longues : Les voyelles longues sont la version allongée des courtes. Pour représenter une voyelle longue, la langue arabe adopte une méthode particulière *la lettre d'allongement*: on ajoute une lettre juste après la consonne et sa voyelle courte. Chaque voyelle courte a son signe d'allongement propre.

Voyelle courte	Exemple de sons courts	Prononciation selon API	Elongation de signe	Résultat	Exemple	Prononciation selon API	Sens
◌َ	ب	/ba/	ا	بَا	بَاب	/bāb/	porte
◌ِ	ب	/bi/	ي	بِي	كَبِير	/kabīr/	grand
◌ُ	س	/su/	و	سُو	سُوق	/sūq/	marché

Tableau 3 : Voyelles longues arabes

3. TANWIN (ALFATHA, ALKASRA, ALDAMMA), c'est-à-dire le doublement de voyelle courte : TANWIN ALFATHA et ALDAMMA se trouvent sur la dernière lettre d'un mot mais TANWIN ALKASRA se trouve sous la dernière lettre. Ces types de TANWIN sont utilisés pour indiquer que la prononciation de la dernière lettre est le son de cette lettre influencée par une des voyelles courtes, une FATHA , DAMMA ou KASRA , selon le type de TANWIN trouvé, et la deuxième FATHA , DAMMA ou KASRA désigne le son /n/.

> *Exemple :"بَاب " (trans.BAB- /ba/bun/ -Trad. La porte : La deuxième partie /bun/ désigne la lettre /b/+ DAMMA /u/+le son /n/ indiqué par la deuxième DAMMA.*

Voici les différents types de TANWIN :

- TANWIN ALFATHA :(ـً);
- TANWIN ALKASRA :(ـٍ)
- TANWIN ALDAMMA :(ـٌ)

4. SHADDA (ـّ): désigne un doublement de la lettre sur la quelle elle se trouve.

5. SUKUN (°): indique que la prononciation de la lettre où se trouve le SUKUN devrait être clair et aigu.

Par ailleurs, la langue arabe n'utilise pas de lettres majuscules (pour les noms propres ou au début d'une phrase) ce qui rend la tâche de désambigüisation plus difficile.

Le *mot* au sens graphique (suite de graphèmes entre deux blancs) se compose souvent en arabe de plusieurs formes collées : on parle alors *d'agglutination*. La structure du mot arabe est décomposable en cinq éléments : proclitique, préfixe, base, suffixe et enclitique. Cette caractéristique peut être source de difficultés pour l'analyse morphosyntaxique du texte arabe, car la distinction entre les formes agglutinées et les lettres radicales d'un mot est très difficile à faire. Certaines décompositions peuvent causer des ambiguïtés. La segmentation des unités lexicales (opération nommée tokenisation en TAL) est déjà un problème en soi.

En ce qui concerne la morphologie, l'arabe se compose essentiellement de trois sous-ensembles : le nom, le verbe, les particules qui se subdivisent elles-mêmes en différentes sous-catégories : préposition, conjonction, pronom, article, interjection et adverbe.

Les verbes et les noms sont le plus souvent dérivés d'une racine à trois consonnes radicales *(Baloul, Alissali, Baudry & de Mareuil, 2002).* Les déclinaisons, qui s'attachent au début ou à la fin de radical, permettent par exemple de distinguer le mode du verbe ou la nominalisation.

En outre, un nom au pluriel prend une autre forme morphologique différente de sa forme initiale du singulier. Par exemple, le mot (امرأة) (trans.<mr>p/trad.femme) au singulier prend la forme (نسوة) (trans.nswp/trad.femmes) au pluriel.

Mais ces ensembles n'ont pas de comportement syntaxique suffisamment homogène, ce qui rend la tâche des certains outils, par exemple d'un étiqueteur morphologique, plus ardue.

I.3.4.2 Wordnet arabe

Les outils TAL génériques sont encore rares et peu développés au regard de l'importance de cette langue en terme de diffusion et de nombre de locuteurs. Il est donc motivant de développer un wordnet arabe, une ressource lexicale générique qui pourrait refléter la richesse de la langue arabe, comme décrit dans (Elkateb, 2005*),* et aboutir à de nombreuses applications telles que la désambigüisation sémantique.

Peu de travaux, à notre connaissance, ont contribué à l'élaboration d'un wordnet pour l'arabe (que nous noterons désormais par arWN). Parmi ces travaux, nous citerons la contribution la plus importante, qui est celle de (Alkateb et al., 2006).

Notons toutefois que (AWN : arab wordnet) proposé par (Alkateb et al, 2006*)* est une des rares ressources pour la langue générale arabe consultable en ligne. Les auteurs ont élaboré un wordnet basé sur la conception et le contenu du Princeton WordNet (PWN 2.0) et qui peut être relié directement avec EWN, les deux wordnets étant indexés par les ILI-RECORDS et l'ontologie SUMO (*Suggested Upper Merged Ontology*). Rappelons brièvement que SUMO : Suggested Upper Merged Ontology (Niles and Pease 2001) est une ontologie supérieure formelle créée par le IEEE Standard Upper Ontology (SUO) Working Group (2001), elle contient 1 000 termes et 4 000 formules définitionnelles exprimée dans la logique du 1er ordre. SUMO possède un modèle de génération en langage naturel et un lexique multilingue qui permet à SUMO d'être exprimé en plusieurs langues. Les relations d'hyperonymie et d'hyponymie sont strictement distinguées dans l'ontologie SUMO.

En fait, certains synsets de AWN ont été reliés aux concepts SUMO correspondants afin d'enrichir la base de donnée de AWN et de maximiser la compatibilité entre les deux WNs (AWN et PWN). Ces synsets peuvent être relié avec un concept SUMO équivalent ou parent qui englobe ce synset (*ex. TimeInterval est un enfant de TimeDuration et de TimePosition*). Les lacunes lexicales dans la relation d'hyperonymie sont remplies manuellement.

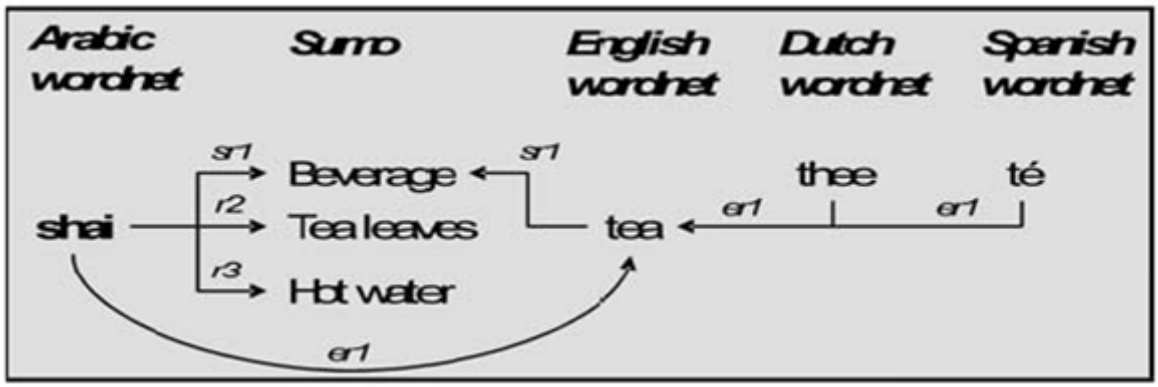

Figure 8 : Exemple du SUMO et ILI (Sabri Elkateb et al., 2006)[9]

Un grand nombre de termes et définitions SUMO correspondant à certains synsets arabes permettrait d'utiliser SUMO à la place des ILI-RECORDS (qui relient plusieurs WNs avec PWN). Comme c'est illustré dans Figure 8 : L'hypothèse suivi s'agit que si le terme arabe (shai) possède des multiples relations de différentes types avec des termes SUMO alors ces relations (ou cette définition) peuvent

[9] Sr : relation de subsomption er :relation d'équivalence

remplacer l'ILI-RECORD existant entre (shai) et le synset de PWN (tea) afin d'avoir un découpage du sens plus précis entre les deux langues.

La base de données relationnelle établie pour leur AWN contient quatre entités principales:

- Item : Contient les entités conceptuelles, synsets, classe d'ontologie et des instances. Chaque entité a un identifiant unique et des informations descriptives comme le gloss.
- Word : Contient le sens du mot, où la *forme* du mot est relié à *élément* via son identifiant.
- Form : contient des informations lexicales (ex. variation flexionnelle, la racine et/ou la forme plurielle pour l'arabe).
- Link (entre items) contient aussi le type de relation (ex. équivalence, subsomption…etc.).

Etant donnée que l'arabe est une langue sémitique, centrée autour des racines, ce réseau a été appliqué sur l'arabe utilisant les voyelles courtes, chaque mot arabe est relié avec sa racine non voyellée (sans les voyelles courtes).

Les auteurs ont d'abord déterminé les concepts de base puis détaillé ces concepts en élargissant SUMO afin d'enrichir la base sémantique d'AWN :

- Les concepts principaux communs de 12 langues sont codés comme des synsets dans arWN, mais les sens spécifiques pour chaque langue sont traduits et ajoutés manuellement aux synsets arabes les plus proches. La même étape est effectuée pour tous les synsets anglais qui ont une relation d'équivalence dans SUMO.
- Les synsets arabes sons ensuite reliés avec des relations hyperonymiques pour former une hiérarchie sémantique. Ce travail a été fait manuellement. SUMO est utilisé pour maximaliser la cohérence sémantique des liens d'hyperonymie (voir Figure 9). Des nouveaux termes formels seront définis afin de couvrir un plus grand nombre d'associations d'équivalence. Ces définitions des nouveaux termes, à leur tour, sont dépendentes de l'existence de concepts fondamentaux dans SUMO.

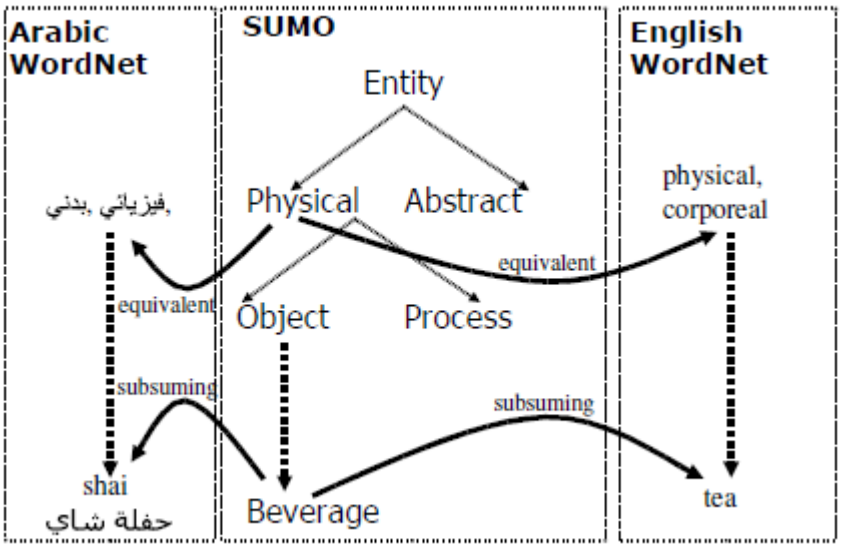

Figure 9 : Association entre SUMO et PWN

Les auteurs ont appliqué des procédures heuristiques sur un dictionnaire bilingue afin d'en tirer des candidats d'association entre mots arabes et synsets anglais. Pour chaque association (ou *mappage*) l'information attachée contient le mot arabe et la racine, le synset anglais, la catégorie, les fréquences relatives, un score de mappage, la profondeur absolue dans AWN, le nombre d'écarts entre le synset et le sommet de la hiérarchie AWN, et des sources contenant le paire de mots (arabe/anglais). L'ensemble marqué des paires du (mot arabe / synset anglais) constituent l'entrée d'un processus de validation manuelle. Les auteurs procèdent par *chunks* d'unités connexes (ensembles de synsets connexes de WN, par exemple les chaînes d'hyponymie, et des ensembles de mots arabes liés, c'est-à-dire ayant la même racine), au lieu d'unités individuelles (synsets, sens, ou mots).

En effet, la première partie des hyponymes est choisie en se basant sur des critères linguistiques (en fonction de critères telles que connectivité maximale, pertinence et généralité (Alkateb et al, 2006); la phase finale complète l'ensemble visé de concepts/synsets y compris les domaines spécifiques et les entités nommées. Chaque étape de construction d'un synset est suivie par une phase de validation, où la cohérence formelle est vérifiée et la couverture est évaluée en termes de fréquence d'occurrence et de distribution de domaine.

Mais malheureusement les auteurs ne donnent pas d'informations sur la précision de leurs résultats.

Bien qu'AWN ait été construit manuellement, des travaux sont en cours pour automatiser partiellement le processus. Cet AWN a été utilisé dans plusieurs applications de TAL : par exemple, le travail de la construction d'une ontologie baptisée Amine AWN (Abouenour et al., 2008). Ce travail est une contribution visant la mise en place d'un prototype d'un système de Question/Réponse arabe, qui illustre l'apport de cette ontologie pour la recherche et l'extraction de la réponse finale.

D'ailleurs, dans le cadre de la Recherche d'Information (RI) pour les textes en langue arabe, AWN a été utilisé pour amorcer le processus d'expansion des requêtes. Cela constitue un premier pas vers l'utilisation des méthodes linguistiques et des ressources lexicales pour l'enrichissement de requête dans un système de RI arabe (Abderrahim, 2009).

Mais comme nous avons déjà mentionné, l'ambigüité lexicale constitue toujours un problème initial lors de la construction d'un WN même pour l'arabe (Alkatib et al., 2006), qui affecte l'organisation du sens des mots. Il est vrai que le recours à la traduction de synsets (comme dans le travail de (Alkatib et al, 2006)) peut proposer une solution pour ce type de problème, ainsi pour l'indisponibilité des équivalents ou le problème des termes intraduisibles en comblant les lacunes qui peuvent exister entre les langues, mais la structure et l'extension différente de langues compliquent dans certains cas la tâche d'encodage de relations sémantiques entre les sens. Par exemple :

Le mot anglais (en-door) a deux sens différents :

1.knocking on the door : objet physique

2.going through the same door: ouverture

Tandis que le mot arabe équivalent qui est (ar-باب) (trans.bAb/trad.porte) a trois sens différents:

1.objet physique

2.ouverture

3.chapitre en sujet particulier dans un livre (qui ne se trouve pas dans la langue anglaise)

En conclusion, l'extension sémantique varie de la langue arabe à la langue anglaise.

I.4 Enjeux et limitations pour la construction de wordnet

I.4.1 Enjeux et limitations

Les travaux mentionnés auparavant, qui ont tiré avantage de cette ressource sémantique (WordNet) soulèvent tout un ensemble de problèmes.

En ce qui concerne les wordnets construits à partir de dictionnaires conventionnels, des problèmes spécifiques se posent : la circularité des définitions, les incohérences (par exemple lorsqu'une équivalence traductionnelle n'est proposée que dans un sens, et pas dans l'autre - ou lorsque les descriptions sémantiques sont de granularité plus fine en langue source qu'en langue cible, ou réciproquement), les problèmes de couverture (absence d'une entrée, absence de caractérisation d'une certaine acception), les problèmes de spécialisation (certains dictionnaires ne couvrent qu'un domaine de spécialité), etc.

Par ailleurs, la structure même des WN pose problème. La relation d'hyperonymie, quand elle touche l'ensemble du lexique d'une langue, se heurte à certaines limites. Par exemple, dans l'organisation des verbes dans PWN, les liens hiérarchiques sont moins élaborés, et l'on passe rapidement d'un sens spécialisé à des sens très généraux. En outre, il n'y a aucune catégorisation hiérarchique actuellement définie pour les adjectifs et les adverbes présents dans réseau.

En ce qui concerne les noms, enfin, on trouve des hiérarchies déséquilibrées, puisque certains noms sont liés à une grande chaîne de sens finement gradués, tandis que, d'autres sont très proches des concepts les plus généraux.

Ensuite, de nombreuses définitions de sens sont manquantes, et certaines relations font défaut. Prenons l'exemple du wordnet français : un petit ensemble de sens structurés, dans le domaine du *traitement de données*, a été construit, mais cet ensemble est restreint et surtout les sens ne sont pratiquement pas liés les uns aux autres : c'est une hiérarchie assez plate. (P.ex. *le lexème fr-citronnier qui existe dans wordnet français, n'est relié à aucun sens, alors qu'il aurait dû être relié par la relation d'hyperonymie/hyponymie "arbre fruitier" qui existe bien dans ce wordnet (Jacquin et al, 2006*)).

Comme PWN, EWN qui est probablement devenu le plus connu et le plus largement utilisé dans le domaine du TAL, souffre de problèmes liés à sa conception même. En effet, la plupart des travaux réalisés reposent aussi sur des dictionnaires traditionnels, ou des ressources électroniques comme PWN. Le problème est que les dictionnaires ont été réalisés pour un usage humain et non pas automatique. Ils manquent donc d'informations pragmatiques utiles à la désambiguïsation " *Véronis (2001) pense qu'on ne pourra pas progresser en désambiguïsation sémantique tant que les dictionnaires n'incluront pas dans leurs définitions des critères distributionnels ou des indices de surface (syntaxes, collocations ,…)* ". (Venant F., 2004)

D'ailleurs, à cause des différences structurales entre les langues, chaque langue possède un jeu de polysémie qui lui est propre. On peut également noter une polysémie importante et une granularité trop fine dans la langue anglaise (ex. le lexème *en-break* qui est répertorié avec 63 sens); cela a comme conséquence le fait que les équivalents sémantiques dans les wordnets ne sont pas reliés exactement avec le même sens de cet équivalent anglais.

Le point le plus délicat dans le passage de PWN et EWN réside dans la traduction des mots polysémiques. En effet, les traductions données par les dictionnaires bilingues ne correspondent pas forcément à tous les synsets d'un même mot en anglais. Pour relier ces traductions, il faut donc déterminer la ou les traduction(s) adaptée(s) à chaque synset.

Plusieurs recherches dans cette direction se sont appuyées sur des dictionnaires bilingues et des corpus parallèles pour y sélectionner les traductions les plus pertinentes selon diverses heuristiques. (Sagot & Fiser, 2008) utilise des corpus parallèles pour lesquels ils effectuent la désambiguïsation du corpus anglais à l'aide des synsets de WordNet et proposent les mots alignés de la langue cible comme nouveaux termes (Mouton & Chalender, 2010*)* utilisent un dictionnaire bilingue en caractérisant les relations sémantiques du réseau lexical par des propriétés syntaxiques distributionnelles. D'abord ils exploitent une mesure de similarité sémantique dans des espaces sémantiques afin de trouver des relations proches de la synonymie entre un synset et ses traductions candidates. Ils exploitent également les relations d'hyponymie et d'hyperonymie pour déterminer quel est le candidat de traduction le plus adapté, en se basant sur deux hypothèses:

" (1) les contextes syntaxiques d'un mot général apparaissent souvent comme contexte syntaxique de ses hyponymes (e.g. : la vitesse du véhicule,et la vitesse du train, du bateau, du camion) et (2)

l'éventail des contextes syntaxiques d'un mot spécifique est plus grand que ceux de ses hyperonymes (e.g. : la quille du bateau mais pas la quille du véhicule)". (Mouton & Chalender, 2010).

Mais la polysémie reste une question difficile à traiter, notamment lorsqu'elle est décrite avec une granularité très fine. Par exemple, c'est le cas du lexème *en-break* répertorié avec 63 sens dans WN. L'utilisation d'un tel réseau dans les applications de TAL peut devenir difficile (Kilgarriff, 1997).

Cependant la caractérisation de la polysémie n'est pas une faiblesse de WordNet, mais plutôt un atout, qui peut ouvrir des perspectives riches et éclairantes sur la structure lexico-sémantique, du moment qu'elle soit traitée d'une manière correcte.

Par conséquent, pour pouvoir réaliser l'objectif de cette thèse et constituer notre ressource sémantique arabe, nous devons commencer par résoudre le problème de l'organisation sémantique des mots polysémiques arabes en fonction des sens, en prenant en compte la granularité parfois trop fine des WN des autres langues.

I.5 Conclusion

Le succès des réseaux sémantiques tels que PWN dans plusieurs domaines a motivé l'élaboration des nouveaux WN pour d'autres langues.

La construction d'un WN pour une langue quelconque peut être abordée de différentes façons selon les ressources lexicales disponibles. Il est vrai que la méthode manuelle conduit aux meilleurs résultats, mais elle est coûteuse à mettre en œuvre. Par conséquent, des approches ont été réalisées de façon automatique ou semi-automatique, en profitant des ressources disponibles.

Généralement, quatre types de ressources ont été utilisés :

1. Wordnet anglais (PWN) qui a été utilisé comme une structure sémantique de base, à laquelle sont reliés les mots de la langue cible. Cette approche se base sur l'hypothèse qu'il existe une similitude conceptuelle et sémantique étroite entre l'anglais et la langue cible.
2. Les taxonomies lexicales et sémantiques déjà existantes pour la langue considérée.
3. Des dictionnaires bilingues.
4. Des dictionnaires monolingues.

Les dictionnaires monolingues ont été utilisés essentiellement comme source pour l'extraction de liens de taxonomie entre les mots ou les sens (Bruce Guthrie, 1992), (Rigau et al., 1997). Quelques autres approches, plus rares, ont utilisé ces dictionnaires pour extraire d'autres types de relations sémantiques, comme les relations méronymiques (lien 'partie-tout') (Richardson, 1997).

Mais les systèmes utilisant les dictionnaires électroniques souffrent du manque d'informations car leurs ressources se basent de manière directe ou indirecte sur le contenu des dictionnaires classiques dont la couverture est parfois limitée. Afin de surmonter cette limite, il est indispensable d'utiliser des ressources plus complètes.

D'après les résultats de Yarowsky, nous remarquons qu'un système peut distinguer les différents sens d'un mot dans une langue en les comparant avec ses différents équivalents éventuels dans une autre langue, en utilisant les *corpus parallèles* comme ressource. (Yarowsky, 1995).

Si nous nous basons sur des corpus de grande taille et de large couverture, plutôt que sur des dictionnaires, on pourrait éviter les problèmes mentionnés précédemment quant aux travaux de (Bruce Guthrie, 1992), (Rigau et al., 1997) et (Richardson, 1997).

L'approche qui consiste à utiliser des corpus monolingues afin d'en extraire des relations entre mots de la même langue est basée sur l'hypothèse que les occurrences d'un même mot au sein de différents contextes correspondent approximativement aux différents sens de ce mot (Weaver 1949/1955, Gale, Church et Yarowsky 1992, Resnik 1997). Par exemple, dans une étude de Yarowsky, les deux sens du mot (plant) se distingue selon qu'il s'agit de (biologie ou usine) alors certains mots pouvant exister dans différentes contextes comme (growth, flower, fruit, car, union, nuclear, job) peuvent être utilisés comme indices pour une tâche de désambiguïsation du sens de ce mot. Mais cette approche distingue les différents sens manuellement, de plus elle demande d'avoir des connaissances complètes sur ces mots et les relations entre eux.

En complément, une approche multilingue avec des *corpus parallèles* permettrait de construire de nouvelles entrées à partir des lexèmes monosémiques ou même polysémiques. Cette approche permet une meilleure automatisation car on peut distinguer les différents sens d'un mot dans une langue en les comparant avec les différents mots éventuels dans une autre langue.

En fait, nous croyons que les corpus parallèles peuvent donner des renseignements utiles tant sur le plan de la synonymie (deux lexèmes possédant les mêmes équivalents dans une langue cible étant susceptibles de partager un même sens) que sur celui de la polysémie (un lexème possédant des équivalents différents étant susceptible d'avoir différentes acception).

Une relation d'équivalence valide entre des mots de langues différentes peut maximiser la compatibilité entre les wordnets de différentes langues en fixant les relations d'ILI-RECORDS déjà existés ou en rajoutant des nouveaux ILI-RECORDS manquants.

Ainsi, afin de gérer le recouvrement lexical et la polysémie qui varient d'une langue à l'autre, nous nous inspirons du principe formulé par (Ploux et Ji, 2003), qui ont montré que le mot n'est pas une entrée adaptée pour l'organisation des sens, notamment pour l'établissement des relations d'équivalence traductionnelles, mais qu'il faut plutôt s'appuyer sur *des cliques* de mots, c'est-à-dire des ensembles de lexèmes qui partagent tous, pris deux à deux, un certain contenu sémantique.

En associant des cliques avec des définitions, Ploux et Ji affirment constituer un niveau de granularité de sens plus fin que les mots, et plus fin même que les acceptions des dictionnaires classiques, car les cliques qu'ils obtiennent sont généralement plus nombreuses que les acceptions. Deux cliques très proches peuvent ne varier que par une ou deux unités lexicales, et permettre de distinguer des significations qui se confondent habituellement.

Par ailleurs, si l'on arrive à constituer des cliques multilingues, celles-ci permettent d'identifier à la fois de possibles équivalents multilingues ainsi que des synonymes potentiels dans la langue cible. Ces cliques pourraient constituer la pierre angulaire de la ressource sémantique que nous voudrions construire pour l'arabe.

Dans le chapitre suivant, nous détaillons plus précisément le mode de construction et l'utilité des cliques, en fonction des objectifs de notre travail.

Chapitre II : Cliques multilingues et organisation des sens

II.1 Introduction

Nous avons vu dans le premier chapitre que les synsets d'un réseau sémantique de type wordnet représentent l'intersection sémantique d'un ensemble d'unités, et constituent l'identification implicite d'une acception (sens), en organisant les unités selon deux propriétés (synonymie et polysémie). Par les expérimentations qui suivront dans ce chapitre, nous voudrions tenter d'organiser les lexèmes selon ces deux propriétés en produisant des cliques sur une base multilingue, c'est-à-dire à partir des ensembles d'unités liées par des relations d'équivalences. Les synsets ainsi produits seraient apparentés aux synsets des réseaux sémantiques multilingues tels qu'EWN. Par une expérimentation préliminaire s'appuyant sur des ressources dictionnairiques, nous étudierons la faisabilité de la méthode, avant même de l'appliquer sur des corpus, en caractérisant de façon précise le concept de clique multilingue pour des équivalences a priori valides (celles des dictionnaires).

Mais avant cette première expérience, examinons d'un point de vue formel ce que sont les cliques en tant qu'objets mathématiques.

II.2 Définition générale des cliques

La définition formelle d'une clique fut introduite par (Festinger, 1949*)* ainsi que (Luce et Perry, 1949), pour désigner un sous ensemble maximal d'au moins trois agents où chaque agent est en relation binaire (adjacence) *sociale* et symétrique. Ce type de relation relie les agents deux par deux. Dans ce contexte, le terme maximal veut dire que chaque agent est en relation avec tous les autres agents de la clique et tel qu'il n'existe pas d'autre agent en relation avec tous les autres agents de la clique (chaque agent est donc en relation avec tous les autres agents de l'ensemble). Cela signifie qu'on ne peut pas y ajouter un autre agent avec les relations qui lui appartiennent, sans perdre la propriété précédente (le graphe est dit *complet*).

Dans les termes de la théorie actuelle des graphes, une clique est un sous graphe *maximal*, *complet* et *connexe*. Dans un graphe complet chaque sommet (on les nommera désormais *nœud*) est relié à tous les autres. On note G_n un graphe complet à *n* nœuds.

Le terme *connexe* signifie que pour chaque paire de nœuds (a,b), il existe *une chaîne* reliant chacun des deux nœuds a et b (chaîne éventuellement vide si a et b sont reliés). Une *chaîne* est une suite de nœuds adjacents, c'est-à-dire une suite de nœuds reliés par des arêtes. On peut passer de l'un à l'autre directement sans passer par d'autres nœuds. La longueur de la chaîne est le nombre d'arêtes visitées pour passer du premier nœud de la chaîne au dernier.

Le concept de clique s'inscrit dans la théorie des *graphes non-orientés*. Il convient d'examiner les caractéristiques générales de ce type de graphe avant de montrer les caractéristiques particulières des cliques-mêmes:

Un graphe non orienté est un graphe G(N, A) où :
N est un ensemble fini d'éléments appelés *nœuds.* $N = \{a_1, a_2, ..., a_n\}$ $(|N| = n)$
A est un ensemble fini de paires de nœuds appelés *arêtes.* $A = \{a_1, a_2, ..., a_m\}$ $(|A| = m)$

Une arête *a* de l'ensemble *A* est définie par une paire *non-ordonnée* de nœuds, appelés les extrémités de *a*. Si l'arête *a* relie les nœuds b et c, on dira que ces nœuds sont adjacents, ou incidents avec *a*, ou encore que l'arête *a* est incidente avec les nœuds *b* et *c*. Le nombre de nœuds (n) de ce graphe constitue ce qu'on appelle l'*ordre du graphe*. On appelle *voisinage d'un nœud*, l'ensemble des nœuds adjacents du nœud ou bien la liste des nœuds que l'on peut directement accéder depuis le nœud courant.

On appelle un graphe partiel de G = (N, A) un graphe de la forme G' = (N,A') avec A'⊂ A. En d'autre terme, c'est un graphe qui a les mêmes nœuds que G mais dont on a enlevé certaines arêtes. Mais si on a enlevé des nœuds et les arêtes qui faisaient référence aux nœuds, alors on obtient un *sous-graphe d'un graphe*.

Un sous-graphe de G est donc un graphe de la forme G'= (N',$A_{N'}$) où N' ⊂S N et $A_{N'}$ = {a = (x, y) ∈ A : x, y ∈ N'}.

La Tableau 7 illustre ces notions.

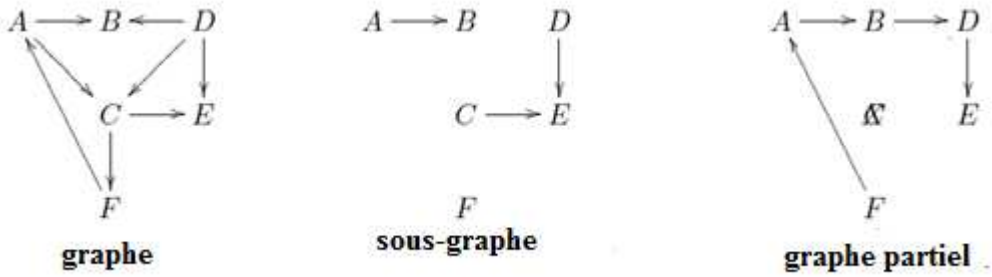

Figure 10 : Graphes, sous-graphes et graphes partiels

Si deux cliques ont au moins une unité en commun, on dit qu'elles se *chevauchent*. Les nœuds qui font partie d'une clique qui en chevauche une autre forment un groupe à part avec les nœuds de l'autre clique, et les nœuds qui font partie d'une clique disjointe forment un groupe bien distinct. Ainsi, on peut définir des groupes distincts partitionnant l'ensemble des nœuds qui font partie d'au moins une clique.

Le chevauchement est réflexif et symétrique, mais pas nécessairement transitif. En effet, il est possible qu'une clique c_i en chevauche une autre c_k et que c_k en chevauche une troisième c_j tandis que c_i et c_j ne se chevauchent pas.

Prenons, à titre d'illustration, le graphe de la Figure 11 dont les lettres A, B,…, C représentent des nœuds, et dont les arêtes indiquent une relation quelconque.

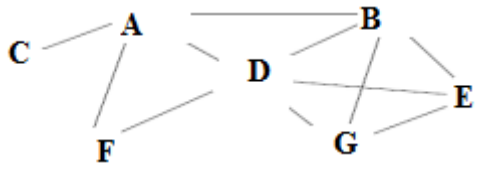

Figure 11. Exemple de cliques avec chevauchement

Les ensembles {A, B, D}, {B, D, E, G}, {A,F,D} sont des cliques qui se chevauchent. L'ensemble {A, B, D, F} n'est pas une clique, puisque il n'existe pas une relation entre B et F.

Après avoir introduit ces quelques termes et concepts concernant les cliques, nous pouvons esquisser un panorama des travaux s'appuyant sur des cliques pour l'organisation du lexique et la représentation du sens.

II.3 L'utilisation des cliques dans le domaine des réseaux sémantiques

Certains travaux portent précisément sur le traitement des unités polysémiques. En effet, la polysémie est un phénomène omniprésent dans le langage, et son traitement reste un problème pour de nombreuses applications du TAL. La polysémie, bien que facile à comprendre intuitivement, est difficile à formaliser, et soulève de très importantes questions sémantiques (Kleiber 2000, Soutet 2005).

Dans ce cadre, (Ploux, 1997), (Ploux et Victorri, 1998) ont développé Visusyn, un logiciel permettant de construire automatiquement un espace sémantique multidimensionnel associé à une unité polysémique. Visusyn associe en fait à une unité non plus un vecteur mais un domaine (des aires délimitées constituant une carte dénotant les tendances de sens d'une unité considérée) à deux niveaux: 1- des formes orthographiques et 2- celui du contenu sémantique. L'accès au contenu sémantique se fait à travers des cliques.

Une analyse factorielle des correspondances est appliqué (Benzécri, 1980) sur la matrice des cliques et des unités. Cet analyse permet de calculer les coordonnées des cliques dans cet espace.

Dans ces travaux, les auteurs représentent les sens d'un mot par des cliques de synonymes qui permettent de distinguer ses acceptions avec une granularité très fine, en s'appuyant sur la *similarité sémantique* entre les unités lexicales qui composent ces cliques. La similarité sémantique est mesurée par le recouvrement entre les zones occupées par ses unités dans l'espace ainsi construit, et par la distance qui les sépare dans l'espace multidimensionnel. Ainsi, certaines cliques représentent

les sens les plus typiques des unités, et d'autres des sens intermédiaires, qui illustrent très précisément l'existence d'un continuum entre des sens typiques parfois très éloignés les uns des autres. L'algorithme utilisé repose sur l'analyse et le traitement d'un grand graphe de relations synonymiques. Exemple :

"L'adjectif insensible : deux ensembles de sens s'organisent autour de deux constructions possibles de sentir, dont le sujet peut désigner le siège de la sensation (d'où insensible = « qui ne peut pas éprouver de sensation ») ou la source de la sensation (d'où insensible = « qui ne peut pas causer de sensation »). En fait, ces deux sens sont reliés par une série de cliques intermédiaires :

endormi, engourdi, indolent

engourdi, froid, inerte

frigide, froid, glacé

apathique, indifférent, indolent

flegmatique, froid, impassible, imperturbable, indifférent

dur, froid, inaccessible, indifférent

impénétrable, inaccessible, insaisissable, sourd

imperméable, impénétrable, inabordable, inaccessible, indifférent

imperceptible, indiscernable, insaisissable, invisible

indifférent, insignifiant, neutre

imperceptible, inapparent, invisible

insignifiant, léger, négligeable

imperceptible, insignifiant, léger " (Ploux et Victorri, 1998 : P.9)

Les auteurs font l'hypothèse que l'appartenance d'une unité à des cliques voisines permet de caractériser les différents sens de cette unité, et de les organiser dans son espace sémantique. En

outre, chaque unité polysémique associée à l'espace sémantique d'une unité vient apporter une sorte de désambigüisation pour cette unité.

L'idée sous-jacente à la construction des espaces sémantiques est qu'un synonyme seul n'est généralement pas suffisant pour caractériser précisément une acception d'une unité, tandis qu'une constellation de synonymes formant un ensemble homogène, telle qu'une clique, peut permettre l'obtention d'une nuance de sens dans cet espace.

Voici quelques exemples de cliques obtenues par (Ploux, 1997), qui illustrent plusieurs valeurs sémantiques pour le mot *insensible.*

Valeurs 'morales' :

(cruel, dur, impitoyable, implacable, inexorable, inflexible, inhumain, insensible)
(cruel, dur, impitoyable, implacable, inexorable, inflexible, insensible, sévère)

Valeurs 'physiques' :

(*endormi, engourdi, inerte, insensibl)e*
(engourdi, froid, inerte, insensible)

Valeurs 'perceptives':

(*imperceptible, inapparent, insensible, invisible)*
(*imperceptible, indiscernable, insaisissable, insensible, invisible)*

On note dans les exemples précédents qu'une unité connexe peut appartenir à différentes cliques : cette caractéristique est due à la non-transitivité de la relation de synonymie. On pourra par exemple retrouver le mot *type* dans deux cliques très différentes, au voisinage de *amant* ou *bonhomme* pour l'une, dans un sens lié au couple, ou avec *exemple* ou *archétype* pour l'autre, dans une acception dénotant la catégorie (Jacquemin et al., 2008).

Plus formellement, si on a trois éléments *x*, *y* et *z* de *E* tels que *x* et *y* sont en relation, ainsi que *y* et *z*, alors *x* et *z* ne sont pas forcement en relation. On note :

$$\forall\, x, y, z \in E, (xRy \wedge yRz) \Rightarrow \neg\, (xRz)$$

Dans un contexte multilingue, l'hypothèse de (Ploux, 1997), a été appliquée sur des lexiques bilingues (français – anglais) (Ploux et Victorri, 1998) afin de projeter les deux systèmes linguistiques sur un espace sémantique commun.

Pour ce faire, les auteurs utilisent trois bases lexicales: 1-Une base de synonymes de la langue source, 2-une base de traduction entre langue source et langue cible, et 3- une base de synonymes de la langue cible.

D'abord le système calcule l'espace sémantique d'une unité à partir des unités existantes dans la base de traduction, Ensuite, l'ensemble des cliques de synonymes de la langue source qui sont liées sémantiquement aux cliques de la langue cible sera calculé. Puis le résultat sera projeté sur une carte qui met en évidence les appariements entre les valeurs sémantiques de la langue source et celles de la langue cible.

En fait, ce modèle fournit une ressource sémantique qui comporte l'information extraite de sept dictionnaires. Mais cette application souffre de son besoin de disposer de plusieurs dictionnaires préexistants, coûteux en temps et en argent, pour construire les graphes.

(Ji et Ploux, 2008) ont remplacé les dictionnaires par le recours à un corpus pour dénoter le sens lexical. Les liens sémantiques dans le corpus peuvent facilement être représentés sous la forme d'un graphe, à partir desquels on peut extraire des cliques. Les auteurs utilisent pour ce faire les associations récurrentes sur le plan syntagmatique, c'est-à-dire la cooccurrence dans le voisinage des unités. Plutôt que ses synonymes, on utilise donc le vocabulaire typiquement associé à un mot, qu'on suppose comme étant lié à ses différents sens.

La relation choisie entre les unités de même clique est donc l'appartenance à un même contexte, le contexte étant défini par une fenêtre prédéterminée. Tous les unités appartenant à cette fenêtre sont interconnectées et appartiennent virtuellement à la même clique.

En fait, pour chaque unité lexicale[10] identifiée, ils construisent l'ensemble des cliques disponibles à projeter sous forme de carte sémantique. Ces ensembles de cliques comportent des contextes syntaxiques. Un dictionnaire bilingue français-anglais comportant environ 40 000 entrées et 250 000

[10] Unité lexicale : Une unité du lexique réunissant une certaine forme lexicale (plan de l'expression) et une certaine signification (plan du contenu).

traductions pour chaque langue est ensuite utilisé dans le but de rechercher la ou les traductions de chaque contexte présent dans une clique de la carte dans la langue-source, et de rechercher leur présence dans des cliques présentes dans une ou plusieurs des cartes de la langue-cible. Alors, le but d'une traduction des contextes syntaxiques est de faire correspondre autant que possible les cartes sémantiques. D'ailleurs, les contextes les plus typiques de l'unité considérée dans un sens donné permettent d'en distinguer les différentes tendances de sens.

L'objectif des auteurs étaient la mise en rapport d'espaces sémantiques comparables, propres à chaque corpus, afin de rapprocher non seulement des mots traduits, mais également des sens et des informations contenues dans différents corpus de langues différentes.

Mais le problème posé par cette méthode est que certains contextes présents dans des cliques de la langue source sont massivement traduites dans la langue cible, c.à.d. les traductions de chaque contexte sont présent dans beaucoup plus de cliques dans la langue-cible que dans la langue-source. Tandis que d'autres le sont beaucoup moins, et de ce fait certaines cliques cible ne peuvent être considérées comme la correspondance d'une clique d'origine, il arrive également que les composants d'une clique trouvent leurs traductions dispersées dans différentes cliques-cibles, qui se recouvrent peu. Ce qui prouve que les cliques obtenues manquent de cohésion sémantique.

Par exemple, ils existent en anglais, un grand nombre de termes (ex. *inert, numb, sluggish, chilly…*) qui représentent dans leurs cliques plusieurs valeurs sémantiques (ex. émotionnelles, physiques…etc.) mais les traductions françaises (ex. *dur, sans-coeur, ...*) de chaque contexte sont présent dans beaucoup moins de cliques dans la langue-française que dans la langue-anglaise.

(Jacquet G. et al., 2010) s'intéressent aussi à la construction du sens d'une unité lexicale au sein d'un corpus de textes. Ils caractérisent un sens par un ensemble de synonymes issus d'un grand corpus. Leur modèle est appliqué à l'étude de la polysémie des prédications verbales. L'algorithme proposé s'appuie sur des cliques construites à partir du graphe de synonymie. Les auteurs cherchent ainsi à construire des espaces sémantiques basés sur des cliques correspondant, en première approximation, à des nuances de sens possibles pour un mot considéré. Mais les auteurs ne donnent pas d'informations précises sur la performance de leur modèle.

Dans les travaux précédents, ils définissent le potentiel sémantique d'une unité comme un espace sémantique assimilable à un nuage de points et les sens des différentes unités lexicales présentes dans une clique s'influencent mutuellement, se spécifient tout en construisant un sens global.

De ce fait, une clique peut être considérée d'un certain point de vue comme le point de convergence des mots qui la composent. En effet, il semble apparaître que la clique recèle un sens qui soit commun à tous ses membres. On peut se rendre compte ainsi qu'il est possible de classer certaines cliques de manière à glisser petit à petit d'un sens à un autre.

Globalement, on peut dire que l'utilisation des cliques comme unité minimale de groupements de sommets a fait ses preuves dans l'exploration de petits graphes de synonymie, constitués par un mot vedette et l'ensemble de ses synonymes. Pour notre objectif de construire une ressource sémantique arabe, en tenant compte du manque de disponibilité des dictionnaires arabes et du fait que les dictionnaires électroniques souffrent du manque de relations sémantiques, nous pensons qu'un ensemble de cliques contenant des équivalents multilingues peut être la solution pour proposer plusieurs équivalents à la fois et couvrir la richesse sémantique des unités.

Ainsi, nous chercherons à créer des cliques en nous basant non sur des relations de synonymie ni des relations de cooccurrence dans un corpus monolingue, mais sur des relations de correspondances lexicales (entre des textes alignés), ce qui présente l'intérêt de renseigner à la fois sur la synonymie et la polysémie des unités, et d'apporter une forme de désambiguïsation sémantique pour ces unités lorsqu'elles apparaissent dans un contexte traductionnel. Et de cette manière, par construction, on évitera le problème de la dispersion des traductions des composants d'une clique dans différentes cliques qui se recouvrent peu.

Dans notre travail, ces relations d'équivalence seront extraites par transitivité des équivalents multilingues (nous les nommerons désormais des équivalents traductionnels), deux lexèmes possédant les mêmes équivalents dans une langue cible étant susceptibles de partager un même sens. Nous appliquerons le principe de *triangulation* (i.e. par l'utilisation de trois langues ou plus), qui consiste à renforcer des hypothèses concordantes issues de plusieurs sources d'information différentes : lorsqu'un choix de traduction spécifie le sens d'une unité, le choix d'un équivalent de traduction dans une langue tierce devrait renforcer ou préciser cette spécification. Si cette triangulation est valide, plus de deux langues devraient donner des indices sémantiques encore plus fins. Dans l'idéal, on devrait pouvoir dégager des cliques monosémiques, chaque clique exprimant un sens différent.

Par ailleurs, nous espérons que les unités de nos cliques puissent se rapprocher des synsets d'EWN partageant un lien d'ILI-RECORD. Comme nous l'avons noté précédemment (voir dans chapitre 1 partie 1.3.2) dans EWN un lien d'ILI-RECORD marque le partage d'un contenu sémantique entre

les synsets des plusieurs langues. L'obtention de cliques proches des synsets d'EWN permettrait de rattacher certaines unités arabes à des sous-sens d'EWN, et d'hériter ses relations sémantiques présentes, afin de construire une ressource originale et compatible avec ce réseau.

Nous allons d'abord effectuer une étude préliminaire, et tester la méthode en nous basant uniquement sur des liens d'équivalences traductionnelles extraites de dictionnaires multilingues. En effet, les relations d'équivalences signalées dans les dictionnaires présentent moins de bruit que celles que nous extrairons dans un second temps des corpus : elles représentent des équivalences plus génériques et prototypiques, moins sensibles aux particularités d'un contexte traductionnel donné. En revanche, elles risquent de ne pas faire état de certaines nuances sémantiques jugées peu fréquentes. En d'autres termes, elles constituent des données a priori moins bruitées mais plus silencieuses que celles issues d'un corpus, et représentent un bon point de départ pour tester notre méthode.

II.4 Cliques et données dictionnairiques

II.4.1 Une étude préliminaire

Nous avons choisi le dictionnaire Larousse pour effectuer notre étude préliminaire et tester la méthode d'extraction automatique des cliques en nous basant uniquement sur des données dictionnairiques. La méthode appliquée consiste à télécharger des entrées du Larousse pour les couples de cinq langues : français, anglais, espagnol, italien, allemand (tous les couples sont représentés sauf les couples espagnol-italien et italien-espagnol)[11].

Le téléchargement se fait en cinq étapes :

1. Télécharger d'abord les traductions directes de l'entrée.
2. Télécharger ensuite les traductions de traductions.
3. Pour chaque retro-traduction récoltée en langue source, télécharger à nouveau les traductions, et les traductions de traductions.
4. A partir des unités obtenues en langue source, calculer un ensemble de pseudo-synonymes de l'entrée : on retient toutes les unités qui partagent une traduction commune avec l'entrée pour au moins deux langues différentes. Pour chaque pseudo-synonyme, télécharger à nouveau les traductions, et les traductions de traduction.

Le principe de l'algorithme consiste à extraire un sous-graphe suffisamment dense autour de l'entrée initiale, sachant que ce n'est qu'une portion incomplète du graphe total constitué par ces dictionnaires. Notons qu'en retenant des traductions de traduction, on a tôt fait de trouver des unités qui divergent complètement de l'unité initiale sur le plan sémantique.

Par exemple, on trouve la séquence : *fr-N-disque -> en-N-record -> fr-N-note -> en-N-bill*

Il s'agit donc d'un graphe très ramifié, qu'il paraît inutile - et peu envisageable - de le parcourir en profondeur : par construction, les unités constituant des cliques autour de l'entrée ne peuvent se situer plus loin que 2 arcs (puisqu'elles partagent leurs traductions dans les autres langues). Ce qui

[11] Les données ont été extraites depuis le site http://www.larousse.fr/dictionnaires/français-anglais, grâce au script extractLarousse.pl, **Annexe 8**, page 244

importe, c'est de récupérer de la façon la plus complète possible le voisinage de l'entrée ainsi que celui de ses éventuels synonymes.

A titre d'illustration, prenons l'exemple du nom *économie* :

1. Nous recherchons d'abord tous les équivalents directs de *fr-N-économie*. Par exemple :

   en-N-economy, en-N-economics, de-N-Wirtschaft, de-N-Volkswirtschaft, it-N-economia, it-N-risparmio, etc.

2. Puis tous les équivalents de ces équivalents. Par exemple :

   de-N-Sparsamkeit : en-N-economy, fr-N-économie, fr-N-action d'économiser, it-N-parsimonia, etc.

3. Puis tous les équivalents des retro-traductions en français obtenu en 2 (telles que *fr-N-macroéconomie*, ou *, fr-N-action d'économise*), et les équivalents de ces équivalents.

4. On constitue ensuite un ensemble de pseudo-synonymes (*fr-N-épargne*, *fr-N-gain*). Puis on recherche tous les équivalents, et les équivalents des équivalents de ces pseudo-synonymes.

Ensuite, nous appliquons plusieurs fonctions :

- Extraire les cliques autour de l'entrée initiale.
- Extraire les cliques autour de ses pseudo-synonymes.
- Regrouper les cliques les plus ressemblantes.
- Proposer des associations entre cliques ressemblantes.

Nous cherchons d'abord toutes les cliques susceptibles d'intégrer *fr-N-économie*, et nous examinons si elles sont ambiguës au regard des principaux sens identifiés pour le mot. On obtient les résultats suivants (l'exemple suivant n'est qu'une petite portion des résultats obtenus), avec des cliques sur 2-4 langues (les cliques sur 5 langues ne peuvent être obtenus car le couple it-es est absent du dictionnaire).

Les trois sens principaux pour *économie,* qui se dégagent des équivalents trouvés sont :

1 Sens de 'système économique'.
2 Sens de 'sciences économiques'.

3 Sens de 'épargne', avec deux sous-sens :

a. (3a) 'économie réalisée : gain, épargne, avantage'

b. (3b) 'action d'économiser : esprit d'économie, parcimonie, etc. '

Voici les cliques obtenues rattachées si possible à un des trois sens précédents :

Avec 4 langues

(fr-N-économie fr-N-épargne de-N-Einsparung en-N-saving it-N-risparmio) →3b[12]

(fr-N-économie fr-N-épargne de-N-Einsparung en-N-saving es-N-ahorro) →3b

(fr-N-économie de-N-Sparsamkeit en-N-thrift en-N-economy es-N-economía) →3b

(fr-N-économie de-N-Sparsamkeit de-N-Wirtschaft en-N-economy it-N-economia) →3b?

(fr-N-économie de-N-Wirtschaft de-N-Sparsamkeit de-N-Volkswirtschaft en-N-economy es-N-economía) →Ambiguë

(fr-N-économie de-N-Volkswirtschaft en-N-economy en-N-economics es-N-economía) →2

Avec 3 langues

(fr-N-économie en-N-economy en-N-economics it-N-economia) →2

(fr-N-économie fr-N-gain fr-N-épargne en-N-saving es-N-ahorro) →3a

(fr-N-économie en-N-economy it-N-economia it-N-risparmio) →3b

(fr-N-économie en-N-saving en-N-economy it-N-risparmio)→3b

(fr-N-économie en-N-saving es-N-economía es-N-ahorro)→3b

(fr-N-économie en-N-thrift en-N-economy en-N-economics en-N-saving es-N-economía) → Ambiguë

Avec 2 langues

[12] Nous ordonnons les langues par ordre alphabétique de leur codage (de en es it) sauf pour le français que nous mettons en premier, afin de commencer par l'unité pivot de la recherche de clique.

(fr-N-économie fr-N-restaurant fr-N-café de-N-Wirtschaft) → Ambiguë

(fr-N-économie fr-N-aspect économique fr-N-sciences économiques en-N-economics) →2

(fr-N-économie fr-N-action d'économiser de-N-Sparsamkeit) → 3b

(fr-N-économie fr-N-micro-économie fr-N-gestion des entreprises de-N-Betriebswirtschaft) →1

(fr-N-économie fr-N-esprit d'économie en-N-thrift) →3b

(fr-N-économie fr-N-macroéconomie de-N-Volkswirtschaft) →1

(fr-N-économie fr-N-économies fr-N-épargne fr-N-gain es-N-ahorro) →3a

(fr-N-économie de-N-Wirtschaft de-N-Volkswirtschaft de-N-Sparsamkeit de-N-Betriebswirtschaft de-N-Einsparung) → Ambiguë

On constate que la plupart des cliques sont non ambiguës, au regard des trois sens identifiés. Nous pouvons remarquer aussi une organisation fine du sens, avec des usages plus génériques et d'autres plus spécifiques :

1. Sens général de /économie/

   (fr-N-économie de-N-Wirtschaft en-N-economy it-N-economia)

   (fr-N-économie de-N-Wirtschaft en-N-economy es-N-economía)

2. Sens de /économie d'un pays/

   (fr-N-économie de-N-Volkswirtschaft en-N-economy es-N-economía)

3. Sens de /science économique/

   (fr-N-economie de-N-Volkswirtschaft en-N-economics es-N-economía)

4. Sens de /épargne/, /action d'économiser/

   (fr-N-économie de-N-Sparsamkeit en-N-economy it-N-economia)

(fr-N-économie de-N-Sparsamkeit en-N-economy es-N-economía)

5. Sens de /parcimonie/, /esprit d'économie/

   (fr-N-économie de-N-Sparsamkeit en-N-thrift es-N-economía)

6. Sens de /ce qui est économisé/

   (fr-N-économie de-N-Einsparung en-N-saving it-N-risparmio)

   (fr-N-économie de-N-Einsparung en-N-saving es-N-ahorro)

Pour éviter la dispersion des sens sur plusieurs cliques, il est possible, dans un second temps, d'appliquer un algorithme de clustering en définissant un indice de similarité entre cliques.

L'algorithme de clusterisation, que nous avons appliqué ici, est rattaché à l'algorithme de CAH (Classification Ascendante Hiérarchique) qui repose sur :

- une *mesure de dissimilarité* entre les individus à regrouper (les cliques) : Pour notre travail, plutôt d'utiliser une mesure de dissimilarité on préfére prendre une mesure de similarité, cela revient au même (au lieu de commencer par regrouper les deux cliques les moins dissemblables, on regroupe les deux cliques les plus semblables).

Le coefficient Dice semble le plus adapté pour cette tâche. Pour deux cliques C_1 et C_2, on le calcule ainsi :

$$Dice = \frac{2(|\ C_1 \cap C_2|)}{(|\ C_1|+|C_2|)}$$

Un algorithme de clustering permet de regrouper itérativement toutes les cliques présentant un degré de similarité supérieur à un certain seuil :

```
Le tableau C[0..n] contient les cliques
# Initialiser tableau Cclust[0..n] :
Pour toute clique C[i] {
       Cclust[i]=C[i]
}
Pour toute clique C[i] {
       Pour toute clique C[j], j > i{
            si Dice(C[i],C[j])>= seuil {
            Cclust[i]<- Cclust[i] ∪ Cclust[j]
```

```
                    Cclust[j]<- Cclust[i] U Cclust[j]
                    }
                }
}
Eliminer les doublons du tableau Cclust.
```

- un *indice d'agrégation* qui permet de calculer la distance entre deux classes de cliques (et non entre deux cliques). Nous avons besoin alors d'une règle d'agrégation pour déterminer le moment où deux classes de cliques seront suffisamment similaires pour n'en former qu'une seule.

La règle que nous avons choisie est *la moyenne non pondérée des groupes associés*[13]**.** La distance entre deux classes est calculée comme la distance moyenne de tous les éléments de chaque classe pris deux à deux. Cette méthode est efficace lorsque les éléments forment déjà naturellement des "groupes" bien distincts :

```
# Initialiser tableau Clust [0..n] :
#Seuil→10
Pour toute cluster Cclust[i] {
   Clust[i]=Cclust [i]
   }
Fin pour
Pour toute cluster Cclust[i] {
   Pour toute cluster Ccluster[j], j > i{
   #Calcul de la distance entre deux clusters
          Si (Cclust [i] ∩ Cclust[j]/ Cclust [i] * Cclust
          [j])<=Seuil{
            clust[i]<- Cclust[i] U Cclust[j]
            clust[j]<- Cclust[i] U Cclust[j]
Fin si
Eliminer les doublons du tableau clust.
```

Avec un tel clustering, il se peut que l'on perde la décomposition fine du sens : avec un seuil élevé de 0,75 les cliques correspondant aux sens 1, 2, 4 et 5 seraient toute de même regroupées.

En revanche, avec un seuil de 0,8, l'étape de clusterisation conserverait les cliques ci-dessus, et permettrait de regrouper correctement les cliques suivantes :

(fr-N-économie fr-N-épargne de-N-Einsparung en-N-saving it-N-risparmio)

(fr-N-économie fr-N-épargne de-N-Einsparung en-N-saving es-N-ahorro)

[13] Cf. http://www.statsoft.fr/concepts-statistiques/classifications/classifications.htm.

(fr-N-économie fr-N-épargne fr-N-gain en-N-saving es-N-ahorro)

Il faut donc régler finement le paramétrage de la clusterisation en effectuant des séries de tests afin de n'effectuer que les regroupements nécessaires. Mais il faut noter qu'il y a toujours un grand risque d'obtenir des cliques ambiguës en regroupant des sens distincts, car il n'est pas rare que la spécification du sens soit liée à une seule unité. On perd donc d'un côté ce que l'on gagne de l'autre, et il n'y a pas de réglage parfait dans ce domaine.

Nous pouvons remarquer dans nos résultats l'absence de la clique suivante:

(fr-N-économie de-N-Wirtschaft en-N-economy es-N-economía it-N-economia)

Ceci est dû à l'absence de relations directes entre espagnol et italien. De même, pour les cliques apparaissant dans les sens 1, 4 et 6 : on obtient à chaque fois deux cliques au lieu d'une, du fait de l'absence de traductions directes entre espagnol et italien.

Afin de combler cette lacune, on peut ajouter des relations supplémentaires en appliquant le principe de transitivité :

Si deux formes f1 et f2 sont telles qu'il existe une forme dans une langue tierce f3, telle que f1 ⇔ f3 et f3 ⇔ f2, alors f1 ⇔ f2

Du fait de la polysémie des unités, la transitivité est bien évidemment inapplicable dans la plupart des cas. *es-N-historial* et *it-N-disco* sont tous deux reliés à *en-N-record*, sans être eux-mêmes équivalents. La relation d'équivalence traductionnelle n'est pas une relation d'équivalence au sens mathématique du terme. Cependant, si cette relation peut être inférée à partir de deux langues tierces, ou plus, cette forme de triangulation permet de consolider sérieusement l'hypothèse d'équivalence. Nous avons donc appliqué un algorithme de complétion par transitivité, en ajoutant des liens à deux conditions :

- Si les deux unités partagent une traduction dans 3 autres langues ;

- ou si les deux unités partagent une traduction dans 2 autres langues et obtiennent un score Dice (mesurant l'intersection de leurs ensembles d'équivalents) supérieur à 0,3.

Avec ces paramétrages, entre le français et l'italien, on obtient par transitivité les liens suivants (dans le calcul autour de *fr-N-économie*) :

it-N-vantaggio⇔ fr-N-avantage

it-N-beneficio⇔ fr-N-avantage

it-N-profitto⇔ fr-N-bénéfice

it-N-profitto⇔ fr-N-avantage

Notons que seuls les liens primaires donnent lieu à transitivité, mais pas les liens secondaires eux-mêmes issus de la transitivité. Entre l'italien et l'espagnol, on obtient les correspondances suivantes (toujours à partir des liens récupérés autour de *fr-N-économie*) :

it-N-parsimonia⇔ es-N-economía

it-N-guadagno⇔ es-N-ganancias

it-N-guadagno⇔ es-N-ganancia

it-N-vantaggio⇔ es-N-ventaja

it-N-risparmio⇔ es-N-economía

it-N-risparmio⇔ es-N-ahorro

it-N-economia⇔ es-N-economía

it-N-macroeconomia⇔ es-N-macroeconomía

it-N-profitto⇔ es-N-ganancia

it-N-profitto⇔ es-N-beneficio

it-N-ristorante⇔ es-N-restaurante

it-N-vincita⇔ es-N-ganancia

Tous les liens obtenus sont pertinents : tous ces couples peuvent être des équivalents traductionnels dans un certain contexte. On voit que la méthode permet de compléter les données sources sans aboutir à un surcroît de bruit.

En appliquant maintenant l'algorithme de constitution des cliques pour les données enrichies avec le couple italien, espagnol, on obtient les cliques suivantes pour 5 langues :

1. Sens général de /économie/

   (fr-N-économie de-N-Wirtschaft en-N-economy es-N-economía it-N-economia)

2. Sens de /ce qui est économisé/

   (fr-N-économie de-N-Einsparung en-N-saving es-N-ahorro it-N-risparmio

3. Sens de /action d'économiser/, /épargne/

   (fr-N-économie de-N-Sparsamkeit en-N-economy es-N-economía it-N-economia)

On constate un certain appauvrissement des résultats. En élargissant aux cliques de 4 langues, on obtient :

4. Sens de /Science économique/

   (fr-N-économie de-N-Volkswirtschaft en-N-economics en-N-economy es-N-economía)

   (fr-N-économie en-N-economics en-N-economy es-N-economía it-N-economia)

5. Sens de /ce qui est économisé/ -> cf. sens 3

   (fr-N-économie en-N-saving es-N-ahorro es-N-economía it-N-risparmio)

   (fr-N-économie fr-N-épargne en-N-saving es-N-ahorro it-N-risparmio)

6. Sens de /esprit d'économie/

   (fr-N-économie de-N-Sparsamkeit en-N-thrift en-N-economy es-N-economía)

7. Cliques ambiguës

   (fr-N-économie it-N-economia en-N-economy es-N-economía it-N-risparmio)

   (fr-N-économie en-N-saving en-N-economy es-N-economía it-N-risparmio)

(fr-N-économie de-N-Wirtschaft de-N-Volkswirtschaft de-N-Sparsamkeit en-N-economy es-N-economía)

L'enrichissement des données par transitivité n'a pas produit de résultats concluants, ni en terme de réduction de la dispersion (plusieurs cliques pour un même sens) ni en terme d'ambiguïtés (plusieurs sens pour une même clique).

Dans l'idéal, l'appartenance d'une même unité à des cliques différentes devrait permettre d'identifier des sens différents pour cette unité donnée. Si l'unité est monosémique, elle devrait n'appartenir qu'à une seule clique. Mais la dispersion des cliques, comme si dessus pour le sens 5, contredit cette hypothèse. Par ailleurs si une unité possède différents sens, elle devrait appartenir à plusieurs cliques différentes. Mais les cliques ambiguës, à leur tour, invalident cette hypothèse. Enfin, chaque sens devrait être représenté et devrait appartenir à au moins une clique complète contenant toutes les langues. Quand ce n'est pas le cas, il s'agit d'une lacune du dictionnaire : un sous-sens un peu marginal (p.ex. *microéconomie*) peut n'apparaître que pour certains couples de langues.

II.4.1.1 Cause de l'ambiguïté des cliques

Pour expliquer la présence des cliques ambiguës, on peut examiner l'exemple ci-dessous, extrait autour de l'entrée *fr-N-ordre* :

(fr-N-ordre de-N-Auftrag de-N-Befehl de-N-Orden de-N-Ordnung de-N-Reihenfolge en-N-order es-N-orden it-N-ordine)

On constate qu'un certain nombre de lexèmes allemands, portant des sens différents, sont venus s'agréger dans la même clique. On y trouve notamment les sens de 'commandement', de 'séquence' et de 'propreté'. L'explication du phénomène est simple : les formes en espagnol, français, italien et anglais sont ambiguës, et susceptibles de porter chacun de ses sens. On a donc affaire à une ambiguïté parallèle sur 4 langues. Entre des langues apparentées, ce cas de figure n'est malheureusement pas exceptionnel. La seule manière d'éviter cette configuration consisterait à faire intervenir une 6ème langue ne partageant pas les mêmes ambiguïtés sur les équivalents de *fr-N-ordre*.

II.4.1.2 Cause de la dispersion des cliques

Il est aussi fréquent que deux cliques différentes contiennent les mêmes sens, comme dans l'exemple obtenu suivant:

Clique 1 : (fr-N-ordre de-N-Befehl en-N-command es-N-orden fr-N-commandement it-N-comando)

Clique 2 : (fr-N-ordre de-N-Befehl en-N-command es-N-orden it-N-comando en-N-order es-N-mandato it-N-ordine)

C'est dû à la présence variable de synonymes, pas forcément reliés entre eux pour toutes les langues (p.ex. *commandement n'est pas relié aux mêmes traductions que ordre, même s'il est pris dans le même sens*).

II.4.1.3 Evaluation

A la lumière de ces premiers résultats, nous remarquons qu'il existe des lacunes dans les dictionnaires. Il suffit qu'un certain sens soit omis dans une certaine langue pour qu'on ne le retrouve pas dans les cliques constituées (et c'est le cas la plupart du temps car ces dictionnaires indiquent seulement les sens principaux, et omettent les sous-sens lorsqu'ils correspondent à des unités plus spécifiques comme p.ex. (*économie→micro-économie*).
En outre, il reste des cliques ambiguës, car certaines ambiguïtés sont partagées dans presque toutes les langues. C'est le cas par exemple pour :

(fr-N-économie de-N-Sparsamkeit de-N-Wirtschaft en-N-economy es-N-economía it-N-economia)

Ici, seule l'unité allemande effectue la distinction entre le sens de 'épargne' et celui de 'système économique'.
Il faudrait alors trouver le moyen d'identifier cette ambiguïté pour séparer (*de-N-Sparsamkeit*) et (*de-N-Wirtschaft*) et créer deux cliques non maximales. Nous pourrions par exemple caractériser l'ambiguïté d'une forme par le nombre des correspondants ambigus (p.ex. *économie a 5 correspondants en allemand*). Inversement, on pourrait caractériser le pouvoir désambiguïsateur de deux unités à l'intérieur d'une clique par leur éloignement sémantique. On peut ainsi montrer que dans cluster a/ :

Cluster a : {fr-N-économie de-N-Sparsamkeit de-N-Wirtschaft en-N-economy es-N-economía it-N-economia}

L'unité *fr-N-économie* n'est pas désambigüisée, cela se fait en comparant l'éloignement sémantique de *fr-N-économie* avec *es-N-economía*, *en-N-economy* etc. (en prenant pour les unités allemandes la réunion de (*de-N-Sparsamkeit* et *de-N-Wirtschaft*).

Par contre, dans b/ il y a bien une désambigüisation :

Cluster b : {*fr-N-économie fr-N-épargne de-N-Einsparung en-N-saving es-N-ahorro it-N-risparmio}*

Ainsi, il faudrait assouplir le concept de clique pour construire de vrais synset multilingues : en effet, il apparaît que les cliques caractérisent l'intersection des sens d'une langue à l'autre, mais l'union lorsqu'on considère les unités dans la même langue. Il faudrait donc identifier les combinaisons de traduction dont l'intersection est la plus réduite, en opérant une sélection dans les cliques existantes. Puis, dans un deuxième temps, on identifierait les synonymes correspondant à ces intersections réduites et on complèterait les ensembles ainsi créés en relâchant le concept de clique.

Pour ce faire, il faudrait d'abord identifier les couples les plus discriminants sur le plan sémantique, tels que :

fr-N-économie en-N-thrift

fr-N-économie en-N-economics

fr-N-économie es-N-ahorro

Ensuite, il faudra réorganiser les cliques en fonction de ces couples. Cette réorganisation permettrait également d'éliminer les cliques redondantes, c'est-à-dire portant à peu près les mêmes sens. En effet, dans l'évaluation des distances entre cliques, lors de la clusterisation, mieux vaudrait s'appuyer sur les unités reconnues comme étant discriminantes plutôt que sur l'ensemble des unités, en incluant toutes celles qui comportent les mêmes ambiguïtés (*fr-N-économie en-N-economy es-N-economía).*

II.4.1.4 Bilan

Nous pouvons constater qu'au bout du compte on est loin d'avoir, par ces données dictionnairiques, des cliques maximales ressemblant à des synsets. En revanche, l'extraction des cliques permet d'identifier des sous-sens, et de les organiser partiellement.

Globalement, les dictionnaires et ressources lexicales telles qu'EWN, Larousse, ou Sensagent, sont fortement lacunaires, et ne fournissent pas une référence totalement stable pour notre méthode. La

méthode appliquée au dictionnaire paraît néanmoins intéressante pour enrichir et compléter les dictionnaires, en appliquant la transitivité triangulée : nous avons montré qu'il était ainsi possible de créer avec une assez bonne précision un dictionnaire italien-espagnol, dans les deux sens, qui n'existait pas dans les données initiales.

Par ailleurs, cette méthode permet la récupération de synonymes et de lexèmes de sens voisins. En partant du nom français (économie), nous avons trouvé, distribués dans les cliques finalement obtenues:

> *Sciences économiques, macroéconomie, micro-économie, épargne, gain, économies, esprit d'économie, aspect économique, action d'économiser, gestion des entreprises*

Chacune de ces unités permet d'éclairer une facette sémantique de notre entrée, et toutes pourraient figurer comme synonyme dans certains contexte.

Nous allons maintenant étendre notre étude aux équivalents obtenus sur *corpus parallèle*. Nous pensons que le fait de travailler sur corpus présente plusieurs avantages, dont deux qui nous paraissent spécialement intéressants : 1/ l'accès aux fréquences des traductions (ce qui permet de pondérer les traitements), et 2/ la possibilité de restreindre le corpus à un domaine particulier (par exemple les notices techniques), ce qui permettra peut-être d'éliminer certaines ambiguïtés.

Une application intéressante d'un travail sur corpus consisterait à créer un dictionnaire multilingue où les sens seraient organisés grâce aux cliques extraites du corpus. Celles-ci donneraient, pour une entrée donnée, l'arbre de ses différents sous-sens, avec des exemples issus des corpus alignés. Dans un tel dictionnaire, nous n'aurions pas besoin de glose, car les exemples en contexte seraient suffisants pour repérer le sens. Une telle application pourrait également fournir des synonymes ou des entrées avec des sens voisins ou connexes.

II.5 Corpus parallèles

Nous définissons d'abord les notions de *corpus parallèle* et d'*alignement* d'un point de vue général. Nous décrirons ensuite les différentes étapes et les traitements mis en œuvre dans la constitution de notre corpus.

II.5.1 Etat de l'art

Un corpus parallèle est composé de deux textes ou plus qui sont des traductions les uns des autres, auxquels on associe des informations sur les parties correspondantes des deux textes. Le parallélisme est donc une relation d'équivalence traductionnelle dont on suppose qu'elle peut se décomposer au niveau d'éléments textuels plus petits : section, chapitres, paragraphes, ou phrases. Les informations de correspondance, qui permettent d'apparier les sous-parties équivalentes, sont généralement calculées automatiquement.

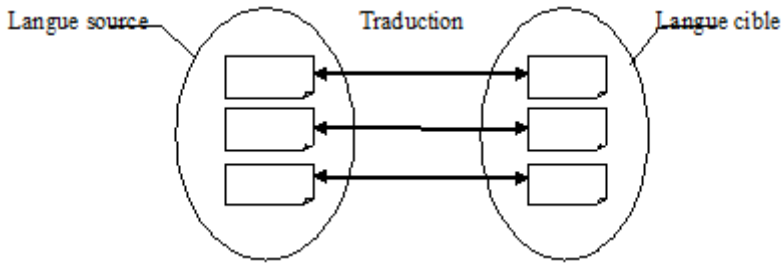

Figure 12 : Schéma d'un corpus parallèle

Outre les traductions littéraires, on trouve de nombreuses sources de corpus parallèles, telles que les sites Web d'entreprise, les documentations techniques traduites en Open Source, les textes législatifs et réglementaires, les rapports parlementaires et tous les documents officiels des pays ou fédérations plurilingues (Canada, Suisse, etc.) ainsi que des organisations trans- ou internationales (ONU, UE, OMS, OCDE, etc.).

Les corpus parallèles sont généralement utilisés dans le domaine du traitement automatique des langues pour développer des systèmes statistiques de traduction automatique.

Ils sont également utilisés par les traducteurs humains, pour alimenter ce qu'on appelle des Mémoires de Traduction (MT). Celles-ci s'intègrent aux outils de Traduction Assistée par

Ordinateur (TAO) pour proposer des ébauches de traductions à partir de fragments de traductions trouvées dans la MT.

Que l'on se situe dans le domaine de la TA ou de la TAO, pour être utile, un corpus parallèle doit receler des exemples de traductions similaires aux phrases constituant le nouveau texte à traduire. La valeur d'un corpus parallèle croît donc avec sa taille : plus un tel corpus est massif, plus on a de chance d'y trouver l'information cherchée. En outre, plus un corpus parallèle contient de langues, plus il permet de traiter un grand nombre de couples, ce qui le rend d'autant plus intéressant.

Dans un corpus parallèle, l'information textuelle seule n'est pas suffisante pour la plupart des applications de TAL. En effet, la mise en correspondance entre les différentes parties équivalentes, à un niveau de granularité suffisamment fin - par exemple les paragraphes ou les phrases - est indispensable pour mettre en œuvre les applications utiles. Cette information de correspondance est appelée *alignement*.

Plusieurs niveaux d'alignement peuvent être définis dans un corpus parallèle. Chaque niveau correspond à un niveau structurel dans le corpus. Par exemple, pour une œuvre littéraire et sa traduction, on pourra définir un alignement entre les chapitres, puis entre les paragraphes à l'intérieur des chapitres, puis entre les phrases de ces paragraphes, et même, dans certains cas, au niveau de certains lexèmes à l'intérieur des phrases.

Les niveaux plus élevés dans la hiérarchie, comme les chapitres, sections et paragraphes, sont généralement en correspondance biunivoque (i.e. avec des correspondances 1-1), et faciles à aligner en s'appuyant sur les marques externes de segmentation. L'alignement phrase à phrase est plus complexe, car les frontières de phrase sont relativement instables avec la traduction. Nombre de phrases sources sont traduites par 0 ou plus de 1 phrase cible. Mais on peut obtenir des résultats très satisfaisants avec des algorithmes d'alignement simples, se basant sur des critères superficiels comme les longueurs de phrases (Gale & Church, 1991) ou la présence de nombres, noms propres ou cognats (Simard, Foster & Isabelle, 1992 ; Mc Enery & Oakes, 1995).

Le niveau qui présente un intérêt particulier pour notre travail est le niveau lexical. L'alignement lexical repose sur l'hypothèse que la relation d'équivalence traductionnelle se décompose aussi au niveau des mots présents dans les paires de phrases alignées du corpus parallèle. Bien qu'on ne puisse établir une relation d'équivalence traductionnelle entre tous les mots d'un couple de phrases la plupart des mots pleins, la traduction mot-à-mot étant en général considérée comme un critère de

mauvaise qualité de la traduction. On pourra trouver une discussion du concept d'alignement lexical dans Kraif (2002), qui préfère utiliser le terme de *correspondance lexicale*. D'après l'auteur, le concept d'alignement (présupposant la décomposition de la relation d'équivalence et le parallélisme) s'applique mal au niveau lexical, du fait des reformulations inhérentes à la traduction du sens. Nous garderons néanmoins le terme d'alignement lexical, consacré par la littérature dans le domaine.

Formellement, l'alignement lexical peut être représenté comme un graphe biparti entre les mots des deux phrases.

Plusieurs approches d'alignement lexical sont proposées : 1-approches purement linguistiques, 2-combinaison des méthodes statistiques de telles approches (Jacquemin, 1991, Bourigault ,1992 ; Smadja, 1993 ; Daille, 1994) qui se basent sur la reconnaissance de patrons et modèles (patterns et templates) en utilisant des grammaires locales ou expressions régulières. Par exemple, (Melamed, 1997) et Kraif (2001a) combinent aussi plusieurs indices, en utilisant des heuristiques adaptées pour réduire l'espace de recherche et minimiser les chances d'erreur.

(Chen, 1993) a utilisé une liste des lexiques bilingues dans le processus d'alignement afin de construire des correspondances entre des unités de textes parallèles.

Il existe aujourd'hui de nombreux logiciels d'alignement automatique disponibles gratuitement comme K-vec++, Giza++, Plug aligner, ou Alinea.

Nous décrirons plus en détail, dans la partie 5.3.1 de ce chapitre, les techniques d'alignement que nous choisissons dans notre travail.

II.5.2 Expérimentation

II.5.2.1 Choix du corpus parallèle

Notre corpus provient des archives des Nations Unies (NU), ses documents sont clairs, cohérents et de qualité supérieure, ils représentent des rapports, lettres et déclarations…etc., de discours officiels traduits en plusieurs langues y compris l'arabe, la traduction est de bonne qualité et certains textes sont volumineux.

Notre corpus parallèle est constitué de 185 textes traitant de sujets différents (ex. commerce international, droit de la femme, santé…etc.) (cf. Le fichier récapitulatif. Corpus3.xls.**Annexe 4**, page 210) dans chacune des quatre langues suivantes : français, anglais, espagnol et arabe classique

(non voyellée). Il contient en moyenne *3 713 665* mots par langue. Nous avons tenté de constituer un corpus de grande dimension, car cela semblait nécessaire pour valider la méthode proposée et ajuster plus finement les algorithmes utilisés. Bien que de taille modeste au regard d'un corpus monolingue, cela constitue une taille honorable pour un corpus parallèle. Constituer un tel corpus seul, avec nos faibles moyens matériels, constituait un véritable défi.

Pour avoir de la quantité, nous avons sélectionné des documents riches en données textuelles, avec le moins possible d'objets complexes tels que graphiques, schémas, tableaux ou listes. Ces éléments peuvent en outre poser problème pour le système d'alignement.

Ces textes sont téléchargeables depuis le système documentaire de la bibliographie officielle de l'ONU : le site UNBISNet (United Nations Bibliographic Information System)[14]. Les documents les plus systématiquement traduits dans les 4 langues (français, anglais, espagnole et arabe) qui nous intéressent sont ceux relatifs à l'Assemblée Générale des Nations Unies. La référence de ces documents est préfixée par la lettre A. Nous nous sommes limitée au téléchargement des documents de la 61ème session, qui s'échelonnent entre 2006 et 2007.

[14] Cf. http://unbisnet.un.org:8080/ipac20/ipac.jsp?profile=bib&menu=search#focus, consulté en mai 2012.

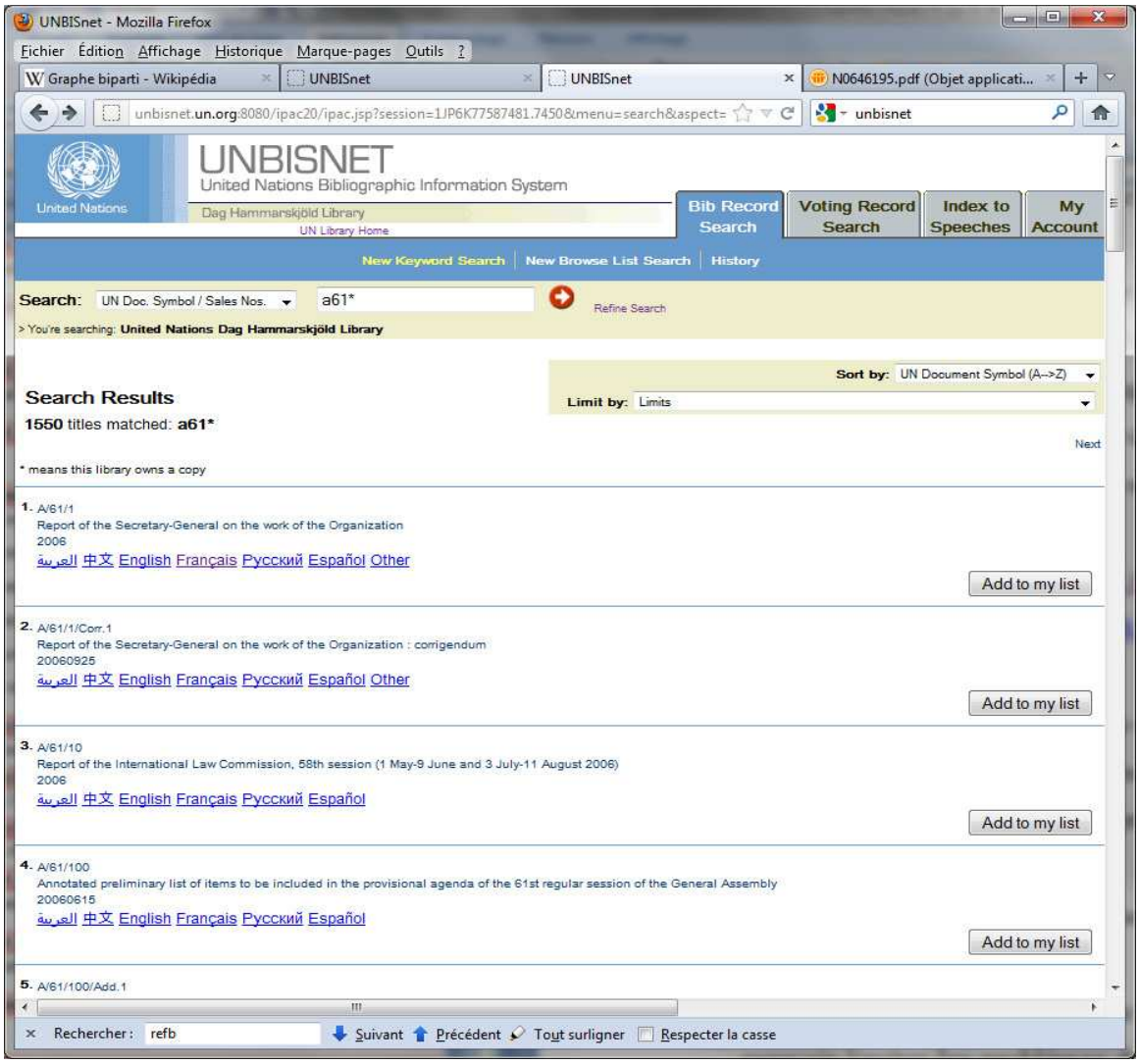

Figure 13 : Résultat d'une recherche des documents de l'Assemblée générale pour la 61ème session[15]

Étant donné que ces textes sont accessibles sous le format WORD, il était indispensable de convertir les fichiers Word en texte brut pour la préparation de la phase d'alignement, ce qui exige un prétraitement important.

II.5.2.2 Prétraitements

Les premiers prétraitements des textes ont été effectués selon les trois étapes suivantes :

15

Cf.http://unbisnet.un.org:8080/ipac20/ipac.jsp?&menu=search&aspect=power&npp=50&ipp=20&spp=20&profile=bibga&index=.UD&term=a61*&sort=3100035&x=6&y=6#focus, consulté en mai 2012.

1. **Conversion & filtrage** : Nous avons d'abord converti les fichiers WORD en HTML afin de pouvoir manipuler le contenu du fichier tout en conservant la structuration du document. En effet, dans le format HTML, nous avons posé des balises de préalignement avant les encadrés, les graphiques, les figures, les tableaux…etc., c'est-à-dire tous les objets non linéaires (du point de vue du texte) susceptibles de nuire à l'alignement. Ces balises, qui serviront de point d'ancrage à l'alignement, permettent en quelque sorte d'isoler les zones non alignables et de limiter leur impact sur l'alignement du reste. Enfin nous avons converti les fichiers HTML en format TXT encodé en UTF8.

   Ensuite, dans le but d'obtenir un alignement de meilleure qualité, nous avons procédé à une phase de filtrage:

   - Suppression de certaines informations numériques apparaissant à la fin du texte (dans l'original, elles apparaissent dans les entêtes et les pieds de page)
   - Suppression des encadrés :
     - Dans certaines paires de fichiers (*ex. anglais-français*) les encadrés ne sont pas formatés de façon identique. En général, les tableaux anglais se présentent sous forme d'images, et lors de la conversion en TXT le texte qu'elles contiennent disparaît avec les images. Dans les fichiers français, les encadrés sont le plus souvent codés sous forme de tableaux, dont le texte est conservé lors de la conversion en TXT.
     - Dans certains encadrés, les phrases n'ont pas de retour de marque de paragraphe, à la conversion en TXT les fins de phrase se collent au début de la phrase suivante. Ainsi, les phrases qui n'ont pas de marque de ponctuation posent problème lors de la segmentation.
   - Eliminer les données qui se présentent sous forme de nomenclature (glossaire, les listes d'abréviations…etc.)
2. **Segmentation phrastique** : Dans le but d'avoir un corpus multilingue parallèle alignable il faut appliquer des règles de segmentation adaptées pour optimiser la contrainte suivante : nombre de phrases de la langue source ≈ nombre de phrases de la langue cible. Nous avons procédé à la segmentation phrastique grâce à un script qui applique un découpage automatique du texte en phrases (cf. Le script seg_phrase.pl.**Annexe 9**, page 279).

Cette procédure produit des segmentations très voisines sur les textes du corpus. Ainsi, elle permettra d'éviter les problèmes générés lors de l'utilisation des outils d'étiquetage différents et de mettre de côté, automatiquement, les corpus avec un mauvais parallélisme, c'est-à-dire pour lesquels le rapport des nombres de phrases s'écarte d'un certain seuil : En fait nous calculons le nombre de couples de phrases, *si* on trouve que le nombre de couples non 1-1 à la suite requis est égale à 3 on commence l'élimination jusqu'à l'on trouve 3 couples 1-1 à la suite requis pour repasser en mode de calcule.

Pour les raisons précédentes, nous avons appliqué le même script de segmentation phrastique sur toutes les langues. La segmentation phrastique repose sur la notion de *séparateur.* En parcourant le texte dans le sens de la lecture, ce sont bien les signes de ponctuation (le point, le point-virgule, les deux points, le point d'interrogation, le point d'exclamation) le majuscule et le retour à la ligne qui permettent de découper le texte et insérer la balise de segmentation. Nous avons mis en œuvre les règles suivantes :

-On insère une balise de segmentation après les signes (.?!) s'ils sont suivi par une espace et une lettre en majuscule

-On insère une balise de segmentation après le signe (.) suivi par un retour de marque de paragraphe

-On insère une balise de segmentation après les signes (:;)

-On ne segmente pas après les abréviations (une liste prédéfinie) suivies par le signe (.)

Le nombre de phrases obtenu par exemple pour la langue française est de 149 836 et le format de sortie est en XML, suivant la norme cesAna, avec un encodage en UTF8. Exemple :

<s id="s32">L'ssemblée générale</s>

<s id="s33">Adopte la Déclaration suivante:</s>

<s id="s34">Déclaration du Millénaire</s>

<s id="s35">I. Valeurs et principes</s>

3. **Etiquetage, lemmatisation et alignement** : Ces derniers fichiers (au format cesAna) ont ensuite subi divers prétraitements : étiquetage, lemmatisation et alignement phrastique.

Avant de détailler ces différentes phases du traitement, examinons quelques outils existants qui peuvent être utiles pour la réalisation de ces tâches.

II.5.2.3 Etiquetage et lemmatisation

II.5.2.3.1 *Etiquetage morphosyntaxiques pour les langues latines*

Dans les applications du TAL, avant de mettre en œuvre des traitements concernant les niveaux syntaxiques et/ou sémantiques, il est en général indispensable de procéder à une phase d'étiquetage morphosyntaxique des textes. L'étiquetage morphosyntaxique (en anglais *Part-of-speech tagging* abrégé en *POS tagging*) est un traitement qui associe à chaque "*mot*" (éventuellement des unités polylexicales) des informations d'ordre morphologique et grammatical. L'étiquetage morphosyntaxique constitue souvent la deuxième étape de l'analyse d'un corpus, après la *tokenisation*, consistant à découper le texte en unités lexicales, signes de ponctuation, etc. Cet étiquetage consiste en l'ajout d'une *étiquette* (ou *tag*) pour chaque token, indiquant sa catégorie grammaticale (partie du discours) ainsi que, le cas échéant, certains traits flexionnels. Voici par exemple les étiquettes éventuellement associées à la phrase : *Il écrit un livre*

Il-> *pronom + personnel + masculin + 3ème personne + singulier + nominatif*

écrit -> verbe + indicatif + présent + 3ème personne + singulier

un -> article + indéfini + masculin + singulier

livre -> nom + commun + masculin + singulier

Généralement, l'étiquetage morphosyntaxique automatique s'effectue en trois étapes (Minh et al 2003)(Rajman et al 2000) :

- *Tokenisation* : La segmentation du texte en unités lexicales est le traitement préalable indispensable. Notons que le problème de la segmentation en mots n'est pas le même pour toutes les langues : dans des langues comme l'anglais ou le français, l'espace constitue un indice relativement fiable pour isoler les unités lexicales (bien qu'en français, les tirets et l'apostrophe soient des séparateurs ambigus). Pour certaines langues comme l'arabe ou le chinois, le découpage au niveau des espaces ne produit pas une tokenisation satisfaisante, et il faut mettre en œuvre des stratégies plus complexes.

- L'*étiquetage a priori* : c'est-à-dire l'association à chaque occurrence de mot de toutes ses étiquettes possibles (cette étape s'appuie en général sur un dictionnaire de formes fléchies).
- La *désambigüisation morphosyntaxique*: qui permet d'attribuer, pour chacune des unités lexicales et en fonction de son contexte, l'étiquette morphosyntaxique la plus probable. De nombreuses formes graphiques sont en effet ambigües : *Ex. "porte" peut être une forme verbale (je porte), nominale (la porte) ou adjectivale (la veine porte)*.s

Plusieurs méthodes ont été développées pour pallier le problème de l'ambigüité lexicale. Celles-ci s'appuient en général sur des informations trouvées dans le contexte immédiat des mots. Ces méthodes d'étiquetage peuvent être classées en trois groupes :

- **Approche linguistique** *:* Elle consiste à codifier des connaissances linguistiques nécessaires en utilisant un ensemble de règles écrites par le linguiste. Exemple de cette approche, le système pionnier TAGGIT (Greene et Rubin, 1971) qui a été utilisé pour créer l'étiquetage initial du corpus Brown, qui a ensuite été révisé manuellement. Une des principales illustrations de cette approche est le développement des *Grammaires de contraintes* (Karlsson et al., 1995) et leur application à l'étiquetage morphosyntaxique (Voutilainen, 1995), qui ont donné parmi les meilleurs étiqueteurs (au-dessus de 99% de précision). L'approche linguistique produit des modèles de langue de haute qualité, mais elle est très coûteuse, plusieurs années de développement étant nécessaires pour pouvoir réaliser un bon modèle linguistique dans une langue donnée.
- **Approches statistiques** *:* ces approches nécessitent beaucoup moins d'effort humain. Les modèles les plus répandus, tels que les Modèles de Markov cachés (HMM) associés à des algorithmes d'optimisation tels que l'algorithme EM (Expectation, Maximisation, Dempster et al. 1977), permettent de déterminer sans intervention humaine, pour un corpus donné, la suite d'étiquettes maximisant globalement la probabilité du corpus. Notons que ces algorithmes n'effectuent que la désambiguïsation, mais nécessitent tout de même de connaître, pour chaque forme, quelles sont ses étiquettes possibles (il faut donc au minimum un dictionnaire des formes fléchies). Les résultats produits par les étiqueteurs statistiques peuvent donner jusqu'à 95%-97% de précision (Manning et Schütze, 1999). Il existe également des méthodes hybrides, qui utilisent à la fois de connaissances linguistiques et des ressources statistiques (Tzoukermann, 1995).
- Le troisième groupe **utilise des algorithmes d'apprentissage** : avec ses méthodes, également de nature probabiliste, on acquiert un modèle de langue à partir d'un corpus

d'apprentissage préalablement étiqueté (en général de façon semi-manuelle). (Dealemans et al, 1992) utilisent une technique d'apprentissage à partir d'exemples (instances) afin de pouvoir déterminer, pour un nouveau contexte, quels sont les exemples les plus semblables du mot à étiqueter. Les approches proposées par (Brill, 1992, et Brill, 1995) peuvent également être considérées comme appartenant à ce groupe : elles sont basées sur un apprentissage supervisé (c'est-à-dire à partir d'un corpus manuellement annoté). L'apprentissage des règles est effectué à travers deux modules successifs : dans le module lexical, on détermine des règles morphologiques pour étiqueter les mots inconnus en fonction de leur forme. Dans le second module, le système construit des règles contextuelles pour désambiguïser l'étiquetage des mots selon leur contexte, et améliore ainsi la précision de l'étiquetage.

De nombreux étiqueteurs morphosyntaxiques ont été développés pour répondre à divers besoins d'étiquetage. Ils atteignent des performances très satisfaisantes pour certaines langues. Les résultats publiés sont d'environ 96-97% de précision.

II.5.2.3.2 *Choix d'un étiqueteur morphosyntaxique pour les langues occidentales (fr-en-es)*

Dans cette recherche, nous avons utilisé l'étiqueteur *Treetagger* (Schmid, 1995), très utilisé dans le domaine du TAL, car performant (en terme de vitesse) et disponible pour de nombreuses langues. Treetagger est en outre facile à intégrer dans une chaîne de traitement, du fait de la simplicité de ses entrées/sorties et de son fonctionnement en ligne de commande. Des modèles de langage pour Treetagger sont disponibles, et gratuitement téléchargeables, pour le français, l'anglais et l'espagnol (entre autres). Le fonctionnement de Treetagger est probabiliste : il s'appuie sur les probabilités conditionnelles d'apparition d'un mot étiqueté en fonction des mots précédents. Les probabilités sont construites à partir des séquences de tri-grammes (constitués de trois étiquettes grammaticales consécutives) observés sur le corpus d'apprentissage.

Présentation générale :

Treetagger est un outil gratuit qui a été développé par Helmut Schmid dans le cadre du *TC Project*, au sein de l'*Institut Für Maschinelle Sprachverarbeitung* de l'Université de Stuttgart.

Treetagger est un système d'étiquetage morphosyntaxique automatique indépendant des langues. De nombreux modèles de langage ont été calculés, notamment pour l'anglais, l'allemand, l'italien, le

néerlandais, l'espagnol, le bulgare, le russe, le chinois, le français et l'ancien français. Il étiquette la *catégorie grammaticale* (information morphosyntaxique) de chaque mot après une étape de tokenisation. Il permet en outre d'effectuer la lemmatisation, en se basant sur un lexique (fourni en sus du corpus d'apprentissage).

Fonctionnement général :

Le choix des étiquettes se fait à partir d'un arbre de décision. Les arbres de décisions permettent de modéliser graphiquement un processus de choix, et sont utiles notamment pour l'aide à la décision. Un arbre de décision binaire est utilisé afin d'estimer les probabilités de transition (probabilité d'obtenir une certaine étiquette morphosyntaxique) en calculant la taille du contexte à utiliser. Ces contextes de taille variable peuvent comprendre des bi-grammes, tri-grammes, …etc. Les contraintes contextuelles peuvent être positives (*ex.* $tag_{-1} = DET$) mais aussi négatives (comme $tag_{-1} \neq$ DET). La probabilité d'une transition est déterminée par le chemin à travers l'arbre qui correspond aux "réponses" de son contexte, jusqu'à ce qu'une feuille soit atteinte.

Treetagger, contrairement aux étiqueteurs reposant sur les modèles de Markov, n'a pas besoin d'un large ensemble d'apprentissage.

Le lexique utilisé pour construire le modèle de langage doit contenir la liste des possibilités d'étiquetage pour chaque mot. Il se divise en deux parties :

- Un lexique de formes pleines (p.ex. pour l'anglais, on utilise les vocables tirés du corpus Penn Treebank). Pour chaque entrée, on a la forme par ligne, chaque occurrence du mot étant suivie par une tabulation et un ensemble de paires tag-lemme, elles-mêmes séparées par des espaces.

Exemple:

Aback RB aback

Abacuses NNS abacus

Abandon VB abandon VBP abandon

abandoned JJ abandoned VBD abandon VBN abandon

- Un lexique de suffixes, organisé dans une structure arborescente. Chaque nœud de l'arbre, à l'exception du nœud racine, correspond à un caractère. Les feuilles de l'arbre, qui correspondent à une certaine terminaison, contiennent des vecteurs de probabilité des étiquettes pour cette terminaison ;

Les jeux d'étiquettes sont codés dans ce lexique, et peuvent être différents pour chaque langue. Par exemple pour la langue française, on a :

ABR : abréviations

ADJ : adjectifs

ADV : adverbes

DET : ART articles

Fichier de sortie de l'étiqueteur :

Une fois le modèle de langage créé (ce qu'on appelle les *paramètres linguistiques*), on peut procéder à l'étiquetage d'un texte quelconque. Le fichier de sortie produit par Treetagger est en format CSV tabulé. Ce fichier est découpé en tokens (ponctuations ou unités lexicales), avec un token par ligne. Il est organisé en 4 colonnes : la première colonne reporte la forme de surface, la seconde son étiquette (catégorie), la troisième le lemme en minuscules, et la quatrième (souvent vide) propose une analyse du mot. Treetagger fait donc la lemmatisation, contrairement à d'autres étiqueteurs comme l'étiqueteur de Brill.

Exemple de sortie :

une DET:ART un

grande ADJ grand

commission NOM commission

Le taux de précision pour l'anglais a été évalué entre 96 et 97 %, avec un corpus d'environ 2 000 000 mots pour l'apprentissage et 100 000 mots tirés d'une autre partie du corpus Penn-Treebank pour le test. En effet Treetagger, en raison de sa nature probabiliste et de l'utilisation du

lexique des suffixes, donne systématiquement une catégorie même si le mot ne fait pas partie de son lexique et qu'il est incapable de lui associer son lemme.[16]

Le fichier de paramètres pour le français a été développé par Achim Stein, qui ne donne pas d'évaluation quant au taux d'étiquetages corrects (il donne par contre un taux d'environ 92% pour le fichier qu'il a développé pour l'ancien français, soit 4% de moins que ce qui est annoncé pour l'anglais).

II.5.2.3.3 *Etiquetage morphosyntaxique pour la langue arabe*

Les recherches utilisant des jeux d'étiquettes dérivés de la théorie grammaticale de l'arabe sont rares et très récentes (Mourad et al, 2008) et il n'y a pas à notre connaissance de système complet et disponible pour l'étiquetage des textes arabes.

Pour certaines langues telles que l'anglais, la plupart des mots ont une ou deux étiquettes possibles, et quand il y a ambiguïté, le système d'étiquetage peut généralement faire un choix en se basant sur les mots / étiquettes du contexte immédiat, juste avant ou juste après le mot traité. Pour l'arabe, du fait de sa grande complexité morphologique, et de son système alphabétique consonantique, la difficulté est incomparable, car les ambiguïtés sont bien plus fréquentes et le jeu d'étiquettes peut se révéler beaucoup plus vaste.

Il se trouve que la langue arabe a un système syntaxique, morphologique et sémantique assez éloigné de celui des langues européennes, et a une tradition grammaticale originale et séculaire, ce qui rend difficile l'adaptation de ses catégories grammaticales aux jeux d'étiquettes utilisés pour les langues indo-européennes.

Pour un texte arabe non voyellé, il existe de nombreuses analyses morphologiques possibles à cause des multiples voyellations alternatives, et ces analyses , qui consistent à déterminer les valeurs d'un grand nombre de fonctionnalités telles que les parties du discours (c.à.d. nom, verbe...etc.), gérondif, genre, nombre, des informations sur le clitiques, correspondent à une véritable "compréhension" du texte puisque l'on ne peut en principe désambigüiser le texte que si on a préalablement interprété le sens.

---

[16] Cf. http://www.crim.fr/travaux_etudiants/2006-2007/alignalco/treetagger.html

Du fait de la morphologie flexionnelle très riche, environ 333 000 combinaisons de traits morphologiques possibles existent pour la langue arabe, tandis qu'en anglais on peut trouver une cinquantaine de combinaisons de traits au maximum. Voici un tour d'horizon rapide de quelques recherches visant à la création d'un analyseur morphosyntaxique de l'arabe (Atwell et al., 2004).

L'analyseur morphologique PROLOG de Shaalan (Shaalan, 1989)

D'abord, rappelons que l'analyse morphologique est un processus qui vise à segmenter la forme de la surface d'un mot en composants dérivationnels et morphèmes flexionnels. Dans une langue telle que l'arabe, qui présente une morphologie flexionnelle et dérivationnelle, les étiquettes morphologiques sont très nombreuses, au contraire des étiquettes morphosyntaxiques qui sont beaucoup moins nombreuses.

Le système de Shaalan est un système fondé sur des règles écrites en Prolog SICStus, demandant des connaissances préalables qui sont difficiles à mettre en œuvre par un linguiste. Ces règles sont antérieures aux standards d'encodage modernes, utilisant un ancien système de translittération.

L'étiqueteur morphologique pour la langue arabe de Khoja (Khoja 2001, Khoja et al, 2003)

Ce système utilise une combinaison de techniques statistiques et de règles linguistiques qui, d'après les auteurs, permettent d'atteindre *le plus haut taux de précision.* Les étiquettes employées sont essentiellement dérivées du jeu d'étiquettes du British National Corpus (BNC), et modifiées pour intégrer quelques notions de la grammaire arabe traditionnelle. Après adaptation, 131 étiquettes différentes sont utilisées. Un corpus de 50 000 mots tiré du journal saoudien *Al-Jaziira* a été utilisé pour entraîner le système. Les résultats fournis atteignent 90% de précision.

Les étiquettes utilisées sont basées sur la classification traditionnelle où les mots se répartissent en cinq classes (voir Figure 14). Nous voyons dans Figure 14 que l'adjectif, le pronom, le nom propre, le nom commun et le numéral sont mis dans une même classe. Mais ces éléments n'ont pas de comportement syntaxique suffisamment homogène, ce qui rend la tâche du choix des étiquettes difficile. Compte tenu du jeu d'étiquettes utilisé, ce logiciel est plutôt un analyseur morphologique qu'un étiqueteur morphosyntaxique.

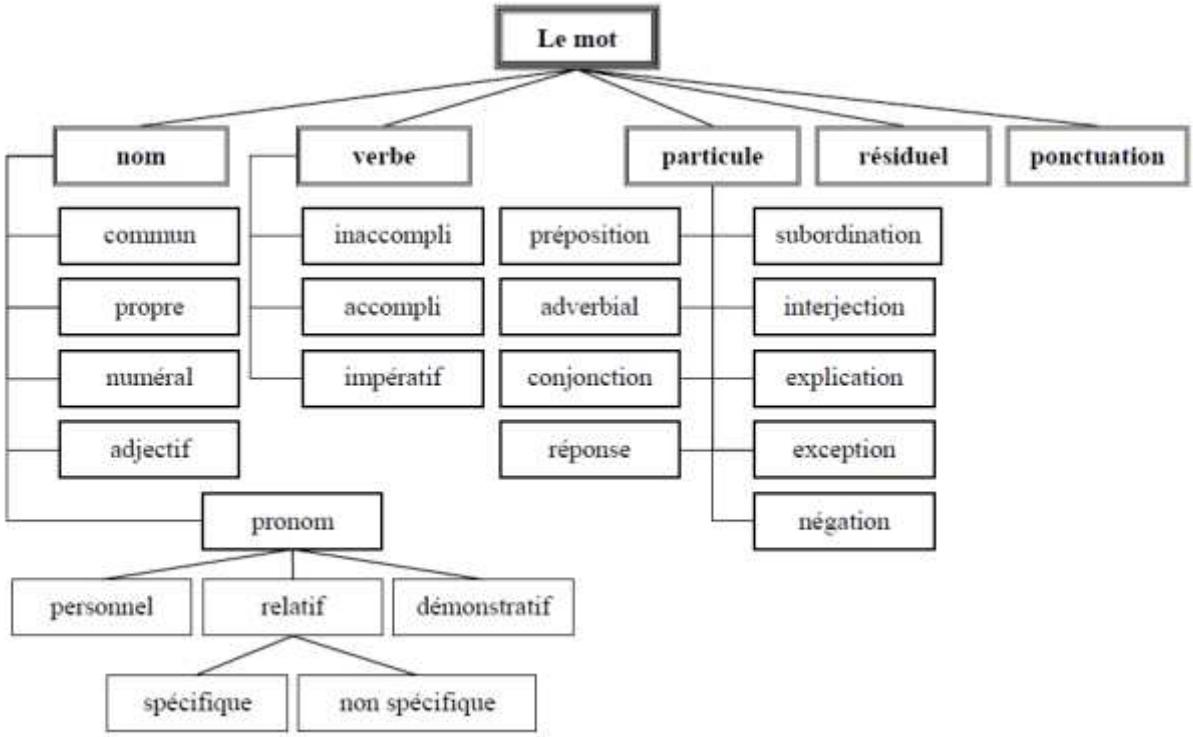

Figure 14 : Classification proposée par Khoja (2001)[17]

Adaptation de la méthode d'Eric Brill sur la langue arabe par Freeman (Freeman 2001, 2002)

Cet étiqueteur est basé sur l'étiqueteur d'Eric Brill, qui peut être entraîné sur un corpus déjà étiqueté. Il utilise 146 étiquettes affectées à des lexèmes. Là encore, il ne s'agit pas d'un jeu d'étiquettes proprement arabe : ces étiquettes sont basées sur celles du Brown corpus pour l'anglais. Il en résulte que ces jeux d'étiquettes comprennent des étiquettes pour des catégories que la grammaire arabe traditionnelle ne reconnaît pas, ou qui correspondent à des catégories différentes en arabe. Par exemple (l'étiquette *(VNCFF désignant un verbe infinitif)* est inexistante en arabe).

L'analyseur morphologique à états finis de Xerox

(Beesley 2001, 2003) a développé un analyseur morphologique arabe utilisant les outils génériques de Xerox, basé sur des modèles de langages implémentés par des automates à états finis. Cet analyseur morphologique est conçu comme un outil pédagogique, et peut également constituer une étape de traitement dans un système plus vaste de traitement automatique de la langue.

L'analyseur morphologique pour la langue arabe de Buckwalter (Buckwalter 2002)

[17] Cf. http://olst.ling.umontreal.ca/pdf/PhDElKassas2005.pdf

L'analyseur morphologique arabe v1.0. est téléchargeable gratuitement à partir du site du Linguistic Data Consortium (LDC[18]).

Le texte en entrée doit être translitéré en caractères ASCII[19] avant tout traitement, et la sortie doit être reconvertie en arabe pour retrouver la graphie originale. De ce fait, le système ne permet pas de mélanger des mots arabes et des mots en alphabet latin dans le même document. Cela peut poser problème, par exemple, quand on a des noms propres ou des emprunts non translittérés.

Lettre	Nom	Buckwalter
ا	ALEF	A
ب	BEH	b
ت	TEH	t
ث	THEH	v
ج	JEEM	j
ح	HAH	H
خ	KHAH	x
د	DAL	d
ذ	THAL	*
ر	REH	r
ز	ZAIN	z
س	SEEN	s
ش	SHEEN	$
ص	SAD	S
ض	DAD	D
ط	TAH	T
ظ	ZAH	Z
ع	AIN	E
غ	GHAIN	g
ف	FEH	f
ق	QAF	q
ك	KAF	k
ل	LAM	l
م	MEEM	m
ن	NOON	n
ه	HEH	h
ة	TEH MARBUTA	p
و	WAW	w
ي	YEH	y
ى	ALEF MAKSURA	Y
ـَ	FATHA	a
ـُ	DAMMA	u
ـِ	KASRA	i
ـً	FATHATAN	F
ـٌ	DAMMATAN	N
ـٍ	KASRATAN	K
ـّ	SHADDA	~
ـْ	SUKUN	o
ء	HAMZA ON LINE	'
أ	HAMZA ON ALEF	>
إ	HAMZA UNDER ALEF	<
ؤ	HAMZA ON WAW	&
ئ	HAMZA ON YEH	}
آ	MADDA ON ALEF	—
ٱ	WASLA ON ALEF	{
ـ	KASHIDA	_

Figure 15 : Translittération Buckwalter

[18] Cf. http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2002L49, consulté en mai 2012.

[19] Les caractères ASCII comptent un peu moins d'une centaine de caractères et incluent les caractères de l'anglais, c'est-à-dire les caractères latins non accentués, ainsi que les chiffres, les principaux signes de ponctuation, les principaux symboles tels que # & @ % $ < >, etc.

Analyseur morphologique Sakher[20]

La société Sakher (entreprise koweitienne implantée en Egypte http://www.sakhr.com/ qui développe les logiciels de traitement automatique de la langue arabe, de la traduction automatique et des outils d'aide à la traduction) a également produit un analyseur morphologique, nommé *Multi-Mode Morphological Processor* (MMMP). Cet analyseur couvre l'arabe classique et moderne, et il détermine la racine possible d'un mot en supprimant tous les affixes et suffixes, et en décrivant la structure morphologique de celui-ci. Malheureusement, il n'existe pas de version d'essai pour cet analyseur.

L'analyseur morphologique Sebawai (Darwish, 2003)

Sebawai est un analyseur morphologique arabe développée par Kareem Darwish. Cet analyseur est utilisé dans une application de recherche d'information. Il a deux modules principaux: Le premier utilise une liste de paires du mot-racine arabe (1) pour établir une liste des préfixes et suffixes, (2) pour construire des modèles de stemmes, et (3) pour calculer la probabilité d'apparition d'un préfixe, un suffixe, ou un modèle. Le seconde traite les mots arabes comme entrée, tente de construire des combinaisons possibles de préfixe-suffixe-radical, et donne comme sortie les racines possibles d'un mot arabe donné. Il permet de trouver la racine avec un taux de réussite de 84%.

Mais pour notre travail, nous cherchons un logiciel qui peut segmenter et étiqueter les textes arabes automatiquement au contraire des logiciels mentionnés ci-dessus, qui sont des analyseurs morphologiques plutôt que des étiqueteurs morphosyntaxiques.

A notre connaissance, ASVM est le seul système disponible au téléchargement qui fait une tokenisation automatique et un étiquetage morphosyntaxique pour des textes arabes.

II.5.2.3.4 *L'étiqueteur ASVM 1.0*

Dans notre travail, nous avons utilisé ASVM 1.0, un étiqueteur de l'arabe librement distribué par l'université de Columbia.

Fonctionnement général d'ASVM:

[20]Cf. http://www.sakhr.com/, consulté en mai 2012.

Le logiciel ASVM de (Diab et al 2004) est une adaptation à l'arabe du système anglais YamCha, basé sur les *Machines à vecteurs de support* (traduction de *Support Vector Machines* ou SVM, parfois nommé *Séparateurs à vastes marges*), un modèle très répandu dédié à l'apprentissage automatique. L'*ASVM Toolkit* propose une série d'outils pour translittérer, tokeniser, lemmatiser et étiqueter des textes. Cette boîte à outil est téléchargeable sur le site de Mona Diab[21] . Une description de son fonctionnement et de ses résultats a été publiée dans les actes d'ACL 2004 (Diab et al., 2004).

Le logiciel est entraîné sur un corpus annoté nommé *Arabic TreeBank* (Maamouri, 2004). Ce corpus se compose de trois parties: • Partie 1: 140K mots de l'Agence France Presse • Partie 2: 144K mots de Al Hayat • Partie 3: 340K mots de Al Nahar. Les fichiers de données sont analysés morphologiquement en utilisant l'analyseur Buckwalter (2002), qui, pour un mot donné, produit toutes les analyses morphologiques possibles. L'analyse comprend des informations sur la stemme et les affixes composant le mot. Ce corpus relie l'étiquette POS arabe avec l'étiquette Penn anglaise, ce qui permet de regrouper plusieurs étiquettes arabes dans une seule étiquette anglaise, par exemple de relier tous les adjectifs à une seule classe. TreeBank contient également des représentations syntaxiques pour les fichiers de fil de presse.

Ce logiciel prend en entrée un ou plusieurs texte(s) translittéré(s) selon les conventions du lexicographe T. Buckwalter, qui a proposé une translittération de l'orthographe de l'arabe standard moderne. Cette translittération est intéressante pour le TAL car elle est réversible et s'appuie sur le jeu de caractères ASCII : elle permet un véritable transcodage orthographique de type biunivoque, avec des correspondances 1-1 entre les caractères arabes et les caractères ASCII, manipulables sous tout type d'ordinateurs et de systèmes d'exploitation.

La boîte à outil se compose de plusieurs scripts :

(utf82buck_unix.pl):

Ce script convertit le fichier original en translittération *Buckwalter* (nous nommerons ainsi, par commodité, cette convention de translittération).

(TOKrun.pl):

[21] Cf. http://alignalco.free.fr/alignalco/ASVM.html

Ce script lance la tokenisation. Du fait des particularités morphologiques de l'arabe (agglutination des clitiques et affixes au radical) cette tokenisation est déjà en quelque sorte une analyse morphologique, puisqu'elle consiste en la segmentation de tous les constituants du *mot* graphique : proclitiques (entre 0 et 2), préfixe (optionnel), radical, suffixe (optionnel), enclitique (optionnel).

(LEMrun.pl):

Ce script remplace la marque de nom féminin singulier "ت"(t) par "ة"(p) lorsque cette marque est attachée à un pronom. En effet, les noms féminins arabes ont souvent des suffixes composés de "ت" (t) et un pronom attaché (par exemple, le mot (*ar-*معالجتها) (trans.mEAjlthA/trad.son traitement) qui exige la substitution du suffixe lors de la génération du lemme. La règle alors est de remplacer le caractère de fin "ت"(t) par le caractère "ة"(t), après avoir enlevé les suffixes (pronoms) attachés (le lemme devient معالجة). Il s'agit d'une normalisation des mots féminins uniquement (standardisation de l'écriture des mots qui varie à cause du style, de l'orthographe peu soignée ou d'une utilisation de diacritiques non-uniformes), mais ce n'est pas vraiment une lemmatisation complète. La lemmatisation ne peut pas être effectuée par de simples expressions régulières effectuant des opérations de recherche/remplacement, mais nécessite un dictionnaire. Le problème vient notamment de la formation du pluriel, qui affecte la structure interne de la plupart des noms et des adjectifs arabes, et ceci de façon difficilement prédictible.

Voici les autres traitements appliqués par ce script :

-Les mots clitiques qui s'écrivent attachés à leur hôte - comme les conjonctions de coordination ف (FA) et و (WAW), la préposition ب (BA), etc. - sont étiquetés séparément, ce qui simplifie l'extraction de patrons.

-Le proclitique article ال AL n'est pas séparé du nom, car il n'est pas tokenisé séparément dans le corpus annoté Arabic Treebank.

(POSrun.pl):

C'est la dernière étape de la chaîne de traitement. L'étiqueteur utilise la méthode basée sur les SVM d'étiquetage, afin de trouver l'étiquette correcte pour un token parmi un jeu de 24 étiquettes possibles.

Ces étiquettes sont celles utilisées dans *Penn Arabic Treebank*, corpus sur lequel l'étiqueteur a été entraîné. Notons que *Penn Arabic Treebank* est composé de textes extraits de dépêches de *l'Agence France Presse*. Le corpus, de taille assez modeste, se compose de 119 phrases pour le développement, 400 phrases pour le test et 4000 phrases pour l'apprentissage.[22]

L'algorithme d'apprentissage supervisé SVM présente l'avantage d'être robuste dans son traitement d'un grand nombre de comportements (caractéristiques), avec une bonne performance de généralisation.

Les étiquettes morphosyntaxiques (tagset) arabes sont les mêmes qu'en français : Nom-Adj, ce qui correspond, dans la sortie de l'étiqueteur ASVM, aux balises NN-JJ. NN pour nom, JJ pour adjectif.

Fichier de sortie

Dans le fichier de sortie du format (POS), on retrouve une phrase par ligne, chaque mot étant suivi d'un slash et de sa catégorie, p. ex. : (*w/CC lm/RP yHtsb/VBP Al/DT Hkm/NN (arabe : ولم يحتسب الحكم)*).

D'après les auteurs (Diab, Hacioglu & Jurafsky, 2004), les performances de SVM-TOK (le tokeniseur TOKrun.pl) atteignent un score 99,12% de F-mesure, et celles de SVM-POS (l'étiqueteur) atteignent une précision de 95,49%. L'évaluation s'est fait selon le corpus Arabic TreeBank.

D'un point de vue qualitatif, nous notons quelques erreurs typiques du tokeniseur et étiqueteur, ce que pourrait poser problème dans la suite de notre travail :

- Certains préfixes et suffixes ajoutés à l'unité lexicale ne sont pas décelés :

Par exemple *le duel (ان)(trans. An) dans (ar-معلمان) (trans.mElmAn/trad.deux professeurs)*

Ainsi l'article d'interrogation أ (>, est-ce que) qui se colle à un verbe n'est pas tokenisé séparément.

[22]Le corpus de développement permet de choisir les méta-paramètres de modèles et notamment le paramètre de régularisation des modèles, le corpus de test sert à vérifier la qualité de l'apprentissage à partir du corpus d'apprentissage, et le corpus d'apprentissage sert à retirer un modèle ou un classement à partir d'un nombre suffisant d'informations.

- Le suffixe (ي) qui s'adjoint à la base nominale (*ex. volcan*) pour créer un adjectif (volcanique) est étiqueté de manière erronée comme un pronom (trad. mon volcan).

II.5.2.4 Reformatage de sortie étiquetée

Après l'application de l'étiqueteur Treetagger sur notre corpus (français, anglais et espagnol), la sortie des textes étiquetés a été reformatée pour les trois langues (cf. Le script ttg2ces.pl.**Annexe 10**, page 282), afin d'obtenir une sortie au format XML étiquetée pour chaque langue, utilisable dans le système d'alignement *Alinea*. De la sorte, nous avons pu garder les balises de segmentation phrastique[23] qui sont utiles lors de la phase d'alignement.

Etant donné que, pour la langue arabe, la segmentation phrastique a été appliquée sur les fichiers arabes tokenisés et translittérés Buckwalter, et que l'ASVM ne distingue pas les balises de segmentation phrastique (au contraire de Treetagger), nous étions obligé d'adapter les fichiers Buckwalter segmentés et les fichiers POS (la sortie d'ASVM) en utilisant un script Perl (cf. Le script mergeTags.pl. **Annexe 11**, page 284) afin d'avoir des fichiers (cesAna) segmentés et étiquetés à la fois.

Ensuite, pour une normalisation d'orthographe, nous avons appliqué les règles suivantes sur certains mots qui se sont différenciés lors du passage entre les fichiers:

p à la fin d'un mot → t
>w→ w (ex. Albrwt>wkwl→ Albrwt>wkwl)
y→a
t à la fin d'un mot→ par p
A au début d'un mot→ " " (ex.Altfawp→ltfawt)

En ce que nous concerne, nous nous intéressons plus à l'extraction des correspondances entre les lemmes afin de regrouper les différentes formes d'une même unité. Mais la lemmatisation d'ASVM n'est pas complète, par exemple ;

-Les verbes : le verbe arabe prend différentes formes suivant plusieurs facteurs (le temps, le nombre des sujets, le genre des sujets, la personne et le mode) mais la lemmatisation d'ASVM ne tient pas

[23] La segmentation phrastique a été effectué dans les premiers prétraitements du corpus, voir partie II.5.2.2

compte de ces différences et il considère la forme agglutinée de verbe (avec les suffixes et préfixes) comme un lemme.

-Les noms et les adjectifs : La *déclinaison* (la flexion du nom ou d'adjectif) dépend du genre (masculins ou féminins) et du nombre (pluriels ou singulier). L'étiqueteur d'ASVM, dans les cas du pluriel irrégulier, n'applique pas des règles de lemmatisation puisque le nom ou l'adjectif arabe suit une diversité de règles complexes et dépend du mot-même; *" Le phénomène du pluriel irrégulier dans l'arabe pose un défi à la morphologie, non seulement à cause de sa nature non concaténative, mais aussi parce que son analyse dépend fortement de la structure comme pour les verbes irréguliers* " (Ben Youssef, 2008).

II.5.3 Format XML et composition globale du corpus étiqueté

Pour les 4 langues, chaque token porte donc, sous forme d'attributs XML, les informations suivantes: identifiant, catégorie et lemme.

Exemple de sortie XML :

```
<s id="s2">
        <tok id="t4" ctag="NP" base="Resolution">Resolution</tok>
        <tok id="t5" ctag="V" base="adopt">adopted</tok>
        <tok id="t6" ctag="PRP" base="by">by</tok>
        <tok id="t7" ctag="DET" base="the">the</tok>
        <tok id="t8" ctag="NP" base="General">General</tok>
        <tok id="t9" ctag="NP" base="Assembly">Assembly</tok>
</s>
```

Pour la langue arabe où la lemmatisation n'est pas complète, la forme et le lemme sont souvent les mêmes.

Les tokens arabes dans la sortie XML étiquetée sont affichés également en Buckwalter. Nous avons adapté quelques règles de transcodage pour que la sortie arabe soit compatible avec la sortie XML de notre script. Nous avons suivi les recommandations de Buckwalter lui-même. Nous avons remplacé :

< par I (pour hamza-sous-alif)

> par O (pour hamza-sur-alif)

Le tableau ci-dessous donne la structure du corpus en sortie :

Langue	Français	Anglais	Espagnol	Arabe
Nombre de phrases	149 836	156 061	153 903	131 711
Nombre de tokens	4 411 728	3 713 665	4 318 070	4 120 719

Tableau 4 : Structure du corpus en sortie

Harmonisation des étiquettes :

Avant de passer à la phase d'alignement, nous avons procédé à l'*harmonisation* des étiquettes des quatre langues pour n'utiliser qu'un seul jeu d'étiquettes (cf. Le script harm_ctag.pl. **Annexe 12**, page 287) avec la table de correspondance des étiquettes). En ce que concerne la différence des étiquettes de Treetagger, cela n'est pas du à la prise en compte des phénomènes qui existent dans une langue et pas dans l'autre, mais le plus souvent à des choix différents pour noter la même chose. Nous supposons que cette harmonisation peut augmenter le taux d'appariement des unités, lors de la phase d'alignement lexical.

Français	Arabe	Anglais	Espagnol
NUM	CD	CD	CARD

Etiquette commune : *CD*

Tableau 5 : Harmonisation des étiquettes

Nous avons appliqués des règles de remplacement de certaines étiquettes dans chaque langue (cf. Le script harm_ctag.pl. **Annexe 12**, page 287): 46 règles pour l'arabe, 33 règles pour le français, 46 règles pour l'anglais, 75 règles pour l'espagnol.

II.5.3.1 Alignement du corpus

Dans la perspective d'effectuer un alignement automatique de notre corpus au niveau lexical, nous avons procédé en deux étapes :

II.5.3.1.1 *Alignement phrastique*

Le programme d'alignement phrastique choisi est *Alinea*, un aligneur gratuit développé par Olivier Kraif (Kraif, 2007) et disponible au téléchargement[24]. Il effectue des regroupements successifs pour les phrases correspondantes entre la source et le cible. On obtient des regroupements parmi 8 possibilités : 1-1, 1-0, 0-1, 1-2, 2-1, 1-3, 3-1, 2-2. Une phrase source peut avoir une ou plusieurs phrases comme équivalents dans le cible. En outre, on pourrait trouver des phrases qui n'ont aucun équivalent. Alinea se base sur des indices superficiels suivants (Kraif, 2001) :

Transfuges: Des chaînes invariantes comme les anthroponymes, les données numériques, les emprunts, etc.

Cognats : Des couples de mots apparentés présentant des ressemblances graphiques;

Rapports des longueurs des phrases : suivant le modèle proposé par (Gale & Church, 1991).

L'alignement phrastique a été lancé sur chaque paire de documents indépendamment. Cela nous a permis de circonscrire les éventuels problèmes liés à certains documents difficilement alignables. Dans un deuxième temps, nous avons pu tout regrouper dans un unique corpus aligné. Le format des fichiers de sortie est le format cesAlign.

Résultats de l'alignement :

A l'issue de cette phase, nous avons observé différents cas de figure, de l'alignement correct à l'alignement partiel, manquant ou erroné. Voici quelques exemples de ces différents cas de figure :

Alignement correct : les deux groupes de phrases correspondent exactement : voici un exemple d'alignement phrastique correct entre l'anglais et l'espagnol :

[24]Cf. http://w3.u-grenoble3.fr/kraif/index.php?Itemid=43&id=27&option=com_content&task=view, consulté en mai 2012.

[s11] Supervening impossibility of performance;	**[s10]** De la imposibilidad subsiguiente de cumplimiento;

Alignement partiel : Une partie de la phrase source est aligné avec la phrase équivalente. Voici un exemple d'alignement phrastique partiel entre le français et l'arabe:

[s19 s20 s21] On estime normal que deux salariés français perdent leur vie au travail chaque jour, et que huit autres soient sacrifiés par minute au bien - être des entreprises ¤ Mais pas que celles - ci, ni le capital, participent davantage aux retraites des personnels ¤ Comment ne pas comprendre la colère des citoyens[25]	**[s14]** يعتبر من الطبيعي ان يموت في العمل اجيران يوميا ويضحى ب ثمانية آخرين في الدقيقة من اجل رفاهية الشركات لكن لا يطلب منها في المقابل ولا من الرأسمال المساهمة اكثر في تعويضات تقاعد العاملين

Alignement erroné : la phrase équivalente n'est pas du tout alignée avec la phrase source. Voici un exemple d'alignement phrastique erroné entre l'anglais et le français:

[s14 s15] Income and sources of funding	**[s12]** Le budget des projets de l' Office était de 47,1 millions de dollars

Filtrage des paires de textes correctement alignées :

Le but du filtrage est de mettre de côté les paires de textes qui dysfonctionnent. Le filtrage s'applique sur chaque sortie phrastique de chaque couple de langues.

Nous avons adapté les paramètres de filtrage en nous appuyant sur le taux d'appariements 1-1 obtenus : en effet, un mauvais parallélisme des fichiers se traduit en général par des suites chaotiques d'appariements 1-0, 0-1, 2-1, etc. Nous avons donc appliqué la règle suivante :

[25] La phrase française soulignée n'est pas alignée dans la langue arabe.

Si trois appariements de phrases non biunivoques sont trouvés à la suite, l'élimination des couples s'applique jusqu'à ce que l'on trouve trois appariements 1-1 à la suite. Dans ce dernier cas on repasse à la validation des appariements qui suivent, jusqu'à la prochaine rupture de parallélisme.

Enfin, nous avons considéré comme non suffisamment parallèles tous les textes ayant moins de 50% de couples conservés. Ces textes ont donc été écartés de la sélection.

Lorsqu'une paire de texte est éliminée, tous les autres couples impliquant ce même texte sont éliminés, même s'ils ne dysfonctionnent pas. Cela permet au final d'obtenir des textes parallèles en 4 langues.

Après ce filtrage, on obtient 168 textes sélectionnés dans chacune des quatre langues (français, anglais, espagnol et arabe). Le nombre de tokens moyen pour chaque langue est d'environ *3 270 000* tokens.

Ensuite nous avons relancé l'alignement phrastique sur 100 fichiers (choisis aléatoirement) filtrés et harmonisés d'une manière automatique, Le nombre de tokens moyen pour chaque langue est d'environ 1 600 000 tokens, et nous avons calculé une estimation de la précision et du rappel pour cet alignement.

En fait, pour calculer cette estimation, nous avons sélectionné un échantillon du corpus, et corrigé manuellement la sortie de l'alignement, afin de construire un alignement de référence. Nous avons ensuite confronté cette référence à l'alignement obtenu de façon purement automatique, afin d'estimer la précision et le rappel.

Pour le français et l'arabe on obtient les valeurs suivantes, arrondies au 1/100 :

	Précision	**Rappel**
Estimation pour le couple fr-ar	86,6 %	82,2 %

Tableau 6 : Estimation du résultat de l'alignement phrastique

Nous pouvons constater qu'avec les indices utilisés (chaînes identiques et longueurs de phrase), combiné à une phase de filtrage, on obtient des résultats suffisamment bons pour procéder à la suite du traitement.

II.5.3.1.2 *Alignement lexical*

L'alignement lexical consiste à apparier les tokens (unités lexicales) équivalents à l'intérieur des phrases alignées.

Pour une mise en œuvre rapide de l'extraction des correspondances entre les différentes langues, nous avons d'abord utilisé le script *mot@mot* développé par (Boxing Chen & Olivier Kraif, 2003).

Mot@mot est un script Perl permettant d'extraire des correspondances lexicales à partir de plusieurs textes bilingues parallèles préalablement alignés au niveau des phrases. Une interface graphique réalisée en TK permet d'accéder aux différents paramétrages.

Mot@mot procède en deux étapes :

Calcul des paramètres : un jeu de paramètres, intégrant les statistiques d'occurrences et de cooccurrences pour tous les couples de tokens susceptibles d'être appariés, est d'abord construit à partir d'une liste de textes parallèles. Le script prend en entrée une série de textes parallèles, ainsi que les fichiers d'alignement phrastique correspondants.

Dans notre travail, nos textes parallèles sont au format cesAna (obtenus en sortie d'Alinea) et contiennent une segmentation de type phrastique avec un étiquetage comprenant pour chaque token la forme de surface, le lemme et sa catégorie. L'intégralité du corpus aligné au niveau des phrases a été utilisée pour calculer les paramètres.

Alignement : nous nous intéressons plus spécifiquement à cette partie puisque cet alignement nous permettra d'extraire les équivalents traductionnels entre les unités lexicales.

L'extraction des équivalents traductionnels est basée sur une combinaison d'indices : fréquence des occurrences et des cooccurrences au sein des phrases alignées, positions dans les phrases, ressemblance graphique et identité des parties du discours (c'est pourquoi les corpus doivent être étiquetés avec un jeu d'étiquettes harmonisées).

Lors de l'extraction des correspondances lexicales, Mot@mot ignore toutes les unités faisant partie des *stoplists* (ou antidictonnaire) définis pour chaque langue. Ces *stoplists* contiennent les mots outils (préposition, articles, conjonctions, …) les plus fréquents, qui n'apportent pas une information utile lors de l'appariement des unités.

Les *stoplists* française, anglaise et espagnole sont déjà fournies avec Mot@mot. En ce que concerne les stoplists arabes, nous avons créé une liste contenant 142 mots. Cette liste a été créée à partir des sites spécialisés contenant la liste des principaux mots vides (articles, prépositions, auxiliaires, etc.), et encodée en translittération Buckwalter pour qu'elle convienne à notre corpus (cf.Le fichier stopword.ar. **Annexe 1**, page 193).

La sortie de cette étape de l'alignement lexical est constituée de fichiers contenant les correspondances lexicales sous deux formats CesAlign (Pal) et (Wal) qui permettent d'accéder aux couples d'équivalents bilingues triés par ordre décroissant des valeurs d'indice de similarité entre les tokens.

Exemple de fichier fr-ar du format Pal	**Exemple de fichier fr-ar du format Wal**
<linkGrp targType="t"> <link xtargets="t5 ; t2"/> <link xtargets="t4 ; t3"/> <link xtargets="t6 ; t4"/> <link xtargets="t8 ; t5"/>	fr-t5-session:ar-t2-Aldwrp 6083.8220307546: 1/2 fr-t4-Soixantième:ar-t3-Alstwn 8.64899336585518: 2/2 fr-t6-projet:ar-t4-m$rwE 4470.2930307546: 1/8 fr-t8-ordre:ar-t5-jdwl 1832.0960307546: 2/8

Figure 16 : Exemple de fichiers Pal et Wal

Nous avons effectué ces calculs pour les 6 couples fr-en, fr-ar, fr-es, es-ar, es-en, en-ar (notons que l'ordre des langues n'intervient pas dans un couple).

II.5.3.2 Premiers résultats

Ici aussi, avant de passer à l'étape suivante, il nous paraît important de pouvoir donner une estimation des résultats obtenus.

Nous avons tiré d'une manière aléatoire 100 couples alignés de chaque paire de langues, afin d'évaluer manuellement sur un échantillon les résultats de l'alignement lexical obtenu par mot@mot.

Nous avons malheureusement constaté un taux de bruit (cas d'erreurs) élevé pour certains couples de langues (voir Tableau 7):

Couple de langues évaluée	Taux de bruit
français – anglais	31%
français – arabe	41%
espagnol – français	71%
espagnol – arabe	59%
anglais – espagnol	29%
anglais – arabe	23%

Tableau 7: Résultats de l'alignement mot@mot

Voici quelques exemples d'alignement lexical erronés obtenus:

en_United → fr_nations
en_Nations→ fr_unies
en_Secretary-General→es_secretario
fr_incertitude→ar_Alyqyn (trad.certitude)

Nous avons remarqué, comme dans les exemples précédents, que les erreurs d'alignement lexical sont souvent dues à une mauvaise prise en compte des unités polylexicales : *United Nations* est bien correctement alignée avec *nations unies*, mais comme Mot@mot ne prend en compte que des appariements 1-1 entre tokens, et que notre tokenisation ne prend pas en compte les unités polylexicales, des appariements partiels (c'est-à-dire impliquant des fragments d'unités polylexicales) aboutissent à des erreurs.

Un alignement de type *n-n* qui prendrait en compte des unités telles (*en_secretary-General →es-secretario general)* permettrait d'éviter ce type de bruit.

C'est pour cette raison que nous avons testé un autre outil, très utilisé dans le domaine de la traduction automatique, nommé GIZA++ (Och, 2003). L'intérêt de GIZA++, par rapport à *Mot@mot* est qu'il peut donner des alignements de 1 à n ou de n à 1 entre le texte source et le texte cible. En l'appliquant dans les deux sens, p.ex. du français vers l'arabe puis de l'arabe vers le français, on peut dès lors obtenir un alignement *n-n*, susceptible de grouper les unités polylexicales qui doivent rester solidaires lors du passage à la traduction, ce qui devrait permettre de diminuer le bruit concernant l'alignement lexical.

II.5.3.3 Utilisation de GIZA++

GIZA++ est un outil Open Source basé sur les principes de traduction statistique automatique de Brown et al. (1991). C'est une extension du programme GIZA qui comprend un grand nombre de fonctionnalités supplémentaires. Les extensions de GIZA++ ont été conçues et implémentées par Franz Josef Och (2003)[26]. Dans ce modèle de traduction, la phase d'alignement lexical permet d'établir des liens entre chaque mot d'une phrase dans la langue source avec zéro, un ou plusieurs mots de la phrase alignée dans la langue cible. En calculant l'alignement dans un sens, puis dans l'autre, on peut obtenir des liens entre des groupes de mots.

Il existe plusieurs stratégies pour fusionner les alignements obtenus dans un sens puis dans l'autre. Pour établir des alignements lexicaux basés sur les deux alignements GIZA++, un certain nombre d'heuristiques peuvent être appliquées[27] : Intersection ($A_1 \cap A_2$)[28], Grow (uniquement ajouter des points de voisins dans le bloc), Grow-diag (ajouter des liens adjacents en diagonale, sans étape finale), Union ($A_1 \cup A_2$), srctotgt (considère seulement les alignements mot-à-mot dans le sens d'alignement source-cible de GIZA++), tgt2src (considère seulement les alignements mot-à-mot dans le sens d'alignement cible-source de GIZA++). La stratégie que nous avons appliquée, pour notre travail, sera expliquée à la fin de cette partie.

Avant le lancement de GIZA++ sur notre résultat de phrases alignées (fichiers cesAlign), nous avons adapté nos fichiers en appliquant certains prétraitements :

[26] http://www.statmt.org/moses/giza/GIZA++.html
[27] http://www.statmt.org/moses/?n=FactoredTraining.AlignWords
[28] A_1 : Les mots appariès dans le premier sens. A_2 : Les mots appariés dans le deuxième sens.

Le script Txs2raw.pl : Ce script permet de construire des fichiers (txt) et (raw) à partir d'une paire de fichiers txs (name.L1.txs et name.L2.txs) qui sont des fichiers segmentés, étiquetés, lemmatisés et filtrés, et du fichier cesAlign correspondant (name.l1-l2.ces) qui est la sortie de l'alignement phrastique. En sortie de ce script, on trouve un fichier d'alignement (all.L1-L2.txt) contenant l'alignement en texte brut (avec les formes) et deux fichiers (all.L1-L2.L1.raw et all.L1-L2.L2.raw) où chaque token apparaît avec la concaténation de ses attributs.

Exemple de contenu des fichiers (RAW) :

Résolution|résolution|NN|t4 adoptée|adopter|V|t5 par|par|PRP|t6 l'|le|DET|t7 Assemblée|assemblée|NN|t8 générale|général|ADJ|t9

Le script (cleaning.pl) : Ce script permet le filtrage du corpus (les fichiers raw) avant d'appliquer le traitement par Giza++. Il élimine tous les couples de phrases inutilisables pour Giza++ (i.e. tous les couples jugés trop longs, contenant plus de 150 mots, et tous les couples dont le rapport des longueurs est supérieur à 3). En sortie on obtient des fichiers portant l'extension raw.clean.

Exemple de phrases éliminées :

Phrase source en arabe	**Phrase cible en français**
w^CC t$ml^v On$Tp^NN dEm^NN AlmhAm^NN AlOxrY^ADJ fy^PRP Alqsm^NN AlmEny^ADJ b^PRP OfDl^ADJ mmArsAt^NN HfZ^NN AlslAm^NN ,^PUN w^CC ZA}f^NN IdAryp^ADJ ,^PUN w^CC wZA}f^NN ttElq^v b^PRP Altwvyq^NN w^CC AlTbAEp^NN ,^PUN b^PRP AllDAfp^NN IlY^PRP mhAm^NN txTyT^NN w^CC tnZym^NN Eqd^NN AlAjtmAEAt^NN REF^SYM AlHlqAt^NN AldrAsyp^ADJ w^CC gyr^ART hA^PR mn^PRP AlmhAm^NN AlskrtAryp^ADJ REF^SYM w^CC	NULL ({ 7 10 15 16 23 28 37 46 47 }) le^DET ({ 1 2 }) assistance^NN ({ 3 }) au^PRP ({ 4 }) membre^NN ({ }) de^PRP ({ }) le^DET ({ }) section^NN ({ 8 }) du^PRP ({ }) pratique^NN ({ }) optimal^ADJ ({ 9 11 12 13 14 }) charger^V ({ 5 }) de^PRP ({ }) autre^ADJ ({ 6 }) fonction^NN ({ 17 21 }) consister^V ({ 29 }) Ã ^PRP ({ 30 }) effectuer^V ({ }) du^PRP ({ }) tâche^NN ({ 31 }) administratif^ADJ ({ 18 }) ,^PUN ({ 19 }) à ^PRP ({ }) se^PR ({ }) occuper^V ({ }) de^PRP ({ }) le^DET ({ 20 22 }) documentation^NN ({ 24 }) et^CC ({ 25 }) de^PRP ({ }) le^DET ({ }) impression^NN ({ 26

AlIdAryp^ADJ @card@^CD	}) de^PRP ({ }) document^NN ({ }) ,^PUN ({ 27 }) à ^PRP ({ }) planifier^V ({ 32 }) et^CC ({ 33 }) à ^PRP ({ }) organiser^V ({ 34 35 }) du^PRP ({ }) réunion^NN ({ 36 }) et^CC ({ }) du^PRP ({ }) séminaire^NN ({ 38 39 }) et^CC ({ 40 }) à ^PRP ({ }) effectuer^V ({ }) de^PRP ({ }) autre^ADJ ({ 41 }) travail^NN ({ }) de^PRP ({ 42 43 }) secrétariat^NN ({ }) ou^CC ({ }) tâche^NN ({ 44 45 }) administratif^ADJ ({ 48 }) @card@^CD ({ 49 })

Figure 17 : Exemple de phrases alignées éléminées lors du filtrage

Le script (extractlemmeCat.pl) : Ce script n'extrait que les suites lemme^catégorie dans les fichiers (raw.clean) et il ignore la forme et le nombre de token, , afin de pouvoir garder le lemme et la catégorie lors du traitement par GIZA++. La sortie de ce script est en format .lemcat.

Exemple du contenu de fichier (lemcat) :

Resolution^NP adopt^V by^PRP the^DET General^NP Assembly^NP

Le script plain2snt.pl : c'est le dernier script, il extrait des fichiers (.lemcat) les fichiers d'entrée de GIZA++ (vcb et snt). Dans les deux fichiers de vocabulaire (L1.vcb et L2.vcb), on trouve chaque mot type, pour une langue donnée, avec sa fréquence d'occurrence et son identifiant. Le fichier (snt) remplace chaque phrase alignée de deux langues par les identifiants des mots qui la contiennent, les identifiants faisant référence au fichier de vocabulaire (.vcb)

Exemple de fichier (vcb) et (snt):

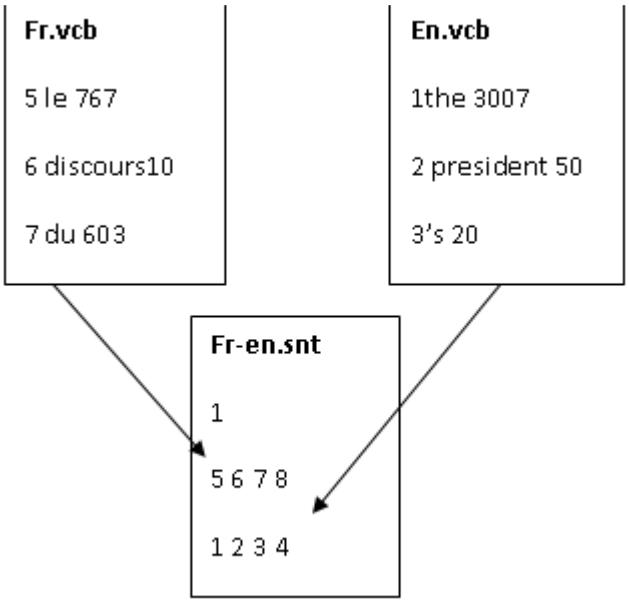

Figure 18 : Exemple de fichier de format VCB et SNT

Voici les statistiques obtenues à l'issu de ces scripts pour chaque couple de langues:

	En-ar	**En-fr**	**En-es**	**Es-fr**	**Es-ar**	**Fr-ar**
paires traitées	80 013	83 641	98 303	85 714	77 992	73 823
Vocabulaire source	43 076 formes	44 011 formes	45 295 formes	42 525 formes	40 415 formes	35 070 formes
Vocabulaire cible	64 231 formes	37 093 formes	43 738 formes	36 532 formes	62 657 formes	62 120 formes
Taille du corpus source après filtrage	2 011 067	2 073 854	2 376 650	2 360 338	2 183 682	2 041 559
Nombre	1 166 137	1 105 149	686 110	1 374 373	1 524 446	1 746 391

d'occurrences éliminées						
Taille du corpus cible après filtrage	2 246 833	2 414 186	2 635 245	2 372 867	2 156 367	2 095 085
Nombre d'occurrences éliminées	1 302 398	1 372 666	823 685	1 415 641	1 377 153	1 448 064

Tableau 8 : Statistiques obtenues à l'issu du script plain2snt.pl pour chaque couple de langues

Nous avons ensuite lancé GIZA++ sur les fichiers vcb et snt. Les fichiers de résultats sont des fichiers du format A3.Final dans le deux sens (source-cible et cible-source). Ces fichiers de résultats contiennent les phrases alignés (source-cible) avec leur score d'alignement.

Exemple de sortie du format (A3.Final) entre (ar-fr):

 # Sentence pair (3) source length 6 target length 5 alignment score: 2.20583e-07

*qrAr^NN Atx*t^v h^PRP AljmEyp^NN AlEAmp^ADJ*

NULL ({ }) résolution^NN ({ 1 }) adopter^V ({ 2 }) par^PRP ({ 3 }) le^DET ({ }) assemblée^NN ({ 4 }) général^ADJ ({ 5 })

La phrase française (cible) est listée token par token, avec des référence aux tokens sources alignés, par exemple le token *résolution* est aligné avec ({ 1 }) i.e le premier token dans la phrase source qui est *qrAr*, et ainsi de suite. Notons que chaque token français (de langue cible) peut être aligné avec plusieurs mots dans la langue source, mais chaque mot arabe (de la langue source) peut être aligné à au plus un mot anglais. Alors l'alignement obtenu est de 1-n.

Afin de fusionner les alignements obtenus dans les deux sens (s->t et t->s), nous avons implémenté un algorithme (cf. Le script gizaOutput.pl.**Annexe 15**, page 307) appliquant trois heuristiques : 1.Union (Toutes les relations sont prises dans les deux sens) 2.Intersection (ne garder que les couples avec des relations dans les deux sens) 3.Compatibilité : (La relation s→t est conservé si t→0

ou t→s) (Idem dans l'autre sens). Ensuite un principe de clôture transitive entre les tokens appariés est calculé dans les deux sens afin de regrouper les alignements (1->n) et avoir une fusion de résultats (n→n). On opère le regroupement à condition d'avoir une distance faible entre les groupes (distance <5) :

```
Alignement = intersection S2T⋂T2S
Alignement= Union S2T⋃T2S
Alignement= Compatibilité (S2T, T2S) #La relation s→t est conservé si
t→0 ou t→s) (Idem dans l'autre sens)
# Regroupements itératif en fonction de la cloture transitive
iterate jusqu'à plus aucun nouveau groupement n'est possible
Pour tokens sources  S = 0 ... n
       Pour tokens cibles T = 0 ... n
                        Si (S[0], T[0]) ∈ alignement {
                            Si (S[0], T[0]) exist dans groupe {

                              Si distance entre groupes < DistanceFaible {
                                    Regroupement des groupes
                                    Enregistre (S[0], T[0]) dans groupe
                              Sinon
                                     enregistre       simplement       la
correspondance
                              Fin si
                            Fin si
                         Fin si
Fin Pour
Incrémenter le nombre de paires traitées
```

La sortie de liste de tokens appariés est affichée en formats pal :

Exemple se sortie :

adopt^V aprobar^V

by^PRP por^PRP

the^DET el^DET

General^NNgeneral^ADJ

Assembly^NNasamblea^NN

Nous avons pris d'une manière aléatoire 100 couples alignés de chaque couple de langues afin d'évaluer manuellement ce nouvel alignement lexical obtenu par GIZA++. Le taux de bruit (pourcentage d'erreurs) figure dans le tableau suivant :

Couples de langues évaluées	Taux de bruit
français - anglais	7%
français - arabe	53%
espagnol - français	5%
espagnol - arabe	29%
anglais - espagnol	2%
anglais- arabe	11%

Tableau 9: Taux de bruit obtenu par GIZA++

II.5.4 Evaluation

Les résultats sont meilleurs qu'auparavant : le taux de bruit global baisse, et est en moyenne d'environ 28%, quoique le taux d'erreur reste très important pour certains couples (comme le couple fr-ar). Afin de comprendre quelles sont les erreurs les plus fréquentes, nous avons relevé les principaux cas de figure d'alignement erroné dans les résultats de GIZA++:

-Unité alignée avec son équivalent correct (qui est souligné dans les exemples suivants) et d'autres équivalents erronés, ce qui altère du coup le reste de l'alignement :

en_embodiment^NN →(ar_hmA^PRP tjsydA^NN) (trad. Littérale : Sont incarnation)
en_mandate^NN→fr_ lui^PR donner^V mandat^NN
fr_pratique^NN→(ar_ OfDl^ADJ AlmmArsAt^NN) (trad. Littérale : meilleur pratiques)

fr_participation^NN→(ar_ AlnAmyp^ADJ m$Arkp^NN) (trad. Littérale : développement participation)
fr_bannière^NN→es_ servir^V bajo^PRP bandera^NN

-Expression alignée en plusieurs unités séparées, ce qui aboutit à un alignement erroné :

Exemple 1 :

en_bear^V →aligné avec→fr_prendre^V
in^PRP→ aligné avec→fr_ en^PRP
mind^NN→ aligné avec→ fr_considération^NN

Exemple 2 :

*ar_ jdwl^NN (trad.tableau) →aligné avec→ fr_*
ar_ AlOEmAl^NN (trad.les travaux) →aligné avec→ fr_ordre^NN du^PRP jour^NN

Couple de phrases mal aligné, ce qui génère des alignements erronés au niveau lexical :

Phrase source, en: income and sources of funding
Phrase cible, fr : le budget des projets de l'office était de 47,1 millions de dollars

On note cependant de nombreux alignements corrects, même pour des cas de figure difficiles :

Alignements n-n :

en_secretary-general^NP→aligné avec→es_ Secretario^NP general^NN
en_we^PR welcome^V→aligné avec→ar_ nrHb^V
en^PRP développement^NN→aligné avec→ developing^ADJ
fr_Cinquante-huitième^NPsession^NN→aligné avec→ar-Aldwrp^NNAlvAmnp^ADJ Alxmswn^ADJ
fr_ toutefois^ADV→aligné avec→es_ sin^PRP embargo^NN
en_within^PRP→aligné avec→es_ dentro^ADV limite^NN
es_ prestara^V asistencia^NN→aligné avec→en_ assist^V

Sigles alignés correctement :

fr_ ONU^Autre→aligné avec→ar_ AlOmm^NP AlmtHdp^ADJ

Notons que l'alignement lexical fournit des équivalents traductionnels en contexte, qui ne sont pas toujours des équivalents d'un point de vue général, c'est-à-dire susceptibles de figurer dans un dictionnaire.

Par ailleurs, l'alignement permet de regrouper des traductions correspondant à des unités mal segmentées : c'est le cas pour une unité agglutinée comme (nrHb^V) (trad. nous nous félicitons) qui a été bien aligné avec ses deux équivalents (en_we^PR welcome^V), alors que si la tokenisation avait été bien faite, on aurait pu avoir une correspondance mot à mot.

Pour éliminer le bruit dû aux alignements erronés, un simple filtrage supprimant les cas de figures peu fréquents n'est pas suffisant car certains alignements erronés sont récurrents dans notre corpus (ex. *fr_ordre du jour qui est mal aligné avec ar_AlOEmA , trad. les travaux*).

Deux problèmes se posent : comment filtrer le bruit lié aux alignements erronés, afin d'avoir des équivalents traductionnels corrects dans les quatre langues et utiliser des données fiables dans les étapes ultérieures de notre expérimentation ? Et comment exploiter ces résultats afin d'obtenir des groupes d'équivalents susceptibles d'être rapprochés des synsets de Wordnet ?

II.6 Bilan d'étape

Nous avons montré au début de ce chapitre que les dictionnaires généraux grand public, tels que les dictionnaires Larousse en ligne, ne fournissent pas une référence totalement stable, et ne permettent pas de réaliser notre objectif d'extraire des cliques maximales ressemblant à des synsets.

Une autre approche a été proposée en se basant sur un corpus parallèle : nous supposons qu'à partir de ce corpus il est possible d'extraire des cliques susceptibles d'organiser empiriquement le sens des unités en mettant en évidence les relations de synonymie et la polysémie.

Dans le but de la création de ces cliques, nous avons d'abord procédé à l'alignement phrastique du corpus, puis à l'extraction des équivalences traductionnelles au sein de notre corpus aligné, pour 4 langues : français, arabe, anglais et espagnol. Mais nous avons constaté que les résultats obtenus contiennent une part importante de bruit, notamment pour certains couples de langues (comme le français et l'arabe). Que faire de ce bruit ?

Notre hypothèse est que la méthode de construction des cliques multilingues, qui contiennent des unités équivalentes en plusieurs langues à la fois peuvent filtrer une part importante de ce bruit. Nous pensons qu'il est en effet peu probable qu'une erreur d'alignement induise une intersection non vide, au sein d'une clique, entre plusieurs langues à la fois.

Dans le chapitre suivant, nous allons exposer les étapes du traitement effectué en vue d'extraire des cliques multilingues fiables à partir d'un corpus parallèle partiellement bruité.

Chapitre III : Expérimentation : extraction des cliques multilingues à partir d'un corpus parallèle

III.1 Introduction

Ce chapitre détaille la suite de notre travail expérimental, à savoir l'extraction des cliques multilingues à partir de notre corpus parallèle, chaque clique contenant un ou plusieurs lexèmes simples et (ou) composés dans chaque langue (fr, en, ar, es). Ces lexèmes de langues différentes au sein d'une même clique sont des équivalents traductionnels présumés, extraits grâce à Giza++ (voir II.5.3.3).

Nous faisons l'hypothèse, pour des cliques impliquant 3 langues ou plus, qu'il est probable que les relations d'équivalences deux-à-deux dénotent en fait (au moins) un sens commun partagé par tous les eq-sets.

En effet, grâce au principe de *triangulation*, on peut s'appuyer sur les équivalents potentiels dans une tierce langue afin d'éliminer les équivalents hors contexte, car deux unités appariées par erreur n'ont probablement pas les mêmes équivalents dans d'autres langue. Le recours à une quatrième langue de projection, par le surcroît d'information apportée, doit logiquement renforcer cette hypothèse.

Outre le filtrage des erreurs, la triangulation permet une forme de désambiguïsation mutuelle des unités de la clique, seule les acceptions communes à toutes les unités étant considérées comme valides pour une clique donnée.

Par exemple dans la clique : (*fr-N-économie, en-N-saving, it-N-risparmio, de-N-Einsparung*), il est probable que les sens partagés par (*fr-N-économie, en-N-saving*) et (*fr-N-économie, it-N-risparmio*) aient une intersection commune. En effet si tel n'était pas le cas, cela signifierait que les équivalences (*fr-N-économie, en-N-saving*) et (*fr-N-économie, it-N-risparmio*) correspondent à deux acceptions distinctes de *fr-N-économie*. Et du coup il serait peu probable que *en-N-saving* et *fr-N-it-N-risparmio* soient eux-même des équivalents potentiels. L'ajout d'un équivalent commun à ces trois unités, avec l'allemand *de-N-Einsparung,* renforce encore cette hypothèse de convergence des intersections. L'appartenance à une clique, qui implique une relation avec tous les éléments de la clique, révèle, lorsqu'un grand nombre de langues est mis en jeu, une propriété centripète de la clique, le "centre" qui en assure la cohésion pouvant être interprété comme l'intersection commune à tous ses membres.

Nous nommerons désormais cette supposition *l'hypothèse de centralité des cliques*. Une conséquence de cette hypothèse est que l'ajout d'une langue supplémentaire devrait aboutir, en général, à des cliques plus fines sur le plan sémantique, pour lesquelles l'intersection est plus étroite ou égale.

Mais que peut-on dire pour deux éléments d'un même eq-set (c'est-à-dire deux éléments de la même langue au sein d'une clique) ? Par définition, ils ne sont pas en relation d'équivalence. Doivent-ils nécessairement être synonymes, c'est-à-dire avoir une intersection sémantique (un sens dénotationnel commun) correspondant au centre de la clique ? On peut imaginer certains cas où une langue opère une distinction non marquée dans les autres, utilisant par exemple deux lexèmes concurrents là où les autres n'en n'utilisent qu'un seul. Dans ce cas, les deux lexèmes peuvent être considérés comme cohyponymes, et ce n'est pas leur intersection mais plutôt leur union qui doit correspondre au centre de la clique. La figure ci-dessous montre ce type de configuration pour une clique (U1, U2, U3a et U3b), correspondant à 3 langues L1 (fond gris), L2 (fond transparent) et L3 (ligne hachurée). Les unités U3a et U3b partitionnent en deux sous-sens l'intersection des sens commune à L1 et L2.

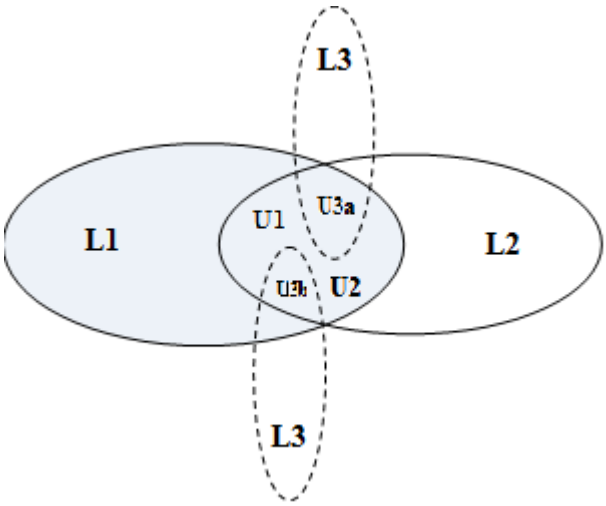

Figure 19 : Union de deux lexèmes correspondant au centre de leur clique

Il est bien sûr possible, d'un point de vue général, que les eq-sets soient également synonymes (c'est-à-dire que U3a et U3b aient une intersection non nulle) mais ce n'est pas une condition obligatoire. Cette configuration peut naturellement concerner plusieurs langues à la fois (et non seulement L3 comme dans la figure ci-dessus).

Ainsi les cliques seraient caractérisées par l'intersection des sens d'une langue à l'autre et par leur union pour la même langue. Si deux unités possèdent la même distribution de leur sens, ont pourra parler de polysémie *parallèle* : par exemple *fr-N-disque* et *it-N-disco* ont des acceptions communes distribuées parallèlement (/plaque circulaire/, /disque intervertébral/, /CD/, /disque dur/, etc.). Tandis que pour *fr-N-disque* et *en-N-record* on peut parler de polysémies *orthogonales*. De telles polysémies *orthogonales* aboutissent donc à des intersections réduites. Or il est peu probable que des polysémies parallèles se retrouvent dans de nombreuses langues différentes et génétiquement éloignées[29]. Il en découle que l'utilisation de plusieurs langues différentes dans une clique peut permettre de désambigüiser les lexèmes polysémiques, l'intersection aboutissant probablement à une réduction de la *surface* considérée[30]. Par ailleurs le recours à plusieurs langues peut permettre de limiter les conséquences des erreurs d'alignement dans la construction des cliques, dans la mesure où il est peu probable qu'une erreur d'alignement induise une intersection non vide au sein d'une clique multilingue.

Partant de ces hypothèses, nous avons implémenté une méthode d'extraction des cliques sur nos équivalents traductionnels en quatre langues :

[29] Même si ce n'est pas à exclure, car il y a tout de même une certaine motivation dans la structuration de la polysémie.
[30] Cette vision géométrique est bien entendu métaphorique, mais elle permet d'illustrer de façon claire et intuitive l'organisation des identités et des différences dans la comparaison des sens.

III.2 Construction des tableaux de stockage des données

Pour mieux délimiter notre champ de recherche, nous ne nous intéressons ici qu'aux unités lexicales appartenant aux classes ouvertes suivantes : noms, verbes, adjectifs et adverbes.

A partir de nos fichiers contenant les équivalents traductionnels alignés pour chaque couple de langues (les fichiers du format pal) nous avons construit deux tableaux de stockage des données, c'est-à-dire un tableau qui enregistre la fréquence des unités et un autre qui enregistre les occurrences des couples de lexèmes appariés dans le corpus (les cooccurrences). Ces lexèmes appariés sont de (n-n) c.à.d. on peut y trouver des expressions polylexicales et des mots simples. On effectue ensuite un filtrage afin de ne pas surcharger les tableaux avec des informations inutiles. Les informations retenues sont stockées dans des bases de données qui nous serviront à construire nos cliques sans devoir reparser à chaque fois les fichiers (du format pal).

Voici une description de l'algorithme appliqué au filtrage et à l'enregistrement de ces données (cf. Le script Stock-infos-v4.pl. **Annexe 13**, page 292).

Première étape : analyse des fichiers contenant les équivalents traductionnels alignés (format pal) *:*

```
Initialiser tableau associatif occ
Initialiser tableau associatif corr
Pour chaque paire (U1,U2) alignée
     Si (U1 ≠ Ø et U2 ≠ Ø )
          Si occ[U1] n'existe pas
                Initialiser : occ[U1] <- 0
          Fin si
          Incrémenter : occ[U1] <- occ[U1] +1
          Si occ[U2] n'existe pas
                Initialiser : occ[U2] <- 0
          Fin si
          Incrémenter : occ[U2] <- occ[U2] +1

          Si corr[U1][U2] n'existe pas {
                Initialiser : corr[U1][U2] <- 0
          Fin si
          Incrémenter : corr[U1][U2] <- corr[U1][U2] +1
     Fin si
Fin pour
```

Deuxième étape : filtrage

Le filtrage est effectué en éliminant toutes les correspondances dont la fréquence absolue est trop faible, ou dont la fréquence relative (par rapport au nombre d'occurrences des unités du couple) est au dessous d'un certain seuil :

```
Initialiser variable de fréquence minimale Freqmin
Pour chaque clé corr [U1][U2]
          Corr[U1][U2]<-Nbocc
      Pour chaque unité occ[U1]
                Freqmin= Occ[U1]
           Pour chaque occ[U2]
                      Si occ[U2]<freqMin {
                          freqMin = occ[U2]
                      Fin si
                   Si (Nbocc <= 2 ou Nbocc/$freqmin  <0.05){
                         Supprimer Corr[U1][U2]
                         Supprimer Corr[U2][U1]
                   Fin si
                     Fin pour
       Fin pour
Fin pour
```

Troisième étape : Stockage de correspondances avec leur nombre de cooccurrences dans une base de données:

Le stockage dans la base de données (stock_corr) est fait uniquement pour les correspondances de catégories suivantes : (Nom, verbe, adjectif, adverbe).

En sortie de notre algorithme nous obtenons :

- Une table de hachage %occ contenant le nombre d'occurrences pour les clés de type *lang-catégorie-lemme*.

- Une table de hachage %stock_corr, enregistrée sur le disque sous forme de DBM Perl, contenant tous les couples *langue1-Catégorie-lemme#langue2-Catégorie-lemme* comme clés avec leur nombre de cooccurrence, les clés sont symétrisées (c.à.d. quand on a clé1#clé2 on a aussi clé2#clé1). Ce tableau sera stocké en base de données pour être réutilisé.

Les nombres de couples obtenus pour chaque paire de langues sont les suivants: en-fr (1 727 375) en-es(2 009 179) es-fr (1 896 386) ar-fr (1 109 287) ar-en (1 046 048) ar-es(978 266).

Pour donner une idée de la qualité des résultats obtenus, voici un extrait d'une centaine de clés de ce tableau (9 clés par paire de langues=9*12) tirés aléatoirement :

es-fr	es-el-DET#fr-le-DET es-cambio-NOUN#fr-changement-NOUN es-no-Autre#fr-sans-PRP es-ser-V#fr-se-PR es-nada-DET nuevo-Adj#fr-nouveauté-NOUN es-para-PRP#fr-à-PRP es-el-DET#fr-le-DET es-Organización-NOUN#fr-organisation-NOUN es-nunca-Adv haber-V#fr-jamais-Adv
en-fr	en-review-NOUN#fr-examen-NOUN effectuer-V en-by-PRP#fr-par-PRP en-the-DET#fr-le-DET en-Nordic-Adj#fr-scandinave-Adj en-government-NOUN#fr-gouvernement-NOUN en-of-PRP#fr-du-PRP en-PRSP-NP#fr-DSRP-NP en-process-NOUN#fr-processus-NOUN en-reveal-V#fr-révéler-V
ar-es	ar-yzdAd-V#es-prever-V ar-AstvmAr-NOUN#es-inversión-NOUN ar-Almktb-NOUN#es-UNOPS-NOUN ar-OTls-V#es-el-DET Atlas-NOUN ar-b-PRP#es-en-PRP ar-nsbp-NOUN 50-CD fy-PRP#es-@card@-CD %-SYM ar-EAm-NOUN#es-en-PRP ar-mqArnp-NOUN#es-con-PRP ar-EAm-NOUN#es-@card@-CD
en-es	en-socio-economic-Adj#es-socioeconómicas-Adj en-issue-NOUN#es-cuestión-NOUN en-at-PRP#es-en-PRP en-the-DET#es-el-DET en-intergovernmental-Adj#es-intergubernamental-Adj en-level-NOUN#es-plano-NOUN en-.-SENT#es-.-PUN en-have-V#es-haber-V en-help-V#es-contribuir-V a-PRP
ar-fr	ar-AlqDAyA-NOUN#fr-problème-NOUN ar-Alr}ysyp-Adj#fr-principal-Adj ar-ObEAd-NOUN#fr-portée-NOUN ar-AlwTnyp-Adj#fr-national-Adj ar-AlIqlymyp-Adj#fr-régional-Adj ar-w-CC#fr-et-CC ar-w-CC yHdd-V#fr-le-DET ar-Altqryr-NOUN#fr-rapport-NOUN présenter-V intervention-NOUN de-PRP ar-OrbEp-Adj#fr-quatre-CD

ar-en	ar-h*A-ART#en-of-PRP ar-Altqryr-NOUN#en-Report-NP ar-fy-PRP#en-the-DET ar-ElY-PRP#en-identify-V area-NOUN ar-AlnA$}p-Adj#en-of-PRP ar-Alty-PR#en-particular-Adj ar-bAhtmAm-NOUN#en-worldwide-Adj ar-Alwqwf-NOUN#en-four-CD ar-AlqDAyA-NOUN#en-and-CC
fr-es	fr-groupe-NOUN #es-grupo-NOUN fr-criminel-Adj #es-delictivo-Adj fr-organiser-V #es-organizado-Adj fr-soit-CC #es-o-CC fr-de-PRP#es-de-PRP fr-son-DET #es-suyo-PR fr-intention-NOUN#es-intención-NOUN fr-de-PRP #es-de-CC fr-commettre-V #es-cometer-V
en-fr	en-Government-NP#gouvernement-NOUN en-of-PRP#de-PRP en-the-DET#le-DET en-Organization-NOUN#organisation-NOUN en-of-PRP#de-PRP en-African-NOUN#africain-Adj en-Unity-NP#unité-NOUN en-,-PUN#NULL en-at-PRP#à-PRP
es-ar	es-asistencia-NOUN # ar-AlmsAEdp-NOUN es-judicial-Adj # ar-AlqAnwnyp-Adj es-recíproco-Adj # ar-AlmtbAdlp-Adj NULL # ar-fy-PRP Es-antes-Adv # ar-Oqrb-Adj NULL # ar-wqt-NOUN Es-posible-Adj # ar-mmkn-Adj NULL # ar-,-PUN Es-y-CC # ar-w-CC
es-en	es-tener-V # en-take-V es-en-PRP # en-into-PRP es-cuenta-NOUN # en-account-NOUN es-el-DET # en-the-DET es-finalidad-NOUN # en-purpose-NOUN es-de-PRP # en-of-PRP es-ese-DET # en-that-DET es-protocolo-NOUN # en-protocol-NOUN es-.-PUN # en-.-PUN
fr-ar	fr-concerner-V #ar-AlmtElqp-Adj b-PRP fr-infraction-NOUN #ar-AljrA}m-NOUN fr-dont-PR #ar-Alty-PR fr-pouvoir-V être-V #ar-yjwz-V fr-un-DET #ar-hy}p-NOUN

	fr-persoNoune-NOUN moral-Adj #ar-AEtbAryp-Adj fr-responsable-Adj #ar- Alms&wlyp-NOUN En-PRP fr-conformément-Adv #ar-mqtDY-NOUN fr-article-NOUN #ar-AlmAdp-NOUN
en-ar	en-Secretary-General-Noun #ar-AlOmyn-Noun AlEAm-Adj en-Nations-Noun #ar-AlOmm-Noun en-United-NOUN #ar-AlmtHdp-Adj en-copy-Noun #ar -nsx-Noun mn-PRP en-law-Noun #ar -qwAnyn-Noun en-its-PR #ar -hA-PR en-that-PR #ar -Almnf-Adj en-furnish-V #ar -*p-Noun en-to-PRP #ar -l-PRP

Figure 20: Exemple du contenu de %stock_corr (échantillon avec une centaine de clés)

III.3 Extraction des cliques

Dans cette partie, nous allons utiliser la sortie de l'étape précédente afin de construire des cliques pour des unités qui nous intéressent. En fait, avec un grand volume de données, il n'est plus possible de tous afficher, et l'évaluation devient impossible, c'est pourquoi nous avons choisi de tester cet étape sur des mots particuliers.

L'algorithme utilisé ici pour l'extraction des cliques (cf. La script Extract_clique.V2.pl.**Annexe 14**, page 296) a été développé au début de ce travail, c'est pourquoi nous n'avons pas utilisé celui de chapitre 2 (cf. Le script extractLarousse. pl. **Annexe 8**, page 244) qui est plus générique.

Argument : on spécifie comme argument du script l'unité U0 pour laquelle on veut extraire toutes les cliques qui la contiennent.

Entrées : les entrées de notre script sont les tables de hachage %stock_corr et %occ (la sortie du script Stock-infos-v4.pl).

Principe :

- Chercher toutes les unités de même catégorie avec lesquelles U0 est alignée dans notre corpus. Par exemple, pour une U0 arabe, on obtient trois listes :
    - C_{en}(U0) : liste de toutes les unités anglaises appariées avec U0
    - C_{fr}(U0) : Liste de toutes les unités françaises appariées avec U0
    - C_{es}(U0) : Liste de toutes les unités espagnoles appariées avec U0
      Par C(U0) on désigne l'union de ces trois listes (on retient dans tous les cas les trois listes correspondant aux langues différentes de celles de U0).
- S'appuyant sur le nombre d'occurrences et de correspondances de chaque unité alignée avec l'unité U0, on construit toutes les cliques minimales, c'est-à-dire les quadruplets avec une unité dans chaque langue, tels que toutes les unités soient correspondantes deux à deux. Par exemple, pour (ar-Verb-كتب)(trans.ktb), on obtient :
    - ar-Verb-ktb, en-Verb-write, fr-Verb-écrire, es-Verb-escribir
    - ar-Verb-ktb, en-Verb-edit, fr-Verb-écrire, es-Verb-redactar
    - ar-Verb-ktb, en-Verb-compose, fr-Verb-composer, es-Verb-componer
    - …

- Pour chaque clique minimale :
    - Construire les ensembles de candidats pour l'augmentation de la clique : ces candidats sont constitués par l'intersection des unités des correspondants obtenus dans les 3 autres langues. Pour une clique minimale (U0,U1,U2,U3) on a donc : Cand0=C(U1)∩C(U2)∩C(U3), Cand1=C(U0)∩C(U2)∩C(U3), Cand2=C(U0)∩C(U1)∩C(U3) et Cand3= C(U0)∩C(U1)∩C(U2).
    - Initialiser l'ensemble des cliques en construction avec la clique minimale courante.
    - Augmenter itérativement les cliques en construction, en tentant d'ajouter les unités issues des ensembles de candidats. Dans notre exemple, Cand3 contient *fr-V-rédiger* et *fr-V-produire*, ce qui nous permet de créer les deux nouvelles cliques suivantes, dès la première itération :
        - Ktb , write, écrire, escribir, rédiger
        - Ktb , write, écrire, escribir, produire

      A l'itération suivante, par l'ajout de ces candidats, on obtient une seule clique :

        - Ktb , write, écrire, escribir, rédiger, produire

Toute clique n'étant plus augmentable par l'ajout d'un nouveau candidat est considérée comme maximale.

Les itérations s'arrêtent lorsque plus aucune clique n'a pu être augmentée. L'ensemble des cliques en constructions contient alors toutes les cliques maximales issues de la clique minimale courante.

Algorithme :

Etape 1 : Construire les tableaux corr et freq pour l'unité U0 pour laquelle on veut extraire toutes les cliques qui la contiennent.

```
Initialiser tableau associatif corr
Initialiser tableau associatif freq
Pour chaque unité [U0]
          nbocc1 = stock_corr[U0][U1..n]
          Initialiser :   corr[U0][U1..n] <-nbocc1
               corr[U1..n][U0] <-nbocc1
Fin pour
Pour chaque clé corr[U0][U1..n]
          Initialiser : freq[U0] <-occ[U0]+1
          freq[U0] <-occ[U1..n]+1
```

```
Fin pour
```

Etape2 : Filtrage du tableau corr pour ne garder que les relations pertinentes.

Il s'agit d'un filtrage plus fin, par rapport au filtrage précédent. Ce dernier avait pour but de supprimer le gros du bruit afin de rendre les fichiers plus maniables. Le présent filtrage, effectué au moment de la création des cliques, fait partie des paramètres de l'extraction des cliques :

```
Initialiser variables :
seuilfreq=0,05
seuilnbocc=10
      Pour chaque clé corr[U0][Un]
                nbocc2= corr[U0][Un]
                nbocc3= corr[Un][U0]
                freq1=freq[U0]
                freq2=freq[Un]
           Si ((nbocc2>2) et ((nbocc2/ freq1) >seuilfreq)) ou ((nbocc3>2)
           et ((nbocc3/ freq2) >seuilfreq)))
                 Si nbocc2< seuilnbocc
                                        supprimer      le      couple       :
      corr[U0][Un]<-0
                       supprimer le couple: corr[Un][U0]<-0
                                  Fin si
           Sinon on conserve la relation
                                  corr[U0][Un]<-nbocc2
                                   corr[Un][U0]<-nbocc3
                      Fin si
            Fin pour
```

Le seuil de cooccurrences choisi (seuilnbocc) n'est pas très élevé afin de ne pas perdre beaucoup de couples lors du filtrage. Et en même temps, n'est pas très faible afin de ne pas avoir beaucoup de bruit dans notre résultat.

Etape 3 : Construire des cliques minimales en construisant tous les quadruplets possibles contenant notre mot de départ U0 et augmenter ces cliques itérativement.

```
Initialiser la liste clique
Initialiser le tableau associatif Clique_max
      Pour chaque clé corr[u0][U1..n]
         • Cand0=C(U1)∩C(U2)∩C(U3),
         • Cand1=C(U0)∩C(U2)∩C(U3),
         • Cand2=C(U0)∩C(U1)∩C(U3)
         • Cand3= C(U0)∩C(U1)∩C(U2).
         Pour chaque Cand0..3 ∉ clique
            clique =Cand0..3
               Fin pour
```

```
      Fin pour
Clique_max<-clique
```

La sortie de notre algorithme est une table de Hachage nommée %clique_max qui contient la liste des cliques maximales dans ses clés. Les clés de cette table sont construites de telle manière que les doublons produits par l'algorithme soient regroupés. Pour ce faire, les éléments constitutifs de la clique ont simplement été triés par ordre alphabétique.

Exemple d'une liste de quelques cliques maximales obtenues pour l'unité *fr-Noun-question*:

(fr-Noun-question ar-Noun-Als&Al en-Noun-question es-Noun-pregunta)

(fr-Noun-point fr-Noun-question ar-Noun-Albnd en-Noun-item es-Noun-tema)

(fr-Noun-sujet en-Noun-topic fr-Noun-question ar-Noun-AlmwDwE en-Noun-subject es-Noun-tema)

III.3.1 Examen des résultats

Etant donné que l'étiqueteur ASVM 1.0 ne fait pas de lemmatisation complète, nous avons choisi de ne pas partir de la langue arabe puisqu'on sera obligé d'effectuer l'extraction à partir de toutes les formes fléchies possibles d'un même lemme arabe cherché. En outre, le fait d'avoir plusieurs formes fléchies pour un même lemme peut dégrader les résultats. La langue choisie est le français.

Ainsi, nous avons appliqué notre méthode sur les 100 noms et 102 verbes français les plus fréquents dans notre corpus (y compris 90 noms et 79 verbes polysémiques). Chaque nom a plus de 1500 occurrences dans le corpus (cf. **Annexe 2**, page 198), tandis que chaque verbe en a plus de 300 (cf. **Annexe 3**, page 204). La sélection de ces unités a été faite en se basant sur les résultats de notre script (stock-info.pl).

Pour chaque unité étudiée, nous avons obtenu une ou plusieurs clique(s), où chaque clique contient au moins une unité dans chaque langue. Cette différence du nombre de cliques obtenus est due à la variation du nombre d'occurrences et de cooccurrences de chaque unité.

Nous avons remarqué que dans certains cas, nous trouvons, dans une même clique, plusieurs formes fléchies pour la même unité arabe, comme dans l'exemple suivant qui montre une partie de la sortie des cliques pour le lemme français *question* :

(fr-Noun-question ar-Noun-AlOs}lp ar-Noun-Als&Al en-Noun-question es-Noun-pregunta)

(fr-Noun-question ar-Noun-AlqDyp ar-Noun-msOlp ar-Noun-qDAyA ar-Noun-msA}l ar-Noun-AlmsA}l ar-Noun-AlmsOlp ar-Noun-AlqDAyA en-Noun-issue es-Noun-cuestión)

(fr-Noun-point fr-Noun-question ar-Noun-Albnd en-Noun-item es-Noun-tema)

(fr-Noun-question ar-Noun-AlmsA}l en-Noun-issue en-Noun-matter es-Noun-cuestión)

(fr-Noun-question ar-Noun-AlmsA}l en-Noun-matter es-Noun-cuestión)

(fr-Noun-question ar-Noun-msOlp ar-Noun-AlmsOlp en-Noun-issue en-Noun-question es-Noun-cuestión)

(fr-Noun-sujet fr-Noun-question ar-Noun-AlmwDwE en-Noun-topic en-Noun-subject es-Noun-tema)

(fr-Noun-question ar-Noun-msOlp en-Noun-question es-Noun-cuestión)

(fr-Noun-question ar-Noun-msOlp en-Noun-question en-Noun-issue en-Noun-question es-Noun-cuestión)

Nous avons remplacé manuellement toutes les formes fléchies d'un même lemme arabe, existant dans la même clique, par leur lemme, afin d'éviter ces problèmes liés à une lemmatisation parcellaire du corpus arabe (puisque l'étiqueteur ASVM 1.0 ne fait pas de lemmatisation complète) et la dispersion artificielle des cliques.

L'application de la lemmatisation manuelle sur les unités arabes est faites après le lancement d'extraction des cliques, puisque les unités arabes dans nos cliques extraites sont beaucoup moins nombreuses que celles de notre base de données (stock_corr), ce qui facilite notre tâche de lemmatisation.

Nous avons procédé par lemmatiser manuellement toutes les sorties arabes en suivant les principes suivants:

- La lemmatisation est faite sur la base du lemme (stem-based) et pas sur la base de la racine (root-based) (Dilekh, 2008). Par exemple la racine du mot arabe (الأعراب) (trans. Al>ErAb/trad. les bédouins) est (عرب)(trans.Erb/trad.les arabes). Un lemme est défini simplement comme un mot sans

préfixe ou/et suffixe. Par exemple, le lemme du mot arabe (الحيوانات , les animaux) est (حيوان , animal).

- Nous avons retiré tous les affixes (préfixes, infixes, ou/et suffixes) des mots pour ramener ces derniers à leurs lemmes.

- Nous avons égalisé ou combiné certaines formes variables du même mot comme (papier, papiers) et (pli, plis, plié, pliant…).

Ainsi, les regroupements effectués pour l'exemple précédent de cliques de l'unité *question* sont :

ar-Noun-AlOs}lp (forme en pluriel+ l'article ال est agglutiné au mot) et ar-Noun-Als&Al (l'article ال est agglutiné au lemme) sont des formes fléchies pour le lemme *s&Al* (سؤال) ar-Noun-AlqDyp (l'article ال est agglutiné au lemme), ar-Noun-qDAyA (forme en pluriel) et ar-Noun-AlqDAyA (forme en pluriel+ l'article ال est agglutiné au mot) sont des formes fléchies pour le lemme *qDyp(قضية)*.

ar-Noun-msOlp, ar-Noun-AlmsOlp(l'article ال est agglutiné au lemme), ar-Noun-msA}l (forme en pluriel) et ar-Noun-AlmsA}l(forme en pluriel+ l'article ال est agglutiné au mot) sont des formes fléchies pour le lemme *msOlp (مسألة)*.

ar-Noun-Albnd (l'article ال est agglutiné au lemme) est une forme fléchie pour le lemme *bnd(بند)*.

ar-Noun-AlmwDwE (l'article ال est agglutiné au lemme) est une forme fléchie pour le lemme *mwDwE (موضوع)*.

Notons qu'une unité peut appartenir à différentes cliques : cette caractéristique est due à la non-transitivité de la relation d'équivalence et de synonymie. Mais dans ces premiers résultats, nous remarquons la redondance de plusieurs groupes d'unités dans des cliques où parfois il y a un seul mot qui diffère. Prenons par exemple les deux cliques suivantes :

*(fr-Noun-question ar-Noun-**msOlp** en-Noun-matter es-Noun-cuestión en-Noun-issue)*

*(fr-Noun-question ar-Noun- **msOlp** en-Noun-matter es-Noun-cuestión)*

La seule différence entre ces deux cliques est l'unité *en-Noun-issue*, qui est absente de la deuxième clique. De fait la deuxième clique n'est pas maximale, et l'algorithme n'a pu l'identifier du fait de la non-lemmatisation.

Prenons l'exemple suivant des cliques extraites pour l'unité *fr-Noun-travail*:

(fr-Noun-travail ar-Noun-Eml en-Noun-work es-Noun-trabajo)

(fr-Noun-travail ar-Noun-Eml en-Noun-labour es-Noun-trabajo es-Adj-laboral)

On peut trouver des cas de cliques ressemblantes, comme dans l'exemple précédent, et qui diffèrent par une ou deux unités. On peut penser que cela est dû à notre corpus (la relation est inexistante ou ne semble pas pertinente au niveau statistique à cause de degré de spécialisation ou la taille du notre corpus).

Ces phénomènes (ex. non-lemmatisation, limitation du corpus…etc.) aboutissent à une prolifération de petites sous-cliques qui devraient pourtant appartenir à des cliques plus grandes maximales.

Etant donné que nous cherchons dans cette partie à avoir des cliques maximales où toutes les composantes d'une clique sont interconnectées, nous pensons qu'une phase de clusterisation appliquée sur nos résultats peut aider à regrouper des cliques voisines et éliminer les cliques non maximales. En outre, si l'on veut coller à l'hypothèse que les cliques représentent des *sens élémentaires* reflétant la partie commune à leurs composants, nous pensons que la clusterisation a pour effet de diluer le sens, c'est ce qu'ont montré les expériences du chapitre 2 (voir Chapitre II :) et ce que nous allons montrer plus loins dans l'évaluation de notre résultat.

Dans cette étape de clusterisation, nous avons appliqué une phase de clusterisation identique à celle appliqué dans le chapitre 2 (voir Chapitre II :) afin de réunir les cliques qui sont très proches et qui n'ont pu être identifiées du fait des limitations du corpus ou de la non-lemmatisation dans le cas de l'arabe.

Toutefois, comme précédemment (voir Chapitre II :), il faut éviter de regrouper trop de cliques, ce qu'aboutirait à multiplier les sous-sens dans le même cluster, et pourrait affaiblir l'hypothèse de centralité des cliques.

Dans l'idéal, les clusters devraient permettre de caractériser l'intersection des sens d'une langue à l'autre, ainsi que leur union pour la même langue.

III.4 Résultat de la clusterisation : étude de quelques cas de figure

La phase de clusterisation a été appliquée sur les résultats obtenus pour les 100 noms et 102 verbes français les plus fréquents dans notre corpus (y compris 90 noms et 79 verbes polysémiques).

Dans le résultat de clusterisation, nous avons obtenu de nombreux clusters pertinents comme dans l'exemple suivant :

*{(fr-Noun-gestion **fr-Noun-administration** ar-Noun-IdArp en-Noun-management es-Noun-gestión)*

*(fr-Noun-gestion ar-Noun-IdArp **en-Noun-administration** en-Noun-management es-Noun-gestión)}*

Où on a une fusion de deux cliques qui diffèrent légèrement dans leurs composants mais dénotent la même acception : /gérance/.

La clusterisation aboutit dans certains cas à des regroupements de sens, comme dans l'exemple suivant qui montre une fusion erronée des cliques de l'unité *fr-Noun-examen* :

{(fr-Noun-examen ar-Noun-AstErAD en-Noun-examination en-Noun-review en-Noun-consideration es-Noun-examen), (fr-Noun-examen fr-Noun-épreuve ar-Noun-AmtHAn en-Noun-examination es-Noun-examen)}

On voit qu'un seul cluster contient deux cliques dénotant deux acceptions différentes (/observation/ et /épreuve/), ce qui aboutit à un élargissement de sens. Cela est dû à la polysémie parallèle de plusieurs unités *es-Noun-examen*, *fr-Noun-examen*, *en-Noun-examination*, qui pèse en faveur du regroupement.

En outre, certains regroupements de cliques ne sont pas effectués alors qu'ils pourraient être opportuns, comme dans l'exemple suivant :

Cluster1:

{(fr-Verb-viser en-Verb-target ar-Verb-yhdf es-Verb-apuntar)} -> /viser un objectif/

Cluster2 :

{(fr-Verb-viser en-Verb-aim ar-Verb-yhdf ar-Verb/PRP-yrmy lIY ar-Verb/PRP-yhdf lIY ar-Verb-yrmy ar-Adj/PRP-Alm$Ar lIY es-Verb/PRP/Noun-tener por objeto)} -> /cible/

On voit dans les deux clusters précédents une clusterisation erronée qui a produit deux clusters dénotant la même acception (/cibler/).

Cet éparpillement est dû au caractère arbitraire de nos seuils de similarité, qui sont parfois trop élevés pour opérer certains regroupements. Mais des seuils plus bas aboutiraientt par ailleurs à de nombreux regroupements erronés, comme dans l'exemple précédent : en jouant sur ces seuils, on perd d'un côté ce que l'on gagne de l'autre, et il convient de trouver le juste équilibre entre l'éparpillement des cliques et leur regroupement intempestif.

Rappelons que le recours à la méthode de clusterisation est directement lié aux limitations du corpus: idéalement, avec un corpus suffisamment étendu pour représenter les principales acceptions des unités avec leurs différentes traductions, on devrait pouvoir compléter les cliques sans recourir à cette étape.

III.5 Evaluation préliminaire des résultats après clusterisation

L'évaluation présentée ici n'est qu'une évaluation préliminaire pour nos résultats après clusterisation, nous allons effectuer une évaluation complète dans le chapitre 4 dans la tâche de confrontation des cliques avec les synsets d'EWN.

L'obtention des clusters est faite, comme décrit précédemment, en mesurant la proximité des cliques par l'intersection et l'union de leurs correspondants. D'après nos hypothèses, chaque cluster devrait regrouper des unités synonymes autour d'un sens commun, et l'appartenance à des clusters différents devrait identifier des sens différents pour une unité donnée.

On peut s'attendre à ce que le nombre de clusters associés à une unité varie beaucoup selon l'unité lexicale considérée : quand l'unité est monosémique, on n'aura vraisemblablement qu'un seul cluster (sauf dans le cas où deux cliques restent artificiellement disjointes du fait de l'absence de liens d'équivalence dans le corpus) ; alors que lorsque l'unité possède différentes acceptions, il est vraisemblable qu'elle appartiendra à plusieurs clusters différents.

Afin de donner une évaluation qualitative sous la forme d'une étude de cas, nous essayons, dans cette première étape d'évaluation, de dégager manuellement les acceptions communes à l'intérieur des clusters obtenus.

Pour l'unité *fr- Noun-situation*, nous obtenons deux clusters cohérents dénotant deux acceptions différentes (/condition/ et /état/):

Cluster 1 :

(fr-Noun-situation en-Noun-condition ar-Noun- AlwDE es-Noun-situación) -> /condition/

Cluster 2 :

(fr-Noun-situation fr-Noun-état en-Noun-status ar-Noun-HAlp es-Noun-situación) -> /état/

Par ailleurs, la distribution des cliques d'une même entrée permet aussi d'identifier ses principales acceptions dans le corpus.

Pour l'unité *fr-Noun-droit* nous avons obtenu les cliques suivantes où chaque clique correspond à une acception bien distincte pour cette unité :

(fr-Noun-droit en-noun-right en-Noun-prerogative es-Noun-derecho ar-Noun-Hq) -> /prérogative/

(fr-Noun-droit en-Noun-law es-Noun-ley es-Noun-derecho ar-Noun-qAnwn) -> /loi/

*(fr-Noun-droit en-Noun-permission es-Noun-derecho ar-Noun-smAH ar-Noun-<*n) -> /permission/*

Notons que l'unité *fr-Noun-droit* partage la même polysémie que l'unité *es-Noun-derecho* mais la langue anglaise et la langue arabe effectuent la distinction entre ces 3 acceptions, d'où une intersection sémantique plus précise pour chacune des cliques. En outre, l'union des deux unités espagnoles *es-Noun-ley* et *es-Noun-derecho* a permis d'identifier une acception commune lié au sens de /loi/.

Par ailleurs, nos clusters permettent d'identifier des sens absents de certains dictionnaires, et que l'on trouve pourtant dans certains contextes du corpus. Prenons l'exemple de l'unité *fr-Noun-mot* :

{(fr-Noun-mot en-Noun-speech es-Noun-palabra es-Noun-discurso ar-Noun-xTAb)} -> /discours/

Nous pouvons remarquer que l'unité *fr-Noun-mot* ne se trouve pas comme équivalents à l'unité *en-Noun-speech* dans le dictionnaire Larousse. Ce sens est pourtant d'un usage courant, comme on le voit dans l'exemple suivant, tiré de notre corpus :

Le mot du Ministre est intervenu la cérémonie de décoration…

Il s'agit dans ce cas d'une lacune du dictionnaire.

III.6 Conclusion

Notre méthode d'extraction de cliques, ou des sous-graphes maximaux complets connexes, s'appuie sur l'extraction automatique des équivalents traductionnels, en se basant sur l'observation des occurrences et cooccurrences en corpus parallèle. Les correspondances extraites à partir de tous les alignements deux à deux des textes du corpus forment un immense graphe reliant des unités des quatre langues considérées. Pour ne retenir que les arcs les plus pertinents de ce graphe, nous avons d'abord procédé à un filtrage des correspondances lexicales. Dans un deuxième temps, nous avons procédé à l'extraction de toutes les cliques autour d'une unité donnée. Enfin, pour éviter l'éparpillement des cliques voisines mais disjointes du fait de l'absence d'une ou deux relations dans notre corpus, une phase de clusterisation a été mise en œuvre. Probablement, l'utilisation d'un corpus plus vaste aurait permis d'obtenir une couverture plus satisfaisante de la langue générale.

Nous avons fait l'hypothèse de la "centralité des cliques", l'interrelation entre tous les éléments de la clique étant probablement une conséquence de l'existence d'une intersection sémantique non vide. Par les quelques cas de figure présentés dans l'évaluation préliminaire des résultats, nous avons ainsi pu vérifier que des unités de même langue apparaissant dans une même clique étaient généralement sémantiquement voisines, cohyponymes ou synonymes (p.ex. *es-Noun-ley* et *es-Noun-derecho*). Par ailleurs, comme les différents sens d'un mot polysémique dans une langue donnée donnent souvent lieu à des traductions différentes dans les autres langues, on peut construire des cliques qui manifestent la différenciation et le partage des acceptions : ce que nous avons constaté pour *fr-Noun-droit*, dont les 3 cliques manifestaient les 3 acceptions principales trouvées dans le corpus.

On constate que les cliques présentent l'intérêt de renseigner à la fois sur la synonymie et la polysémie des unités.

On peut noter par ailleurs deux utilisations intéressantes des cliques obtenues :

- d'une part, elles constituent une forme de filtrage efficace des correspondances lexicales, la condition d'interrelation permettant d'éliminer pratiquement toutes les correspondances erronées. S'il reste des cliques contenant des mauvais équivalents traductionnels, il s'agit le plus souvent de correspondances fragmentaires entre des unités polylexicales qui n'ont pas été identifiées comme telles (c'est le cas p.ex. pour *en-Noun-emergency* et *fr-Noun-situation*). On peut donc réutiliser les lexiques de traductions obtenus pour l'alignement traductionnel, notamment lorsque l'on doit aligner

plus de deux langues en même temps (comme p.ex. *les textes de l'ONU ou de l'UE*). En effet, des séries d'équivalents traductionnels correctement appariés permettent de consolider l'alignement phrastique en étant réinjectés dans le calcul de similarités entre les phrases alignables.

- par ailleurs, nous avons vu qu'une même clique peut contenir plusieurs formes fléchies du même lemme arabe. Ce regroupement des formes fléchies d'un même lemme à l'intérieur d'une même clique pourrait être utilisé comme indice intéressant dans un processus de lemmatisation de l'arabe : cette piste n'a pas été, à notre connaissance, encore explorée.

Enfin, ces cliques maximales où toutes les unités sont en interrelation, du fait d'une probable intersection sémantique (des sens voisins ou connexes), ressemblent aux synsets d'un réseau sémantique tel que Wordnet. En effet, dans Wordnet, les sens sont caractérisés de manière similaire, par l'intersection sémantique d'un ensemble de nœuds fortement liés et activés simultanément : chaque synset dénote un "concept" différent situé au croisement d'un ensemble d'unités lexicales susceptible de porter ce sens, décrit par une courte définition appelée *gloss*. De la même manière qu'avec nos cliques, l'appartenance d'une unité lexicale à plusieurs synsets constitue une manifestation explicite de sa polysémie.

C'est pourquoi nous allons maintenant tenter de relier nos cliques avec le lexique sémantique d'EuroWordNet, afin d'évaluer la possibilité de récupérer pour les unités arabes des relations sémantiques déjà déclarées pour des unités en anglais, français et espagnol, dans leurs réseaux respectifs. Ces relations, une fois projetées sur le lexique arabe, permettraient de construire automatiquement un réseau utile pour certaines applications de traitement de la langue arabe, comme les moteurs de question-réponse, la traduction automatique, les systèmes d'alignement, la recherche d'information, etc. Le chapitre suivant est consacré à la mise en œuvre de cette expérimentation.

Chapitre IV : Rattachement des cliques à WordNet

IV.1 Introduction

Dans ce chapitre, nous allons utiliser nos clusters afin de tester la possibilité de rattacher les sens et les relations de wordnet (WN) à des unités en arabe, et afin de voir si nos eq-sets correspondent peu ou prou à des synsets.

En effet, par construction, les équivalents traductionnels (les lexèmes de langues différentes) se trouvant dans la même clique partagent deux-à-deux un lien d'eq-synonymie, tel qu'il est défini dans un wordnet multilingue comme EuroWordNet (EWN).

Nous faisons l'hypothèse que si tous les équivalents traductionnels d'un même cluster sont associés au même identifiant d'ILI-RECORD dans EWN, et si la fréquence de ces lexèmes est élevée, alors nous pouvons dire que le résultat est fiable.

Par ailleurs, cette approche permet de traiter les lexèmes polysémiques, très fréquents dans notre corpus. Si l'ambigüité n'est pas partagée par tous les lexèmes de langues différentes dans une même clique, cela permet de désambigüiser sémantiquement ce cluster et, par la suite, de rattacher les unités arabes de façon plus précise au sens partagé.

Par exemple, prenons la clique suivante :

(fr-Verb-cesser en-Verb-terminate ar-Verb-OnhY es-Verb-acabar)

Les deux unités *en-Noun-terminate* et *es-Verb-acabar* sont polysémiques mais l'unité française *fr-Verb-cesser* est monosémique (en référence aux WNs et au dictionnaire Larousse), donc l'ambiguïté n'est pas partagée par toutes les langues ce qui facilite le fait de désambigüiser sémantiquement notre clique. En effet, la désambigüisation est ici liée au fait que les unités dans la clique ont des polysémies orthogonales, et non pas parallèles, ce qui aboutit à une intersection réduite.

Il convient, avant de décrire notre méthode de rattachement de cliques au sens et relations d'EWN et la manière dont nous avons désambigüisé nos cliques, de donner une description plus complète de la structure des wordnets d'EWN.

IV.2 Structure des wordnets d'EWN

La Figure 21, empruntée à la documentation technique d'EuroWordNet, donne l'architecture d'ensemble des données dans EuroWordNet. On voit que les réseaux pour chaque langue sont constitués de synsets reliés entre eux par des liens internes à chaque langue (*language dependant links*), et sont également reliés à des unités référencées par un numéro d'ILI-RECORD, ces unités conceptuelles étant considérées comme indépendantes des langues. Ces ILI-RECORDS sont par ailleurs reliés à deux ontologies également indépendantes des langues (*Domain Ontology* et *Top Ontology*).

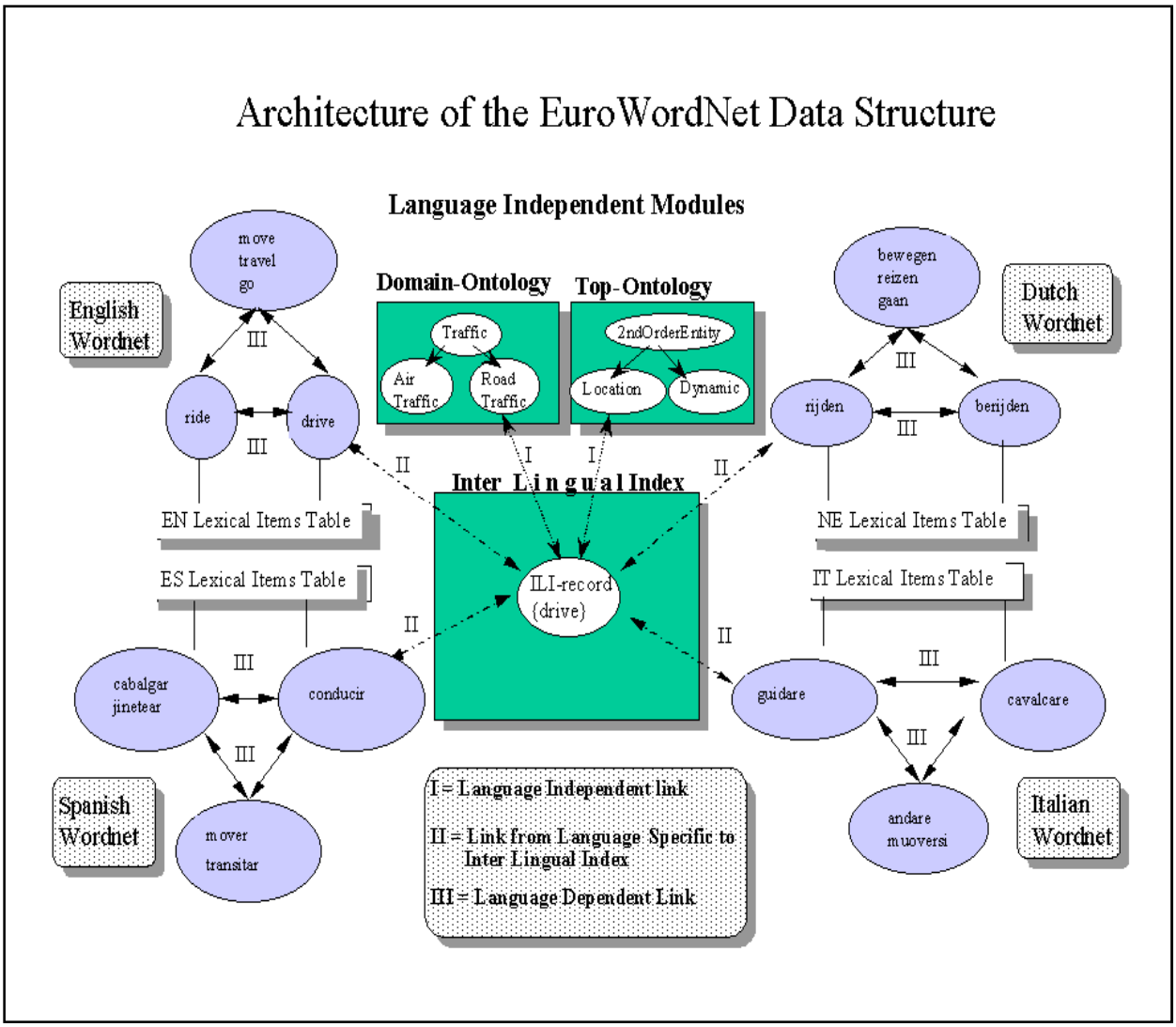

Figure 21 : Architecture d'ensemble d'EuroWordNet (extrait de la documentation technique) [31]

[31] Cf. http://vossen.info/docs/2002/EWNGeneral.pdf, consulté en juin 2012.

Comme nous l'avons déjà vu, la notion centrale permettant de relier les différents wordnets d'EWN est le *concept*, référencé par un ILI-RECORD. Les *synsets* quant à eux correspondent à des significations dépendantes des langues. Comme dans WordNet, ils sont caractérisés par la liste des lexèmes susceptibles d'exprimer cette acception dans la langue concernée, et entretiennent des liens sémantiques avec d'autres synsets de la langue. Les lexèmes d'un même synset sont considérés comme *synonymes*.

Notons que si nous faisons la distinction entre signification (linguistique) et concept (extra-linguistique), celle-ci n'est pas faite par les concepteurs d'EuroWordNet, qui utilisent indifféremment le terme de concept pour les synsets et les ILI-RECORDS. Cette ambiguïté se retrouve dans la construction même du réseau, puisque les concepts des ILI-RECORDS proviennent en fait de WordNet 1.5, et correspondent donc initialement aux synsets de l'anglais. Cette confusion a été l'objet de nombreuses critiques (Rastier, 1991).

La figure ci-dessous montre comment est structurée une entrée pour un wordnet d'EWN :

```
0 @3467@ WORD_MEANING
  1 PART_OF_SPEECH "n"
  1 VARIANTS
    2 LITERAL "academic term"
      3 SENSE 1
      3 EXTERNAL_INFO
        4 SOURCE_ID 1
          5 TEXT_KEY "09143888-n"
    2 LITERAL "school term"
      3 SENSE 1
      3 EXTERNAL_INFO
        4 SOURCE_ID 1
          5 TEXT_KEY "09143888-n"
    2 LITERAL "session"
      3 SENSE 4
      3 EXTERNAL_INFO
        4 SOURCE_ID 1
          5 TEXT_KEY "09143888-n"
```

```
1 INTERNAL_LINKS
  2 RELATION "has_hyperonym"
    3 TARGET_CONCEPT
      4 PART_OF_SPEECH "n"
      4 LITERAL "term"
        5 SENSE 5
  2 RELATION "has_hyponym"
    3 TARGET_CONCEPT
      4 PART_OF_SPEECH "n"
      4 LITERAL "semester"
        5 SENSE 2
  2 RELATION "has_hyponym"
    3 TARGET_CONCEPT
      4 PART_OF_SPEECH "n"
      4 LITERAL "trimester"
        5 SENSE 1
  2 RELATION "has_hyponym"
    3 TARGET_CONCEPT
      4 PART_OF_SPEECH "n"
      4 LITERAL "quarter"
        5 SENSE 9
  2 RELATION "has_holo_part"
    3 TARGET_CONCEPT
      4 PART_OF_SPEECH "n"
      4 LITERAL "academic year"
        5 SENSE 1
1 EQ_LINKS
  2 EQ_RELATION "eq_synonym"
    3 TARGET_ILI
      4 PART_OF_SPEECH "n"
      4 WORDNET_OFFSET 9143888
```

Figure 22 : Extrait d'EWN pour le synset nominal anglais (academic term, school term, session)

Un synset est décrit dans ce dictionnaire par les champs suivants :

- WORD_MEANING :

Dans la base d'EWN, chaque synset commence par l'étiquette WORD_MEANING contenant l'identifiant de ce synset. Une structure de données suit cette étiquette et forme une arborescence de plusieurs sous-champs :

- PART_OF_SPEECH :

Ce champ désigne la catégorie de chaque synset, avec les étiquettes suivantes :

  - a – adjectif
  - n – nom
  - v – verbe
  - b – adverbes
  - pn – noms propres

- VARIANTS :

Ce champ enregistre les différents lexèmes du synset. Il contient plusieurs sous-champs :

  - LITERAL : Le lemme correspondant au lexème
  - SENS : Le numéro permettant de distinguer les diverses acceptions d'un même lexème
  - EXTERNAL_INFO : Contient des informations de fréquence liées à des corpus utilisés dans la confection de la ressource.

- INTERNAL_LINKS :

Ce champ exprime les relations sémantiques (autres que la synonymie) entre un synset et un (ou plusieurs) autre(s) synset(s) de la même langue.

Cet attribut comprend plusieurs sous-champs :

  - RELATION : C'est le type de relation qui relie ce synset à un autre. Tous les types classiques de relations sémantiques sont représentés : hyponymie, hyperonymie, méronymie, holonymie, antonymie, etc. La plupart du temps, il s'agit de relations entre lexèmes au sein d'une même partie du discours (verbe-

verbe, adjectif-adjectif, etc.) mais il est possible de spécifier des relations entre parties du discours différentes.

- TARGET_CONCEPT: Comprend le triplet (PART_OF_SPEECH + LITERAL + SENSE) qui permet d'identifier le synset cible de la relation (à travers un lexème particulier).

- EQ_LINKS :

Ce champ exprime les relations sémantiques entre un synset et un (d') autre(s) synsets d'autres langues. Cet attribut comprend plusieurs sous-attributs :

- EQ_RELATION : C'est le type de relation qui relie ce synset avec une référence d'ILI-RECORDS (en général 'eq-synonym')
- TARGET_ILI : Comprend la catégorie cible (PART_OF_SPEECH) et l'identifiant qui renvoie à un ILI-RECORDS, c'est-à-dire à une référence d'ILI-RECORD qui joue un rôle pivot entre les synsets de différentes langues. On constate que cette référence n'est autre qu'un identifiant de synset de WordNet 1.5 (WORDNET_OFFSET).

Outre les wordnets spécifiques à chaque langue, l'Index Interlingue constitue une structure de réseau pivot, et rassemble les entrées nommées ILI-RECORDS, comme le montre la Figure 23.

```
0 @1@ ILI_RECORD

  1 PART_OF_SPEECH "n"

  1 WORDNET_OFFSET 2403

  1   GLOSS   "something   having   concrete
existence; living or nonliving& 03"

  1 VARIANTS

    2 LITERAL "entity"

      3 SENSE 1



0 @2@ ILI_RECORD

  1 PART_OF_SPEECH "n"

  1 WORDNET_OFFSET 2728
```

```
  1   GLOSS   "any   living   entity&   03
1stOrderEntity Living Natural Origin"

  1 VARIANTS

    2 LITERAL "life form"

      3 SENSE 1

    2 LITERAL "organism"

      3 SENSE 1

    2 LITERAL "being"

      3 SENSE 1

    2 LITERAL "living thing"

      3 SENSE 1
```

Figure 23 : Extrait de l'index interlingue (ILI-RECORDS)

Cet index a été construit à partir de WordNet 1.5, et complété par des entrées supplémentaires, afin d'ajouter des concepts génériques pour mieux remplir son rôle de dictionnaire-pivot, et compléter certaines lacunes.

Voici par exemple, dans la Figure 24, une nouvelle entrée permettant d'ajouter un concept lié à la terminologie informatique :

```
0 ILI_RECORD

  1 PART_OF_SPEECH "n"

  1 ADD_ON_ID 8001

  1 GLOSS "COMPUTER_TERMINOLOGY Redefining
in a child class a method or function
member defined in a parent class."

  1 VARIANTS

    2 LITERAL "overriding"

      3 SENSE 1
```

Figure 24 : Exemple d'enregistrement additionnel dans l'ILI-RECORD

Chaque entrée de cet index commence par l'étiquette (ILI_RECORD) et comprend les champs suivants:

- PART_OF_SPEECH: Catégorie morphosyntaxique (partie du discours) du ILI-RECORD.
- WORDNET_OFFSET: L'identifiant du synset anglais équivalent dans WordNet 1.5. Pour les nouvelles entrées n'apparaissant pas dans WordNet 1.5, on utilise le champ ADD_ON_ID.
- GLOSS: Une définition explicite en anglais.
- VARIANTS : Rassemble les sous-champs :
    - LITERAL : Le lexème anglais qui joue le rôle de pivot.
    - SENSE : Le numéro d'acception de ce lexème
- EQ_RELATION : Indique le type de relation sémantique qui relie ce synset à d'autres WORDNET_OFFSET.

IV.3 Expérimentation

IV.3.1 Algorithme de rattachement des cliques à EWN

Dans le but de rattacher nos cliques à EWN, nous avons enregistré d'abord les informations existant dans les dictionnaires et le fichier ILI-RECORD, dans des tables de hachage (cf. Le script hash-WN4.pl. **Annexe 16**, page 315).

En ce qui concerne le fichier ILI-RECORD, chaque identifiant du synset anglais équivalent (WORDNET_OFFSET) est relié dans une table de hachage à son champ PART_OF_SPEECH, GLOSS et ses WORDNET_OFFSET reliés :

- Clé : WORDNET-OFFSET
- Valeur : PART_OF_SPEECH#GLOSS#liste de WORDNET_OFFSET

Ensuite, pour chaque dictionnaire (pour chaque langue), l'enregistrement est fait comme suit:

- Chaque clé d'un synset est relié à l'identifiant de synset correspondant :
  - Clé : LITTERAL#PART_OF_SPEECH#SENS
  - Valeur : WORD_MEANING
- Chaque identifiant d'un synset (WORD_MEANING) est relié à touts les triplets (LITTERAL# PART_OF_SPEECH#SENS) correspondants.
  - Clé : WORD_MEANING
  - Valeur : liste de LITTERAL#PART_OF_SPEECH#SENS
- Chaque identifiant d'un synset (WORD_MEANING) est relié à touts les synsets anglais équivalents (WORDNET_OFFSET)
  - Clé : WORD_MEANING
  - Valeur : liste de WORDNET_OFFSET
- Chaque identifiant d'un synset, associé à un type de relation sémantique (WORD_MEANING#RELATION) est relié à l'identifiant des synsets cibles de cette relation (WORD_MEANING).
  - Clé : WORD_MEANING#RELATION

- Valeur : liste de WORD_MEANING

Ces tables de hachage ont servi au rattachement de nos cliques à l'EWN :

Pour chaque unité (*langue-catégorie-lemme*) simple ou composée d'une clique (ou d'un cluster issu de la fusion d'une ou plusieurs cliques) :

> On cherche dans le dictionnaire de la langue de cette unité les synsets qui ont ce lemme comme LITTERAL et cette catégorie comme PART-OF-SPEECH.
>
> Pour chaque synset trouvé (correspondant à un certain numéro de SENS), on cherche le ou les équivalent(s) de ce synset (WORDNET_OFFSET). Pour chaque équivalent trouvé, on enregistre l'ILI-RECORD correspondant et on récupère sa définition (GLOSS). On obtient donc, pour chaque unité du cluster, un ensemble d'ILI-RECORDS qui correspond à la liste des concepts qui lui sont potentiellement rattachés.

Dans un deuxième temps, on compare les identifiants ILI-RECORDS de chaque unité de la même clique :

> Si toutes ses unités partagent un unique ILI-RECORD alors nous considérons la clique (ou le cluster) comme valide et désambigüisé. Nous nous basons sur l'hypothèse que toutes les unités d'une même clique non ambiguë doivent être reliées à au moins un synset partageant un lien d'équivalence avec le même ILI-RECORD.
>
> Sinon, si toutes ses unités partagent des liens d'équivalence avec plusieurs ILI-RECORDS, nous considérons qu'une ambigüité est partagée par toutes ces unités et que la clique est non-désambigüisée.
>
> Enfin, en l'absence d'ILI-RECORDS communs, la clique apparaît comme invalide : ce peut-être un indice de bruit (une unité non équivalente s'étant agrégée par erreur dans l'ensemble).

Mais ce dernier cas est relativement fréquent, et n'est pas toujours lié au bruit : Il peut s'agir d'une relation absente liée à la couverture incomplète des dictionnaires. Afin de

palier ce problème et de rattacher les unités arabes aux sens partagés par leurs cliques désambigüisées, nous adoptons plusieurs principes que nous expliquons dans la suite.

IV.3.2 Rattachement des sous-sens et des relations de WN à des unités arabes

Après l'application de notre méthode de rattachement des unités de même clique (ou cluster) au sens partagé d'une manière précise, nous procédons maintenant à étendre ce rattachement vers les unités arabes.

Notre objectif est la projection des sous-sens et des relations sémantiques vers l'arabe, ce qui vise à offrir la possibilité de structurer automatiquement les sens des unités arabes par l'intermédiaire de WN, afin de construire une ressource originale pour l'arabe à partir de nos corpus.

Mais suivant notre résultat de rattachement (présenté plus loin), nous trouvons que certaines cliques obtenues sont 1.valides (leur unités partagent un seul ILI-RECORD) 2.invalides (leur unités ne partagent pas le même ILI-RECORD) ou 3.valides partiellement (certaines unités de la clique partagent un sens plus général ou plus spécifique que l'ILI-RECORD partagé par le reste d'unités). Ces résultats différents nous conduisent à définir certains principes qui prennent en compte les différents cas obtenus et qui nous permettent d'étendre le rattachement des sous-sens ou des relations vers l'arabe.

Plusieurs principes sont appliqués dans cette direction :

IV.3.2.1 Principe de clôture transitive intra-clique

Ce principe permet de compléter les liens manquants vers les unités arabes dans une clique. En effet, si toutes les unités d'une clique (sauf bien entendu l'unité arabe), partagent un lien ILI-RECORD dans EWN, cette clôture transitive intra-clique permet de rattacher l'unité arabe à cet ILI-RECORD commun et donc d'avoir un lien d'eq-synonymie entre l'unité arabe et les autres unités. Autrement dit, si tous les éléments non arabes d'une clique partagent un et un seul ILI-RECORD alors on suppose que le sens correspondant peut être étendu à l'élément (ou aux éléments) arabe(s) de la clique. Mais dans le cas où les

éléments non arabes partagent plusieurs ILI-RECORDS, on traite la clique comme ambigüe et on n'applique pas ce principe.

Afin de clarifier le rattachement de sous-sens vers l'arabe, prenons l'exemple de la clique précédente:

(fr-Verb-cesser en-Verb-terminate ar-Verb-OnhY es-Verb-acabar)

Suivant le principe de clôture transitive intra-clique, l'unité arabe *ar-Verb-OnhY* (أنهى) peut être rattachée à l'ILI-RECORD suivant (nous indiquons seulement la glose) : */The war ended after three months""\have an end, in a temporal or spatial sense: "My property ends by the bushes"; "The symphony ends in a pianissimo"& 2ndOrderEntity 42 Property SituationType Static"* /.parce que cet ILI-RECORDS est partagé par toutes les autres unités de cette clique.

Afin de vérifier si ce lien ILI-RECORD partagé correspond bien à un des sens de l'unité arabe *ar-Verb-OnhY*, nous nous référons à un des dictionnaires arabes les plus réputés, *Alwaseet*. Nous avons utilisé ce dictionnaire puisqu'à notre connaissance, il n'existe pas actuellement de WN arabe complet et librement disponible.

Le sens identifié par notre clique sera considéré comme valide et pertinent pour rattacher le lemme arabe (composé ou simple) à l'ILI-RECORD partagé si un des sens de cette unité arabe dans le dictionnaire *Alwaseet* correspond bien à l'ILI-RECORD partagé.

Pour l'exemple précédent, nous trouvons qu'un seul des sens de l'unité arabe *ar-Verb-OnhY* proposé par *Alwaseet* est : */arrêter de faire quelque chose/atteindre la fin de quelque chose/*, ce qui correspond bien à l'ILI-RECORD partagé et qui valide donc le rattachement de l'unité arabe.

IV.3.2.2 Principe de clôture transitive inter-clique

Maintenant nous voulons tester si nous pouvons projeter aussi *les relations sémantiques* partagées vers les unités arabes :

En vue de projeter ces relations vers les unités arabes, nous appliquons ici un principe de *clôture transitive inter-clique* que nous formulons ainsi : Si deux cliques sont chacune

rattachées à au plus un ILI-RECORD, et si pour une langue donnée il existe une relation sémantique entre deux unités appartenant à ces deux cliques, pour des acceptions liées aux ILI-RECORDS retenus, alors la relation peut être étendue pour les unités arabes contenues dans ces cliques, sauf si une relation contradictoire peut-être inférée à partir d'une autre langue.

Nous pensons que cette méthode de projection des relations sémantiques vers l'arabe permet de structurer automatiquement les sens des unités arabes par l'intermédiaire d'EWN.

A titre d'illustration, prenons l'exemple suivant :

> *Clique 1 pour l'unité arabe ar-Noun-qsm (قسم) : (ar-Noun-qsm fr-Noun-fragment en-Noun-snippet es-Noun-recorte).*
>
> *Clique 2 pour l'unité arabe ar-Noun- Hsp (حصة) : (ar-Noun- Hsp fr-Noun-morceau es-Noun-pedazo en-Noun-piece).*

Toutes les unités de la clique 1 partagent le lien ILI-RECORD suivant: */a small piece of anything (especially a piece that has been snipped off)& 03 06 1stOrderEntity Artifact Form Object Origin Part/.*

De plus, les unités de la clique 2 partagent le lien ILI-RECORD suivant: */an instance of some kind; "it was a nice piece of work" or "he had a bit of good luck"& 03 11 2ndOrderEntity BoundedEvent Dynamic SituationType/.*

Suivant le principe de *clôture transitive intra-clique*, le lien ILI-RECORD partagé par la clique 1 peut être étendu à l'unité arabe *qsm* (قسم) et le lien ILI-RECORD partagé par la clique 2 peut être étendu à l'unité arabe *Hsp*(قسم).

En nous référant au dictionnaire *Alwaseet*, nous trouvons que les deux unités arabes *qsm* et *Hsp* portent en effet le sens correspondant au sens retenu pour les cliques :

> *qsm→ /une partie de quelque chose/*
>
> *Hsp→/un morceau, une partie ou une période temporelle/*

En outre, les unités *fragment*, *snippet* et *recorte* sont reliées dans leurs WNs avec les unités de même langue de la clique 2 suivant la relation sémantique 'has_hyperonym' :

fragment 'has_hyperonym' *morceau*

recorte 'has_hyperonym' *pedazo*

snippet 'has_hyperonym' *piece*

Ces relations précédentes nous indiquent alors qu'il est possible qu'une relation d'hyperonymie existe aussi entre les deux unités arabes *qsm* et *Hsp*.

Par conséquent, en appliquant le principe de clôture inter-clique, on peut dire que la relation est projetée vers l'arabe comme suit :

qsm 'has_hyperonym' *Hsp*

Maintenant imaginons un cas de projection de relations où une clique est reliée à plusieurs ILI-RECORDS dans EWN, et donc est ambiguë, comme dans l'exemple suivant :

Clique 1 extraite pour l'unité arabe ar-Noun-mwDwE (موضوع) : (ar-Noun-mwDwE fr-Noun-sujet en-Noun-content en-Noun-subject es-Noun-tema).

Toutes les unités précédentes partagent le lien ILI-RECORD : */topic of subject matte/*

En outre, *subject*, *sujet* et *tema* dans la clique 1 sont reliées avec les unités de même langue de la clique 2 suivant la relation sémantique *est* 'synonyme de' :

Clique 2 : (ar-Noun-mDmwn es-Noun-contenido en-Noun-content fr-Noun-contenu)

Subject 'est synonyme de' *content*

Sujet 'est synonyme de' *contenu*

Tema 'est synonyme de' *contenido*

Or la clique 1 est sémantiquement plus large que la deuxième puisque selon EWN, la clique 1 est reliée à d'autre ILI-RECORDS comme par exemple: */something (a person or*

object or scene) selected by an artist or photographer for graphic representation/, ainsi la clique 1 est ambiguë et donc suivant notre principe de clôture transitive inter-clique la projection de la relation de synonymie est non valide.

Mais cet exemple suggère alors qu'il faut préciser que la relation de projection est entre des cliques associées à un certain ILI-RECORD. On pourrait donc étendre la relation pour la combinaison clique1-ILI1 et non clique1-ILI2.

Ainsi, on pourrait projeter cette relation de synonymie (que l'unité arabe *ar-Noun-mwDwE* 'est synonyme de' mDmwn) seulement lorsque *ar-Noun-mwDwE* est relié à l'ILI-RECORD suivant : */topic of subject matter/.*

On peut en conclure que la contrainte d'avoir une clique non ambiguë est inutile, puisque de toute façon on rattache les relations à des unités désambigüisées, c'est-à-dire associées à un certain ILI-RECORD.

IV.4 Résultats du rattachement

Dans ce travail, nous avons testé seulement les cliques contenant des unités de deux catégories (Noms et Verbes) puisque le FWN (French wordnet) ne comporte ni adjectifs ni adverbes. Nous n'avons pas traité non plus les unités pour lesquelles les catégories Noms et Verbes n'étaient pas complètement désambiguïsées ex. *(Noun/Adj),(V/Noun/Adj)*…etc.

Résultats des clusters pour les noms français :

Voici les statistiques concernant la partie française du corpus :

Catégorie	Nombre total de types	Nombre de type de mots non-traités[32]	Nombre d'occurrences des mots traités
Noms (simples)	6 226	3 635	794 818
Noms (composés)	807	0	16 147
Verbes (simples)	1 484	385	220 406
Verbes (composés)	263	0	7 543
Total	**8 781**	**4 009**	**1 201 043**

Tableau 10 : Statistiques concernant la partie française du corpus

Le nombre total de types français (noms simple + noms composés) alignés avec les autres langues (ar, es et en) dans notre corpus est: 7 033.

Le nombre total de types français (verbes simple + verbes composés) alignés avec les autres langues (ar, es et en) dans notre corpus est: 1 747.

Nous avons appliqué notre approche sur un échantillon de 100 clusters obtenus pour les 100 noms les plus fréquents dans notre corpus (y compris 90 noms polysémiques). Chaque cluster est obtenu pour un nom différent. Chaque nom a au moins 1500 occurrences dans le

[32] Tous les mots étiquetés par Treetagger n'ayant pas comme catégories Noun ou Verbe sont écartés de l'évaluation. Les mots non traités sont donc les chiffres, les noms propres, les abréviations, les mots mal étiquetés, les mots outils,…etc.

corpus (cf. **Annexe 2**, page 198). La sélection de ces noms est faite en se basant sur le résultat de notre script (cf. Le script stock-info-v4.pl. **Annexe 13**, page 292).

Le tableau ci-dessous indique le nombre de clusters traités pour les 100 noms simples français : le nombre des clusters monosémiques (i.e. ne contenant que des unités rattachées à un seul ILI-RECORD), le nombre de cluster désambiguïsées et le nombre de clusters non-désambiguïsées (les unités étant rattachées à plusieurs ILI-RECORDS communs ou à aucun ILI-RECORD).

clusters traités	clusters monosémiques	clusters désambigüisées	clusters non désambigüisées
100	3	53	44

Tableau 11: Evaluation du nombre de clusters désambiguïsés pour les noms simples en français

Résultats des clusters des verbes français :

Nous avons appliqué notre approche sur 100 verbes (y compris 79 verbes polysémiques), ce sont les verbes les plus fréquents dans le corpus. Chaque verbe a au moins 300 occurrences (cf. **Annexe 3**, page 204).

Le tableau ci-dessous indique le nombre de clusters traités (un échantillon de 100 clusters obtenus, chaque cluster est obtenu pour un verbe différent) pour les 100 verbes français : le nombre des clusters monosémiques (i.e. ne contenant que des unités rattachées à un seul ILI-RECORDS), le nombre de clusters désambiguïsés et le nombre de clusters non-désambigüisées (les unités étant rattachées à plusieurs ILI-RECORDS communs ou à aucun ILI-RECORD).

clusters traitées	clusters monosémiques	clusters désambigüisées	clusters non désambigüisées
100	2	27	71

Tableau 12: Evaluation du nombre de clusters désambiguïsés pour les verbes simples en français

Résultats pour des lexèmes composés :

Prenons maintenant quelques exemples des noms composés étudiés :

Noms français (lexèmes composés)	Nombre d'occurrences
agent de police (polysémique)	10
vice-président	192
bien-être (polysémique)	144

Tableau 13 : Le nombre d'occurrences des noms composés français étudiés

Parmi ces cliques, 2 sur 3 ont été désambiguïsées :

(Fr-Noun-agent de police en-Noun-policeman es-Noun-policía ar-Noun-$rTy)→ */a member of a police force& 03 18 1stOrderEntity Form Function Human Living Natural Object Origin/*

(fr-Noun-bien-être en-Noun-welfare en-Noun-well-being es-Noun-bienestar Ar-Noun-rfA)→ */the state of being happy and healthy and prosperous& 03 26 2ndOrderEntity Condition Property SituationType Static/*

Mais pour l'unité *vice-président*: nous avons obtenu la clique suivante: (fr-Noun-vice-président en-Noun-vice-president es-Noun-vicepresidente ar-Noun-nA}b r}ys).

Les deux unités (fr-es) partagent l'ILI-RECORD suivant: */one ranking below or serving in the place of a chairman& 03 18 1stOrderEntity Form Function Human Living Natural Object Occupation Origin)/*

Mais l'unité anglaise *vice-president* n'existe pas dans WN anglais.

Examinons maintenant quelques exemples des verbes composés étudiés :

Verbes français (lexèmes composés)	Nombre d'occurrences
faire remarquer (monosémique)	27

faire savoir (polysémique)	3
faire connaître (polysémique)	4

Tableau 14 : Le nombre d'occurrences des verbes composés français étudiés

Parmi les cliques obtenues pour ces trois unités, aucune n'a été désambiguïsée. On ne trouve pas de liens ILI-RECORDS partagés par les unités de ces dernières cliques pourtant les trois unités existent dans EWN.

IV.5 Evaluation qualitative des résultats

IV.5.1 Validité sémantique des clusters

Dans cette partie de l'évaluation, nous cherchons à examiner la validité au plan sémantique des clusters obtenus.

Plusieurs cas de figure ont été rencontrés, que nous listons ci-dessous.

Cas n°1 : identification correcte de plusieurs acceptions

D'abord, nous avons obtenu des clusters qui peuvent permettre d'identifier différents sens pour une même unité arabe. Prenons l'exemple suivant qui illustre ce point :

Nous avons obtenu deux clusters pour le lemme arabe *Elm* (علم):

1. Cluster 1: {(ar-Noun-AlElwm ar-Noun-AlElm en-Noun-science fr-Noun-science es-Noun-ciencia)}.

Il se trouve que les noms arabes *AlElwm, AlElm* (العلوم، العلم) sont des formes fléchies pour le lemme *Elm* (علم).

Par ailleurs, les trois unités *en-Noun-science, es-Noun-ciencia* et *fr-Noun-science*, en se référant aux WNs de chaque langue pris indépendamment, sont polysémiques. Mais toutes les unités (fr-en-es) de ce cluster partagent le seul lien ILI-RECORD suivant: */a particular branch of scientific knowledge/*

Un des sens de l'unité arabe *ar-Noun-Elm*(علم) mentionné dans le dictionnaire Alwaseet est : */un groupe de connaissances scientifiques dans un domaine particulier/*[33] ce que correspond bien au lien ILI-RECORDS partagé par les autres unités (fr-en-es) existant dans le même cluster.

Donc le sens identifié par notre clique est pertinent et l'unité *ar-Noun-Elm* peut être rattachée au sens : */a particular branch of scientific knowledge/*.

[33] Les sens des unités arabes indiqués ici sont une traduction personnelle de l'arabe (la langue de dictionnaire Alwaseet) vers le français.

2. Cluster 2 pour le même lemme arabe *Elm*: {(ar-Noun-Elm ar-Noun-tElm fr-Noun-apprentissage en-Noun-learning es-Noun-aprendizaje)}.

Toutes les unités arabes du cluster précédent sont des formes fléchies du même lemme *Elm* :(*ar-Noun-Elm ar-Noun-tElm*).

Ensuite, l'ILI-RECORD commun des trois unités (en-es-fr) est : */the cognitive process of acquiring skill or knowledge; "the child's acquisition of language"& 03 09 2ndOrderEntity Agentive Cause Dynamic Experience Mental Property SituationType Static/*.

Cet ILI-RECORD est assez proche de l'un des sens de l'unité arabe *Elm* dans le dictionnaire Alwaseet qui est lié aussi à la notion d'apprentissage: */l'acquisition et la connaissance de la vérité des choses/*.

1. Le sens identifié par ce cluster semble donc pertinent pour rattacher les trois unités arabes à un deuxième sens. Ainsi, nous avons obtenu, par nos deux clusters précédents, deux sens différents pour l'unité arabe *Elm* :

1. /A particular branch of scientific knowledge/

2. /the cognitive process of acquiring skill or knowledge; "the child's acquisition of language"& 03 09 2ndOrderEntity Agentive Cause Dynamic Experience Mental Property SituationType Static/

Notons que dans certains cas les sens représentés par les clusters sont incomplets, puisque dans la langue il y a d'autres acceptions très communes qui ne sont pas représentées, du fait des limitations de nos ressources, wordnets et corpus :

Cas n°2 : insuffisance de couverture des WNs

a. absence d'un lexème dans les WNs

C'est le cas, par exemple, des lexèmes suivants :

Les verbes français qui se trouvent dans nos cliques, pourtant communes, n'apparaissent pas dans FWN : *adjoindre, approprier, figurer, spécialiser...*

Idem pour les verbes espagnols : *adjuntar, coordinar, promover...*

b. absence de catégories entières dans un WN (comme les adjectifs du FWN)

Par exemple, dans la clique suivante, le lexème écologique n'est lié à aucun ILI-RECORD : (*ar-Adj-by}p fr-Noun-environnement en-Adj-environmental es-Adj-ambiental* ***fr-Adj-écologique***).

Une autre limite découlant de WN est l'impossibilité de rattacher une unité d'une catégorie différente à un ILI-RECORD partagé puisque WN ne contient pas des liens transcatégoriels.

c. Absence d'un sens dans les WNs

Prenons l'exemple suivant qui montre une des limitations concernant nos WNs :

Le cluster obtenu pour le mot arabe *flsfp* (فلسفة) est : {*(ar-Noun- flsfp es-Noun-filosofía fr-Noun-philosophie en-Noun-philosophy)}*

Notons que les deux unités *en-Noun-philosophy* et *es-Noun-filosofía*, se référant au WN de chaque langue, sont polysémiques alors que l'unité *fr-Noun-philosophie* serait monosémique (d'après FWN). L'ambigüité n'étant pas partagée par toutes les langues, toutes les unités (fr, en et es) de ce cluster ne partagent que le lien ILI-RECORD suivant : */the rational investigation of questions about existence and knowledge and ethics/.*

Mais cela est simplement dû à une lacune du FWN (French wordnet) puisque l'on trouve dans la langue française d'autres sens pour le mot *philosophie* (p.ex. */sagesse/*). Ainsi il faut distinguer entre deux types de désambigüisation des cliques :

- Une vraie désambigüisation liée au fait que les unités dans la clique ont des polysémies orthogonales, et non pas parallèles, ce que aboutit à une intersection réduite (comme p.ex. *en-Noun-record, fr-Noun-disque*).
- Une désambigüisation artificielle liée au fait qu'une des ressources est incomplète, et aboutit à une intersection réduite uniquement par l'absence de certains sens (comme dans notre exemple de *philosophie).*

d. Spécificité du découpage des sens dans les WNs

Dans certains cas, assez marginaux, il se peut que l'hypothèse de centralité des cliques ne soit pas vérifiée. Par exemple chaque couple d'unités dans la clique suivante (en-Noun-fund fr-Noun-fonds es-Noun-fondo) partage un lien ILI-RECORD, mais aucun ILI-RECORD n'est partagé par les trois unités considérées ensemble :

- *en-Noun-fund* ET *fr-Noun-fonds*: */a reserve of money set aside for some purpose& 03 1stOrderEntity 21 Artifact Function MoneyRepresentation Origin Possession/.*
- *en-Noun-fund* ET *es-Noun-fondo*: */a supply of something available for future use& 03 1stOrderEntity 21 Function Possession/.*
- *fr-Noun-fonds* ET *es-Noun-fondo*: */assets in the form of money& 03 1stOrderEntity 21 Function Possession/*

On constate que les sens 1 et 3 sont cependant assez voisins, et qu'avec un découpage un peu moins spécifique des sens on pourrait cependant les identifier (tout découpage comportant une certaine part d'arbitraire).

e. rattachement à un sens trop générique (*top-ontology*)

Dans certains cas, l'ILI-RECORD commun est beaucoup trop générique pour donner une indication utile sur l'acception commune des unités. Par exemple, la clique : (es-Noun-disposición en-Noun-provision fr-Noun-disposition) est liée au ILI-RECORD suivant de la *top-ontology : /& 03 10 2ndOrderEntity 3rdOrderEntity Agentive BoundedEvent Cause Communication Dynamic Mental Purpose Relation Situation Type Social Static/.*

En dehors de ces cas évidents de lacune, il existe deux autres cas de non-résolution des ambiguïtés, liées aux faits suivants:

Cas n°3 : insuffisance de couverture du corpus

Prenons les deux clusters obtenus pour le mot arabe *mAdp* (مادة):

Cluster1: {(ar-Noun-AlmAdp ar-Noun-AlmwAd ar-Noun-AlmAdtyn ar-Noun-mAdp fr-Noun-article en-Noun-article es-Noun-artículo)}.

Les trois unités *fr-Noun-article en-Noun-article es-Noun-artículo* partagent l'ILI-RECORD suivant: */one of a class of artifacts; "an article of clothing"/.*

Cluster 2: (ar-Noun-mwAd fr-Noun-matériaux en-Noun-material es-Noun-material).

Les trois unités *fr-Noun-matériaux en-Noun-material es-Noun-material* partagent l'ILI-RECORD suivant: */Information (data or ideas or observations) that can be reworked into a finished form;"the archives provided rich material for a definitive biography"/.*

Mais il manque d'autres sens pour l'unité *ar-Noun-mAdp* qui ne sont pas représentés du fait des limitations du corpus comme le sens: */chose physique, corporelle, par opposition à l'esprit/*. La taille trop réduite du corpus n'a pas permis à notre méthode d'extraire un second cluster contenant d'autres équivalents susceptibles de se référer à cette autre acception.

C'est pour cette raison que l'utilisation d'un corpus multilingue de grande dimension, et suffisamment varié, est nécessaire, si l'on veut obtenir une couverture suffisante de la langue générale. En effet, il faut de très nombreuses occurrences pour donner la possibilité de représenter la plupart des sens possibles pour un mot donné. De plus, une grande variété dans les équivalents de traduction démultiplie les possibilités de désambigüisation, comme l'illustre la suite de notre étude.

Cas n°4 : ambiguïtés dues à des polysémies parallèles

Certaines ambigüités sémantiques sont partagées par plusieurs langues considérées, c.à.d. qu'on trouve des sens différents partagés par plusieurs unités d'une même clique (ou cluster). Les cliques ambigües qui en découlent empêchent parfois le rattachement d'un unique sens aux unités arabes. Voici un exemple des cas des cliques ambiguës:

Clique : (en-Noun-topic fr-Noun-sujet ar-Noun-mwDwE en-Noun-subject es-Noun-tema)

L'unité *en-Noun-subject* est polysémique et elle partage avec *fr-Noun-sujet* et *es-Noun-tema* plusieurs ILI-RECORDS:

1. */some situation or event that is thought about; "he kept drifting off the topic"; "it is a matter for the police"/*

2. */something (a person or object or scene) selected by an artist or photographer for graphic representation/*

On peut dire que l'ambigüité est partagée par les trois langues. En outre, en ce que concerne les deux unités *en-Noun-topic* et *en-Noun-subject* du fait de l'hypothèse de centralité, pour une même langue on considère l'union : deux termes au sein du même eq-set ne sont pas nécessairement synonymes, si la clique est ambiguë. Il se peut que chacun corresponde à des sens différents liés à cette clique.

Ce dernier cas, assez courant, n'est pas lié aux ressources mais aux langues impliquées. On peut supposer que plus le nombre de langues est grand, plus ces cas devraient être rares, car la probabilité d'obtenir des polysémie parallèles diminue avec la variété des langues mises en jeu.

Cas n°5 : bruit lié à la non reconnaissance d'unités polylexicales dans les équivalents traductionnels

Par ailleurs, quelques clusters obtenus montrent d'autres types de problèmes entravant la désambigüisation:

- Prenons le cluster suivant :{(ar-Noun-lgp fr-Noun-langue en-Noun-language es-Noun-dioma fr-Noun-linguistique)}

Les deux unités *fr-Noun-langue* et *en-Noun-language* qui sont polysémiques, en se référant à leur WN, partagent les ILI-RECORDS suivants :

1. /a systematic means of communicating by the use of sounds or conventional symbols/

2. /communication by word of mouth; "his speech was slurred" or "he used harsh language"/

Tandis que l'unité *es-Noun-dioma* qui est polysémique a comme ILI-RECORDS:

1. */a human written or spoken language used by a community; opposed to e.g. a computer language/*
2. */a characteristic language of an occupational group/*

Mais l'unité française *fr-Noun-linguistique* qui est monosémique et qui appartient à un synset totalement différent de celui de *fr-Noun-langue* a comme ILI-RECORD : */the scientific study of language/.*

On peut penser que cette différence est due au bruit lié au découpage des unités (*language study→ linguistique*) ou à l'ambigüité morphologique (p.ex. *language research→recherche linguistique*, l'adjectif *linguistique* étant par erreur étiqueté comme un nom).

Cas n°6 : ambiguïtés liées à une sur-clusterisation

Un autre problème qui peut être rencontré est la clusterisation des deux cliques qui devraient rester séparées, comme dans le cluster suivant pour le nom français *droit*:

{(fr-Noun-droit en-Noun-right es-Noun-derecho ar-Noun-Hq) (fr-Noun-droit en-Noun-Law es-Noun-derecho ar-Noun-qAnwn)}

Deux sous-sens existent dans ce cluster :

1. (fr-Noun-droit en-Noun-right es-Noun-derecho ar-Noun-hq)→ */ an abstract idea of that which is due to a person or governmental body by law or tradition or nature: "they are endowed by their Creator with certain unalienable Rights, that among these are Life, Liberty and the pursuit of Happiness"; "Certain rights can never be granted to the government but must be kept in the hands of the people"- Eleanor Roosevelt; "it is his right to say what he pleases"& 03 07 09 2ndOrderEntity 3rdOrderEntity Dynamic Experience Mental Property SituationType Social Static/.*
2. (fr-Noun-droit en-Noun-law es-Noun-derecho ar-Noun-qAnwn)→ */the collection of rules imposed by authority; "civilization presupposes respect for the law"& 03 14 Group/.*

En somme, certains problèmes qui ne sont pas directement liés à notre méthode, mais plutôt aux limitations des ressources mise en œuvre, peuvent faire échouer, dans quelques cas, notre méthode d'identification du sens au sein d'un cluster (ou d'une clique).

IV.6 Evaluation du rattachement à EWN pour les langues présentes dans le réseau (en-es-fr)

Il se peut qu'un cluster soit bien formé, valide sémantiquement, désambiguïsé et pas rattaché à EWN pour certaines questions de lacune (voir tableau 6 et tableau 7). Par ailleurs, on pourrait avoir des clusters rattachés à EWN mais ambigus à cause des certains problèmes indiqués auparavant dans la partie (IV.5.1).

Le tableau ci-dessous montre les statistiques liées à la typologie des cas de non-rattachement pour les noms:

Cause du non-rattachement	%
Cas 2. Insuffisance de couverture des WNs	18%
Cas 3, Cas 5, Cas 6. Pas d'ILI-RECORD commun à toutes les unités	9%
Cas 4. Ambiguïtés dues à des polysémies parallèles	17%
Total	44%

Tableau 15 : Répartition en % des cas de non-rattachement pour les noms

Quant aux verbes, nous avons eu des résultats très faibles.

Le tableau ci-dessous montre les statistiques liées à la typologie des cas de non-rattachement pour les verbes :

Cause du non-rattachement	%
Cas 2. Insuffisance de couverture des WNs	24%
Cas 3, Cas 5, Cas 6. Pas d'ILI-RECORD commun à toutes les unités	30%
Cas 4. Ambiguïtés dues à des polysémies parallèles	17%
Total	71%

Tableau 16: Répartition en % des cas de non-rattachement pour les verbes

Ainsi, pour les verbes nous avons obtenu des clusters correctement désambiguïsées et rattachées à EWN dans seulement 29% des cas. La clique suivante est un exemple de clique correctement désambiguïsée et rattachée : (fr-Verb-participer es-Verb-participar en-

Verb-participate ar-Verb-$Ark) où toutes les unités partagent le lien ILI-RECORD : */be involved in something& 2ndOrderEntity 41 Agentive Cause Dynamic SituationType Social/.*

IV.7 Evaluation manuelle des sens rattachés aux unités arabes

Jusqu'à présent nous avons examiné, d'un point de vue quantitatif, seulement les possibilités de rattachement des clusters pour les trois langues en-es-fr. Reste à effectuer une validation manuelle des unités arabes pour les clusters qui ont été correctement rattachés à des synsets d'EWN et désambiguïsés. Pour les noms et les verbes, les résultats figurent respectivement dans le Tableau 3 (cf. **Annexe 5**, page 231 et **Annexe 6**, page 239) et le Tableau 4 (cf. **Annexe 7**, page 240). Notons que ces tableaux ne contiennent que les lemmes de toutes les formes fléchies arabes trouvées dans les clusters obtenus.

Le nombre total de lemmes nominaux arabes existant dans les 56 clusters (construits à partir de noms simples français), désambigüisés au regard des trois langues (en-fr-es), est de 74.

Nous avons obtenu 9 clusters avec 0 lemmes arabes, 27 clusters avec un seul lemme arabe, 20 clusters avec 2 lemmes arabes ou plus.

Le nombre total de lemmes arabes existant dans les 29 clusters (construits à partir de verbes simples français), désambigüisés au regard de trois équivalents (trois langues), est de 37.

Nous avons obtenu 3 clusters avec 0 lemmes arabes, 18 clusters avec un seul lemme arabe, 8 clusters avec 2 lemmes arabes ou plus.

En ce que concerne les trois unités arabes trouvées dans les clusters construits à partir des noms français composés, leurs sens sont validés complètement par le dictionnaire *Alwaseet*.

Dans la validation du sens nous distinguons trois cas de figure :

1. **Exactitude** : L'unité en arabe est bien subsumée par le sens de l'ILI-RECORD. Pour ce cas de figure, le taux de correction des unités arabes validées complètement au sein des cliques de noms français est de 59/74. Le taux des unités arabes validées au sein des cliques de verbes français est de 21/37.

   Par exemple, prenons la clique suivante : (fr-Verb-autoriser en-Verb-authorize es-Verb-autorizar ar-Verb-smH en-Verb-permit es-Verb-permitir).

Toutes les unités précédentes (sauf l'arabe) partagent le lien ILI-RECORD suivant: */grant authorization or clearance for& 2ndOrderEntity 32 Agentive BoundedEvent Cause Communication Dynamic Purpose SituationType Social/.*

L'unité arabe *ar-Verb-smH* a comme sens dans le dictionnaire *Alwaseet: /Donner à quelqu'un le droit de faire quelque chose/*.

On peut considérer ces deux sens comme suffisamment similaires pour valider la clique.

2. **Spécificité** : Si le sens de l'ILI-RECORD correspond à une catégorie plus générale ou plus spécifique (c'est ce qu'on a dans certains cas), on peut parler de validité partielle. Dans ce cas :
   Le taux des unités arabes partiellement valides au sein de clusters de noms français est de 8/74.

   Le taux des unités arabes partiellement valides au sein de clusters de verbes français est de 6/37.

   Prenons l'exemple de la clique suivante : (Fr-Noun-mission en-Noun-mission es-Noun-misión ar-Noun-mhmp).

   Toutes les unités précédentes (sauf l'arabe) partagent le lien ILI-RECORD suivant : */ an operation that is assigned by a higher headquarters; "the planes were on a bombing mission"& 03 04 2ndOrderEntity Agentive BoundedEvent Cause Dynamic Purpose SituationType Social /.*

   L'unité arabe *ar-Noun-mhmp* (مهمة) a comme sens dans le dictionnaire Alwaseet: */opération à faire/.*

   On voit bien que le sens arabe est plus général que le sens partagé.

3. **Sens invalides** : Le dernier cas de figure correspond à des sens non validés, car trop éloignés :
   Le taux des unités arabes incorrectes au sein des cliques de noms français est de 7/74 (≈ 9%).

Le taux des unités arabes incorrectes au sein des cliques de verbes français est de 10/37(≈27%).

L'examen minutieux des cas invalides nous révèlent plusieurs types de cas :

1. Le sens arabe est complètement différent de celui des autres unités, car l'unité a été regroupée par erreur lors de la phase de clusterisation.

   Considérons l'exemple du cluster suivant :

{(fr-Noun-commission fr-Noun-comité en-Noun-commission en-Noun-committee es-Noun-comisión ar-Noun-ljnp)

(fr-Noun-commission en-Noun-commission es-Noun-comisión ar-Noun-ArtkAb)}

   Les deux cliques de ce cluster portent des sens différents. Toutes les unités de la première clique partagent le lien ILI-RECORDS suivant : */a special group delegated to consider some matter& 03 14 1stOrderEntity Function Group Human Living Natural Origin/.*

   L'unité arabe *ar-Noun-ljnp* partage aussi le sens précédent, tandis que l'unité arabe *ar-Noun-ArtkAb* partage avec les unités de la deuxième clique le sens suivant : */the act of committing a crime/.*

   Mais en raison de nos critères de clusterisation l'unité arabe *ar-Noun-ArtkAb* a été mal reliée avec le 1er sens */a special group delegated to consider some matter& 03 14 1stOrderEntity Function Group Human Living Natural Origin/.*

2. Le mot arabe est un lexème composé, mais un seul lexème se trouve appartenir à la clique, à cause d'une mauvaise tokenisation :

   Exemple :

Prenons la clique suivante : (fr-Noun-conflit en-Noun-conflict es-Noun-conflicto ar-Noun-nzAE ar-Noun-SrAE ar-Noun-n$wb)

Le sens partagé par les éléments de cette clique est validé par le dictionnaire *Alwaseet*, sauf pour l'unité arabe *ar-Noun-n$wb*(نشوب), qui n'est qu'un fragment de l'unité composée *n$wb AlnzAE,* qui signifie */le déclenchement du combat ou de la lutte/.* Prise isolément, cette unité signifie : /déclenchement/.

3. Lacune dans le dictionnaire arabe :

   Exemple :

Prenons la clique suivante : (en-Verb-evaluate fr-Verb-évaluer en-Verb-assess es-Verb-evaluar ar-Verb-yqym).

Toutes les unités précédentes partagent le lien ILI-RECORD suivant : */judge the worth of something& 2ndOrderEntity 31 Agentive BoundedEvent Cause Dynamic Mental Purpose SituationType/.*

L'unité arabe *ar-Verb-yqym* (يقيم) a comme sens dans le dictionnaire *Alwaseet* : */Fixer la valeur de quelque chose/.* Nous pouvons pourtant considérer le sens partagé comme étant juste puisque le verbe *yqym* peut prendre ce sens dans certains contextes. Bien entendu, ici se pose le problème de notre référence d'évaluation : un dictionnaire ne peut décrire tous les sens de manière exhaustive. Nous aurions pu, bien entendu, nous appuyer uniquement sur un jugement de locuteur arabophone, plutôt que sur ce dictionnaire. Mais n'étant pas en mesure d'effectuer une validation auprès de plusieurs locuteurs, ce type d'évaluation nous a paru trop subjectif. C'est pourquoi nous avons préféré nous en tenir aux sens donnés par le dictionnaire (même si une certaine part - inévitable - de subjectivité est intervenue dans l'interprétation des gloses et des définitions).

D'un point de vue général, dans nos résultats expérimentaux, on voit que 94 / 111 unités arabes (verbes et noms) sont validées partiellement ou complètement par le dictionnaire *Alwaseet* (voir Tableau 17), ce que nous semble être un résultat tout à fait encourageant pour la méthode. Si celle-ci semble mieux fonctionner pour les noms que pour les verbes, c'est peut-être dû au fait que beaucoup de verbes présentent un sens très général, et sont plus polysémiques que les noms. En outre, de nombreux verbes font partie de locutions verbales ou jouent le rôle de collocatifs dans des collocations verbes nom, et ne prennent leur sens précis qu'au sein d'une expression plus large.

	Noms arabes	Verbes arabes
N° clusters traités	100	100
N°clusters valides	56	29
N° lemmes arabes dans les clusters valides	74	37
N° lemmes validés complètement	59	21
N° lemmes validés partiellement	8	6
N° lemmes non validés	7	10
Total d'unités arabes validés	94/111 (67/74+27/37)	
Pourcentage d'unités arabes (noms et verbes) validées (complètement et partiellement)	≈ 84,7%	

Tableau 17 : Tableau récapitulatif pour le résultat arabe

IV.8 Conclusion

Nous avons présenté une méthode de projection des relations sémantiques vers l'arabe qui offre la possibilité de structurer automatiquement les sens des unités arabes par l'intermédiaire de réseaux de type WordNet et de construire une ressource originale pour l'arabe à partir d'un corpus multilingue. En outre, les cliques multilingues extraites permettent la mise à jour et l'enrichissement des relations sémantiques qui sont manquantes dans les WNs mis en œuvre.

Nous avons relié non pas des mots, mais des cliques, aux WNs utilisés, parce qu'elles représentent un grain plus fin dans la décomposition du sens, parce qu'elles sont moins sensibles au découpage d'une langue donnée, et parce qu'elles forment de meilleurs candidats pour un ajustement mutuel.

Nos cliques clustérisées, analogues à des synsets, permettent théoriquement l'identification et l'organisation des sous-sens selon les deux propriétés de synonymie et de polysémie.

En outre, nous avons constaté que la qualité du rattachement des sens aux unités arabes qui apparaissent dans les cliques nominales est suffisante pour envisager la construction d'une ressource sémantique arabe d'une manière automatisée et moins coûteuse que s'il fallait partir de zéro. Une révision manuelle des résultats semble néanmoins nécessaire pour éliminer les erreurs et compléter les lacunes. L'idée de s'appuyer sur des ressources sémantiques élaborées pour d'autres langues semble donc prometteuse.

Cependant, les résultats pour les cliques verbales sont nettement moins satisfaisant (29%). Plusieurs critères doivent être pris en compte dans le but d'avoir un résultat plus précis et plus riche :

- Utilisation d'un corpus multilingue de grande dimension, et suffisamment varié afin d'obtenir une couverture suffisante de la langue générale.
- Utilisation d'un grand nombre de langues afin de diminuer la probabilité d'obtenir des polysémies parallèles.
- Prise en compte des lacunes des wordnets mis en œuvre. Pourquoi pas combler les lacunes de wordnets (absence de lexèmes, sens, certaines catégories…) en

bénéficiant de cliques pour l'enrichissement de base de données et la maximisation de compatibilité entre les WNs.

- Réglage de l'indice de similarité entre les cliques afin d'améliorer le regroupement en clusters et la structuration sémantique des cliques fusionnées.
- Amélioration de l'alignement lexical afin d'éviter les incohérences de découpage concernant les lexèmes composés. Le grain de découpage de la tokenisation devrait dans l'idéal s'arrêter aux limites de la compositionnalité sémantique (i.e. toute unité non-compositionnelle devrait être considérée comme un tout).

Chapitre V : Conclusion générale et perspectives

Nous avons présenté dans cette thèse la méthodologie employée en vue de construire une ressource sémantique pour la langue arabe. Dans cette approche, nous nous sommes inspirés de l'architecture de WordNet, qui présente l'avantage de structurer les sens par la prise en compte de deux phénomènes fondamentaux et complémentaires dans l'articulation du lexique, la polysémie (une unité pouvant porter des sens différents) et la synonymie (deux unités différentes pouvant porter un sens commun du point de vue de la désignation). Le synset, unité de base de ce type de réseau, s'avère en effet être un modèle à la fois simple et efficace pour représenter l'articulation entre les deux plans de l'expression et du contenu.

Nous avons montré par une étude préliminaire, consistant à créer des cliques à partir de correspondances extraites de dictionnaires multilingues (les Larousse en ligne), que les dictionnaires ne fournissent pas une référence suffisamment stable et complète pour construire des synsets sur une base multilingue. La méthode n'étant toutefois pas invalidée par ces premières observations, nous avons fait l'hypothèse que l'exploitation des corpus multilingues de grandes dimensions pouvait permettre d'obtenir une meilleure couverture des sens lexicaux.

Nous avons commencé nos expérimentations par l'extraction d'équivalents traductionnels à partir d'un corpus multilingue aligné, étiqueté et lemmatisé. L'extraction des équivalents s'est faite en se basant sur l'observation des occurrences et cooccurrences en corpus. Ensuite, à partir de ces équivalences, nous avons créé des "cliques", dont toutes les unités sont en interrelation, du fait d'une probable intersection sémantique : l'hypothèse qui est au centre de notre travail est que ces cliques sont analogues à des synsets, mais dans une perspective multilingue.

Lors de nos observations, nous avons constaté que les cliques créées automatiquement à partir de notre corpus multilingue constituent un guide intéressant pour l'organisation des sens. En effet, ces cliques présentent l'intérêt de renseigner à la fois sur la synonymie et la polysémie des unités, et d'apporter une forme de désambiguïsation sémantique - les équivalents traductionnels apportant souvent des indices intéressants pour la désambiguïsation, puisqu'une même polysémie a peu de chances de se retrouver dans de nombreuses langues différentes.

Un premier avantage lié à la constitution des cliques multilingues est qu'elles contiennent des unités connexes dans plusieurs langues : cette cohésion interne liée a leur propriété de complétude et de connexité permet de filtrer les résultats de l'alignement lexical et d'éliminer la plupart des erreurs d'alignement (de nombreuses correspondances obtenues avec Giza++ étant soit erronées, soit incomplètes, soit difficilement isolables de leur contexte traductionnel particulier). Il est en effet peu probable qu'une telle erreur d'alignement induise une intersection non vide entre plusieurs langues à la fois.

Nous avons pu vérifier, par l'étude minutieuse de nos cliques, que l'hypothèse de centralité des cliques était le plus souvent réalisée : les relations deux à deux de tous les éléments d'une clique dénotent un "sens" précis situé, en faisant une métaphore géométrique, à l'*intersection* de tous ces éléments. Mais la situation peut être parfois un peu plus complexe: d'une part on constate que les équivalents traductionnels de même clique partagent deux à deux un lien d'eq-synonymie tel qu'il est défini dans les wordnets multilingues de type EuroWordNet ; d'autre part, les sous-ensembles constitués des lexèmes d'une même langue au sein d'une même clique, que nous avons nommés eq-sets, ne correspondent que partiellement à la définition des synsets : dans certains cas, deux unités d'un même eq-set peuvent porter des sens voisins mais complémentaires, si la langue en question réalise un découpage sémantique plus fin que les autres langues de la même clique, comme les deux unités allemandes (*de-N-Sparsamkeit* et *de-N-Wirtschaft*) effectuant la distinction entre le sens de 'épargne' et celui de 'système économique' (cf. section II.4.1.3).

Il en résulte que ces cliques, analogues aux synsets d'un réseau sémantique, peuvent être reliées à un lexique sémantique préexistant de type Wordnet - à savoir, dans notre travail, EuroWordNet. Cela nous a donné la possibilité de récupérer, pour les unités arabes issues de notre corpus, des relations sémantiques définies pour des unités équivalentes en anglais, en français ou en espagnol.

Au vu de l'évaluation que nous avons effectuée, en ce que concerne la possibilité d'étendre les connaissances sémantiques dans d'autres langues vers la ressource arabe que nous voudrions construire *in fine*, les résultats obtenus apparaissent plutôt satisfaisants.

La méthode de projection des connaissances sémantiques vers l'arabe offre la possibilité de structurer automatiquement les sens des unités arabes par l'intermédiaire de WN, et confirme la possibilité d'en tirer une ressource originale pour la langue arabe.

En retour, l'ensemble des cliques multilingues obtenu s'est révélé utile pour mettre à jour et enrichir les relations sémantiques qui sont manquantes dans les WNs. En effet, les cliques offrent la possibilité de maximiser la compatibilité entre les wordnets de différentes langues, en consolidant les relations d'équivalence existantes et en les complétant par de nouvelles relations pour certaines langues.

Cette méthode automatique, et donc peu coûteuse, s'avère suffisamment générale pour être appliquée à d'autres langues que l'arabe, qui est sans doute un cas de figure parmi les plus difficiles étant donné les difficultés posées par cette langue en terme d'ambiguïté graphique et de segmentation des mots. Cette technique présente un intérêt pour les langues dites "peu dotées", qui ont besoin de rattraper l'écart technologique concernant les traitements informatiques et les usages sur les réseaux de l'information. La constitution d'une ressource sémantique complète peut en effet être très utile, dans un second temps, pour alimenter et améliorer diverses applications de TAL, comme les moteurs de question-réponse, la traduction automatique, les systèmes d'alignement, la recherche d'information, etc.

Enfin, nous avons vu qu'assez fréquemment une même clique peut contenir plusieurs formes fléchies pour le même lemme arabe. Vu les limitations actuelles des lemmatiseurs arabes, nous pensons que le fait d'avoir plusieurs formes fléchies d'un même lemme dans la même clique pourrait permettre de développer également le processus de la lemmatisation arabe, par la constitution progressive d'un *dictionnaire de formes fléchies* de plus en plus complet.

Cependant, nos expérimentations ont aussi permis de dégager certaines limites inhérentes aux données investies dans notre méthode. A cause de certaines insuffisances au niveau de la couverture de notre corpus, une phase de clusterisation a été rendue nécessaire pour regrouper des cliques artificiellement éparses. Cette phase a engendré des erreurs dans nos résultats, certaines cliques étant indûment regroupées. Afin d'éviter ce type de bruit, probablement, le recours à un corpus plus vaste et plus varié permettrait d'obtenir une couverture suffisante de la langue générale, de façon à capter de manière plus complète

toutes les virtualités sémantiques des unités et toute l'étendue de leurs possibilités de traduction. Pour diminuer l'effet de ce que nous avons dénommé des polysémies parallèles, le recours à un plus grand nombre de langues différentes ne peut qu'améliorer la finesse des résultats.

Nous avons constaté par ailleurs que les limites d'un alignement lexical trop souvent mot à mot, et négligeant la prise en compte des lexèmes composés et des expressions polylexicales, aboutissent à une augmentation importante du bruit dans les résultats. Pour améliorer la prise en compte de ces expressions, il n'est pas nécessaire, d'après nous, d'effectuer un prétraitement spécifique pour chaque langue mise en jeu. Les algorithmes d'alignement lexical classiques mis en œuvre dans des outils tels que Giza++ sont susceptibles d'identifier les expressions polylexicales, à condition que l'alignement soit effectué de façon bi-directionnelle, et que le corpus soit suffisamment vaste pour que les expressions polylexicales possèdent un nombre d'occurrences statistiquement significatif (5 occurrences ou plus sont généralement considérées comme suffisantes). En ce qui nous concerne, nous aurions sans doute pu améliorer nos résultats en réinjectant les résultats de l'alignement lexical dans une seconde étape, pour recalculer un alignement phrastique de meilleure qualité, et par conséquent augmenter la taille du corpus correctement aligné. En effet, les lexiques d'équivalents traductionnels permettent de consolider l'alignement phrastique en retour.

Toutes ces limitations, liées aux ressources utilisées (corpus et wordnets), génèrent des problèmes qui ne sont pas directement liés à notre méthode, mais à l'insuffisance des données. En vue d'obtenir une ressource sémantique arabe générique et réutilisable dans d'autres applications, un prolongement de cette recherche nous paraît indispensable, dans les directions suivantes :

- Etendre le corpus multilingue utilisé afin d'obtenir une couverture suffisante de la langue générale et d'éviter l'étape de clusterisation. Cela implique, bien sûr, de trouver des corpus plus variés, en terme de domaine et de genre, que ceux de l'ONU - qui représentent une variété trop spécifique pour en tirer une ressource générale.
- Diminuer le bruit en se basant sur une meilleure segmentation et une lemmatisation plus complète de l'arabe.

- Améliorer les résultats de l'alignement phrastique en réinvestissant les équivalences traductionnelles déjà identifiées sur un corpus de départ (tel que notre corpus).
- Utiliser des wordnets d'autres langues afin de pouvoir évaluer les cliques obtenues concernant les adjectifs et les adverbes, et compléter les résultats pour l'arabe.
- Etendre à l'ensemble du lexique, pour les verbes, les noms, les adverbes et les adjectifs, la projection des relations sémantiques vers les cliques obtenues pour l'arabe (pour les relations d'hyperonymie, de synonymie, d'antonymie, ….etc.).
- Enfin, du point de vue de l'évaluation, comparer nos résultats avec les relations obtenues dans d'autres travaux visant à relier des synsets arabes à l'ontologie SUMO, afin de maximaliser la cohérence sémantique des liens d'hyperonymie (Elkateb et al., 2006).

Ce processus peut-être progressif et incrémental, est une ressource sémantique imparfaite pouvant néanmoins être utile pour améliorer les résultats sur des nouveaux corpus.

Au final, toute la difficulté est de collecter des corpus multilingues parallèles suffisamment grands, et pour de nombreuses langues : mais la croissance exponentielle des données échangées sur Internet, et l'augmentation régulière des traductions collaboratives mises à la disposition des internautes (concrétisée par des projets tels que *OPUS Corpus*) laisse entrevoir une amélioration rapide de l'accès à ces traductions, qui constituent un véritable gisement de données pour le TAL.

Glossaire

Clique désambigüisée : Une clique où l'ambiguité sémantique n'est pas partagée par toutes les unités de langues différéntes. La polysémie des unités est orthogonale et non pas parallèle ce qu'aboutit à une intersection réduite.

Clique valide : Une clique dont toutes les unités sont rattachées au même ILI-RECORD dans EuroWordNet.

Clusterisation : Processus consistant à réunir des données voisines dans des ensembles constituant une partition de l'ensemble des ses données.

Concept : Un concept n'est pas une unité linguistique, mais un objet extralinguistique défini au sein d'un certain domaine (physique, mathématique, biologie… etc.), par des procédures spécifiques (un concept mathématique ne se construit pas de la même manière qu'un concept en biologie, où il est associé - par exemple - à tout un ensemble de procédures expérimentales). Il faut noter ici que pour certains réseaux sémantiques utilisés pour représenter des contenus linguistiques, tels que les lexiques sémantiques de type WordNet, on utilise le terme de concept pour désigner des contenus linguistiques.

Dénotation : Selon le Petit Larousse, "ensemble des éléments fondamentaux et permanents du sens d'un mot". Nous préférons utiliser le terme "signification", la notion de notion de sens étant pour nous liée à l'interprétation du mot en contexte.

Dictionnaires de formes fléchies : Un dictionnaire comportant toutes les formes possibles implique que les traits soient attachés à toutes les formes fléchies d'un même mot.

eq-sets : Par cette notation nous désignons les sous-ensembles constitués des lexèmes d'une même langue, au sein des cliques.

EWN: EuroWordNet.

Intersection sémantique : Désigne les acceptions communes à plusieurs unités lexicales. Pour des unités de langues différentes, l'intersection est constituée des acceptions équivalentes (i.e. susceptibles d'être à la base d'une équivalence traductionnelle dans un contexte donné).

SUMO: Suggested Upper Merged Ontology.

Synonymie : Relation entre des lexèmes ayant des formes différentes mais ayant des sens voisins. Notons que la synonymie n'implique pas la dénomination multiple (deux synonymes peuvent désigner le même référent, mais sans avoir les mêmes significations). Ex. *Le vilain mari* et *le prince charmant*.

TAL : Traitement automatique des langues.

Taxonomie : Discipline qui a pour objet de décrire les organismes vivants et de les regrouper en entités appelées taxons (familles, genres, espèces, etc.) afin de pouvoir les nommer et les classer. Par extension, en sémantique, classification d'un ensemble d'entités au moyen d'une hiérarchie de classes et sous-classes.

Unité désambigüisée : Une unité rattachée à un et un seul ILI-RECORD au sein d'une clique.

wordnet : Ce terme (écrit en minuscule) est utilisé dans notre travail comme un nom commun désignant un réseau lexical électronique qui est organisé selon des sens lexicaux, c.à.d. des sens structurés par le lexique de la langue. Les lexèmes partageant un même sens sont regroupés dans des *synsets*. L'intersection sémantique entre plusieurs lexèmes d'un même synset constitue une caractérisation linguistique du sens complémentaire de la glosse qui en est donnée.

Annexes

Annexe 1: Le fichier Stopword.ar contenant les stoplists arabes :

*lk

{

}

¤

<*A

<*F

<*n

<DAftF

<lx

<lY

<mA

>Hsn

>HyAnA

>HyAnF

>jl

>l*yn

>llty

>mAm

>mAm

>w

>wl}k

>xrY

>y

Al*Any

Al>wl

Al>xyr

AlEaSr

AlrAbE

AlsAbE

AlsAds

AltAsE

AlvAmn

AlxAms

Aql

b

b<stvnA'

bAldAxl

bdwn

bEd

bEydF En

bfDl

bHdwd

bjAnb

blqrb mn

blY

bsbb

bxSwS

dA}mA

dA}mF

Dd

dwmA

dwmF

EAdtF

Ebr

EdA

Eks

ElY

ElY Twl

En EwDF

EndmA

EsY

f

fqT

fwq

fy

h&lA'

h*A

h*h

HASA

hk*A

hl

hm

hn

hnA

Hsn

Ht~Y

hw

Hwl

hy

hyA

HyvmA

jmE

jmyE

jyd

k

kAn

kl~mA

kmA

kvyr

kvyrF

ky

kyf

l

lA

lEl

lkl

lkn

lmA*A
ln
lys
lyt
mA*A
mE
mEF
mEtbrF
mkAn
mn
mn blrgm
mn byn
mn jdyd
mn*
mqAbl
mstvnyF
mvl
nAqS
nAtj En
nEm
nfs
nfsh
qbl
qbyl
qd
qlyl
qlylF
qryb
rgm
rgm
rqm
SEr
swf

tArAtF

tHt

tlk

tqrybF

vm

w

wfqA l

wrA'

xArj

xlAl

yjb

ykwn

Annexe 2: Les noms français existant dans notre corpus et ayant un nombre d'occurrences superieur à 1500 :

Nom français	Nombre d'occurrences
Rapport	12024
Pays	11252
Droit	11094
Organisation	10817
Session	9471
Programme	8385
Projet	7853
Membre	7784
Service	6721
Conseil	6594
Convention	6240
Commission	5995
Partie	5971
Question	5944
Article	5816
Groupe	5528

Travail	5502
Ressource	5102
Bureau	4858
Gouvernement	4758
Paragraphe	4672
Information	4599
Mesure	4546
Femme	4349
Paix	4122
Recommandation	4072
Mission	3993
Gestion	3953
Dollar	3729
Application	3564
Homme	3329
Personnel	3094
Arme	3023
Postes	2905
Document	2898

Personne	2861
Appui	2776
Protection	2711
Territoire	2685
Plan	2585
Formation	2551
Institution	2520
Ministre	2501
Coordination	2488
Enfant	2420
Objectif	2381
Fonds	2348
Office	2336
Budget	2328
Assistance	2324
Action	2304
Situation	2287
Tableau	2270
Domaine	2222

Accord	2197
Secteur	2167
Organe	2143
Principe	2028
Examen	2011
Niveau	1982
Financement	1963
Effort	1961
Peuple	1953
Disposition	1948
Processus	1863
Demande	1837
Directive	1809
Centre	1785
Compte	1778
Annexe	1729
Division	1702
Conflit	1699
Cadre	1678

Montant	1656
Affaire	1648
Base	1625
Pratique	1615
Besoin	1604
Communication	1588
Contribution	1577
Zone	1531
Maintien	1515
Commerce	1473
Loi	1462
Nombre	1459
Cours	1457
Protocole	1444
Population	1439
Sommet	1434
Environnement	1429
Monsieur	1429
Cas	1379

Rapporteur	1378
Exercice	1369
Initiative	1360
Aide	1354
Expert	1337
Mécanisme	1337
Participation	1336
Pêche	1335
Effet	1330
Coût	1329
Technologie	1318

Annexe 3 : Les verbes français existant dans notre corpus et ayant un nombre d'occurrences superieur à 300 :

Verbes français	Nombre d'occurrences
Devoir	5480
Pouvoir	4828
Adopter	3409
Réfugier	3360
Concerner	3229
Prendre	3182
Voir	2949
Noter	2563
Renforcer	2218
Examiner	2127
Faire	1998
Recommander	1983
Suivre	1978
Continuer	1947
Organiser	1581
Tenir	1578

Créer	1543
Fournir	1471
Présenter	1424
Articler	1370
Permettre	1325
Appliquer	1317
Améliorer	1254
Demander	1246
Viser	1246
Promouvoir	1235
Etablir	1214
Approuver	1129
Proposer	1101
Utiliser	1045
Encourager	1024
Assurer	1001
Reconnaître	947
Occuper	933
Aider	930

Participer	904
Représenter	903
Comprendre	879
Constituer	863
Exprimer	846
Faciliter	846
Réduire	835
Intituler	798
Contribuer	764
Souligner	754
Poursuivre	731
Soumettre	731
Membrer	711
Réaffirmer	710
Financer	708
Recevoir	708
Spécialiser	703
Décider	702
Charger	697

Féliciter	677
Prévoir	664
Accepter	659
Considérer	657
Autoriser	655
Intégrer	652
Envisager	646
Lier	634
Partir	616
Formuler	611
Développer	607
Appuyer	601
Soutenir	601
Elaborer	583
Falloir	573
Protéger	566
Coordonner	561
Signer	549
Atteindre	547

Inviter	521
Commettre	519
Figurer	510
Réaliser	488
Répondre	484
Accroître	482
Prévenir	481
Réviser	473
Compter	467
Prier	459
Estimer	456
Rester	452
Consacrer	444
Accorder	437
Donner	434
Conclure	423
Contenir	421
Garantir	419
Evaluer	418

Augmenter	413
Adresser	411
Adjoindre	409
Informer	405
Avancer	401
Déterminer	401
Approprier	395
Ratifier	389
Vivre	387
Respecter	377
Limiter	374

Annexe 4 : Récapitulatif des textes du notre corpus (Les codes de types de textes indiqués dans le tableau sont clarifiés sur le site de l'ONU suivant : http://www.un.org/Depts/dhl/unbisref_manual/bd/codes/c089.htm)

Numero du texte	Titre	Année	Type	Nombre de mots
T115	Déclaration des Nations Unies sur le Nouveau Partenariat pour le développement de l'Afrique (16 septembre 2002)	2002	Afrique	506
T116	Déclaration du Millénaire (8 septembre 2000)	2000	Afrique	3301
T117	Document final du Sommet mondial de 2005 (16 septembre 2005)	2005	Anniversaires	16576
T118	United Nations Declaration on Human Cloning (8 March 2005)	2005	clonage	458
T119	United Nations Convention on the Use of Electronic Communications in International Contracts	2005	commerce international	4282
T120	Model Law on International Commercial Conciliation of the United Nations Commission on International Trade Law (19 November 2002)	2002	commerce international	2177
T121	United Nations Convention on the Assignment of Receivables in International Trade (12 December 2001)	2001	commerce international	9237
T123	United Nations Convention against Corruption (31 October 2003)	2003	crime	19340
T124	United Nations Convention against Transnational Organized Crime: Protocol against the Illicit Manufacturing of and Trafficking in Firearms, Their Parts and Components and Ammunition (31 May 2001)	2001	crime	4233

T125	United Nations Convention against Transnational Organized Crime (15 November 2000)	2000	crime	22129
T126	Vienna Declaration on Crime and Justice: Meeting the Challenges of the Twenty-first Century (4 December 2000)	2000	crime	2228
T127	Global Agenda for Dialogue among Civilizations (9 November 2001)	2001	Culture	1946
T128	Agreement concerning the Relationship between the United Nations and the Organization for the Prohibition of Chemical Weapons (7 September 2001)	2001	désarmement	2759
T130	Convention on Jurisdictional Immunities of States and Their Property (2 December 2004)	2004	droit international	5407
T131	Nationality of Natural Persons in Relation to the Succession of States (12 December 2000)	2000	droit international	2820
T132	Political declaration (10 June 2000)	2000	droits de le femme	706
T133	United Nations Declaration on the Rights of Indigenous Peoples (13 September 2007)	2007	droits de l'homme	3923
T134	Basic Principles and Guidelines on the Right to a Remedy and Reparation for Victims of Gross Violations of International Human Rights Law and Serious Violations of International Humanitarian Law (16 December 2005)	2005	droits de l'homme	3507
T135	Convention against Torture and Other Cruel, Inhuman or Degrading Treatment or Punishment: Optional Protocol (18 December 2002)	2002	droits de l'homme	4865

T136	Convention on the Rights of the Child: Optional Protocol on the Sale of Children, Child Prostitution and Child Pornography (25 May 2000)	2000	droits de l'homme	5721
T137	A world fit for children (10 May 2002)	2002	Enfants	11280
T138	Convention on the Safety of United Nations and Associated Personnel: Optional Protocol (8 December 2005)	2005	Fonction publique internationale	1192
T139	Statute of the United Nations System Staff College (12 July 2001)	2001	Fonction publique internationale	1915
T140	Agreement between the United Nations and the World Tourism Organization (23 December 2003)	2003	Institutions spécialisées	3047
T141	Declaration of Commitment on HIV/AIDS (27 June 2001)	2001	Santé	6372
T144	Contingent liability reserve for the United Nations Postal Administration : report of the Secretary-General	2007	rapport,étude,lette	2471
T145	Letter dated 7 May 2007 from the Permanent Representative of Cuba to the United Nations addressed to the Secretary-General	2007		336
T145	Identical letters dated 2007/05/10 from the Chairman of the Peacebuilding Commission addressed to the President of the General Assembly and the President of the Security Council	2007	rapports,lettres,résumée	4141
T146	Letter dated 2007/05/07 from the Permanent Representative of Cuba to the United Nations addressed to the Secretary-General	2007	B18,A20	936

T147	Note verbale dated 2007/05/10 from the Permanent Mission of Bosnia and Herzegovina to the United Nations addressed to the President of the General Assembly	2007	B18	290
T148	Identical letters dated 2007/05/10 from the Permanent Representative of Israel to the United Nations addressed to the Secretary-General and the President of the Security Council	2007	B18	1405
T150	Letter dated 2007/05/08 from the Permanent Representatives of Egypt and Iraq to the United Nations addressed to the Secretary-General	2007	B18,A20	265
T151	Identical letters dated 2007/05/10 from the Chargé d'affaires a.i. of the Permanent Mission of Lebanon to the United Nations addressed to the Secretary-General and the President of the Security Council	2007	B18	495
T152	Identical letters dated 2007/05/10 from the Chargé d'affaires a.i. of the Permanent Mission of Lebanon to the United Nations addressed to the Secretary-General and the President of the Security Council	2007	B18	994
T153	Identical letters dated 2007/05/14 from the Chargé d'affaires a.i. of the Permanent Mission of Lebanon to the United Nations addressed to the Secretary-General and the President of the Security Council	2007	B18	1953
T154	Letter dated 2006/06/07 from the Permanent Representative of Peru to the United Nations addressed to the President of the General Assembly	2007	B18,A20	439

T155	Identical letters dated 2007/05/16 from the Chargé d'affaires a.i. of the Permanent Mission of Israel to the United Nations addressed to the Secretary-General and the President of the Security Council	2007	B18	2475
T156	Letter dated 2007/05/15 from the Permanent Representative of Egypt to the United Nations addressed to the Secretary-General	2007	B18	353
T157	Identical letters dated 2007/05/16 from the Permanent Representative of Afghanistan to the United Nations addressed to the Secretary-General and the President of the Security Council	2007	B18	1712
T158	Letter dated 2007/05/15 from the Permanent Representative of Cyprus to the United Nations addressed to the Secretary-General	2007	B18	1707
T160	Identical letters dated 2007/05/17 from the Chargé d'affaires a.i. of the Permanent Observer Mission of Palestine to the United Nations addressed to the Secretary-General and the President of the Security	2007	B18	705
T161	Letter dated 2007/05/18 from the Permanent Representatives of Moldova to the United Nations addressed to the Secretary-General	2007	B18	640

T162	Reports of the Secretary-General on the revised estimates resulting from decision S-4/101 adopted by the Human Rights Council at its 4th special session in 2006 (A/61/530/Add.2) and on the revised estimates resulting from resolutions adopted by the Council at its 4th session in 2007 (A/61/30/Add.3) : report of the Advisory Committee on Administrative and Budgetary Questions	2007	B16	1407
T164	Letter dated 2007/05/03 from the Chargé d'affaires a.i. of the Permanent Mission of Saudi Arabia to the United Nations addressed to the President of the General Assembly	2007	B18,A08,A20	1314
T165	Comprehensive proposal on appropriate incentives to retain staff of the International Criminal Tribunal for Rwanda and the International Tribunal for the Former Yugoslavia : report of the Advisory Committee on Administrative and Budgetary Questions	2007	B16	1697
T166	Report of the International Research and Training Institute for the Advancement of Women ; report of the Secretary-General on the financial situation of the International Research and Training Institute for the Advancement of Women : report of the Advisory Committee on Administrative and Budgetary Questions	2007	B16	1155
T167	Letter dated 2006/06/12 from the Permanent Representatives of Azerbaijan and Lithuania to the United Nations addressed to the Secretary-General	2007	B18,A20	1866

T168	Report of the Redesign Panel on the United Nations system of administration of justice : revised estimates relating to the programme budget for the biennium 2006-2007 and the proposed programme budget for the biennium 2008-2009 pursuant to General Assembly resolution 61/261 : report of the Advisory Committee on Administration and Budgetary Questions	2007	B16	2849
T169	Comprehensive report on strengthening the capacity of the United Nations to manage and sustain peace operations ; proposed budget for the support account for peacekeeping operations for the period from 1 July 2007 to 30 June 2008 ; revised estimates relating to the programme budget for the biennium 2006-2007 and the proposed programme budget for the biennium 2008-2009 under sections 5, Peacekeeping operations, 28D, Office of Central Support Services, and 35, Staff assessment ; performance report on the budget for the support account for peacekeeping operations for the period from 1 July 2005 to 30 June 2006 : report of the Advisory Committee on Administrative and Budgetary Questions	2007	B16	23878
T171	Letter dated 2007/06/07 from the Permanent Representative of Azerbaijan to the United Nations addressed to the Secretary-General	2007	B18,A08	2160
T172	Special measures for protection from sexual exploitation and sexual abuse : report of the Secretary-General	2007	B15,B16	4981

T173	Appointment of members of the Joint Inspection Unit : note / by the President of the General Assembly	2007	A10,B18,B15	4552
T174	Administrative and budgetary aspects of the financing of the United Nations peacekeeping operations : report of the 5th Committee : General Assembly, 61st session	2007	B02,B04	11288
T175	Review of the efficiency of the administrative and financial functioning of the United Nations ; programme budget for the biennium 2006-2007 ; report on the activities of the Office of Internal Oversight Services ; administrative and budgetary aspects of the financing of the United Nations peacekeeping operations : report of the 5th Committee : General Assembly, 61st session	2007	B04,B02,A08	2323
T176	Letter dated 2007/05/30 from the Permanent Representative of Pakistan to the United Nations addressed to the Secretary-General	2007	B18,A20	75086
T177	Identical letters dated 2007/07/06 from the Permanent Representative of Italy to the United Nations addressed to the Secretary-General and the President of the Security Council	2007	B18,A16	2257
T178	Note verbale dated 2007/06/08 from the Permanent Missions of Australia and Indonesia to the United Nations addressed to the Secretary-General	2007	B18,A08,20	3198
T179	Report of the Secretary-General on the work of the Organization	2005	B15,B16	31607
T180	Report of the International Law Commission, 57th session (2 May-3 June and 11 July-5 August 2005)	2005	B04,A17	79036

T181	Report of the International Law Commission, 57th session (2 May-3 June and 11 July-5 August 2005)	2005	B19	94250
T182	Annotated draft agenda of the 60th session of the General Assembly : addendum	2005	B19	30413
T183	Letter dated 2006/08/01 from the Permanent Representative of Malaysia to the United Nations addressed to the Secretary-General	2006	B18,A20	40599
T184	Report of the Committee on Contributions, 65th session (6-24 June 2005)	2005	B04	11648
T185	Letter dated 2005/07/05 from the Permanent Representative of Jamaica to the United Nations addressed to the Secretary-Genral	2005	B18,A20	16210
T186	Note [transmitting report of the Board of Auditors on implementation of its recommendations relating to the biennium 2002-2003] Note / by the Secretary-General	2005	B16,B18	44461
T188	Report on the world social situation, 2005	2005	B16	62481
T189	Report of the United Nations High Commissioner for Refugees	2005	B04	8468
T190	United Nations High Commissioner for Refugees : report of the Executive Committee of the Programme of the United Nations High Commissioner for Refugees, 56th session (3-7 October 2005)	2005	B04	10754
T193	Report of the Commissioner-General of the United Nations Relief and Works Agency for Palestine Refugees in the Near East, 1 July 2004-30 June 2005	2005	B04,B18	40828

T195	Follow-up to the implementation of the International Year of Volunteers : report of the Secretary-General	2005	B15,B16	8137
T196	Some measures to improve overall performance of the United Nations System at the country level. Part 1, Short history of United Nations reform in development : note / by the Secretary-General	2006	B16	7211
T197	Report of the Joint Inspection Unit on some measures to improve overall performance of the United Nations System at the country level : note / by the Secretary-General	2005	B19	7639
T199	Establishment of a nuclear-weapon-free zone in the region of the Middle East : report of the Secretary-General	2005	B15,B16	6260
T200	African Institute for the Prevention of Crime and the Treatment of Offenders : report of the Secretary-General	2005	B15,B16	7305
T202	Letter dated 2005/06/27 from the Permanent Representative of Mexico to the United Nations addressed to the Secretary-General	2005	B18,A16,A20	5502
T203	Confidence-building measures in the regional and subregional context : report of the Secretary-General	2005	B15,B16	4652
T204	odalities of the inter-agency coordination of the implementation of the outcomes of the World Summit on the Information Society, including recommendations on the follow-up process : report of the Secretary-General	2006	B15,B16	4997

T206	Administrative expenses of the United Nations Joint Staff Pension Fund : report of the Standing Committee of the United Nations Joint Staff Pension Board	2005	B16	31275
T207	Sustainable fisheries, including through the 1995 Agreement for the Implementation of the Provisions of the United Nations Convention on the Law of the Sea of 10 December 1982 relating to the Conservation and Management of Straddling Fish Stocks and Highly Migratory Fish Stocks, and related instruments : report of the Secretary-General	2005	B15,B16	23779
T208	Curricula vitae of candidates nominated by national groups : note / by the Secretary-General	2005	A10	15697
T211	United Nations Institute for Disarmament Research : note / by the Secretary-General	2005	B16	8701
T213	New Partnership for Africa's Development : 3rd consolidated report on progress in implementation and international support : report of the Secretary-General	2005	B15,B16	10401
T214	Implementation of the International Strategy for Disaster Reduction : report of the Secretary-General	2005	B15,B16	8862
T215	Report on the world social situation, 2005	2005	B16	62481
T216	Note [transmitting report of the Board of Auditors on implementation of its recommendations relating to the biennium 2002-2003]	2005	B16,B18	44461
T217	Letter dated 2005/07/05 from the Permanent Representative of Jamaica to the United Nations addressed to the Secretary-Genral	2005	B18,A20	16210

T218	Annotated preliminary list of items to be included in the provisional agenda of the 60th regular session of the General Assembly	2005	B19	94250
T219	Report of the Secretary-General on the work of the Organization	2005	B15,B16	31607
T220	Report of the Committee on Contributions, 65th session (6-24 June 2005)	2005	B04	11648
T221	Report of the Committee for Programme and Coordination, 45th session (6 June-1 July 2005)	2005	B04	19404
T223	Report of the Commissioner-General of the United Nations Relief and Works Agency for Palestine Refugees in the Near East, 1 July 2004-30 June 2005	2005	B04,B18	40828
T224	Report of the International Law Commission, 57th session (2 May-3 June and 11 July-5 August 2005)	2005	B04,A17	79036
T225	Report of the Committee on the Elimination of Racial Discrimination, 66th session, 21 February-11 March 2005 [and] 67th session, 2-19 August 2005	2005	B04	77875
T226	Report of the United Nations Commission on International Trade Law on its 38th session, 4-15 July 2005	2005	B04,A17	35660
T227	Report of the United Nations High Commissioner for Refugees	2005	B04	8468
T228	United Nations High Commissioner for Refugees : report of the Executive Committee of the Programme of the United Nations High Commissioner for Refugees, 56th session (3-7 October 2005)	2005	B04	10754

T229	Report of the Office of Internal Oversight Services on the inspection of programme and administrative management of the subregional offices of the Economic Commission for Africa	2005	B16	9233
T231	Global analysis and evaluation of national action plans on youth employment : report of the Secretary-General	2005	B15,B16	8567
T232	International cooperation against the world drug problem : report of the Secretary-General	2005	B15,B16	8820
T233	Letter dated 2005/07/25 from the Permanent Representative of Yemen to the United Nations addressed to the Secretary-General	2005	B18,A08,A20	84552
T234	Curricula vitae of candidates nominated by States Members of the United Nations and by non-member States maintaining permanent observer missions at United Nations Headquarters : note / by the Secretary-General	2005	A10	40551
T236	Financial performance report for the period from 1 July 2003 to 30 June 2004 and proposed budget for the period from 1 July 2005 to 30 June 2006 of the United Nations Logistics Base at Brindisi : implementation of the strategic deployment stocks, including the functioning of the existing mechanisms and award of contracts for procurement : report of the Advisory Committee on Administrative and Budgetary Questions	2005	B16	13673

T237	Financial performance report for the period from 1 July 2003 to 30 June 2004 and proposed budget for the support account for peacekeeping operations for the period from 1 July 2005 to 30 June 2006 : report of the Advisory Committee on Administrative and Budgetary Questions	2005	B16	14831
T238	Letter dated 2005/03/24 from the Secretary-General to the President of the General Assembly	2005	B15,B16,A08	19325
T240	Letter dated 2005/03/18 from the Permanent Representative of Azerbaijan to the United Nations addressed to the Secretary-General	2005	B18	14474
T241	Letter dated 2005/03/16 from the Chargé d'affaires a.i. of the Permanent Mission of Armenia to the United Nations addressed to the Secretary-General	2005	B18	15151
T242	Budget for the United Nations Stabilization Mission in Haiti for the period from 1 July 2005 to 30 June 2006 and expenditure report for the period from 1 May to 30 June 2004 : report of the Secretary-General	2005	B15,B16,B11	14007
T243	Budget for the support account for peacekeeping operations for the period from 1 July 2005 to 30 June 2006 : report of the Secretary-General	2005	B15,B16	52021
T245	Gratis personnel provided by Governments and other entities : report of the Secretary-General	2005	B15,B16	7864

T248	Report on the work of the United Nations Open-ended Informal Consultative Process on Oceans and the Law of the Sea : letter dated 9 June 2003 from the Co-Chairpersons of the Consultative Process addressed to the President of the General Assembly	2003	B04,B18	16574
T249	Assessment of the results achieved in realizing aims and objectives of the International Year of Ecotourism : note / by the Secretary-General	2003	B16	8039
T250	Revitalization of the work of the General Assembly : report of the Secretary-General	2004	B16,B15	40985
T251	Letter dated 2004/07/15 from the Permanent Representative of Turkey to the United Nations addressed to the Secretary-General	2004	B18,A08,A20	123357
T252	Note [transmitting report of the Panel of Eminent Persons on United Nations-Civil Society Relations]	2004	B16,B10	33840
T253	Report of the Joint Inspection Unit entitled "Managing information in the United Nations System organizations : management information systems" : note / by the Secretary-General	2003	B16	21638
T254	Report of the Joint Inspection Unit entitled "Evaluation of United Nations System response in East Timor : coordination and effectiveness" : note / by the Secretary-General	2003	B16	18797
T255	Implementation of the Programme of Action for the Least Developed Countries: report of the Secretary-General	2003	B15,B16	12461
T256	Report of the Governing Council of the United Nations Human Settlements Programme, 19th session (5-9 May 2003)	2003	B04,B01,A08	31472

T258	General and complete disarmament : report of the 1st Committee : General Assembly, 55th session	2000	B04,B02,B07	29695
T260	First report on the implementation of the recommendations of the Board of Auditors on the accounts of the United Nations funds and programmes for the biennium ended 31 December 1999 : report of the Secretary-General : addendum	2000	B15,B16	24260
T261	Large-scale pelagic drift-net fishing, unauthorized fishing in zones of national jurisdiction and on the high seas, fisheries by-catch and discards, and other developments : report of the Secretary-General	2000	B15,B16	25097
T262	Implementation of the United Nations New Agenda for the Development of Africa in the 1990s : progress report of the Secretary-General	2000	B15,B16	21878
T263	Illicit traffic in small arms : report of the Secretary-General	2000	B15,B16	22903
T264	Identical letters dated 2000/08/21 from the Secretary-General to the President of the General Assembly and the President of the Security Council	2000	B15,B16	42631
T265	Report of the Committee on the Elimination of Discrimination against Women, 22nd session (17 January-4 February 2000) [and] 23rd session (12-30 June 2000)	2000	B04,B01,B24	59827
T266	Report of the Council of the United Nations University, January-December 1999	2000	B04	18459
T267	Report of the Special Committee on the Charter of the United Nations and on the Strengthening of the Role of the Organization	2000	B04	25278

T268	Report of the Committee on Conferences for 2000	2000	B04	21563
T269	Report of the Committee on the Exercise of the Inalienable Rights of the Palestinian People	2000	B04	13765
T270	Report of the Committee on the Elimination of Discrimination against Women, 22nd session	2000	B04	28331
T272	Report of the Special Committee to Investigate Israeli Practices Affecting the Human Rights of the Palestinian People and Other Arabs of the Occupied Territories : note / by the Secretary-General	2000	B04	24787
T273	Letter dated 2001/06/25 from the Secretary-General to the President of the General Assembly	2001	B15,B16	32450
T274	Letter dated 2005/08/15 from the Permanent Representative of the Bolivarian Republic of Venezuela to the United Nations addressed to the Secretary-General	2005	B16,B18	8496
T277	Summary of the special high-level meeting of the Council with the Bretton Woods institutions and the World Trade Organization (New York, 26 April 2004) / by the President of the Economic and Social Council	2004	B19	4308
T278	Strengthening the coordination of emergency humanitarian assistance of the United Nations : report of the Secretary-General	2004	B15,B16	8926
T279	Implementation of the Programme of Action for the Least Developed Countries for the Decade 2001-2010 : report of the Secretary-General	2004	B15,B16	9357

T280	Letter dated 2004/06/02 from the Permanent Representatives of Finland and the United Republic of Tanzania to the United Nations addressed to the Secretary-General	2004	B18,B16	337
T281	The situation in Afghanistan and its implications for international peace and security : report of the Secretary-General	2004	B15,B16	8969
T283	Report of the Economic and Social Council for 2000	2001	B04,B07	39319
T285	We the peoples :#the role of the United Nations in the 21st century : report of the Secretary-General	2000	B15,B16	29088
T286	Report of the International Law Commission, 54th session (29 April-7 June and 22 July-16 August 2002)	2002	B04,A17	89009
T288	Annotated preliminary list of items to be included in the provisional agenda of the 57th regular session of the General Assembly	2002	B19	98404
T289	Annotated draft agenda of the 57th session of the General Assembly : addendum	2002	B19	16787
T290	Report of the Committee on Contributions, 62nd session (3-21 June 2002)	2002	B04	11591
T292	Report of the Executive Committee of the Programme of the United Nations High Commissioner for Refugees, 53rd session, 30 September-4 October 2002	2002	B04,B01,A08	24836
T293	Report of the Commissioner-General of the United Nations Relief and Works Agency for Palestine Refugees in the Near East, 1 July 2001-30 June 2002	2002	B04,B18	45757

T294	Report of the Trade and Development Board on its 19th special session (Bangkok, 29 April to 2 May 2002)	2002	B04,B02	19479
T295	Measures to eliminate international terrorism : report of the Secretary-General	2002	B15,B16	28294
T297	Report of the Special Committee on the Situation with regard to the Implementation of the Declaration on the Granting of Independence to Colonial Countries and Peoples for 2002	2003	B04,A16,B01,B02	46583
T298	Organization of the 57th regular session of the General Assembly, adoption of the agenda and allocation of items : 1st report of the General Committee	2002	B04	16201
T299	Programme budget for the biennium 2002-2003	2002	A08	32698
T301	Proposed programme budget for the biennium 2002-2003. Part 1, Overall policy-making, direction and coordination, Section 1, Overall policy-making, direction and coordination	2001	A17	17016
T303	Proposed programme budget for the biennium 2002-2003. Part 5, Regional cooperation for development, Section 16, Economic and social development in Africa (Programme 14 of the medium-term plan for the period 2002-2005)	2001	A17	22763
T306	Proposed programme budget for the biennium 2002-2003. Part 5, Regional cooperation for development, Section 19, Economic and social development in Latin America and the Caribbean (Programme 17 of the medium-term plan for the period 2002-2005)	2001	A17	37685

T308	Proposed programme budget for the biennium 2002-2003. Part 5, Regional coooperation for development, Section 20, Economic and social development in Western Asia (Programme 18 of the medium-term plan for the period 2002-2005)	2001	A17	15191
T309	Proposed programme budget for the biennium 2002-2003. Part 6, Human rights and humanitarian affairs, Section 22, Human rights (Programme 19 of the medium-term plan for the period 2002-2005)	2001	A17	22390
T310	Proposed programme budget for the biennium 2002-2003. Part 6, Human rights and humanitarian affairs, Section 25, Humanitarian assistance (Programme 20 of the medium-term plan for the period 2002-2005)	2001	A17	17110
T312	Proposed programme budget for the biennium 2002-2003. Part 2, Political affairs, Section 3, Political affairs (Programme 1 of the medium-term plan for the period 2002-2005)	2001	A17	17820
T313	Proposed programme budget for the biennium 2002-2003. Part 13, Development Account, Section 33, Development Account	2001	A17	33436
T314	Proposed programme budget for the biennium 2002-2003. Part 3, International justice and law, Section 8, Legal affairs (Programme 5 of the medium-term plan for the period 2002-2005)	2001	A17	18068
T315	Budget for the United Nations Organization Mission in the Democratic Republic of the Congo for the preiod from 1 July 2001 to 30 June 2002 : report of the Secretary-General	2001	B15,B16,B11	20423

T316	Proposed programme budget for the biennium 2002-2003. Part 8, Common support services, Section 27D, Office of Central Support Services (Programme 24 of the medium-term plan for the period 2002-2005)	2001	A17	18201
T317	Proposed programme budget for the biennium 2002-2003. Part 2, Political affairs, Section 4, Disarmament (Programme 2 of the medium-term plan for the period 2002-2005)	2001	A17	13228
T318	Proposed programme budget for the biennium 2002-2003. Part 2, Political affairs, Section 5, Peacekeeping operations (Programme 3 of the medium-term plan for the period 2002-2005)	2001	A17	14100
T319	Budget for the United Nations Transitional Administration in East Timor for the period from 1 July 2001 to 30 June 2002 : report of the Secretary-General	2001	B15,B16,B11	10669
				Totale : 3272805

Annexe 5 : Les noms arabes existant dans les cliques désambigüisées au regard de 3 sens :

Noms arabes existant dans les cliques désambigüisées au regard de 3 sens (3 langues)	Sens commun extrait au regard de 3 sens	Validation du sens extrait
ar-Noun-tTbyq (تطبيق)[34] ar-Noun-tnfy (تنفيذ)	the act of bringing something to bear; using it for a particular purpose; "he advocated the application of statistics to the problem"& 03 04 2ndOrderEntity Agentive Cause Dynamic Purpose SituationType	✓[35] ✓
ar-Noun-slAH (سلاح)	weaponry used in fighting or hunting; "he was licensed to carry a weapon"& 03 06 1stOrderEntity Artifact Form Function Group Instrument Object Origin	✓
ar-Noun-msAEdp (مساعدة) ar-Noun-tqdym AlmsAEdp (تقديم المساعدة)	a resource: "visual aids in teaching"; "economic assistance to depressed areas"& 03 07 2ndOrderEntity Property SituationType Static	✓ ✓
ar-Noun-AHtyAjAt (احتجاجات) **ar-Noun-tlbyp AHtyAjAt (تلبية احتجاجات)**[36]	a state of extreme poverty or destitution; "their indigence appalled him"; "a general state of need exists among the homeless"& 03 26 2ndOrderEntity Condition Property SituationType Static	✓ ✓
ar-Noun-mktb (مكتب)	an administrative unit of government; "the Central Intelligence Agency"; "the Census Bureau"; "Office of Management and Budget"; "Tennessee Valley Authority"& 03 14 1stOrderEntity Function Group Human Living	✓X[37]

[34] Les unités dans la même case, existent dans la même clique, donc elles partagent le même sens désambigüisé.

[35] ✓ Le sens est bien trouvé dans le dictionnaire.

[36] Les unités en gras sont des lexèmes composés.

[37] ✓ X le sens est bien trouvé dans le dictionnaire mais il est spécifique ou correspond à une catégorie générale.

	Natural Origin	
ar-Noun-tjArp (تجارة)	an instance of buying or selling; "international trade with Mexico"; "it was a package deal"& 03 04 2ndOrderEntity Agentive Cause Dynamic Purpose SituationType Social	✓
ar-Noun-ljnA (لجنة) ar-Noun-ArtkAb (ارتكاب)	a special group delegated to consider some matter& 03 14 1stOrderEntity Function Group Human Living Natural Origin	✓ X[38]
ar-Noun-nzAE (نزاع) ar-Noun-SrAE (صراع) ar-Noun-n$wb (نشوب)	a state of dissension or open fighting& 03 26 2ndOrderEntity SituationType Static	✓ ✓ X
ar-Noun-m$wrp (مشورة) **ar-Noun-tqdym Alm$wrp (تقديم المشورة)** **ar-Noun-IsdA' Alm$wrp (اسداء المشورة)**	a proposal for an appropriate course of action& 03 10 2ndOrderEntity 3rdOrderEntity Communication Mental Purpose Relation SituationType Social Static	✓ ✓ ✓
ar-Noun-tnsyq (تنسيق)	the regulation of diverse elements into an integrated and harmonious operation& 03 04 2ndOrderEntity Agentive Cause Dynamic Purpose SituationType	✓
ar-Noun-Tlb (طلب)	a formal message requesting something& 03 10 2ndOrderEntity 3rdOrderEntity Communication Mental Purpose Relation SituationType Social Static	✓
ar-Noun-$Ebp (شعبة) **ar-Noun-$Ebp AlIdArp (شعبة الادارة)**	the act of dividing or disconnecting	X

[38] X Le sens n'est pas trouvé dans le dictionnaire mais il est correct

ar-Noun-wvyqp (وثيقة) ar-Noun-mstnd (مستند) ar-Noun-wrqp (ورقة)	writing providing information; esp. of an official nature& 03 10 1stOrderEntity 2ndOrderEntity 3rdOrderEntity Artifact Communication LanguageRepresentation Mental Origin Purpose Relation SituationType Social Static	✓ ✓ ✓
ar-Noun-Hq (حق) ar-Noun-qAnwn (قانون)	an abstract idea of that which is due to a person or governmental body by law or tradition or nature: "they are endowed by their Creator with certain unalienable Rights, that among these are Life, Liberty and the pursuit of Happiness"; "Certain rights can never be granted to the government but must be kept in the hands of the people"- Eleanor Roosevelt; "it is his right to say what he pleases"& 03 07 09 2ndOrderEntity 3rdOrderEntity Dynamic Experience Mental Property SituationType Social Static	✓ ✓
ar-Noun-jhd (جهد) **ar-Noun-b*l jhwd (بذل جهود)**	use of physical or mental energy: "they managed only with great effort"& 03 04 2ndOrderEntity Agentive Cause Dynamic Purpose SituationType Social UnboundedEvent	✓ ✓
ar-Noun-Tfl (طفل)	a human offspring (son or daughter) of any age; "they had three children"; "they were able to send their kids to college"& 03 18 1stOrderEntity Form Function Human Living Natural Object Origin	✓
ar-Noun-AmtHAnAt (امتحانات) ar-Noun-AstErAD (استعراض) ar-Noun-nZr (نظر)	a set of questions or exercises evaluating skill or knowledge& 03 10 2ndOrderEntity Agentive Cause Communication Dynamic Purpose SituationType Social	✓ X X

	UnboundedEvent	
ar-Noun-IdArp (ادارة)	those in charge of running a business& 03 14 1stOrderEntity Group Human Living Natural Origin	✓
ar-Noun-Hkwmp (حكومة)	the exercise of authority over a political unit& 03 04 2ndOrderEntity Agentive Cause Dynamic Purpose SituationType Social UnboundedEvent	✓
ar-Noun-fryq (فريق) ar-Noun-jmAEp (جماعة) ar-Noun-f}p (فئة)	any number of entities (members) considered as a unit& 03 Group	✓ ✓ ✓
ar-Noun-mElwmp (معلومة)	a message received and understood that reduces the recipient's uncertainty& 03 10 2ndOrderEntity 3rdOrderEntity Communication Mental Purpose Relation SituationType Social Static	✓X
ar-Noun-wzyr (وزير)	a person appointed to a high office in the government; "Minister of Finance"& 03 18 1stOrderEntity Form Function Human Living Natural Object Occupation Origin	✓
ar-Noun-bEvp (بعثة)	an operation that is assigned by a higher headquarters; "the planes were on a bombing mission"& 03 04 2ndOrderEntity Agentive BoundedEvent Cause Dynamic Purpose SituationType Social	✓
ar-Noun-mblg (مبلغ)	a quantity of money; "he borrowed a large sum"& 03 1stOrderEntity 21 Function Possession	✓
ar-Noun-slm (سلم)	a treaty to cease hostilities; "peace came on November 11th"& 03 10 1stOrderEntity 2ndOrderEntity 3rdOrderEntity Agentive Artifact	✓

	BoundedEvent Cause Communication Dynamic LanguageRepresentation Mental Origin Purpose Relation SituationType Social Static	
ar-Adj-Trf (طرف)[39] ar-Noun-jz' (جزء)	one of the portions into which something is regarded as divided and which together constitute a whole: "the written part of the exam"; "the finance section of the company"; "the BBC's engineering division"& 03 09 2ndOrderEntity 3rdOrderEntity Dynamic Experience Mental Part Property SituationType Static	✓ ✓
ar-Noun-skAn (سكان)	a group of organisms of the same species populating a given area; "they hired hunters to keep down the deer population"& 03 14 Group	✓
ar-Noun-Emlyp (عملية)	a sustained phenomenon or one marked by gradual changes; "events now in process"; "the process of calcification begins later for boys than for girls"& 03 22 2ndOrderEntity Cause Dynamic Experience Phenomenal Physical SituationType UnboundedEvent	✓X
ar-Noun-msAlp (مسالة) ar-Noun-qDyp (قضية) ar-Noun-mwDwE (موضوع)	some situation or event that is thought about; "he kept drifting off the topic"; "it is a matter for the police"& 03 09 2ndOrderEntity 3rdOrderEntity Dynamic Experience Mental Property SituationType Static	✓ ✓ ✓
ar-Noun-tqryr (تقرير)	the act of informing by verbal report& 03 10 2ndOrderEntity Agentive BoundedEvent Cause Communication Dynamic Purpose SituationType Social	✓

[39] Les unités soulignées sont des unités d'une autre catégorie mais qui partagent aussi le même sens

ar-Noun-twSyp (توصية)	something that recommends (or expresses commendation)& 03 10 2ndOrderEntity 3rdOrderEntity Communication Mental Purpose Relation SituationType Social Static	✓
ar-Noun-xdmp (خدمة)	a stroke (in tennis or squash) that puts the ball in play& 03 04 2ndOrderEntity Agentive BoundedEvent Cause Dynamic Location Manner Physical Purpose SituationType Social	X
ar-Noun-wDE (وضع) ar-Noun-HAlp (حالة)	the general state of things; the combination of circumstances at a given time; "the present international situation is dangerous"; "wondered how such a state of affairs had come about"; "eternal truths will be neither true nor eternal unless they have fresh meaning for every new social situation"- Franklin D.Roosevelt& 03 26 2ndOrderEntity Condition Property SituationType Static	✓ ✓
ar-Noun-qmp (قمة)	a meeting of heads of governments& 03 14 1stOrderEntity Group Human Living Natural Origin	X
ar-Noun-Iqlym (اقليم) ar-Noun-AlArD (الارض)	a territorial possession controlled by a ruling state& 03 1stOrderEntity 21 Function Possession	✓ ✓X
ar-Noun-HmAyp (حماية)	a covering that is intend to protect something from damage& 03 06 1stOrderEntity Artifact Covering Form Function Object Origin	✓
ar-Noun-xbyr (خبير)	a person who performs skillfully& 03 18 1stOrderEntity Form Function Human Living Natural Object Origin	✓
ar-Noun->lyp (الية)	a piece of machinery or a mechanical device; has moving	✓

	parts that perform some function& 03 06 1stOrderEntity Artifact Form Function Instrument Object Origin	
ar-Noun-m$Arkp (مشاركة)	sharing the activities of a group& 03 04 2ndOrderEntity Agentive Cause Dynamic Purpose SituationType Social	✓
ar-Noun-Avr (اثر)	a phenomenon that follows and is caused by some previous phenomenon& 03 19 2ndOrderEntity Cause Dynamic Experience Phenomenal Physical SituationType	✓
Ar-Noun-Eml (عمل) ar-Noun-AmtnAEAt (امتناعات)	something done (usually as opposed to something said); "there were stories of murders and other unnatural actions""\the state of being active; "his sphere of action"; "volcanic activity"	✓ ✓X
ar-Noun-$&wn (شؤون)	a vaguely specified concern; "several matters to attend to"; "it is none of your affair"; "things are going well"& 03 09 2ndOrderEntity Dynamic Mental SituationType Static	✓X
ar-Noun-Almrfq (المرفق)	an addition that extends a main building& 03 06 1stOrderEntity Artifact Form Object Origin Part	✓X
ar-Noun-ITAr (اطار)	a structure supporting or containing something& 03 06 1stOrderEntity Artifact Form Function Object Origin	✓
ar-Noun-dwrA (دورة)	a body of students who are taught together; "early morning classes are always sleepy""\education imparted in a series of lessons or class meetings; "he took a course in basket weaving"; "flirting is not unknown in college classes"	✓X

ar-Noun-m&ssp (مؤسسة) ar-Noun-wkAlp (وكالة)	an organization founded for a specific purpose& 03 14 1stOrderEntity Function Group Human Living Natural Origin	✓ ✓
ar-Noun-hdf (هدف) ar-Noun-gAyp (غاية)	the goal intended to be hit& 03 09 2ndOrderEntity 3rdOrderEntity Cause Dynamic Experience Mental Property Purpose SituationType Static Stimulating	✓ ✓
Ar-Noun-bld (بلد) ar-Adj-qTry (قطري)	the territory occupied by a nation; "he returned to the land of his birth"& 03 15 1stOrderEntity Function Part Place	✓ X
ar-Noun-$Eb (شعب)	the body of citizens of a state or country; "the Spanish people"& 03 14 1stOrderEntity Group Human Living Natural Origin	✓
ar-Noun-wZyfp (وظيفة)	the position where something or someone (as a guard or sentry) stands or is assigned to stand: "a sentry station"& 03 15 1stOrderEntity Function Place	✓

Annexe 6 : Les noms arabes (lexèmes composés) existant dans les cliques désambigüisées au regard des 3 sens (3 langues) :

Noms arabes (lexèmes composés) existant dans les cliques désambigüisées au regard des 3 sens (3 langues)	**Sens commun extrait au regard des 3 sens**	**Validation du sens extrait**
Ar-Noun-$rTy (شرطي)	a member of a police force& 03 18 1stOrderEntity Form Function Human Living Natural Object Origin	✓
Ar-Noun-rfh (رفه)	the state of being happy and healthy and prosperous& 03 26 2ndOrderEntity Condition Property SituationType Static	✓
ar-Noun-nA}b r}ys (نائب رئيس)	one ranking below or serving in the place of a chairman& 03 18 1stOrderEntity Form Function Human Living Natural Object Occupation Origin	✓

Annexe 7 : Les verbes arabes existant dans les cliques désambigüisées au regard des 3 sens (3 langues) :

Verbes arabes existant dans les cliques désambigüisées au regard des 3 sens (3 langues)	**Sens commun extrait au regard des 3 sens**	**Validation du sens extrait**
ar-Verb-zAd (زاد)	The amount of work increased"& 2ndOrderEntity 30 Dynamic Quantity SituationType	✓
ar-Verb-AEtmd (اعتمد)	of theories, ideas, policies, strategies or plans& 2ndOrderEntity 31 40 Agentive BoundedEvent Cause Dynamic Mental Possession Purpose SituationType	X
ar-Verb-tTbq (تطبق) ar-Verb-tnf* (تنفذ)	"apply a principle"; "practice a religion"& 2ndOrderEntity 41 Agentive Cause Dynamic SituationType Social	✓ ✓
ar-Verb-A*n (اذن) ar-Verb-smH (سمح)	grant authorization or clearance for& 2ndOrderEntity 32 Agentive BoundedEvent Cause Communication Dynamic Purpose SituationType Social	✓ ✓
ar-Verb-Artkb (ارتكب)	perform an act, usually with a negative connotation: "perpetrate a crime"; "pull a bank robbery"& 2ndOrderEntity 41 Agentive Cause Dynamic SituationType Social	✓
ar-Verb-$Ark (شارك)	be involved in something& 2ndOrderEntity 41 Agentive Cause Dynamic SituationType Social	✓
ar-Adj-mqrrp (مقررة) **ar-Verb/PRP-ynS ElY (ينص على)**	regard something as probable or likely& 2ndOrderEntity 31 Agentive BoundedEvent Cause Dynamic Mental Purpose SituationType	X X

ar-Verb-qym (قيم)	judge the worth of something& 2ndOrderEntity 31 Agentive BoundedEvent Cause Dynamic Mental Purpose SituationType	X
ar-Verb-sAEd (ساعد)	lend support, give aid to	✓
ar-Noun-tHqyq (تحقيق)	reach a destination or a specific point or level; "We could make Detroit by noon"; "The water reached the doorstep"; "We barely made the plane"& 2ndOrderEntity 38 Dynamic Location Physical SituationType	✓
ar-Verb-rfE (رفع)	The amount of work increased"& 2ndOrderEntity 30 Dynamic Quantity SituationType	✓
Ar-Verb-tDm (تضم) **ar-PRP/ART-fy *lk (في ذلك)**	make sense of a language; "She understands French"; "Can you read Greek?"& 2ndOrderEntity 31 Dynamic Experience Mental Property SituationType Static	X X
ar-Verb-y$kl (يشكل)	"This money is my only income"; "The stone wall was the backdrop for the performance"; "These constitute my entire belonging"; "The children made up the chorus"; "This sum represents my entire income for a year"& 2ndOrderEntity 42 Property SituationType Static	✓X
ar-Verb-qrr (قرر)	reach, make, or come to a decision about something& 2ndOrderEntity 31 Agentive BoundedEvent Cause Dynamic Mental Purpose SituationType	✓
ar-Verb-hn} (هنا)	express congratulations& 2ndOrderEntity 32 41 Agentive BoundedEvent Cause Communication Dynamic Purpose SituationType Social UnboundedEvent	✓

ar-Verb-tSdr (تصدر) ar-Verb-Hdv (حدث)	"make a big stink"; "make revolution"; "do harm"; "do wrong"& 2ndOrderEntity 36 BoundedEvent Cause Condition Dynamic Existence Physical SituationType	X X
ar-Verb-AElm (اعلم)	"I advised him that the rent was due"& 2ndOrderEntity 32 41 Agentive Cause Communication Dynamic SituationType Social UnboundedEvent	✓
ar-Verb-drj (درج)	"She incorporated his suggestions into her proposal"& 2ndOrderEntity 30 35 Cause Dynamic Location SituationType Static	✓X
ar-Adj/PRP-mlzmp b (ملزمة ب) **ar-Adj/PRP-AlmrtbTp b (المرتبطة ب)**	make a logical or causal connection& 2ndOrderEntity 31 Agentive BoundedEvent Cause Dynamic Mental Purpose SituationType	✓ ✓
ar-Verb-tHtl (تحتل)	be present in; be inside of & 2ndOrderEntity 42 Location Relation SituationType Static	✓X
ar-Verb-AqtrH (اقترح)	make a proposal, declare a plan for something& 2ndOrderEntity 32 Agentive BoundedEvent Cause Communication Dynamic Purpose SituationType	✓
ar-Noun/PRP-Hd mn (حد من) ar-Noun-xfD (خفض) ar-Noun-tDyyq (تضييق)	as of sauces; "The cook reduced the sauce by boiling it for a long time"& 2ndOrderEntity 30 Agentive BoundedEvent Cause Dynamic Quantity SituationType	✓X ✓X ✓X
ar-Verb-tlqy (تلقي)	get something; come into possession of; "receive payment"; "receive a gift"; "receive letters from the front"& 2ndOrderEntity 40 Dynamic Possession SituationType	✓
ar-Verb-AwSt (اوصت)	express a good opinion of&	✓

	2ndOrderEntity 31 32 Agentive BoundedEvent Cause Communication Dynamic Mental Purpose SituationType	
ar-Noun-tEzyz (تعزيز)	make strong or stronger& 2ndOrderEntity 30 Cause Dynamic SituationType	✓
ar-Verb-qdm (قدم)	present formally& 2ndOrderEntity 40 Agentive BoundedEvent Cause Dynamic Possession Purpose SituationType	✓
ar-Verb-tstxdm (تستخدم)	incapable of being made smaller or simpler: "an irreducible minimum"; "an irreducible formula"	X
ar-Verb-trAEy (تراعي) ar-Verb-tHtrAm (تحترم)	show respect towards& 2ndOrderEntity 31 41 Agentive BoundedEvent Cause Dynamic Mental Purpose SituationType Social	X ✓

Annexe 8 : script extractLarousse.pl d'extraction de cliques à partir de dictionnaire Larousse:

```
#---------------------------------------- modules

use strict;
use LWP::Simple;
use IO::Handle;
use URI::Escape;
use DB_File; #Bundle-Tie-DB_File-SplitHash

#****************~ use utf8;

STDOUT->autoflush();

#---------------------------------------- variables globales
my $larousseDomain="http://www.larousse.fr";
my $larousseIndexes="$larousseDomain/dictionnaires/";
my %cache;
my $path=".";
my $cachePath=$path."/cache";

if (! -d $cachePath) {
        mkdir $cachePath;
}

#---------------------------------------- PARAMETRES

# traduction des langues pour le Larousse

my %languages=( "fr"=>"francais", "en"=>"anglais", "es"=>"espagnol", "it"=>"italien",
"de"=>"allemand");

# traduction des catégories pour le Larousse

my %categories=( "en-N"=>"noun", "fr-N"=>"nom", "de-N"=>"(der|die|das)", "it-
N"=>"sostantivo", "es-N"=>"sustantivo","fr-V"=>"verbe", "en-V"=>"verb" );

# liste des paires présentes dans le Larousse

my %pairs=("de-en"=>1,"de-es"=>1,"de-it"=>1,"de-fr"=>1,"fr-en"=>1,"fr-es"=>1,"fr-
it"=>1,"fr-de"=>1,"en-fr"=>1,"en-es"=>1,"en-it"=>1,"en-de"=>1,"es-fr"=>1,"es-
en"=>1,"es-de"=>1,"it-fr"=>1,"it-en"=>1,"it-de"=>1,"fr-ar"=>1);
my $compound=0;
                        # indique s'il faut rechercher les mots composés comme entrées du
Larousse (-> produit beaucoup de bruit)
my @langues=("de","en","es","fr","it");
```

```
my $diceLimitForCompletion=0.3;                                                 # indique le seuil de dice entre deux unités pour établir un lien de correspondance entre elles
my $interSizeLimitForCompletion=2;                                              # indique la taille min de l'intersection entre deux unités pour établir un lien de correspondance entre elles
my $diff=0;
                                        # définit une marge de tolérance pour accepter une nouvelle unité dans une clique. Si =0, pas de tolérance, l'unité doit avoir toutes les correspondances
my $diceMean4enrichment=0.2;
        # définit le seuil que le dice moyen doit dépasser pour rajouter une unité de langue source à une clique constituée
my $forceDownload=0;
my $sizeMin=4;
                                        # nombre min de langues dans les cliques
my $sizeMax=4;
                                        # nombre max de langues dans les cliques
my @languagesToCompleteByTransitivity=("it","fr");      # tous les couples parmi ces langues seront complétés par transitivité
my $seed="fr-N-économie";                                                       # note : attention, le fichier doit être encodée en iso-latin-1



#---------------------------------------- FIN PARAMETRES
my %visitedUrl;
my %corr;
my $cliqueMin=1; # paramètre indiquant qu'on démarre les itérations à partir de cliques "minimales" comportant une unité pour chaque langue
my $verbose=0;
my ($sourceLang,$cat,$lemma);

if ($seed=~/^(.*?)-(.*?)-(.*)$/) {
        ($sourceLang,$cat,$lemma)=($1,$2,$3);
} else {
        print "La forme étudiée doit être de la forme LANGUE-CAT-LEMME\n";
        die;
}

print "\n\n===> Chargement des correspondances\n";
if (-f "$path/cliques.$seed.txt" && !$forceDownload) {
        open(IN,"<","$path/cliques.$seed.txt");
        my $i;
        while (<IN>) {
                if (/(.*)<=>(.*)/) {
                        $corr{$1}={} unless exists($corr{$1});
                        $corr{$1}{$2}=1;
                        $corr{$2}={} unless exists($corr{$2});
                        $corr{$2}{$1}=1;
                        $i++;
```

```
                }
        }
        print "$i correspondances lues à partir de $path/cliques.$seed.txt";
} else {
        downloadLarousse($seed);
}
```

 # affichage des correspondances obtenues

```
foreach my $unit (sort keys %corr) {
        foreach my $unit2 (sort keys %{$corr{$unit}} ) {
                $verbose && print $unit."<=>".$unit2."\n";
        }
}
print "\n\n===> Complétion des liens par transitivité
(@languagesToCompleteByTransitivity)\n";

addTransitiveLinks(@languagesToCompleteByTransitivity);

print "\n\n===> Extraction des cliques pour $seed\n";
my @cliques=entry_clique($seed,$sizeMin,$sizeMax);
my $n=@cliques;
print "\n\n$seed --> $n cliques obtenues\n";
print "\n\n===> Calcul des quasi-synonymes de $seed\n";
my @syns=calcSyn($seed);
print "\n\n===> Extraction des cliques pour les synonymes (@syns)\n";

foreach my $syn (@syns) {
        my @cliques_syn=entry_clique($syn,$sizeMin,$sizeMax);
        my $n=@cliques_syn;
        print "\n\n$syn --> $n cliques obtenues\n";
        push(@cliques,@cliques_syn);
}

print "\n\n===> Première passe de clusterisation :\n";
my @clusters=mergeClique(\@cliques);

print "\n\n===> Deuxième passe de clusterisation :\n";
my @clusters2=mergeClique(\@clusters);
```

 # enregistrement des clusters finaux

```
open(OUT,">","clusters.$seed.txt");
foreach my $cluster (@clusters2) {
        print OUT "(".join(", ",@{$cluster}).")\n";
}
close(OUT);

print "\n\n===> calcul des cliques associées :\n";
```

```
associateCliques($seed,@clusters2);

#---------------------------------------------------------- fonctions

#------------------------------------------- fonctions pour l'extraction des données du Larousse

# downloadLarousse()
# Fonction dédié aux téléchargement des données du Larousse autour d'une entrée
# Entrées
# arg1 : entrée du dico p.ex. "fr-N-extrémité"

# Sorties
# Remplissage du tableau %corr avec :
#       - les traductions immédiates de l'entrée
#       - les traductions de traductions
#       - les traductions des pseudo-synonymes (trad de trad en langue source partageant un certain nombre de trad avec l'entrée)
#       - les traduction de traduction des pseudo-synonymes
# Par ailleurs les pages téléchargées sont conservées en cache

sub downloadLarousse {
        my $seed=shift;
        tie(%cache,"DB_File","cache.db");

        # 1. recherche des traductions immédiates
        print "\n\n1. Recherche des traductions immédiates de $seed\n\n";
        my @result=extractLarousse($seed);
        print "->".($#result+1)." équivalents de $seed : ".join(", ",@result)."\n";

        foreach my $unit (@result) {
                $corr{$seed}={} unless exists($corr{$seed});
                $corr{$seed}{$unit}=1;
                $corr{$unit}={} unless exists($corr{$unit});
                $corr{$unit}{$seed}=1;
        }

        # 2. recherche des traductions de traductions
        print "\n\n2. Recherche des traductions de traductions\n\n";
        foreach my $unit (@result) {
                my @result2=extractLarousse($unit);
                print "->".($#result2+1).". équivalents de $unit : ".join(", ",@result2)."\n";
                foreach my $unit2 (@result2) {
                        $corr{$unit}={} unless exists($corr{$unit});
                        $corr{$unit}{$unit2}=1;
                        $corr{$unit2}={} unless exists($corr{$unit2});
                        $corr{$unit2}{$unit}=1;
                }
```

```
	}

	# 3. recherche des traductions des retro-traductions en français
	my @retroTranslations=grep {/^$sourceLang/} keys %corr;
	my $nbRetro=@retroTranslations;
	print "\n\n3. Recherche des traductions des $nbRetro retro-traductions en
français\n\n";
	foreach my $unit (@retroTranslations) {
		my @result2=extractLarousse($unit);
		print "->".($#result2+1).". équivalents de $unit : ".join(", ",@result2)."\n";
		foreach my $unit2 (@result2) {
			$corr{$unit}={} unless exists($corr{$unit});
			$corr{$unit}{$unit2}=1;
			$corr{$unit2}={} unless exists($corr{$unit2});
			$corr{$unit2}{$unit}=1;

			# on recherche les traduction de traduction des retro-traductions
			my @result3=extractLarousse($unit2);

			if (@result3 != 0) {
				print "Nouveaux équivalents de $unit2 : ".join(",
",@result3)."\n";
				foreach my $unit3 (@result3) {
					$corr{$unit2}{$unit3}=1;
					$corr{$unit3}={} unless exists($corr{$unit3});
					$corr{$unit3}{$unit2}=1;
				}
			}
		}
	}

	# 4. recherche des traductions pour tous les synonymes français, afin de les
faire rentrer dans les cliques du français
	print "\n\n4. Complétion itérative des traductions (et traductions de traductions)
pour tous les 'quasi-synonymes' français\n\n";
	my $continue=1;
	# on continue la boucle tant que l'on obtient de nouveaux synonymes à traiter
(la liste des synonymes se  complète itérativement)
	while ($continue) {
		$continue=0;
		my @syns=calcSyn($seed);
		print "->".($#syns+1)." quasi-synonymes identifiés\n\n";
		my $i;
		 foreach my $unit (@syns) {
			$i++;
			my @result2=extractLarousse($unit);
			my $n=@result2;
			if ($n!=0) {
				$continue=1;
```

```
                                        print "\nSynonyme $i - $unit - $n nouveaux
équivalents  : ".join(", ",@result2)."\n\n";
                                        foreach my $unit2 (@result2) {
                                                $corr{$unit}={} unless
exists($corr{$unit});
                                                $corr{$unit}{$unit2}=1;
                                                $corr{$unit2}={} unless
exists($corr{$unit2});
                                                $corr{$unit2}{$unit}=1;
                                                # on recherche les traduction
de traduction des synonymes
                                                my
@result3=extractLarousse($unit2);
                                                if (@result3 != 0) {
                                                        print "Nouveaux
équivalents de $unit2 : ".join(", ",@result3)."\n";
                                                        foreach my $unit3
(@result3) {

$corr{$unit2}{$unit3}=1;

$corr{$unit3}={} unless exists($corr{$unit3});

$corr{$unit3}{$unit2}=1;
                                                        }
                                                }
                                        }
                                }
                        }
                        untie(%cache);
}

# téléchargement des équivalents directs pour une entrée
# Entrées
# arg1 : entrée du dico p.ex. "fr-N-extrémité"
# Sorties
# return : @liste des équivalents trouvés dans Larousse p.ex.
("end_en_N","tip_en_N","estremità_it_N")

sub extractLarousse {
        my $entry=shift;
        my %equivalents=();
        my ($sourceLang,$cat,$lemma);
        if ($entry=~/^(.*?)-(.*?)-(.*)$/) {
                ($sourceLang,$cat,$lemma)=($1,$2,$3);
        } else {
                print "$entry doit être de la forme LANGUE-CAT-FORME\n";
                die;
```

```
	}
	my $sourceLanguage=$languages{$sourceLang};
	my $category=$categories{$sourceLang."-".$cat};

	if (!$compound && $lemma=~/ /) {
		$verbose && print "Requête abandonnée pour le mot composé $lemma\n";
		return ();
	}

	foreach my $lang (keys %languages) {
		if ($lang ne $sourceLang && exists($pairs{"$sourceLang-$lang"})) {
			$verbose && print "Recherche d'équivalent en '$lang' pour
'$entry'\n";
			my $language=$languages{$lang};
			my $indexUrl=$larousseIndexes.$sourceLanguage."-
".$language."/".uri_escape($lemma);
			if (!exists($visitedUrl{$indexUrl})) {
				$verbose && print "Lecture de la page $indexUrl...\n";
				my $indexPage=getPage($indexUrl);
				if (! $indexPage ) {
					print "Page d'index non trouvée !\n";
					die;
				} else {
					while ($indexPage=~/<ul
class="list_resultats">(.*?)<\/ul>/sg) {
						my $links=$1;
						while ($links=~/href="(.*?)"/g) {
							my $link=$1;
							$link=~s/#[^\/]*//;	# suppression du
lien vers ancre
							my
$articleUrl=$larousseDomain.$link;
							if (! exists($visitedUrl{$articleUrl})) {
								$verbose && print "Lecture de
la page $articleUrl...\n";
								my
$articlePage=getPage($articleUrl);
								if (! $articlePage ) {
									print "Article non trouvé
à l'adresse $articleUrl\n";
								} else {
									my
@subArticles=split(/<h1.*?>/,$articlePage);
									shift (@subArticles); #
on jette ce qui précède le premier <h1>
									if (@subArticles==0) {
										print "Pas
d'article à traiter dans $articlePage\n";
									}
```

```
                                                                    foreach my $subArticle
(@subArticles) {
                                                                        if
($subArticle=~/^\s*$lemma\s*<\/h1>/gi) {
                                                                            if
($subArticle=~/<span class="CategorieGrammaticale".*?>\s*(.*?)\s*<\/span>/s) {
        my $catPage=$1;
                                                                            if
($catPage=~/\b$category\b/i) {
        while ($subArticle=~/<span class="Traduction".*?>\s*(.*?)\s*<\/span>(.*)/s) {
                my $content=$1;
                my $rest=$2;
                my $openSpan=($content=~s/<span/<span/g);
                my $closeSpan=($content=~s/<\/span/<\/span/g);
                while ($openSpan>$closeSpan) {
                        $rest=~/^(.*?)<\/span>(.*)/s;
                        $content=$content."</span>".$1;
                        $rest=$2;
                        $openSpan=($content=~s/<span/<span/g);
                        $closeSpan=($content=~s/<\/span/<\/span/g);
                }
                $subArticle=$rest;

                my $translations=$content;
                $translations=~s/<span class="Genre"[^>]*>[^<]*,[^<]*<\/span>/,/sig; #
conservation de la virgule qui sépare deux équivalents et qui est parfois collée au genre
                $translations=~s/<span
class="(Genre|Metalangue.?|Locution.?|Traduction2)".*?>.*?<\/span>//sig;
                $translations=~s/<.*?>//g; # suppression des hypersliens
```

```
                $translations=~s/\(.*?\)\s*//g; # suppression des parenthèses
                $translations=~s/ / /g; # remplacement des  
                $translations=~s/^\s*|\s*$//g; # trim
                $translations=~s/\n//g; # suppressuib des retours chariots

                my @translations=split(/\s*,\s*/,$translations);
                foreach my $trans (@translations) {
                        $trans=~s/[\x0A\x0D]//g;
                        if ($trans!~/^\s*$/) {
                                $trans=~s/  / /g;
                                $equivalents{"$lang"."-$cat"."-$trans"}=1;
                                $verbose && print "$lang"."-$cat"."-$trans\n";
                        }
                }
        }
                                                                                }
else {
        $verbose && print "Catégorie $catPage -> abandonné (<> /$category/)\n";
                                                                                }
                                                                        } else {
        print "Page $articleUrl : catégorie non trouvée pour $lemma !\n";
                                                                                }
                                                                        }
                                                                }
                                                        }
                                                }
                                        }
                                }
                        } else {
                                $verbose && print "$indexUrl déjà parcourue\n";
                        }
```

```
                }
        }
        my @equivalents=keys %equivalents;
        my $n=@equivalents;
        $verbose && print $n." traductions trouvées pour $entry !\n";
        return @equivalents;
}

# fonction de téléchargement des pages avec gestion du cache
# met à jour le cache et le hachage %visitedUrl
# renvoie la page complète
sub getPage {
        my $url=shift;
        my $page;
        my $fileName=$url;
        $fileName=~s/https?:..//g;
        $fileName=~s/[\\\/ ?:]/_/g;

        if (-f $cachePath."/".$fileName) {
                $verbose && print "Page $url trouvée en cache\n";
                open(IN,$cachePath."/".$fileName);
                $page=join("",<IN>);
                close(IN);
                $visitedUrl{$url}=1;
                return $page;
        } else {
                my $page=get($url);
                if ($page) {
                        # transformation de l'url en nom de fichier

                        open(OUT,">",$cachePath."/".$fileName) or die "Impossible
d'enregistrer $cachePath/$fileName\n";
                        print OUT $page;
                        close(OUT);
                        $verbose &&  print "Téléchargement de page $url et sauvegarde en
cache\n";
                        $cache{$url}=$fileName;
                        $visitedUrl{$url}=1;
                        return $page;
                } else {
                        print "Pas de page à l'url $url\n";
                        $visitedUrl{$url}=1;
                        return "";
                }
        }
}

# calcul des synonymes
# deux synonymes sont deux unités de la même langue partageant un sens en commun
```

```
# deux synonymes doivent donc avoir des traductions communes dans toutes les
langues

# pour chaque mot du lexique, et pour chaque langue, on examine le nombre de
correspondant communs
sub calcSyn {
	my $entry=shift;
	$entry=~/^(\w\w)/;
	my $entryLang=$1;
	my @syns;

	# d'abord on cherche le lexique pour la langue source
	my %lexicon;
	foreach my $corr (keys %corr) {
		if ($corr=~/^($entryLang)-/) {
			$lexicon{$corr}=1;
		}
	}

	my %commons;
	foreach my $unit (keys %lexicon) {
		if ($unit ne $entry) {
			$commons{$unit}={};
			foreach my $corr (keys %{$corr{$entry}}) {
				$corr=~/^(\w\w)-/;
				my $langCorr=$1;
				if (exists($corr{$unit}{$corr})) {
					$commons{$unit}{$langCorr}++;
				}
			}
			my $isSyn=1;
			my $nbCom=0;
			foreach my $lang (grep {!/$entryLang/} @langues) {
				if (exists($commons{$unit}{$lang})) {
					$nbCom+=$commons{$unit}{$lang};
				} else {

					$isSyn=0;
				}
			}

			if ($nbCom>=2 or $isSyn)  {
				push(@syns,$unit);
				print "Synonyme de $entry : ".$unit."(nbCom=$nbCom,
dice=".dice($unit,$entry).")\n";
			}
		}
	}
	return @syns;
```

```
}

#---------------------------------------------- fonctions pour l'extraction des cliques

# Extraction des cliques contenant une certaine entrée
# Les correspondances doivent au préalable être enregistrées dans %corr
# Entrées :
#       - arg1 : string de la forme LANGUE-CAT-FORME
#       - arg2 : taille minimale des cliques (en nombre de langues)
#       - arg3 : taille maximale des cliques (en nombre de langues)
# Sorties :
#       - écriture du fichier cliques.$entry.txt qui contient les cliques trouvées

sub entry_clique {
        my $entry = shift (@_);
        my $sizeMin= shift (@_);
        my $sizeMax= shift (@_);
        my $sourceLang;
        if ($entry=~/^(.*?)-.*?-.*$/) {
                $sourceLang=$1;
        } else {
                print "$entry doit être de la forme LANGUE-CAT-FORME\n";
                die;
        }
        my $file_clique = $path."/cliques.$entry.txt";

        my @langues=("fr","en","es","it","de");

        print "Création de $file_clique\n";


        my %cliques_max= ();    #hash permettant de stocker les cliques finales

        # ETAPE 0 : enregistrement des correspondances dans le fichier de sortie

        print "Enregistrement des correspondances dans $file_clique\n";
        open (OUT,">",$file_clique);
        print OUT "Sauvegarde des correspondances :\n";
        foreach my $unit (sort keys %corr) {
                foreach my $unit2 (sort keys %{$corr{$unit}} ) {
                        # pour les correspondances réelle
                        if ($corr{$unit}{$unit2}==1) {
                                print OUT $unit."<=>".$unit2."\n";
                        # pour les correspondances transitives
                        } else {
                                print OUT $unit."<->".$unit2."\n";
                        }
                }
        }
```

```
print OUT "\n";
```

 # ETAPE 1 : consitution de la liste des candidats à l'appartenance dans les cliques autour de $entry

 # Le hachage de hachages ci-dessous enregistre les candidats à l'extension pour chaque langue

```
my %candidats;
foreach my $lang (@langues) {
	$candidats{$lang}={};
}
```

 # tous les correspondants de $entry sont candidats

```
foreach my $corr (keys %{$corr{$entry}}) {
	if ($corr=~/^(\w\w)-/) {
		my $lang=$1;
		$candidats{$lang}{$corr}=1;
		$verbose && print "Ajout du candidat $corr\n";
	} else {
		die "le mot $corr est mal formé\n";
	}
}
```

#pour le moment, pour $sourceLang, seul $entry est candidat à l'appartenance dans une clique (pour la construction des cliques minimales)

```
$candidats{$sourceLang}{$entry}=1;
```

 # ETAPE 2 : constitution de l'ensemble des cliques de départ (singleton avec $entry si $cliqueMin=0, ou bien l'ensemble des cliques minimales complètes, i.e. avec toutes les langues)

```
my @cliques=();     #liste des cliques de départ, que l'on essaiera d'augmenter
```

 # si $sizeMin>1 création de cliques minimales comportant toutes les combinaisons de $sizeMin langues

```
if ($sizeMin>1) {
	my @combin=combin($sizeMin,@langues);
	foreach my $comb (@combin) {
		push(@cliques,cliquesMin(\%candidats,@{$comb}));
	}
} else {
# sinon on initialise seulement avec l'entrée d'origine
	push (@cliques, [$entry]);
}
$verbose && print "Cliques de départ :\n";
foreach my $c (@cliques) {
	$verbose && print "(".join(", ",@{$c}).")\n";
}
```

```
	# ETAPE 2 bis : Ajout des pseudo-synonymes de $entry (i.e. toutes les unités en $sourceLang obtenues par aller-retour)
	foreach my $candidat (keys %{$corr{$entry}}) {
		if ($candidat=~/^(\w\w)-/) {
			my $lang=$1;
			# tous les "pseudo-synonymes", correspondant en langue source des correspondants sont également candidats
			foreach my $syn (keys %{$corr{$candidat}}) {
				if ($syn =~/^($sourceLang)-/) {
					$candidats{$sourceLang}{$syn}=1;
					$verbose && print "Ajout du pseudo-synonyme $syn\n";
				}
			}
		}
	}

	# ETAPE 3 : augmentation itérative des cliques
	# Chaque clique augmentée est ajoutée à @cliques.
	# Quand une clique n'est plus augmentable est ajoutée à %cliques_max (comme clé résultant du tri de ses éléments)
	# on initialise le compteur d'itérations
	my $i=1;
	# si $sizeMin == $sizeMax on a déjà que des cliques maximales. On ne fera pas d'augmentation itérative

	if ($sizeMin == $sizeMax) {
		foreach my $adr_clique (@cliques) {
			my $cle = cle_clique ($adr_clique);
			print OUT "(".join(", ",@{$adr_clique}).")\n";
			print "AJOUT DE LA CLIQUE DE CLE : $cle\n";
			$cliques_max{$cle} = $adr_clique;
		}
		@cliques=();
	}

	#tant que la liste cliques n'est pas vide, on essaie d'augmenter

	while (@cliques != 0){
		my %new_cliques =();	#tableau de stockage des cliques trouvées
		my $n=@cliques;
		print "\n\nItération $i - $n cliques\n";


		foreach my $adr_clique (@cliques) {
			my $notmax =0;
			my $t=@{$adr_clique};
			foreach my $lang (@langues) {
```

```
                        #on teste l'appartenance à la clique pour chaque candidat

                        my $c=keys %{$candidats{$lang}};
                        foreach my $candidat (keys %{$candidats{$lang}}){
                                if (!membre($candidat,@{$adr_clique})) {

                                        # on vérifie que la clique étendue est bien
une clique, et que le nombre de langues ne dépasse pas $sizeMax

                                        if
(isExtendedClique($candidat,$lang,@{$adr_clique}) && langList($candidat,
@{$adr_clique}) <=$sizeMax ) {
                                                $notmax =1;
                                                my $adr_new_clique = [];
                                                push
(@{$adr_new_clique},($candidat, @{$adr_clique}));
                                                my $cle = cle_clique
($adr_new_clique);
                                                if (!exists ($new_cliques{$cle})) {
                                                        $new_cliques{$cle} =
$adr_new_clique;
                                                }
                                                $verbose && print "ajout de $candidat
à la clique ".@{$adr_clique}."\n";
                                        } else {
                                                $verbose && print "non ajout de
$candidat à la clique @{$adr_clique}\n";
                                        }
                                }
                        }
                }

                # si la clique est maximale, on la stocke dans %clique_max

                if ($notmax ==0){
                        my $cle = cle_clique ($adr_clique);
                        if (!exists ($cliques_max{$cle})) {
                                print OUT "(".join(", ",@{$adr_clique}).")\n";
                                print "AJOUT DE LA CLIQUE DE CLE : $cle\n";
                                $cliques_max{$cle} = $adr_clique;
                        }
                }
        }
        #on copie les cliques de new_clique dans la liste de cliques pour les
tester
        #quand new_clique sera vide, on sortira de la boucle

        @cliques=();
        foreach my $key (keys %new_cliques) {
```

```
                push (@cliques,$new_cliques{$key});
            }

            $i++;
        }

        print OUT "calcul en $i itérations\n";
        close (OUT);

        my @result=();
        foreach my $key (keys %cliques_max) {
            push (@result,$cliques_max{$key});
        }
        return @result;
        }
```

 # fonction récursive renvoyant sous forme de liste toutes les combinaisons de $n éléments extraits d'une liste @list

```
sub combin {
        my $n=shift;
        my @list=@_;
        my @combin=();
        if ($n>@list) {
            print "$n dépasse la taille de la liste";
            return @combin;
        }
        # terminaison de la récursivité : la liste comprend $n éléments
        if ($n == @list) {
            push(@combin,[@list]);
            return @combin;
        }
        # deuxième cas de terminaison : $n=1
        if ($n==1) {
            foreach my $elt (@list) {
                push(@combin,[$elt]);
            }
            return @combin;
        }
        # récursivité : on calcule toutes les sous combinaisons de taille n-1 avec le premier élément, plus toutes les combinaisons de taille n sans le premier élément
        my $elt =shift(@list);
        my @subCombin=combin($n-1,@list);
        # ajout de toutes les combinaisons de $n-1 élément, en rajoutant $e ensuite
        foreach my $comb (@subCombin) {
            push(@combin,[$elt,@{$comb}]);
        }
        # ajout de toutes les combinaisons de $n élément, mais sans $e
        push(@combin,combin($n,@list));
        return @combin;
```

```
}

# renvoie la liste des langues dans une clique ou un cluster (liste d'unités)
sub langList {
	my @list;
	my %langs;
	foreach my $unit (@list) {
		$unit=~/^(\w\w)/;
		$langs{$1}=1;
	}
	return keys %langs;
}

# Fonction opérant le filtrage des cliques enregistrée dans un fichier
# arg1 : l'entrée qui doit figurer dans les cliques
# arg2 : nombre minimal d'unité en langue source dans chaque clique (synonymes)
# arg3 : nbLang : nombre minimal de langues représentées dans la clique

sub cliqueFiltering {
	my $cliqueFile=shift;
	my $entry=shift;
	my $nbSourceLangMin=shift;
	my $nbLangMin=shift;
	my ($IN,$OUT);

	open($IN,$cliqueFile) or die "Impossible d'ouvrir $cliqueFile\n";
	open($OUT,">",$cliqueFile.".$nbSourceLangMin-$nbLangMin.txt");

	my @cliques;
	$entry=~/^(\w\w)/;
	my $sourceLang=$1;
	while (<$IN>) {
		if (/$entry/ && /\((.*)\)/) {
			my @clique=split(/, /,$1);
			my $nbSourceLang=0;
			my %lg;
			if (@clique) {
				foreach my $unit (@clique) {
					if ($unit=~/^(\w\w)-/) {
						my $lang=$1;
						$lg{$lang}=1;
						if ($lang eq $sourceLang) {
							$nbSourceLang++;
						}
					} else {
						print "$unit : unité illisible\n";
					}
				}
				my $nbLang=(keys %lg);
```

```
                                if ($nbSourceLang>=$nbSourceLangMin && $nbLang
>=$nbLangMin) {
                                        print $OUT "(".join(", ",@clique).")\n";
                                }
                        }
                }
        }
        close($IN);
        close($OUT);
}
```

 # fonction récursive pour le calcul des cliques minimales de départ, comportant une unité dans chaque langue
 # la fonction est appelée récursivement sur l'ensemble de langue réduit
 # arg1 : l'adresse du tableau %candidats contenant les candidats à l'extension pour chaque langue
 # arg2 : la liste de langue à traiter

```
sub cliquesMin {
        my $adrCand=shift;
        my @langues=@_;

        my $lang=shift @langues;
        my @candidats=keys %{$adrCand->{$lang}};
        my @newCliquesMin=();

        # fin de la récursion : chaque candidat donne un singleton
        if (@langues == 0) {
                foreach my $candidats (@candidats) {
                        push (@newCliquesMin,[$candidats]);
                }
                return @newCliquesMin;
        }

        # appel récursif sur l'ensemble réduit de langues
        my @cliquesMin=cliquesMin($adrCand,@langues);

        # pour chaque clique_min de l'ensemble réduit
        foreach my $adr_clique (@cliquesMin) {
                # pour chaque candidat de la langue courante
                foreach my $candidat (@candidats) {
                        # si en complétant avec $candidat on a toujours une  clique, on
enregistre la clique étendue
                        if (isExtendedClique($candidat,$lang,@{$adr_clique})) {
                                my $newAdr_clique=[$candidat,@{$adr_clique}];
                                push (@newCliquesMin,$newAdr_clique);
                        }
                }
        }
```

```
	return @newCliquesMin;
}

# vérifie que l'ajout d'un candidat à une clique permet d'obtenir une clique étendue
sub isExtendedClique {
	my $candidat=shift;
	my $lang= shift;
	my @clique=@_;

	my $is_clique=1;
	my $nbMatches=0;
	my $nbForeigns=0;
	foreach my $unit (@clique) {
		if ($unit!~/$lang-/) {
			$nbForeigns++;
			$nbMatches++ if exists($corr{$unit}{$candidat});
		}
	}
	# une clique étendue est validée s'il y a autant de corresp que d'unités étrangères
	# pour les cliques de taille supérieure à 5, on accepte néanmoins un certain différentiel
	if (@clique>=5) {
		$is_clique= ($nbMatches>=$nbForeigns-$diff);
	} else {
		$is_clique= ($nbMatches==$nbForeigns);
	}
	return $is_clique;
}

#teste si le mot ne fait pas déjà partie de la clique

sub membre {
	my $e = shift @_;
	my @l = @_;
	foreach my $element(@l) {
		if ($e eq $element) {
			return 1;
		}
	}
	return 0;
}

#calcule les clés du tableau %clique_max

sub cle_clique {
	(my $adr) = @_;
	my @l = sort @{$adr};
	return join ("_",@l);
```

```
}


# compare les cliques 2 à 2 et indique les candidats à la fusion
sub mergeClique {
	my $adr=shift;
	my @cliques=@{$adr};
	my %num_group;
	my $nbGroups=0;

	for (my $i=0;$i<=$#cliques-1;$i++) {
		my @clique1=@{$cliques[$i]};
		my %clique1;
		my $n1=@clique1;
		foreach my $unit1 (@clique1) {
			$clique1{$unit1}=1;
		}
		for (my $j=$i+1;$j<=$#cliques;$j++) {
			my @clique2=@{$cliques[$j]};
			my $n2=@clique2;

			# calcul de l'intersection pour @clique1 et @clique2
			my $inter=0;
			foreach my $unit2 (@clique2) {
				if (exists($clique1{$unit2})) { $inter++ };
			}
			# la condition pour fusionner est de ne différer que d'une unité
au max

			if (2*$inter/(@clique1+@clique2)>0.7) {
				print "$i(".join(", ",@clique1).") et $j(".join(", ",@clique2).")
sont candidats à la fusion\n";
				# si chaque clique est rattachée à un groupe, on fusionne vers
le plus petit
				if (exists($num_group{$i}) && exists($num_group{$j})) {
					my $gi=$num_group{$i};
					my $gj=$num_group{$j};
					# fusion de $gj vers $gi
					foreach my $num (keys %num_group) {
						if ($num_group{$num}==$gj) {
$num_group{$num}=$gi };
					}
				} elsif (exists($num_group{$i})) {
					# fusion de $gj vers $gi
					$num_group{$j}=$num_group{$i};
				} elsif (exists($num_group{$j})) {
					# fusion de $gi vers $gj
					$num_group{$i}=$num_group{$j}
				} else {
```

```
                                        # création d'un nouveau groupe avec les deux
nums
                                        $nbGroups++;
                                        $num_group{$i}=$nbGroups;
                                        $num_group{$j}=$nbGroups;
                                }
                        } else {
                                # si nécessaire, création des singletons
                                if (!exists($num_group{$i})) {
                                        $nbGroups++;
                                        $num_group{$i}=$nbGroups;
                                }
                                # si nécessaire, création des singletons
                                if (!exists($num_group{$j})) {
                                        $nbGroups++;
                                        $num_group{$j}=$nbGroups;
                                }
                        }
                }
        }

        # algorithme de fusion des cliques
        my %group_nums; # enregistre pour chaque groupe le numéro des cliques
correspondantes
        # on extrait %group_nums le tableau réciproque de %num_group
        foreach my $num (keys %num_group) {
                my $group=$num_group{$num};
                $group_nums{$group}=[] unless exists($group_nums{$group});
                push(@{$group_nums{$group}},$num);
        }

        # on créé ensuite la liste des clusters de cliques
        my @clusters;
        foreach my $group ( keys %group_nums ) {
                my %units;
                my @nums=@{$group_nums{$group}};
                foreach my $num (@nums) {
                        my @clique=@{$cliques[$num]};
                        foreach my $unit (@clique) {
                                $units{$unit}=1;
                        }
                }
                my @cluster=sort keys %units;
                push(@clusters,[@cluster]);
        }

        print "Liste des clusters :\n";
        my $n=1;
        foreach my $cluster (@clusters) {
```

```
                print "Cluster $n : (",join(", ",@{$cluster}).")\n";
                $n++;
        }
        return @clusters;
}


#---------------------------------------------------- fonctions diverses

# complète les liens par transitivité (primaire) pour les langues passées en paramètre
sub addTransitiveLinks {
        my @langs=@_;
        my %lang_units;

        # construction du hachage contenant, pour chaque langue, la liste des unités correspondantes
        foreach         my $unit (keys %corr) {
                $unit=~/^(\w\w)-/;
                my $l=$1;
                $lang_units{$l}=[] unless exists($lang_units{$l});
                push(@{$lang_units{$l}},$unit);
        }

        # ensuite on passe en revue chaque couple de @langs
        for (my   $i=0;$i<=$#langs-1;$i++) {
                my $l1=$langs[$i];
                my @units1=@{$lang_units{$l1}};
                for (my  $j=$i+1;$j<=$#langs;$j++) {
                        my $l2=$langs[$j];
                        # une fois déterminé le couple de langue, on passe en revue chaque couple d'unités
                        my @units2=@{$lang_units{$l2}};
                        foreach my $unit1 (@units1) {
                                foreach my $unit2 (@units2) {
                                        if (!exists($corr{$unit1}{$unit2})) {
                                                my $common=0;
                                                # si on trouve un correspondant commun, on incrémente le compteur
                                                foreach my $corrUnit (keys %{$corr{$unit1}}) {
                                                        # pour la transitivité, on exige deux liens primaires (non des liens transitifs égaux à 0.5)
                                                        if (exists($corr{$corrUnit}{$unit2}) && $corr{$corrUnit}{$unit1}==1 && $corr{$corrUnit}{$unit2}==1) {
                                                                $common++;
                                                        }
                                                }
                                                my $dice=dice($unit1,$unit2);
```

```
                                        # si le compteur est supérieur ou égal au
seuil, on ajoute la correspondance
                                        if
(($common>=$interSizeLimitForCompletion && dice($unit1,$unit2) >
$diceLimitForCompletion) or $common>=3) {
                                                $verbose && print "Ajout de la
correspondance $unit1<=>$unit2 par transitivité\n";
                                                print "Ajout de la correspondance
$unit1<=>$unit2 par transitivité\n";
                                                $corr{$unit1}{$unit2}=0.5;
                                                $corr{$unit2}{$unit1}=0.5;
                                        }
                                }
                        }
                }
        }
}

# renvoie les liens internes manquants pour un synset passé en argument
sub getMissingLinks {
        my @synset=@_;
        my @missing;
        for (my  $i=0;$i<=$#synset-1;$i++) {
                my $unit1=$synset[$i];
                for (my  $j=$i+1;$j<=$#synset;$j++) {
                        my $unit2=$synset[$j];

                        if ( substr($unit1,0,2) ne substr($unit2,0,2) && !
exists($corr{$unit1}{$unit2})) {
                                my $dice=dice($unit1,$unit2);
                                push(@missing,"$unit1 <?> $unit2 (dice=$dice)");
                        }
                }
        }
        return @missing;
}

# algorithme de classification ascendante hiérarchique
sub clustering {
        my $distances=shift; # référence vers hachage contenant les distances
        my @set=@_;
        my $seuil=10;

        # les clusters sont enregistrés au moyen du hachage %cluster_num qui fait
correspondre à numéro de cluster la liste de ses éléments
        my %cluster_nums;

        my $nbClusters=@set;
```

```
	# initialisation des clusters : chaque élément est dans un singleton
	for (my $i=0;$i<=$#set;$i++) {
		$cluster_nums{$i}=[$i];
	}

	my $oldNbClusters=$nbClusters+1;
	print "$nbClusters initiaux\n";
	# tant qu'on a encore des regroupements possible et que l'on n'est pas arrivé à stabilité, on regroupe
	while ($oldNbClusters>$nbClusters && $nbClusters>1) {

		# calcul de la demi-matrice de distance entre les clusters
		my @clusters=keys %cluster_nums;
		my $distMin=-1;
		my $i_min;
		my $j_min;
		for (my $i=0;$i<=$#clusters-1;$i++) {
			my @cluster1=@{$cluster_nums{$clusters[$i]}};
			for (my $j=$i+1;$j<=$#clusters;$j++) {
				my @cluster2=@{$cluster_nums{$clusters[$j]}};

				# la distance entre deux clusters est la distance moyenne de ses éléments
				my $distSum=0;
				foreach my $num1 (@cluster1) {
					foreach my $num2 (@cluster2) {
						if (exists($distances->{$set[$num1]." ".$set[$num2]})) {
							$distSum+=$distances->{$set[$num1]." ".$set[$num2]};
						} else {
							$distSum+=10000;
							#print "Pas de distance calculée pour $set[$num1]-$set[$num2]\n";
						}
					}
				}
				my $distMean=$distSum/(@cluster1*@cluster2);
				if ($distMin == -1 or $distMean<$distMin) {
					$distMin=$distMean;
					$i_min=$clusters[$i];
					$j_min=$clusters[$j];
				}
			}
		}
		$oldNbClusters=$nbClusters;
		# si la distance minimale est inférieure à un certain seuil on regroupe
		if ($distMin<=$seuil) {
```

```
                        # regroupement des clusters $i_min et $j_min

        $cluster_nums{$i_min}=[@{$cluster_nums{$i_min}},@{$cluster_nums{$j_min}
}];
                        my @units=map {$set[$_]} (@{$cluster_nums{$i_min}});
                        print "Fusion de $i_min et $j_min ($distMin) -> (". join(",
",@units).") \n";
                        delete ($cluster_nums{$j_min});
                        $nbClusters--;
                }
        }
        print "$nbClusters cluster obtenus !\n";
        my @clusters=keys %cluster_nums;
        my $i=0;
        foreach my $cluster (@clusters) {
                my @cluster=@{$cluster_nums{$cluster}};
                my @units=map {$set[$_]} @cluster;
                print "Cluster $i : (". join(", ",@units).")\n";
                $i++;
        }
        return @clusters;
}

# renvoie le cosinus entre deux vecteurs de correspondance (enregistrés dans
$corr{$unit1} et $corr{$unit2})
sub cosinus {
        my ($unit1,$unit2)=@_;
        my $scalaryProduct=0;
        my @corr1=keys %{$corr{$unit1}};
        my $l1=@corr1;
        my $l2=keys %{$corr{$unit2}};

        foreach my $corr (@corr1) {
                if (exists($corr{$unit2}{$corr})) {
                        $scalaryProduct+=$corr{$unit2}{$corr}*$corr{$unit1}{$corr};
                }
        }

        return $scalaryProduct/(sqrt($l1)*sqrt($l2));
}

# renvoie l'intersection de deux vecteurs de correspondance (enregistrés dans
$corr{$unit1} et $corr{$unit2})
sub inter {
        my ($unit1,$unit2)=@_;
        my $inter=0;
        my @corr1=keys %{$corr{$unit1}};
        my $l1=@corr1;
        my $l2=keys %{$corr{$unit2}};
```

```
	foreach my $corr (@corr1) {
		if (exists($corr{$unit2}{$corr})) {
			$inter++;
		}
	}

	return $inter;
}

sub dice {
	my ($unit1,$unit2)=@_;
	my $inter=0;
	my @corr1=keys %{$corr{$unit1}};
	my $l1=@corr1;
	my $l2=keys %{$corr{$unit2}};

	foreach my $corr (@corr1) {
		if (exists($corr{$unit2}{$corr})) {
			$inter++;
		}
	}

	return 2*$inter/($l1+$l2);
}

# renvoie la distance euclidienne entre deux vecteurs de correspondance
sub euclidianDistance {
	my ($unit1,$unit2)=@_;
	my $dist=0;
	foreach my $corr (keys %corr) {
		if ($corr ne $unit1 && $corr ne $unit2) {
			my $x1=exists($corr{$unit1}{$corr});
			my $x2=exists($corr{$unit2}{$corr});
			$dist+=($x1-$x2)^2;
		}
	}
	return sqrt($dist);
}
# renvoie un hachage avec les distances pour chaque paire d'unités
sub calcDistances {
	my %dist;
	my $distFile=$path."/".$seed.".distances.txt";
	my $DISTFILE;
	if (-f $distFile) {
		open($DISTFILE,$distFile);
		while (<$DISTFILE>) {
			if (/(.*)\t(.*)/) {
				my $key=$1;
```

```
                my $value=$2;
                $dist{$key}=$value;
            }
        }
        close($DISTFILE);
    } else {
        open ($DISTFILE,">",$distFile) or die "impossible d'ouvrir $distFile en
écriture\n";
        foreach my $unit1 (keys %corr) {
            foreach my $unit2 (keys %corr) {
                if ($unit1 ne $unit2 && !exists($dist{$unit1.$unit2})) {
                    my $cos=cosinus($unit1,$unit2);
                    if ($cos>0) {
                        $dist{$unit1." ".$unit2}=1/dice($unit1,$unit2);
                        $dist{$unit2." ".$unit1}=$dist{$unit1."
".$unit2};
                        print $DISTFILE $unit1."
".$unit2."\t".$dist{$unit1." ".$unit2}."\n";
                        print $DISTFILE $unit2."
".$unit1."\t".$dist{$unit1." ".$unit2}."\n";
                    }
                }
            }
        }
        close($DISTFILE);
    }
    return %dist;
}

# calcule, pour une clique, le dice moyen interne
sub averageDice {
    my @clique=@_;
    my %corresp; # hash enregistrant les corresps pour chaque langue représentée dans
la clique

    print "Evaluation de la clique (@clique)\n";

    foreach my $lang (@langues) {
        my @units=grep {/^$lang/} @clique;
        if (@units) {
            $corresp{$lang}={};
            foreach my $unit (@units) {
                #~ $corresp{$lang}{$unit}=1;                # deux modes de
calculs différents (prise en compte ou non des autocorresp)
                foreach my $corr (keys %{$corr{$unit}}) {
                    $corresp{$lang}{$corr}=1;
                    #~ print "$lang => $corr\n";
                }
            }
```

```
		}
	}

	my $diceSum;
	my $n;
	my @langs= keys %corresp;
	# pour chaque couple de langue, on évalue le dice
	for (my $i=0;$i<=$#langs-1;$i++) {
		for (my $j=$i+1;$j<=$#langs;$j++) {
			my $inter=0;
			my $lang1=$langs[$i];
			my $lang2=$langs[$j];

			my @corr1=keys %{$corresp{$lang1}};
			my $l1=@corr1;
			my $l2=keys %{$corresp{$lang2}};

			foreach my $corr (@corr1) {
				if (exists($corresp{$lang2}{$corr})) {
					$inter++;
					print "Corresp commune $lang1 $lang2 -> $corr\n";
				}
			}
			my $dice = 2*$inter/($l1+$l2);
			$diceSum +=$dice;
			$n++;
			print "($lang1,$lang2)=>inter=$inter - dice=".$dice."\n";
		}
	}
	print "Moyenne Dice : ".($diceSum/$n)."\n";
}
```

 # calcule quelles sont les unités partiellement ou totalement désambiguïsées à l'intérieur d'une clique

```
sub evalDisambClique {
	my @clique=@_;
	my %disamb;

	# le hachage %lang contient les langues de la clique dans ses clés
	my %langs;
	foreach my $u (@clique) {
		$u=~/^(\w\w)/;
		 $langs{$1}++;
	}
	for (my $i=0;$i<=$#clique-1;$i++) {
		my $unit1=$clique[$i];
		 $unit1=~/^(\w\w)/;
```

 # on ne traite que les unités qui sont seules représentantes de leur langue dans la clique

```
if ($langs{$1}==1) {
                for (my $j=$i+1;$j<=$#clique;$j++) {
                        my $unit2=$clique[$j];
                        $unit2=~/^(\w\w)/;
                        if ($langs{$1}==1) {
```

 # pour le couple ($unit1,$unit2), on cherche quelle est la langue plus désambiguisatrice (celle qui propose le plus de traductions pour l'une et l'autre - en ne retenant que les traductions hétéromorphes)

```
                                my $max=0;
                                my $langMax;
                                my $dice;
                                my @inter;
                                foreach my $lang (keys %langs) {
                                        if ($unit1 !~/^$lang/ and $unit2 !~/^$lang/) {
                                                # nb de traductions de $unit1 pour $lang
                                                my @corr1=grep {/^$lang/ && dice($unit1,$_)<0.5} keys %{$corr{$unit1}};
                                                my @corr2=grep {/^$lang/ && dice($unit2,$_)<0.5 } keys %{$corr{$unit2}};
                                                my $nbTrad1=@corr1; # on retient tous les correspondants dont la couverture est suffisamment différente
                                                my $nbTrad2=@corr2; # on retient tous les correspondants dont la couverture est  suffisamment différente
                                                if ($nbTrad1+$nbTrad2>$max) {
$max=$nbTrad1+$nbTrad2;
                                                        $langMax=$lang;
@inter=intersection(\@corr1,\@corr2);
$dice=2*@inter/(@corr1+@corr2);
                                                }
                                        }
                                }
```

 # dans cette intersection, on cherche les éléments de la clique

```
                                my @int=intersection(\@inter,\@clique);
                                my $int=@int;
                                my $inter=@inter;
                                print "\n($unit1,$unit2) Langue de désambiguisation commune $langMax. Inter=@inter ($inter). Int=@int ($int). Dice=$dice\n";
```

```
                                        # si la clique ne contient qu'une seule des unités
jugées les plus désambiguisatrice, c'est un bonne indice de désamb pour
($unit1,$unit2)
                                                    if ( (grep {/^$langMax/} @clique) == 1) {
                                                            print "Désambiguisation entre $unit1
et $unit2 grâce à @int\n";
                                                            $disamb{$unit1}=1;
                                                             $disamb{$unit2}=1;
                                                     }
                                                }
                                            }
                                        }
                                    }
                                }
                                 my @disamb=keys %disamb;
                                print "Unités désambiguisées dans la clique: (@disamb)\n";
}
# Pour chaque langue, calcule les intersections des correspondants des unités de la
clique dans cette langue.
# renvoie la taille de l'intersection la plus grande (dans la langue pour laquelle les
ambiguités se manifestent le plus explicitement)
# NB : on ne calcule ces intersections que pour les unités apparaissant seules.
sub evalDisambClique2 {
                my @clique=@_;
                my %disamb;

            # le hachage %lang contient les langues de la clique dans ses clés, et les unités
correspondantes dans ses valeurs

                my %langs;
                foreach my $u (@clique) {
                        $u=~/^(\w\w)/;
                          if (!exists($langs{$1})) {
                                        $langs{$1}=$u;
                                } else {
                                        $langs{$1}.=" ".$u;
                                 }
                }
                my $sommeInter=0;
                 my $n=0;
            my %inter;
            foreach my $langInClique (keys %langs) {
                         my @units=split(/ /,$langs{$langInClique});

                    # pour chaque langue, on calcule l'intersection entre l'intersection déjà
calculée et les correspondant de $unit

                        foreach my $lang (@langues) {
```

```
                                my @corr; # on y met tous les correspondants de @units
dans $lang
                                if ($lang eq $langInClique) {
                                        @corr=@units;
                                } else {
                                        foreach my $unit (@units) {
                                                push(@corr,grep {/^$lang/ } keys
%{$corr{$unit}});
                                        }
                                }
                                # on prend l'intersection entre les correspondants obtenus, et les
correspondants précédents obtenus à partir d'autres langues (s'il existe)
                                if (exists($inter{$lang})) {

$inter{$lang}=[intersection($inter{$lang},\@corr)];
                                } else {
                                        $inter{$lang}=[@corr];
                                }
                        }
        }

        my $max=0;
        my $langMax="?";
        foreach my $lang (@langues) {
                        if (exists($inter{$lang})) {
                                my $inter=@{$inter{$lang}};
                                if ($inter > $max) {
                                        $max=$inter;
                                        $langMax=$lang;
                                }
                        }
        }

        print "Langue = $langMax, Intersection = $max\n";
        return $max;
}
```

 # intersetion (sans doublon) de deux listes

```
sub intersection {
        my $adr1=shift;
        my $adr2=shift;
        my @inter;
            foreach my $e (@{$adr1}) {
                        push(@inter,$e) if (membre($e,@{$adr2}) && !membre($e,@inter));
            }
            return @inter;
}
```

 # enrichissement des cliques avec les unités les plus proches

#v1 : enrichissement pour $lang seulement et création d'une nouvelle clique à chaque ajout

```
sub enrichCliques1 {
        my $lang=shift;
        my @cliques=@_;
        my @newCliques=();

        # on extrait le lexique correspondant à $lang

        my @lexique= grep {/^$lang/ } keys %corr;

        # pour chaque clique

        foreach my $adr (@cliques) {
                my @clique=@{$adr};
                my $enrichment=0;

                # pour chaque unité du lexique

                foreach my $unit (@lexique) {
                        if (!membre($unit,@clique)) {
                                my $diceSum=0;
                                foreach my $unit2 (@clique) {
                                        $diceSum+=dice($unit,$unit2);
                                }
                                my $diceMean=$diceSum/@clique;
                                if ($diceMean>$diceMean4enrichment) {
                                        print "Enrichissement de @clique avec
$unit : $diceMean\n";
                                        push(@newCliques,[$unit,@clique]);
                                        $enrichment=1;
                                }
                        }
                }
                if (!$enrichment) {
                        push(@newCliques,[@clique]);
                }
        }
        return @newCliques;
}
```

#v2 : enrichissement pour $lang seulement et chaque unité n'est rajouté qu'une fois, à la clique la plus proche, plusieurs ajouts étant possibles dans la même clique

```
sub enrichCliques2 {
        my $lang=shift;
        my @cliques=@_;

        # on extrait le lexique correspondant à $lang
```

```
	my @lexique= grep {/^$lang/ } keys %corr;

	# pour chaque unité du lexique
	foreach my $unit (@lexique) {
		my $cliqueMax;
		my $diceMeanMax=0;

		# pour chaque clique
		foreach my $adr (@cliques) {
			my @clique=@{$adr};
			my $enrichment=0;
			    if (!membre($unit,@clique)) {
				my $diceSum=0;
				  foreach my $unit2 (@clique) {
					$diceSum+=dice($unit,$unit2);
				  }
				  my $diceMean=$diceSum/@clique;
				      if ($diceMean>$diceMean4enrichment &&
$diceMean>$diceMeanMax) {
					$diceMeanMax=$diceMean;
					$cliqueMax=$adr;
				    }
			    }
		}
		if ($cliqueMax) {
			print "Enrichissement de @{$cliqueMax} avec $unit :
$diceMeanMax\n";
			 push(@{$cliqueMax},$unit);
		}
	}
	return @cliques;
}
```

#v3 : enrichissement seulement avec les synonymes de seed et chaque unité n'est rajoutée qu'une fois, à la clique la plus proche, plusieurs ajouts étant possibles dans la même clique

```
sub enrichCliques3 {
	my $lang=shift;
	my $adrSyns=shift;
	my $adrCliques=shift;
	my @cliques=@{$adrCliques};

	# on extrait le lexique correspondant à $lang

	my @lexique= @{$adrSyns};
```

```
	# pour chaque unité du lexique
	foreach my $unit (@lexique) {
				my $cliqueMax;
				my $diceMeanMax=0;
				# pour chaque clique
				foreach my $adr (@cliques) {
					my @clique=@{$adr};
					my $enrichment=0;
					if (!membre($unit,@clique)) {
						my $diceSum=0;
						foreach my $unit2 (@clique) {

					$diceSum+=dice($unit,$unit2);
						}
						my $diceMean=$diceSum/@clique;
						if ($diceMean>$diceMean4enrichment
			&& $diceMean>$diceMeanMax) {
							$diceMeanMax=$diceMean;
							$cliqueMax=$adr;
						}
					}
		}
		if ($cliqueMax) {
			print "Enrichissement de @{$cliqueMax} avec $unit :
$diceMeanMax\n";
			push(@{$cliqueMax},$unit);
		}
	}
return @cliques;
}
```

 # associe à chaque clique de @cliques, les cliques les plus proche en terme de dice moyen

```
sub associateCliques {
	my $entry=shift;
	my @cliques=@_;
	foreach my $adr1 (@cliques) {
		if (membre($entry,@{$adr1})) {
			print "\nCliques associées à (".join(", ",@{$adr1}).") :\n";
			foreach my $adr2 (@cliques) {
				my $dice=interCliqueDice($adr1,$adr2);
				if ($dice>0.2) {
					print "\t\t(".join(", ",@{$adr2}).")\n";
				}
			}
		}
	}
}
```

```
sub interCliqueDice {
	my ($adr1,$adr2)=@_;
	my $diceSum=0;
	my $n=0;
	foreach my $unit1 (@{$adr1}) {
		foreach my $unit2 (@{$adr2}) {
			$diceSum+=dice($unit1,$unit2);
			$n++;
		}
	}
	return $diceSum/$n;
}
```

Annexe 9 : script seg_phrase.pl de ségmentation phrastique du notre corpus

#---------- déclaration des variables globales

```
our $abrev="etc|chap|introd|art|adj|adv|apr|J\\.\\-C|art\\-|av|bibliogr|boul|bull|c\\.\\-à\\-
d|cap|cf|Cf|conf|chap|coll|dép|éd|édit|env|etc|étym|ex|fasc|fig|hab|ibid|id|i\\.e|i|e|ill|inf|introd|
ital|loc|cit|M|MM|math|ms|N|N\\.B|N\\.\\-
D|N\\.D\\.A\\.|N\\.D\\.E|N\\.D\\.L\\.R|obs|op|cit|ouvr|cit|P\\.C\\.C|p|ex|p|pp|paragr|p|i|p|l|p\\.\\
-s|Q\\.G|R\\.P|S\\.A|sq|sqq|subst|suiv|sup|suppl|S\\.V\\.P|t|trad|var|vol|zool";
```

 # ne pas oublier d'échapper les points et les tirets

```
our $points1="\\.?!";
our $points2=":;";
our $sep="\\.|:|;|\\?|\\!";
my $entite="&lt|&gt| ";
```

#---------- text2Sent()

```
# Fonction de segmentation réversible en phrases
# Entrées :
#       * arg1 : un texte
# Sorties :
#       * return @phrases une liste des phrases
# Deux règles principales de découpage :
#       point1 espace maj
#       point2
sub text2Sent {
        my $texte=shift (@_);
        my @phrases=();
        my ($tete,$reste,$decoupe);

        # traitement des exceptions : on neutralise le point par une marque <dot>
        $texte=~s/($abrev)\./$1<dot>/gi;                # transformation du point des abrév.
        $texte=~s/(\n\s*[0-9]+)\./$1<dot>/g;# transformation du point des titres
        $texte=~s/($entite);\n/$1<Pvergule>/g; #transformation de virgule suivi d'un   point
virgule
        $texte=~s/($sep)\s|\n/$1<p>/g;          # insertion des balises de segmentation
        $decoupe=1;
        while ($decoupe) {
                if($texte=~/^([^$points2]+?[$points1][\"\)]?)(\s+[A-Z0-9].*)/) {
                                                                                # cas
                 général point esp maj
                  # Ne pas oublier le ^ initial, car le +? n'étant pas 'gourmand' il faut
                 forcer le motif
```

 # à commencer au début de la phrase

 # Par ailleurs l'expr. commence par [^$points2]+? plutôt que .+?

 # car s'il y a un point double dans la phrase, on segmente avant le point (règle suivante)

```
                                                $tete=$1;
                        $reste=$2;
                    } elsif ($texte=~/^(.+?[$points2])(.*)/) {# points doubles
                                        if ($texte=~/($entite)(.|\n*)/) {    #ne pas segmenter
aprés entité;
                                                                        $tete=$texte;
                                        $decoupe=0;
                                } else{
                                        $tete=$1;
                                        $reste=$2;
                                }
                        }else {
                                $tete=$texte;
                                $reste="";
                                $decoupe=0;
                        }
                $tete=~s/<dot>/./g; # rétablissement des points
                        $tete=~s/<Pvergule>/;/g;
                        if ($tete) {
                                push(@phrases,$tete);
                        }
                        $texte=$reste;
                }
                return @phrases;
}
```

#----------Text2Sent

```
$language="fr";
my $n=1;
if ($language eq "fr") { # fr, en, es, ar pour le corpus français, on peut s'appuyer sur les \n
pour segmenter
        $sep.="|\\n";
}
opendir(DIR,".");
while($file=readdir(DIR)) {
        if (! -f $file ||$file!~/(fr.html.txt)$/) { next; }
        open(INPUT,"<",$file);
            open(OUT,">","$file.$language.seg");
        my $text = do { local $/; <INPUT> };
```

```
            close(INPUT);
            my @l;
                  my $phrase;
                 @l=text2Sent($text);
                  foreach $sent (@l) {
                                    @sentences=split(/<p>/,$sent);
                                     foreach $s (@sentences) {
                                                       if ($s !~/^\s*$/) {
                                                                         print OUT "\t\t<s
id=\"s$n\">$s</s>\n";
                                                                         $n++;
                                                        }
                                      }
                           }
              print "$n phrases traitées\n";
     }
    close(OUT);
```

Annexe 10 : script ttg2ces.pl déstiné à reformater la sortie des textes étiquetés par Treetagger et obtenir une sortie au format XML :

```
use IO::Handle;
STDOUT->autoflush(1);
#---------------------------------------------- déclaration des variables globales
#our $points1="\\.?!?:;";
our $nTok=1;
our @t=("","\t","\t"x2,"\t"x3,"\t"x4,"\t"x5);
our $nChunks=0;
our $rest;
our $language="fr";

#-----------------------------------------------------program
opendir(DIR,".");
open(OUT,">:encoding(UTF-8)","Tagged.all.$language.txs");
print OUT "$t[0]<?xml version=\"1.0\" encoding=\"utf-8\"?>\n";
print OUT "$t[1]<txs>\n";
print OUT "$t[2]<chunkList>\n";

while($file=readdir(DIR)) {
   if (! -f $file||$file!~/(ttg)$/) { next; }
                                      open(INPUT,"<",$file);
            $text = do { local $/; <INPUT> };
   close(INPUT);

            print OUT "$t[3]<chunk id=\"$file\">\n";

   $nChunks++;
   while ($text){

      if($text=~/^(<s id=".*?">)\s+?(((.|\n)*)*)$/) {
            my $idph=$1;
            $rest=$2;
            print OUT "$t[4]$idph\n";
      }elsif ($text=~/^(<\/s>)\s+?(((.|\n)*)*)$/) {
            my $finph=$1;
             $rest=$2;
            print OUT "$t[4]$finph\n";
      }elsif($text=~/^(<\www\..+?\..+?\>)\n+?(((.|\n)*)*)$/){   #ex.<www.seeurope.org>\n
               print $1."\n";
               my $tag="";
               my $lemme=$1;
               $rest=$2;
                print OUT "$t[5]<tok id=\"t$nTok\" ctag=\"$tag\"
base=\"$lemme\">".$lemme."</tok>\n";
                $nTok++;
      }elsif($text=~/^(<.+?\@.+?\..+?>)\n+?(((.|\n)*)*)$/){     #ex.<nyambe@un.org>\n
```

```
                print $1."\n";
                my $tag="";
                my $lemme=$1;
                $rest=$2;
                print OUT "$t[5]<tok id=\"t$nTok\" ctag=\"$tag\"
base=\"$lemme\">".$lemme."</tok>\n";
                $nTok++;
        }elsif($text=~/^(.+?)\s(.+?)\s(.+?)\s+?(((.|\n)*)*)$/){
                my $tok=$1;
                my $tag=$2;
                my $lemme=$3;
                $rest=$4;
                print OUT "$t[5]<tok id=\"t$nTok\" ctag=\"$tag\"
base=\"$lemme\">".$tok."</tok>\n";
                $nTok++;

        }else {
           die $text."\n";
        }
        $text=$rest;
    }
    print OUT "$t[3]</chunk>\n";
    print "$nChunks\n";
}
print OUT "$t[2]</chunkList>\n";
print OUT "$t[1]</txs>\n";
close(OUT);
```

Annexe 11 : Script merge-tags.pl destiné à la fusion de deux types d'annotation : 1/ segmentation en phrase du fichier 2/ tokenisation étiquetage avec AMIRA.La sortie est un fichier au format txs :

```
use IO::HANDLE;
STDOUT->autoflush(1);
#------------------------------------------------------------- déclarations

# paramètres du script
our $dir=".";
# autres var globales
our $nTok=0;
our $nLine=0;
my $language="ar";
$nbtxt=0;
#------------------------------------------------------------------ programme principale
opendir(DIR,$dir);
open(OUT,">","Entagged.all.".$language.".txs");
print OUT "<?xml version=\"1.0\" encoding=\"utf-8\"?>\n";
      print OUT "\t<txs>\n";
      print OUT "\t\t<chunkList>\n";


while($file=readdir(DIR)) {

        # si on trouve un fichier étiqueté
        if
($file=~/(.*)(TOK.POS.TOK.POS.TOK.POS.TOK.POS.TOK.POS.TOK.POS.TOK.POS.
TOK.POS.TOK.POS.TOK.POS.TOK.POS.TOK.POS.TOK.POS.TOK.POS.TOK.POS)$/) {

                    my $taggedFile=$1.$2;
                    my $segmentedFile=$1."bw.bw.bw.bw.bw.bw.bw.bw.ar.seg.lolo.seg";
      print OUT "\t\t\t<chunk id=\"$taggedFile\">\n";
                  # on teste l'existence du fichier segmenté correspondant
                  if (-f $dir."/".$segmentedFile) {
                        # on place tout le texte segmenté dans une variable tampon
                          open(SEG,$segmentedFile);
                          my @lines=<SEG>;
                          $segText=join("",@lines);
                         close (SEG);
                        # ouverture du fichier résultat et écriture de l'entête
                       # boucle de lecture du fichier taggé
                         open(TAG,$taggedFile);

                while (my $line=<TAG>) {
                                    $nLine++;
                                    chomp $line;
```

```
                                    my @tokens=split(/ /,$line);
            my $prevOrth="";
                                    # on passe en revue tous les tokens issus
d'AMIRA. Chaque token sera cherché dans le fichier segmenté
                                  foreach my $token (@tokens) {
                                        $token=~/^(.+)\/([^\/]+)$/;
                                        my $orth=$1;
                                        my $tag=$2;
                                        $variante1=$orth;
                                        $variante1=~s/^.(.*)/$1/;
                                    $variante2=$orth;
                                    $variante2=~s/p$/t/;
                                    $variante3=$orth;
                                    $variante3=~s/>w/w/;     #
ex.'Albrwt>wkwl-->Albrwtwkwl
                                    $variante4=$orth;
                                    $variante4=~s/Y/A/;
      $variante5=$orth;
                                    $variante5=~s/t$/p/;
      $variante6=$orth;
                                    $variante6=~s/^A(.*)p$/$1t/;
#ex.Altfawp -->ltfawt

                                    # si on trouve des tag de segmentation
en début de chaine, on les imprime
                                    if ($segText=~/^\s*(<\/s>)?\s*(<s
id="[^"]+"\s*>)\s*(\Q$orth\E|\Q$variante1\E|\Q$variante2\E|\Q$variante3\E|\Q$
variante4\E|\  Q$variante5\E|\Q$variante6\E)(.*)/s) {

            $closeTag=$1;
                                        $openTag=$2;
                                        $rest=$4;
                                        if ($closeTag) {
                                                print OUT
"\t\t\t\t".$closeTag."\n";
                                        }
                                        print OUT
"\t\t\t\t".$openTag."\n";
                                       $segText=$orth.$rest;
                                    }
                                    # synchronisation : recherche de $orth
en tête de chaine

if($segText=~/^\s*(\Q$orth\E|\Q$variante1\E|\Q$variante2\E|\Q$variante3\E|\Q$
variante4\E|\Q$variante5\E|\Q$variante6\E)(.*)$/si) {
                                        # si trouvé, on imprime le
token et on réduit $segText au reste
                                        $segText=$2;
                                        $nTok++;
```

```
                                                    $orth=~tr/<>/IO/;

                                                    print OUT "\t\t\t\t\t<tok
        id=\"t$nTok\" msd=\"$tag\">$orth</tok>\n";

                                                } else {
                                                        # sinon, cela signifie que la
        synchronisation a échoué
                                                 print "Echec : pas de synchronisation, token
        '$token' non trouvé ($nTok ligne          $nLine)\n";
                                                   print "Texte restant à analyser :
        ".$segText;
                                                die;
                                                }
   }
        }
close(TAG);
         print OUT "\t\t\t\t</s>\n";
         print OUT "\t\t\t</chunk>\n";
        $nbtxt=$nbtxt+1;
        print $nbtxt;
        print "Traitement terminé avec succès !\n";
                # fermeture des tags et fin de fichier
      }
   }
}
print OUT "\t\t</chunkList>\n";
print OUT "\t</txs>\n";
close(OUT);
```

Annexe 12 : Script harm.ctag.pl déstiné à l'harmonisation des étiquettes d'étiquetage des quatre langues pour n'utiliser qu'un seul jeu d'étiquettes :

```
use IO::Handle;
STDOUT->autoflush(1);

my $lang="fr"; #(changement de langue pour chaque groupe de règle : fr, ar, en, es)

opendir(DIR,".");
while(my $file=readdir(DIR)) {
   if ( ! -f $file||$file !~/(Tagged.[0-9]*.$lang.txs)$/) { next; }

        open(INPUT,"<",$file); open(OUT,">","$file.harm.txs");
my $text = do { local $/; <INPUT> };
print $file."\n";
        close(INPUT);
```

#L'armonisation des etiquettes françaises (33 règles de remplacement) :

```
      $text=~s/ctag="KON"/ctag="CC"/g;
       $text=~s/ctag="NUM"/ctag="CD"/g;
       $text=~s/ctag="PRO"/ctag="CD"/g;
       $text=~s/ctag="DET:ART"/ctag="ART"/g;
       $text=~s/ctag="ADV"/ctag="ADV"/g;
       $text=~s/ctag="PRP"/ctag="PRP"/g;
       $text=~s/ctag="PRP:det"/ctag="PRP"/g;
       $text=~s/ctag="ADJ"/ctag="ADJ"/g;
       $text=~s/ctag="NAM"/ctag="NP"/g;
       $text=~s/ctag="NOM"/ctag="NN"/g;
       $text=~s/ctag="PUN"/ctag="PUN"/g;
       $text=~s/ctag="PUN:cit"/ctag="PUN"/g;
       $text=~s/ctag="PRO:REL"/ctag="PR"/g;
       $text=~s/ctag="DET:POS"/ctag="PR"/g;
       $text=~s/ctag="PRO:POS"/ctag="PR"/g;
       $text=~s/ctag="PRO:DEM"/ctag="PR"/g;
       $text=~s/ctag="PRO:IND"/ctag="PR"/g;
       $text=~s/ctag="PRO:PER"/ctag="PRP"/g;
       $text=~s/ctag="INT"/ctag="INT"/g;
       $text=~s/ctag="SYM"/ctag="SYM"/g;
       $text=~s/ctag="VER:infi"/ctag="V"/g;
       $text=~s/ctag="VER:pper"/ctag="V"/g;
       $text=~s/ctag="VER:ppre"/ctag="V"/g;
       $text=~s/ctag="VER:pres"/ctag="V"/g;
       $text=~s/ctag="VER:subp"/ctag="V"/g;
       $text=~s/ctag="VER:impe"/ctag="V"/g;
       $text=~s/ctag="VER:simp"/ctag="V"/g;
       $text=~s/ctag="VER:subi"/ctag="V"/g;
       $text=~s/ctag="VER:impf"/ctag="V"/g;
       $text=~s/ctag="VER:futu"/ctag="V"/g;
       $text=~s/ctag="VER:cond"/ctag="V"/g;
```

```
$text=~s/ctag="SENT"/ctag="SENT"/g;
$text=~s/ctag="ABR"/ctag="Autre"/g;
```

#L’armonisation des etiquettes arabes (46 règles de remplacement) :

```
#$text=~s/msd="EX"/ctag="ADV"/g;
#$text=~s/msd="RBS"/ctag="ADV"/g;
#$text=~s/msd="DT"/ctag="ART"/g;
#$text=~s/msd="RBR"/ctag="ADV"/g;
#$text=~s/msd="RB"/ctag="ADV"/g;
#$text=~s/msd=WRB"/"ctag=ADV"/g;
#$text=~s/msd="RP"/ctag="ART"/g;
#$text=~s/msd="FW"/ctag="Autre"/g;
#$text=~s/msd="IN"/ctag="PRP"/g;
#$text=~s/msd="TO"/ctag="PRP"/g;
#$text=~s/msd="JJ"/ctag="ADJ"/g;
#$text=~s/msd="JJR"/ctag="ADJ"/g;
#$text=~s/msd="JJS"/ctag="ADJ"/g;
#$text=~s/msd="NP"/ctag="NP"/g;
#$text=~s/msd="NNP"/ctag="NP"/g;
#$text=~s/msd="NNPS"/ctag="NP"/g;
#$text=~s/msd="NN"/ctag="NN"/g;
#$text=~s/msd="NNS"/ctag="NN"/g;
#$text=~s/msd="LS"/ctag="NN"/g;
#$text=~s/msd="\\"/ctag="PUN"/g;
#$text=~s/msd=","/ctag="PUN"/g;
#$text=~s/msd=":"/ctag="PUN"/g;
#$text=~s/ctag="-"/ctag="PUN"/g;
#$text=~s/msd="\.\.\."/ctag="PUN"/g;
#$text=~s/msd="\""/ctag="PUN"/g;
#$text=~s/msd=""/ctag="PUN"/g;
#$text=~s/msd="’"/ctag="PUN"/g;
#$text=~s/msd=";"/ctag="PUN"/g;
#$text=~s/msd="`"/ctag="PUN"/g;
#$text=~s/msd="\/"/ctag="PUN"/g;
#$text=~s/msd="WDT"/ctag="PR"/g;
#$text=~s/msd="WP"/ctag="PR"/g;
#$text=~s/msd="wp$"/ctag="PR"/g;
#$text=~s/msd="POS"/ctag="PR"/g;
#$text=~s/msd="PRP$"/ctag="PR"/g;
#$text=~s/msd="PDT"/ctag="PR"/g;
#$text=~s/msd="PRP"/ctag="PRP"/g;
#$text=~s/msd="UH"/ctag="INT"/g;
#$text=~s/msd="SYM"/ctag="SYM"/g;
#$text=~s/msd="VB"/ctag="v"/g;
#$text=~s/msd="VBN"/ctag="v"/g;
#$text=~s/msd="VBG"/ctag="v"/g;
#$text=~s/msd="VBP"/ctag="v"/g;
```

```
#$text=~s/msd="VBZ"/ctag="v"/g;
#$text=~s/msd="VBD"/ctag="v"/g;
#$text=~s/msd="MD"/ctag="v"/g;
```

#L’armonisation des etiquettes anglaises (46 règles de remplacement) :

```
#$text=~s/ctag="EX"/ctag="ADV"/g;
#$text=~s/ctag="RB"/ctag="ADV"/g;
#$text=~s/ctag="DT"/ctag="ART"/g;
#$text=~s/ctag="RBR"/ctag="ADV"/g;
#$text=~s/ctag="RBS"/ctag="ADV"/g;
#$text=~s/ctag="WRB"/ctag="ADV"/g;
#$text=~s/ctag="RP"/ctag="ART"/g;
#$text=~s/ctag="FW"/ctag="Autre"/g;
#$text=~s/ctag="IN"/ctag="PRP"/g;
#$text=~s/ctag="TO"/ctag="PRP"/g;
#$text=~s/ctag="JJ"/ctag="ADJ"/g;
#$text=~s/ctag="JJR"/ctag="ADJ"/g;
#$text=~s/ctag="JJS"/ctag="ADJ"/g;
#$text=~s/ctag="NP"/ctag="NP"/g;
#$text=~s/ctag="NNP"/ctag="NP"/g;
#$text=~s/ctag="NNPS"/ctag="NP"/g;
#$text=~s/ctag="NN"/ctag="NN"/g;
#$text=~s/ctag="NNS"/ctag="NN"/g;
#$text=~s/ctag="LS"/ctag="NN"/g;
#$text=~s/ctag="\"/ctag="PUN"/g;
#$text=~s/ctag=","/ctag="PUN"/g;
#$text=~s/ctag=":"/ctag="PUN"/g;
#$text=~s/ctag="-"/ctag="PUN"/g;
#$text=~s/ctag="\.\.\."/ctag="PUN"/g;
#$text=~s/ctag="""/ctag="PUN"/g;
#$text=~s/ctag="""/ctag="PUN"/g;
#$text=~s/ctag="’"/ctag="PUN"/g;
#$text=~s/ctag=";"/ctag="PUN"/g;
#$text=~s/ctag="`"/ctag="PUN"/g;
#$text=~s/ctag="\/"/ctag="PUN"/g;
#$text=~s/ctag="WDT"/ctag="PR"/g;
#$text=~s/ctag="WP"/ctag="PR"/g;
#$text=~s/ctag="WP$"/ctag="PR"/g;
#$text=~s/ctag="POS"/ctag="PR"/g;
#$text=~s/ctag="PRP$"/ctag="PR"/g;
#$text=~s/ctag="PDT"/ctag="PR"/g;
#$text=~s/ctag="PRP"/ctag="PRP"/g;
#$text=~s/ctag="UH"/ctag="INT"/g;
#$text=~s/ctag="SYM"/ctag=SYM"/g;
#$text=~s/ctag="VB"/ctag="v"/g;
#$text=~s/ctag="VBN"/ctag="v"/g;
#$text=~s/ctag="VBG"/ctag="v"/g;
#$text=~s/ctag="VBP"/ctag="v"/g;
```

```
#$text=~s/ctag="VBZ"/ctag="v"/g;
#$text=~s/ctag="VBD"/ctag="v"/g;
#$text=~s/ctag="MD"/ctag="V"/g;
$text=~s/ctag="VVG"/ctag="V"/g;
$text=~s/ctag="VV"/ctag="V"/g;
$text=~s/ctag="VVN"/ctag="V"/g;
$text=~s/ctag="VHZ"/ctag="V"/g;
$text=~s/ctag="VVZ"/ctag="V"/g;
$text=~s/ctag="VVD"/ctag="V"/g;
```

#L'armonisation des etiquettes espagnoles (75 règles de remplacement) :

```
#$text=~s/ctag="CSUBF"/ctag="CC"/g;
#$text=~s/ctag="CSUBI"/ctag="CC"/g;
#$text=~s/ctag="CSUBX"/ctag="CC"/g;
#$text=~s/ctag="CARD"/ctag="CD"/g;
#$text=~s/ctag="ART"/ctah="ART"/g;
#$text=~s/ctag="QU"/ctag="ART"/g;
#$text=~s/ctag="ADV"/ctag="ADV"/g;
#$text=~s/ctag="SE"/ctag="ART"/g;
#$text=~s/ctag="PE"/ctag="Autre"/g;
#$text=~s/ctag="CCAD"/ctag="CC"/g;
#$text=~s/ctag="CCNEG"/ctag="CC"/g;
#$text=~s/ctag="PREP"/ctag="PRP"/g;
#$text=~s/ctag="PREP/ctag="PRP"/g;
#$text=~s/ctag="DEL"/ctag="PRP"/g;
#$text=~s/ctag="ADJ"/ctag="ADJ"/g;
#$text=~s/ctag="NP"/ctag=NP"/g;
#$text=~s/ctag="NMON"/ctag="NP"/g;
#$text=~s/ctag="ORD"/ctag="ADJ"/g;
#$text=~s/ctag="NC"/ctag="NN"/g;
#$text=~s/ctag="NMEA"/ctag="NN"/g;
#$text=~s/ctag="ALFS"/ctag="NN"/g;
#$text=~s/ctag="ALFP"/ctag="NN"/g;
#$text=~s/ctag="BACKSLASH"/ctag="PUN"/g;
#$text=~s/ctag="CM"/ctag="PUN"/g;
#$text=~s/ctag="COLON"/ctag="PUN"/g;
#$text=~s/ctag="DASH"/ctag="PUN"/g;
#$text=~s/ctag="DOTS"/ctag="PUN"/g;
#$text=~s/ctag="FS"/ctag="PUN"/g;
#$text=~s/ctag="QT"/ctag="PUN"/g;
#$text=~s/ctag="SEMICOLON"/ctag="PUN"/g;
#$text=~s/ctag="SLASH"/ctag="PUN"/g;
#$text=~s/ctag="RP"/ctag="PUN"/g;
#$text=~s/ctag="LP"/cyag="PUN"/g;
#$text=~s/ctag="CQUE"/ctag="CC"/g;
#$text=~s/ctag="INT"/ctag="PR"/g;
#$text=~s/ctag="REL"/ctag="PR"/g;
#$text=~s/ctag="PPO"/ctag="PR"/g;
```

```
        #$text=~s/ctag="DM"/ctag="PR"/g;
        #$text=~s/ctag="PPX"/ctag="PRP"/g;
        #$text=~s/ctag="PPC"/ctag="PRP"/g;
        #$text=~s/ctag="ITJN"/ctag="INT"/g;
        #$text=~s/ctag="RP"/ctag="SYM"/g;
        #$text=~s/ctag="LP"/ctag="SYM"/g;
        #$text=~s/ctag="PERCT"/ctag="SYM"/g;
        #$text=~s/ctag="SYM"/ctag="SYM"/g;
        #$text=~s/ctag="VCLInf"/ctag="v"/g;
        #$text=~s/ctag="VSinf"/ctag="v"/g;
        #$text=~s/ctag="VMinf"/ctag="v"/g;
        #$text=~s/ctag="VLinf"/ctag="v"/g;
        #$text=~s/ctag="VHinf"/ctag="v"/g;
        #$text=~s/ctag="VEinf"/cyag="v"/g;
        #$text=~s/ctag="VMadj"/ctag="v"/g;
        #$text=~s/ctag="VLadj"/ctag="v"/g;
        #$text=~s/ctag="VHadj"/ctag="v"/g;
        #$text=~s/ctag="VEadj"/ctag="v"/g;
        #$text=~s/ctag="VSadj"/ctag="v"/g;
        #$text=~s/ctag="VCLIger"/ctag="v"/g;
        #$text=~s/ctag="VEger"/ctag="v"/g;
        #$text=~s/ctag="VHger"/ctag="v"/g;
        #$text=~s/ctag="VLger"/ctag="v"/g;
        #$text=~s/ctag="VMger"/ctag="v"/g;
        #$text=~s/ctag="VSger"/ctag="v"/g;
        #$text=~s/ctag="ACRNM"/ctag="Autre"/g;
        #$text=~s/ctag="VSfin"/ctag="v"/g;
        #$text=~s/ctag="VMfin"/ctag="v"/g;
        #$text=~s/ctag="VLfin"/ctag="v"/g;
        #$text=~s/ctag="VHfin"/ctag="v"/g;
        #$text=~s/ctag="VEfin"/ctag="v"/g;
        #$text=~s/ctag="VCLIfin"/ctag="v"/g;
        #$text=~s/ctag="CODE"/ctag="Autre"/g;
        #$text=~s/ctag="PNC"/ctag="Autre"/g;
        #$text=~s/ctag="PDEL"/ctag="Autre"/g;
        #$text=~s/ctag="PAL"/ctag="Autre"/g;
        #$text=~s/ctag="FO"/ctag="Autre"/g;
        #$text=~s/ctag="NEG"/ctag="Autre"/g;

  print OUT $text;
}
close (OUT);
```

Annexe 13 : Script stock-infos-v4.pl déstiné à la construction des tableaux de stockage des données qui nous permettront d'extraire les cliques:

```
use IO::handle;
STDOUT->autoflush(1);

# ------------------------------------------------------------------------Paramétres
$outPath=".\\";              #à  changer
$nbCorrMin=2;                                          # nombre minimum de
correspondances pour la prise en compte des cooccurrents
$pourcentFreqMin=5/100;                        # on ne conserve que les correspondances
représentant plus de $pourcentFreqMin de min(occ1,occ2)
$verbose=1;
my $cTok2,
my $cTok1;

%corr=();                          # hash de hash contenant pour un couple de cles {"langX-
Tok"}{"langY-Tok"} -> le nombre de corresp
%occ=();                           # hash contenant le nombre d'occurrence correspondant Ã
une cles "lang-Tok"
%stock_corr=();         # hash contenant tous les couples "lang1-Cat1-Tok1#lang2-Cat2-
Tok2"->nb étant le nombre d'occurrences, ce tableau sera stocké en bdd pour être réutilisé

#------------------------------------------------------ fonctions

sub supprDoublons {
        my @l=@_;
        my %dejaVu;
        my @result=();

        foreach my $e (@l) {
                if (!$dejaVu{$e}) {
                        push (@result,$e);
                        $dejaVu{$e}=1;
                }
        }
        return @result;
}

#faire en plusieurs fois pour éviter les plantages
#il faut donc aussi numéroter les bases créees pour ne pas les effacer

dbmopen(%occ,$outPath."occ",0666);             #dbmopen HASH, Pth2sortie, EXPR(nom
de base.pag et dir), MODE

opendir(DIR,".");
while(my $file=readdir(DIR)) {
```

```
    if ( ! -f $file||$file !~/((.*)\.(\w\w)-(\w\w)\.pal)$/) { next; }
            my $lang1=$3;
            my $lang2=$4;
print $file."\n";
            open(IN,"<",$file);

            while ($ligne=<IN>) {
    # d'abord on elimine des caractéres de fin de ligne
      chomp $ligne;             # Rajouter d'autres cats (np ...etc)
                        $ligne=~s/\^ADJ/^Adj/g;
                        $ligne=~s/\^v-/^Verb-/g;
      $ligne=~s/\^V-/^Verb-/g;
                        $ligne=~s/\^NN/^Noun/g;
                        $ligne=~s/\^ADV/^Adv/g;

              #$n++;
              if ($ligne =~/^(.*)\t(.*)$/) {
                                    my $toks1=$1;
                                    my $toks2=$2;

                                    my @toks1=split(/ /,$toks1);
                                    my @toks2=split(/ /,$toks2);

                                    my @cats1=map {if (/\^([a-z]+?)$/i) {$1} else
                 {""}} @toks1;
                                    my @cats2=map {if (/\^([a-z]+?)$/i) {$1} else
                 {""}} @toks2;
                                    #~ print $toks2." : ".join("|",@cats2)."\n";

                                    @cats1=supprDoublons(@cats1);
                                    @cats2=supprDoublons(@cats2);

                                    my @lems1=map {/^(.+)\^/;$1} @toks1;
                                    my @lems2=map {/^(.+)\^/;$1} @toks2;

                                    $lems1=join(" ",@lems1);
                                    $lems2=join(" ",@lems2);

                                    $cats1=join("/",@cats1);
                                    $cats2=join("/",@cats2);


                 $key1="$lang1-$cats1-$lems1";
                  if (exists ($occ{$key1})) {
                    $occ{$key1}++;
                } else {
                    $occ{$key1}=1;
                }
                # incrementation des occurrences pour lang
```

```
                                        $key2="$lang2-$cats2-$lems2";
                    if (exists ($occ{$key2})) {
                      $occ{$key2}++;
                    } else {
                      $occ{$key2}=1;
                    }

                                        if (exists ($corr{$key1}{$key2})) {
                                                $corr{$key1}{$key2}++;
                                                $corr{$key2}{$key1}++;
                                        } else {

                                                # on initialise le hachage pour
                        $key1
                                                if (!exists ($corr{$key1})) {
                                                        $corr{$key1}={};
                                }
                                                # on initialise le hachage pour
                        $key2

                                                if (!exists ($corr{$key2})) {
                                                        $corr{$key2}={};
                                                }
                                                $corr{$key1}{$key2}=1;
                                                $corr{$key2}{$key1}=1;

                  }
                  } else {

                                                $verbose && print "ligne non lue :
                        $ligne\n";
                                        }
              }
}
 # ----------------------------------------------------------------
#premier filtrage afin de ne pas surcharger les tableaux d'informations inutiles
   foreach $mot1(keys %corr) {
      foreach $mot2(keys %{$corr{$mot1}}) {
         $freqMin=$occ{$mot1};
         if ($occ{$mot2}<$freqMin) {
            $freqMin = $occ{$mot2};
         }
         #si le couple apparait moins de 2 fois et si son nombre d'occurrences est
inférieur à  5% de la fréquence du moins fréquent
         if (($corr{$mot1}{$mot2} <=$nbCorrMin) || ($corr{$mot1}{$mot2} / $freqMin
<$pourcentFreqMin)) {
            $corr{$mot1}{$mot2}=0;
            $corr{$mot2}{$mot1}=0;
```

```
        } else {
                                                    print "Conservation du couple
($mot1,$mot2) nb=".$corr{$mot1}{$mot2}."\n";
                                    }
      }
   }

   foreach $mot1(keys %corr) {
      foreach $mot2(keys %{$corr{$mot1}}) {
         if ($corr{$mot1}{$mot2}==0) {
            delete $corr{$mot1}{$mot2};
         }
      }
   }
#aprés filtrage, on stocke les correspondances dans la base stock_corr
#on ne stocke que les paires de même catégorie, uniquement pour Noun, Verb, Adj et
Adv

dbmopen(%stock_corr,$outPath."stock_corr",0666);

foreach $mot1(keys %corr) {

                    $mot1 =~/^[a-z]{2}-(.*?)-.*/;
                    $catmot1 = $1;

        if ($catmot1 =~ /Noun/ || $catmot1 =~ /Verb/ ||$catmot1 =~ /Adv/ ||$catmot1 =~
    /Adj/ ){   #              Rajouter d'autres cats (np ...etc)
                          foreach $mot2(keys %{$corr{$mot1}}) {
                                        $nbocc = $corr{$mot1}{$mot2};
                                        $concat = "$mot1#$mot2";
                                        print "$concat->$nbocc\n";
                                        $stock_corr{$concat} = $nbocc;

                          }
        }
}
dbmclose (%stock_corr) || die ("impossible de refermer");
dbmclose (%occ) || die ("impossible de refermer");
close (IN);
```

Annexe 14 : Script extract.cliques.v2.pl qui permet d'extraire, pour un mot donné, les cliques :

```
use diagnostics;
use IO::handle;
STDOUT->autoflush(1);
$len =$#ARGV;
my @langues;

if ($len==4) {
                @langues= @ARGV;
} else {
        @langues=("ar","fr","en","es");
        ($langue1,$langue2,$langue3,$langue4) = @langues;
}
my $path = ".\\";
#----------------------------------paramètres de la fonction principale
my $motdep = "mentionner"; #mot de départ
my $catdep = "Verb";          #Noun, Adj, Verb
my $cat = "(?:Noun|Verb|Adj)";
my $seuilNbOcc=3;             #seuil du nombre d'occurrences de 2 mots appariés
my $seuilFreq= 0.01;          #seuil de fréquence d'un mot
my $cliqueMinSize = 4;        # nombre de langues impliqué au minimum dans chaque
clique
my $L1 = $langue1;            #langue du mot de départ
my $L2 = $langue2;
my $L3 = $langue3;
my $L4 = $langue4;
my $file_clique="cliquemax_$L1-$catdep-$motdep"; #fichier de sortie des cliques max
my $verbose=1;                # affichage de toutes les traces
#tableaux de stockage
my %freq =();   #on y stocke les fréquences des mots (à  partir de dbm occ).  cle : L-CAT-
LEM -> valeur : freq
my %corr =();   #on y stocke le nombre d'occurrences des couples de mots (à  partir de
dbm stock_corr) cle : L-CAT-LEM ->  L-CAT-LEM -> valeur : cooc
my %occ;                # hachage liÃ© aux dbm sauvegardÃ©es
my %stock_corr; # hachage liÃ© aux dbm sauvegardÃ©es

#-----------------------------on construit les cliques à  partir de notre mot de départ

mot_clique
($motdep,$catdep,$cat,$seuilNbOcc,$seuilFreq,$L1,$L2,$L3,$L4,$file_clique);

#on stocke ensuite toutes les informations nécessaires dans log_clique.txt

open (OUT,">>$path.log_clique.txt");
print OUT "ce test a Ã©tÃ© effectuÃ© le : ";
@date = localtime(time);
```

```
$annee = $date[5]+= 1900;
$mois = $date[4]+1;
print OUT " $date[3]/$mois/$annee ($date[2]h, $date[1]mns, $date[0]s)\n";
print OUT "le mot traitÃ© est $L1-$catdep-$motdep\n";
print OUT "le seuil du nombre d'occurrences est $seuilNbOcc\n";
print OUT "le seuil de frÃ©quence est $seuilFreq\n\n";
print OUT "les cliques rÃ©sultantes sont : \n";
open (IN, "$path$file_clique") or die;
while ($ligne =<IN>){
                print OUT $ligne;
}
print OUT "\n elles sont stockées dans le fichier $file_clique";
print OUT "\n\n";
print OUT"**************************************";
print OUT "\n\n";
close (IN);
close (OUT);
```

#----------fonction principale

```
sub mot_clique {
        @l = @_;
        my $mot_clique = shift (@l);
        my $catdep = shift (@l);
        my $cat = shift (@l);
        my $seuilNbOcc = shift (@l);
        my $seuilFreq = shift (@l);
        my $L1 = shift (@l);
        my $L2 = shift (@l);
        my $L3 = shift (@l);
        my $L4 = shift (@l);
        my $file_clique = shift (@l);
        my $mot= "$L1-$catdep-$mot_clique";
        print "le mot est $mot\n";
        print "le nom du fichier est $file_clique\n";
```

 # ETAPE 1 : on remplit tout d'abord les tableaux corr et freq à partir de stock_corr et occ

 # on part d'abord de notre mot puis il faut répéter la même opération avec les mots qui lui sont appariés puisqu'il nous faudra tester les relations entre $motdep et appariés, et entre mots appariés

 # les données stockées dans ces tableaux permettront de construire les cliques

 # remplissage de %corr

```
print "ouverture de la dbm  ".$path."stock_corr\n";
        dbmopen(%stock_corr,$path."stock_corr",0600) or die;

        print "Ajout dans %corr des correspondances de $mot\n";
```

```
cree_tabl($mot_clique,$catdep,$L1);
        foreach my $cle (keys %corr) {
                if ($cle ne $mot) {
                        if ($cle =~/^(\w\w)-($cat)-(.*)/) {
                                        print "Ajout dans %corr des correspondances de
            $cle\n";
                                        cree_tabl ($3 ,$cat,$1);
                        } else {
                                $verbose && print "La clÃ© $cle est nÃ©gligÃ©e\n";
                           }
    }
}
dbmclose(%stock_corr);
undef %stock_corr;    # effacement du tableau

# remplissage de %freq
dbmopen(%occ,$path."occ",0600);
foreach my $cle(keys %corr) {
                if (!exists($occ{$cle})) {
                        die "Pas de statistique d'occurrences pour $cle\n";
                        $occ{$cle}=0;
                }
                if (exists($freq{$cle})) {
                        $freq{$cle} += $occ{$cle};
                } else {
                        $freq{$cle} = $occ{$cle};
                }
        }
        dbmclose (%occ) || die ("impossible de refermer");
        undef %occ; # effacement du tableau

        #on poursuit par le filtrage du tableau corr (ou utiliser fonction purge) pour ne
garder que les relations pertinentes
        #on peut jouer avec les paramétres pour faire varier les seuils

        foreach $mot1(keys %corr){
                foreach $mot2(keys %{$corr{$mot1}}) {
                        if (($corr{$mot1}{$mot2} < $seuilNbOcc) &&
       verif_freq($mot1,$mot2)){
                                        $corr{$mot1}{$mot2}=0;
                                        $corr{$mot2}{$mot1}=0;
                        }
                        else {
                                        $verbose && print "on conserve la relation entre
          $mot1 et $mot2\n";
                        }
                }
        }
```

```
foreach $mot1(keys %corr) {
        foreach $mot2(keys %{$corr{$mot1}}) {
                if ($corr{$mot1}{$mot2} ==0 ){
                        delete $corr{$mot1}{$mot2};
                        $verbose && print "on efface la relation entre $mot1
  et $mot2\n";
                }
        }
}

# ETAPE 2 : constitution des cliques minimales
# on construit ensuite tous les quadruplets possibles contenant notre mot de
départ
# on récupére ici uniquement les cat N, V, Adv, Adj
# on obtient des cliques composées uniquement de N, ou de V ...

my @list_clique_min;   #liste des groupes de quadruplets

# calcul préliminaire des cliques minimales complétes, si $cliqueMinSize égal
à 4
if ($cliqueMinSize==4) {
        foreach my $cle_fr (keys %corr) {
                #je cherche la premiére clé commençant par fr-
                if ($cle_fr =~/$L1-$catdep-$mot_clique/) {
                        #j'extrais le mot dans $mot_fr
                                $mot_fr= $cle_fr;
                        #je cherche ensuite le premier mot anglais qui lui est
    associé
                        foreach my $cle_en(keys %{$corr{$cle_fr}}) {
                                if (($cle_en =~ /$L2-$cat-.*/) &&
  exists($corr{$cle_fr}{$cle_en})){
                                        #j'extrais le mot qui satisfait la
  condition dans $mot_en
                                        $mot_en=$cle_en;
                                        foreach my $cle_ar (keys
    %{$corr{$cle_en}})  {
                                                #je cherche ensuite le premier
    mot arabe qui lui est associé
                                                if (($cle_ar =~ /$L3-$cat-.*/)
    && exists($corr{$cle_ar}{$cle_en}) && exists($corr{$cle_ar}{$cle_fr})) {
                                                        #j'extrais le mot qui
    satisfait la condition dans $mot_ar
                                                                $mot_ar =
    $cle_ar;
                                                                foreach my
    $cle_es (keys %{$corr{$cle_ar}})   {
                                                                        #je
    cherche ensuite le premier mot espagnol qui lui est associé
```

```
                                                                    if
            (($cle_es=~/$L4-$cat-.*/) && exists($corr{$cle_es}{$cle_fr})&&
            exists($corr{$cle_es}{$cle_en})&& exists($corr{$cle_es}{$cle_ar})) {

                    #j'extrais le mot qui satisfait la condition dans $mot_es

                    $mot_es = $cle_es;

                    $adr_mot_clique=[];

                    push(@{$adr_mot_clique},$mot_fr,$mot_en,$mot_ar,$mot_es);

                    print "Clique minimale complÃ¨te :
                $mot_fr,$mot_en,$mot_ar,$mot_es\n";

                    #je mets la liste dans ma liste globale

                    push (@list_clique_min,$adr_mot_clique);
                                                                        }
                                                                    }
                                                                }
                                                            }
                                                        }
                                                    }
            }

            if (@list_clique_min==0) {
                        # si pas de clique minimal, on commence par le singleton
   [$mot]
                        print "ATTENTION : Pas de clique minimale complÃ¨te
   pour $mot\n";
                        $adr_mot_clique=[$mot];
                        push (@list_clique_min,$adr_mot_clique);
            }
        } else {
            # initialisation des clique min avec le singleton [mot]
            $adr_mot_clique=[$mot];
            push (@list_clique_min,$adr_mot_clique);
        }

        # ETAPE 3 : augmentation itÃ©rative des cliques à  partir des cliques
minimales, pour chacune d'elle
        # Chaque clique augmentée est ajouté à  @cliques.
        # Quand une clique n'est plus augmentable est ajoutée à  %cliques_max
(comme clé résultant du tri de ses éléments)

        my %cliques_max= ();    #hash permettant de stocker les cliques finales
```

```
open (OUT,">$path$file_clique");
$i=1;
#on prend les quadruplets un par un, et on cherche à  les augmenter par palier

foreach my $adr_clique_min (@list_clique_min) {
	print "\n\nTraitement de la clique minimale numéro $i :\n";

	# construction d'un hachage permettant de caractériser l'appartenance
  d'un mot à  la clique min courante
	my %clique_min=();
	foreach my $mot (@{$adr_clique_min}) {
		$clique_min{$mot}=1;
	}

	# Etape 3.a : pour chaque mot de la liste, on crée 4 ensembles de mots
candidats
	# Le hachage de hachages ci-dessous enregistre les candidats à
l'extension pour chaque langue
	my %candidats;
	foreach my $langue (@langues) {
		$candidats{$langue}={};
	}

	# chaque mot de la clique min permet d'obtenir un ensemble de
candidats pour l'extension, avec tous ses correspondants
	foreach my $mot (keys %clique_min) {
		foreach my $corr (keys %{$corr{$mot}}) {
			if (! exists($clique_min{$corr})) {
				if ($corr=~/^(\w\w)-/) {
					$langue=$1;
					$candidats{$langue}{$corr}++;
					$verbose && print "Ajout du candidat
	$corr\n";
				} else {
					die "le mot $corr est mal formé\n";
				}
			}
		}
	}

	# ETAPE 3.b : on va tenter d'augmenter les cliques itérativement à
partir des ensembles de mots-candidats. Chaque nouvelle clique est enregistrée dans
@cliques
	# Les cliques qui ne sont pas augmentables sont supprimées de
@cliques et enregistrées comme clé dans %cliques_max

	@cliques=();	#liste des cliques que l'on essaie d'augmenter
	# on initialise avec la clique min.
	push (@cliques, $adr_clique_min);
```

```
            #tant que la liste cliques n'est pas vide, on essaie d'augmenter
            while (@cliques != 0){
                    %new_clique =();     #tableau de stockage des cliques trouvÃ©es
                    foreach $adresse_clique(@cliques) {
                            $notmax =0;
                            #on teste l'appartenance Ã  la clique pour chaque
    candidat
                            foreach $nv_fr (keys %{$candidats{"fr"}}){
                                    if (!membre($nv_fr,@{$adresse_clique})) {
                                            $is_clique=1;
                                        foreach $mot (@{$adresse_clique}) {
                                            if ($mot!~/fr-/) {
                                                    $is_clique=$is_clique
        && exists($corr{$mot}{$nv_fr});
                                            }
                                        }
                                        if  ($is_clique) {
                                            $notmax =1;
                                            $adr_new_clique = [];
                                            push (@{$adr_new_clique},($nv_fr,
@{$adresse_clique}));
                                            $cle = cle_clique ($adr_new_clique);
                                            if (!exists ($new_clique{$cle})) {
                                                    $new_clique{$cle} =
    $adr_new_clique;
                                            }
                                            $verbose && print "ajout de $nv_fr Ã
    la clique ".@{$adresse_clique}."\n";
                                        } else {
                                            $verbose && print "non ajout de
    $nv_fr Ã  la clique @{$adresse_clique}\n";
                                        }
                                    }
                            }
                            foreach $nv_en(keys %{$candidats{"en"}}){
                                    if (!membre($nv_en,@{$adresse_clique})) {
                                        $is_clique=1;
                                        foreach $mot (@{$adresse_clique}) {
                                                if ($mot!~/en-/) {
                                                    $is_clique=$is_clique
          && exists($corr{$mot}{$nv_en});
                                                }
                                        }
                                        if  ($is_clique) {
                                            $notmax =1;
                                            $adr_new_clique = [];
                                            push (@{$adr_new_clique},($nv_en,
    @{$adresse_clique}));
```

```
                                        $cle = cle_clique ($adr_new_clique);
                                        if (!exists ($new_clique{$cle})) {
                                                $new_clique{$cle} =
$adr_new_clique;
                                        }
                                        $verbose && print "ajout de $nv_en à
la clique ".@{$adresse_clique}."\n";
                                } else {
                                        $verbose && print "non ajout de
$nv_en à  la clique @{$adresse_clique}\n";
                                }
                        }
                }
                foreach $nv_ar(keys %{$candidats{"ar"}}){
                        if (!membre($nv_ar,@{$adresse_clique})) {
                                $is_clique=1;
                                foreach $mot (@{$adresse_clique}) {
                                        if ($mot!~/ar-/) {
                                                $is_clique=$is_clique
        && exists($corr{$mot}{$nv_ar});
                                        }
                                }
                                if  ($is_clique) {
                                        $notmax =1;
                                        $adr_new_clique = [];
                                        push (@{$adr_new_clique},($nv_ar,
@{$adresse_clique}));
                                        $cle = cle_clique ($adr_new_clique);
                                        if (!exists ($new_clique{$cle})) {
                                                $new_clique{$cle} =
    $adr_new_clique;
                                        }
                                        $verbose && print "ajout de $nv_ar à
la clique ".@{$adresse_clique}."\n";
                                } else {
                                        $verbose && print "non ajout
    de $nv_ar à  la clique @{$adresse_clique}\n";
                                }
                        }
                }
                foreach $nv_es(keys %{$candidats{"es"}}){
                        if (!membre($nv_es,@{$adresse_clique})) {
                                $is_clique=1;
                                foreach $mot (@{$adresse_clique}) {
                                        if ($mot!~/es-/) {
                                        $is_clique=$is_clique &&
exists($corr{$mot}{$nv_es});
                                        }
                                }
```

```
                                        if  ($is_clique) {
                                                $notmax =1;
                                                $adr_new_clique = [];
                                                push (@{$adr_new_clique},($nv_es,
        @{$adresse_clique}));
                                                $cle = cle_clique ($adr_new_clique);
                                                if (!exists ($new_clique{$cle})) {
                                                        $new_clique{$cle} =
        $adr_new_clique;
                                                }
                                                $verbose && print "ajout de $nv_es à
    la clique @{$adresse_clique}\n";
                                        }
                                        else {
                                                $verbose && print "non ajout de
        $nv_es à  la clique @{$adresse_clique}\n";
                                        }
                                }
                        }
                        # si la clique est maximale, on la stocke dans
%clique_max
                        if ($notmax ==0){
                                $cle = cle_clique ($adresse_clique);
                                if (!exists ($cliques_max{$cle})) {
                                        print OUT "@{$adresse_clique}\n";
                                        print "AJOUT DE LA CLIQUE DE CLE :
        $cle\n";
                                        $cliques_max{$cle} = $adresse_clique;
                                }
                        }
                }
                #on copie les cliques de new_clique dans la liste de cliques pour
les tester
                #quand new_clique sera vide, on sortira de la boucle
                @cliques = values %new_clique;
        }
        $i++;
    }
    close (OUT);
}

#----------------------------------------fonctions appelées par la fonction mot_clique

#permet de remplir le tableau global %corr avec toutes les correspondances de
$mot_clique
sub cree_tabl {
    my @l = @_;
    my $mot_clique = shift (@l);
    my $cat = shift (@l);
```

```
        my $L = shift (@l);

        my $mot= "$L-$cat-$mot_clique";
        foreach $cle (keys %stock_corr) {
                if ($cle=~/^($mot)#(.*)$/) {

                                my $mot1=$1;
                                my $mot2=$2;
                                $nbocc = $stock_corr{$cle};
                                $corr{$mot1}{$mot2} =$nbocc;
                                $corr{$mot2}{$mot1} =$nbocc;
                }
    }

        return %corr;
}
```

#teste si la fréquence est suffisante
 # Vérifie que le rapport entre les cooccurrences et les occurrences dépasse un certain seuil ($seuilFreq)

```
sub verif_freq {
        my $filtre=0;
        my @l = @_;
        my $mot1 = shift (@l);
        my $mot2 = shift (@l);
        $freq1 = $freq{$mot1};
        $freq2 = $freq{$mot2};
        $freq_ens12 = $corr{$mot1}{$mot2};
        $freq_ens21 = $corr{$mot2}{$mot1};

        if ($freq_ens21 != $freq_ens12) {
                die "ANOMALIE : $freq_ens21 <> $freq_ens12\n";
        }

        print "$mot1 ($freq1), $mot2 ($freq2) => $freq_ens12\n";
        unless ( (($freq_ens12 / $freq1) >$seuilFreq) || (($freq_ens21 / $freq2) >$seuilFreq)
) {
                $filtre = 1;
                print "la fréquence n'est pas assez importante\n";
        }
        return $filtre;
}
```

#teste si le mot ne fait pas déjà partie de la clique

```
sub membre {
        my $e = shift @_;
        my @l = @_;
        foreach $element(@l) {
```

```
                if ($e eq $element) {
                        return 1;
                }
        }
        return 0;
}
```

#calcule les clés du tableau %clique_max

```
sub cle_clique {
        (my $adr) = @_;
        my @l = sort @{$adr};
        return join ("",@l);
}
```

Annexe 15 : Script gizaOutput.pl déstiné à fusionner les alignements obtenus par Giza++ dans les deux sens :

```
use strict;
#use locale;

my $lang1;
my $lang2;
use Encode;
use IO::Handle;

STDOUT->autoflush();

#-----------------------------------Variable pour le fichier pal
our @t=("","\t","\t"x2,"\t"x3,"\t"x4,"\t"x5);

# nom des fichiers d'alignements dans les deux directions
# paramètres

my $seuilDistInterGroup=5;    # distance maximale entre deux unités pour faire partie d'un même groupe (1=contiguité)
my $heuristic="compat";       # 3 valeurs possibles pour le mélange des deux directions s->t et t->s :

                    # "inter" -> ne conserve que les relations communes

                    # "union" -> conserve toutes les relations et les interprète dans les deux sens

                    # "compat" -> une relation s->t n'est gardé que si l'on a t->s ou t->0 (idem dans l'autre sens)

# variables globales

my $pairNum=1;
my @sourceSents;     # liste des phrases sources. Chaque phrase est elle-même une liste de tokens.
my @targetSents;     # liste des phrases cibles. Chaque phrase est elle-même une liste de tokens.
my @t2s;             # liste des hachages contenant les relations target2source
my @s2t;             # liste des hachages contenant les relations source2target
my %s2t;
my %t2s;
my @scores;
my %traites;
my $open1=0;
my $open2=0;
opendir(DIR, ".");
```

```
while(my $file=readdir(DIR)) {

        if ( !-f $file ||$file !~/(A3\.(.*?)-(.*?)\.final)$/ || exists ($traites {$file})) {  next; }
   $lang1=$2;
   $lang2=$3;
   my $fileS2T=$file;
   my $fileT2S= "A3.$lang2-$lang1.final";
   $traites{$fileT2S}=$1;

   open(FILES2T,"<:encoding(utf8)",$fileS2T);
   open(FILET2S,"<:encoding(utf8)",$fileT2S);
   delete $s2t{$_};    #A.A -->vider hash
   delete $t2s{$_};

   while (! eof (FILES2T )) {

      # 1. lecture des trois prochaines lignes de FILES2T

      my $scoreLine=<FILES2T>;
      my $sourceLine=<FILES2T>;
      my $targetLine=<FILES2T>;

       if ($sourceLine=~/(Tagged\.\d+)\.$lang1\.txs/) {
          if ($open1) {
                close(OUT);
          }
          print "écriture de $1.$lang1-$lang2.pal\n";
          open(OUT,">:encoding(utf8)",$1.".$lang1-$lang2.pal");
          $open1=1;
       }

      # lecture du score

      my $scoreS2T=0;
      if ($scoreLine=~/#.*alignment score : (.*)/) {
         $scoreS2T=$1;
      } else {
         die "1. problème avec la ligne score du couple $pairNum\n";
      }

      # découpage de la phrase source

      my @sourceSent=split(/ /,$sourceLine);

      # toutes les phrases commencent par le premier token vide, d'indice 0 : "NULL"

      unshift(@sourceSent,"NULL");
```

```
# boucle d'analyse de la phrase cible avec les correspondances

my @targetSent=();
my %h1;
my $line=$targetLine;

chomp $line;
my $numTokT=0;
while ($line) {

   # découpage en trois éléments : 1/ Token 2/ Correspondances 3/ Reste

   if ($line=~/^(.*?) \(\{ ((?:\d+ )*)\}\) (.*)/) {
      my $tok=$1;
      my $aligned=$2;
      if ($aligned eq"") {
         $aligned="0";
      }
      # le reste sera traité à la prochaine itération
      $line=$3;
      # enregistrement du token courant dans @targetSent
      push(@targetSent,$tok);

      # analyse des correspondances

      my @aligned=split(/ /,$aligned);

      foreach my $s (@aligned) {
         $h1{"$s-$numTokT"}=1;
      }
   } else {

      die "1. problème avec la ligne cible du couple $pairNum\n";
      $line="";
   }
   $numTokT++;
}

$numTokT--;
push(@s2t,\%h1);
push(@sourceSents,\@sourceSent);
push(@targetSents,\@targetSent);

# 2.  lecture des trois prochaines lignes de FILET2S

$scoreLine=<FILET2S>;
$targetLine=<FILET2S>;
$sourceLine=<FILET2S>;
```

```
# lecture du score

my $scoreT2S=0;

if ($scoreLine=~/#.*alignment score : (.*)/) {
  $scoreT2S=$1;
} else {
   die "2. problème avec la ligne score du couple $pairNum\n";
}

# boucle d'analyse de la phrase source avec les correspondances

my %h2;
$line=$sourceLine;
chomp $line;
my $numTokS=0;

while ($line) {
   # découpage en trois éléments : 1/ Token 2/ Correspondances 3/ Reste
   if ($line=~/^(.*?) \(\{ ((?:\d+ )*)\}\) (.*)/) {
      my $tok=$1;
      my $aligned=$2;
      if ($aligned eq"") {
         $aligned="0";
      }
      # le reste sera traité à la prochaine itération
      $line=$3;
      # analyse des correspondances
      my @aligned=split(/ /,$aligned);

      foreach my $t (@aligned) {
         $h2{"$numTokS-$t"}=1;
      }
   } else {
      die "2. problème avec la ligne cible du couple $pairNum\n";
      $line="";
   }
   $numTokS++;
}
$numTokS--;
push(@t2s,\%h2);

# calcul de l'union et de l'intersection

my %union=%h1;    # toutes les relations sont prises dans les deux sens
my %inter=%h1;    # on ne garde que les couples avec des relations dans les deux sens
my %compat=();    # une relation s->t est conservée si t->0 ou t->s

# on complète l'union
```

```
foreach my $key (keys %h2) {
   $union{$key}=1;
}
# on réduit l'intersection
foreach my $key (keys %inter) {
   if (!exists($h2{$key})) {
      delete($inter{$key});
   }

}

# on conserve les relations compatibles

foreach my $key (keys %h1) {
   my ($num1,$num2)=split(/-/,$key);
   if (exists($h2{$key})) {
      $compat{$key}=1;
   } else {
      if (exists ($h2{"$num1-0"})) {
         $compat{$key}=1;
      }
   }
}
foreach my $key (keys %h2) {
   my ($num1,$num2)=split(/-/,$key);
   if (exists ($h1{"0-$num2"})) {
      $compat{$key}=1;
   }
}
# calcul de la closure transitive :
# application de l'heuristique

my %hash;
if ($heuristic eq "compat") {
   %hash=%compat;
}
if ($heuristic eq "inter") {
   %hash=%inter;
}
if ($heuristic eq "union") {
   %hash=%union;
}
```

my @groupesS=(0,1..$numTokS); # chaque token source est rattaché à un numéro de groupe. Initialement, chaque groupe ne contient qu'un seul token

my @groupesT=(0,1..$numTokT); # chaque token cible est rattaché à un numéro de groupe.

```
    my %s2t;                # enregistre les alignements entre groupe S et groupe T
    my %t2s;                # enregistre les alignements entre groupe T et groupe S
    my $continue=1;
```

 # Regroupements itératif en fonction de la cloture transitive des alignements. On sort lorsque plus aucun nouveau groupement n'est possible

```
    while ($continue) {
        $continue=0;        # on ne lance la prochaine itération que si le parcours de hash aboutit à un nouveau regroupement

      foreach my $key (keys %hash) {
        my ($numS,$numT)=split(/-/,$key);
        if ($numS*$numT==0) { next;}    # on ne tient pas compte des relations de non-alignement
        my $groupeS=$groupesS[$numS];
        my $groupeT=$groupesT[$numT];
        # si le groupe image existe déjà et est différent on regroupe $numT dans ce groupe
        if (exists($s2t{$groupeS}) && $s2t{$groupeS} != $groupeT) {
          # si le numéro de groupe actuel est supérieur à l'image du groupe
          if ($groupeT>$s2t{$groupeS} ) {
            # on opère le regroupement à condition d'avoir une distance faible entre les groupes (si =1 alors continuité)
            if ($groupeT-$s2t{$groupeS}<=$seuilDistInterGroup) {
              $groupesT[$numT]=$s2t{$groupeS};
              $groupeT=$s2t{$groupeS};
              $continue=1; # puisqu'il y a eu regroupement il faudra réitérer
            }
          } else {
            # on retient toujours l'image avec le numéro de groupe plus petit
            $s2t{$groupeS}=$groupeT;
            $continue=1; # puisqu'il y a eu permutation il faudra réitérer
          }
        # sinon on enregistre simplement la correspondance entre les 2 groupes
        } else {
          $s2t{$groupeS}=$groupeT;
        }

        # si le groupe image existe déjà et est différent on regroupe $numS dans ce groupe
        if (exists($t2s{$groupeT}) && $t2s{$groupeT} != $groupeS) {
          if ($groupeS>$t2s{$groupeT}) {
            # on opère le regroupement à condition d'avoir une distance faible entre les groupes (si =1 alors continuité)
            if ($groupeS-$t2s{$groupeT}<=$seuilDistInterGroup) {
              $groupesS[$numS]=$t2s{$groupeT};
              $groupeS=$t2s{$groupeT};
```

```
                $continue=1;    # puisqu'il y a eu regroupement il faudra réitérer
              }
            } else {
              # on retient toujours l'image avec le numéro de groupe plus petit (le
regroupement se fera lors d'une prochaine itération)
              $t2s{$groupeT}=$groupeS;
              $continue=1;    # puisqu'il y a eu permutation il faudra réitérer
            }
          # sinon on enregistre simplement la correspondance entre les 2 groupes
          } else {
            $t2s{$groupeT}=$groupeS;
          }
        }
      }
      # on imprime le résultat en sortie
      my %grS; # clé : le num du groupe - valeur : une liste triée avec tous les nums sources
dans le groupe
      for (my $i=0;$i<=$#groupesS;$i++) {
        if (exists( $grS{$groupesS[$i]})) {
          my @l=@{$grS{$groupesS[$i]}};
          push(@l,$i);
          @l=sort {$a <=> $b} @l;
          $grS{$groupesS[$i]}=\@l;
        } else {
          $grS{$groupesS[$i]}=[$i];
        }
      }
      my %grT; # clé : le num du groupe - valeur : une liste triée avec tous les nums sources
dans le groupe
      for (my $i=0;$i<=$#groupesT;$i++) {
        if (exists( $grT{$groupesT[$i]})) {
          my @l=@{$grT{$groupesT[$i]}};
          push(@l,$i);
          @l=sort{$a <=> $b} @l;
          $grT{$groupesT[$i]}=\@l;
        } else {
          $grT{$groupesT[$i]}=[$i];
        }
      }
      # fonction utilisé dans le tri des groupes sources

      sub prem {
        my $numGr=shift;
        if (exists($grS{$numGr})) {
          my @l=@{$grS{$numGr}};
          return $l[0];
        } else {
          return $numGr;
        }
```

```
 }
    # on trie les groupes sources en fonction de la position du premier élément de
chacun d'eux

    my @keys=sort {prem($a) <=> prem($b)} keys %grS;
    foreach my $key (@keys) {
       my @groupeS=@{$grS{$key}};
       if (!exists($s2t{$key})) {
       } else {
          if (exists($s2t{$key})) {
             my $numT=$s2t{$key};

             if (exists($grT{$s2t{$key}})) {
                my @groupeT=@{$grT{$s2t{$key}}};
                my $groupeS=join(" ",map {$sourceSent[$_]} @groupeS);
                my $groupeT=join(" ",map {$targetSent[$_]} @groupeT);
                 print OUT "$groupeS$groupeT\n";
             } else {
                print "Pas de valeur trouvée pour le groupe $numT\n";
                die;
             }
          }
       }
    }

    # incrémentation du nombre de paires traitées

    $pairNum++;
  }
  delete $s2t{$_};    #vider hash
  delete $t2s{$_};
  close(FILET2S);
  close(FILES2T);

}
```

Annexe 16 : Script hash-WN4.pl déstiné au ratachement de cliques à EuroWordNet :

```
use IO::Handle;
STDOUT->autoflush(1);
#----------------------------------------Variables
my $nn=1;
my $offset;
my $caat;
my $GLOSS;
my $concat;
my $Target;
my $OFF;
my $valeur;
my $Rien="";
my $liste="";
my $corresp;
my $relationn;
my $relatio;
my %IDS;
my %TARGETILIS;
my %ILI;
my %catts;
my %offsets;
my %GLOSSES;
my %concattt;
my %concatt;
my %relationns;

#----------------------------------------Lecture de fichier ILI-WN :
my $file1="ILI_WN15.ewn";
open (IN,"<",$file1);
while ($ligne=<IN>){
  if($ligne!~/ILI_RECORD/){
     if($ligne!~/EQ_RELATION\s+?"(.+?)\"/){
        if($ligne!~/TARGET_ILI/){
           if($ligne=~/PART_OF_SPEECH\s+?\"(.*?)\"/){
              $caat=$1;
              $catts{$nn}=$caat;

              # WORDNET_OFFSET ou ADD_ON_ID: Code identifiant le synset par référence au dictionnaire Princeton Wordnet 1.5.//ADD_ON_ID est utilisé quand il y a des relations entre le synset et autres synsets
            }elsif($ligne=~/WORDNET_OFFSET\s+?(\d+)?/){
              $offset=$1;
               $offsets{$nn}=$offset;
            }elsif($ligne=~/ADD_ON_ID\s+?(\d+)?/){
              $ID=$1;
               $IDS{$nn}=$ID;
               $ligne=<IN>;
```

```
                if($ligne=~/GLOSS\s+?\"(\"*.+\"*\"*\\*.+?)\"/){
                   $GLOSS=$1;
                   $GLOSSES{$nn}=$GLOSS;
                   $concatt{$nn}=$caat."#".$GLOSS;
                  $valeur= $concatt{$nn};
                }

            }elsif($ligne=~/GLOSS\s+?\"(\"*.+\"*\"*\\*.+?)\"/){
                  $GLOSS=$1;
                  $GLOSSES{$nn}=$GLOSS;
                  $concattt{$nn}=$caat."#".$GLOSS."\n";
                 $ILI{$offsets{$nn}}=$concattt{$nn};
            }
          }else{    #pour continuer s'il y a encore TARGET-ILI (relation avec autres synsets).....
            $ligne=<IN>;
            if($ligne=~/WORDNET_OFFSET\s+?(\d+)?/){
               $OFF="**".$1;
               $liste=$OFF."**".$Rien;
               $Rien=$liste."\n";
               $TARGETILIS{$IDS{$nn}}=$Rien;
               $ILI{$IDS{$nn}}=$valeur.$relatio.$TARGETILIS{$IDS{$nn}};
            }
          }

      }else{ #pour continuer avec EQ_RELATION (afin d'indentifier le type de relation entre les synsets reliés).....
            $relationn=$1;
            $relationns{$nn}=$relationn;
            $relatio=$relationns{$nn};
      }
   }else{
   $nn++;
   $Rien="";
   }
}
close(IN);
#------------------------------------------------------------ Main
my $h=1;
my $lang1;
my $lang2;
my $lang3;
my $lang4;
my $v;
my $v1;
my $v2;
my $ke;
my $rest;
my $infooffset;
```

```
my $infooffset2;
my $parts;
my $part="";
my $no=0;
my $res="";
my $u="";
my $ligne="";
my $l;
my $c;
my $form;
my $cat;
my $sens;
my $idsynset;
my $memoire;
my $catin;
my $formin;
my $sensin;
my $relationin;
my $idsynsetid;
my $n=0;
my $key;
my $variable="#";
my $val;
my $memoire1; my %concat;
my %catsin;
my %formsin;
my %sensesin;
my %offset;
my %listeoffset;
my %trouves; our @mots;
our @keys;

$file2="cliquemax_fr-Verb-voir";
open (IN,"<:utf8",$file2) or die;

while ($lign=<IN>){    while($lign=~/^(\S+?)-(\S+?)-((.|\n)*)/){
      $l=$1;   #---La langue
      $c=$2;   #---La catgorie
      $bb=$3;
      $lign=$bb;
      $parts="";
      $no=0;
      while ($lign=~/^(.+?)\s*?((.|\n)*)$/ && $no==0){
         $part=$1;
         $res=$2;
         if ($lign=~/^(\S+?)$/){
            $part=$1;
            $no=1;
         }
```

```
        if ($lign=~/^(.+?)\s+?((.|\n)*)/){ #pour prendre en compte le cas des unités composées ex.into account
            $part=$1;
            $res=$2;
        }
        if ($lign=~/^(\S+?)\n*?((.|\n)*)/){
            $u=$parts.$part;
            if ($lign=~/^(\S+?)\n+?((.|\n)*)/){
                $u=$parts.$part;
            }
        }
        if ($res =~/^(\S+?)-(\S+?)-((.|\n)*)/){
            $u=$parts.$part;    #---Le lemme
            $no=1;
            $lign=$res;
        }else {
            $parts=$parts.$part." ";
            $lign=$res;
        }
    }
    $mot=$l.$c.$u;
    print $mot."\n";
    if (exists( $mots{$mot})){  #pour éviter le traitement d'un mot déjà traité
        $mots{$mot}=1;
    }else{
        $mots{$mot}=0;
        if ($c eq "Noun"){
            $c=~s/Noun/n/;   #car La catgorie Noun dans le dico est n
        }elsif ($c eq "Verb"){
            $c=~s/Verb/v/;#car La catgorie Verb dans le dico est v
        }elsif ($c eq "V"){
            $c=~s/V/v/;#car La catgorie V dans le dico est v
        }elsif ($c eq "NP"){
            $c=~s/NP/n/;#car La catgorie NP dans le dico est n
        }elsif ($c eq "Adv"){
            $c=~s/Adv/a/;#car La catgorie NP dans le dico est n
        }
        my $file="wnn-ts-$l.txt";
        print $file."\n";
        open(INN,"<","wnn-ts-$l.txt");
        #----------------------------------------Fonction pour la Lecture de Dico
        my %forms;
        my %cats;
        my %senses;
        my %relationins;
        my %relationseqs;
        my %rel;
        my %RechercheSynsets;
        my %hash;
```

```
my %hash2;
my %offsettts;
my %idsynsets;
while (!eof(INN)) {
   $ligne=<INN>;
   if($ligne!~/@(\d+?)@\sWORD_MEANING/){
      if ($ligne!~/INTERNAL_LINKS((.|\n)*)/){
         if($ligne!~/EQ_RELATION((.|\n)*)/){
            if ($ligne!~/LITERAL\s+?\"(.*?)\"/){

               #----------Enregistrer les catégories dans un hash (%cats)N°-->cat
                  if($ligne =~/PART_OF_SPEECH\s+?\"(.*?)\"/){
                     $cat=$1;
                     $cats{$n}=$cat;
                  }
               #-----------Enregister les formes dans un hash (%forms)N°-->form
               }else{
                  $form=$1;
                  $forms{$idsynsets{$n}}=$form;
                  $ligne=<INN>;

                  #--------------Enregister les senses dans un hash (%senses)N°-->sens
                  if($ligne=~/SENSE\s+?(\d+)?/){
                     $sens=$1;
                     $senses{$n}=$sens;
                        #--------------Enregister chaque synset avec son ID et tous ses
formes, catégories, senses dans un hash (%hash)
                        if (exists $hash{$idsynsets{$n}}){
                           $memoire=$hash{$idsynsets{$n}};
                           $hash{$idsynsets{$n}}=$memoire."-"."-"."-
".$form.$variable.$cat.$variable.$sens;
                        }else{
                           $hash{$idsynsets{$n}}=$form.$variable.$cat.$variable.$sens;
                        }
                        #--------------Enregister chaque forme, cat et sens pour un
synsets précis comme clé d'un hash (hash2) et la valeur est l'ID de synset
                        $hash2{$form.$variable.$cat.$variable.$sens}=$idsynsets{$n};
                  }
               }
            }else{
               #--------------Enregister le type de relation d'ILI relié à chaque synset
dans un hash(%relationeqs) ID de synset-->type de relation
               if ($ligne=~/EQ_RELATION\s+?\"(.*?)\"/){
                  $relationeq=$1;
                  $relationseqs{$idsynsets{$n}}=$relationeq;
                  $ligne=<INN>;
                  $ligne=<INN>;
                  $ligne=<INN>;
```

```
                    #----------Enregister l'ILI (seulement un seul offset est traité
pour chaque synset(on peut trouver plusieurs pour le même synset mais tous
reviennent au même gloss)relié à chaque synset dans un hash  (%offsets) ID de
synset-->offset....note:on ne s'interesse pas au type de relation
                    }
                    if ($ligne=~/WORDNET_OFFSET\s+?(\d+)?/){
                       $offsettt=$1;
                       $offsettts{$idsynsets{$n}}=$offsettt;

                    }elsif ($ligne=~/ADD_ON_ID\s+?(\d+)?/){
                       $offsettt=$1;
                       $offsettts{$idsynsets{$n}}=$offsettt;

                    }
                 }
              }else{
                 $ligne=<INN>;
                 #----------Enregister toutes les relations reliant le synset avec autres
synsets de même langue dans un hash(%relationins) ID synset-->type de relation
interne
                 my $f=0;
                 while($ligne!~/EQ_LINKS((.|\n)*)/ && (!eof (INN)) && ($f!~1)){  #pour
pouvoir sortir de la boucle
                    if ($ligne!~/EQ_LINKS((.|\n)*)/){
                       if ($ligne=~/RELATION\s+?\"(.*?)\"/){
                          $relationin=$1;
                          $relationins{$idsynsets{$n}}=$relationin;
                          #Id-synset##relation-->id synset relié
                          $ligne=<INN>;
                          $ligne=<INN>;

                          #----------Ensuite enregister la catégorie du synset relié dans
un hash(%catins) N°-->catégorie
                          if($ligne =~/PART_OF_SPEECH\s+?\"(.*?)\"/){
                             $catin=$1;
                             #$catins{$n}=$catin;
                             $ligne=<INN>;
                          }
                          #----------Ensuite enregister la forme du synset relié dans un
variable
                          if ($ligne=~/LITERAL\s+?\"(.*?)\"/){
                             $RechercheSynset=$1;
                             $ligne=<INN>;
                          }
                          #----------Ensuite enregister le sens du synset relié sens interne
                          if($ligne=~/SENSE\s+?(\d+)?/){
                             $sensin=$1;
                             #$senseins{$n}=$sensin;
```

#----------Ensuite enregister l'ensemble d'information de relation interne dans un hash (%RechercheSynsets) id-synset##relation Interne-##nombre de fois de relation(juste pour n'avoir pas les mêmes clés)->forme#catégorie#sens

```
                        if (exists($RechercheSynsets{$idsynsets{$n}})){
                            $memoire1= $RechercheSynsets{$idsynsets{$n}};
                            $RechercheSynsets{$idsynsets{$n}}=$memoire1."---
".$relationins{$idsynsets{$n}}.$variable.$variable.$RechercheSynset.$variable.$catin.$va
riable.$sensin;

                        }else{


$RechercheSynsets{$idsynsets{$n}}=$relationins{$idsynsets{$n}}.$variable.$variable.$R
echercheSynset.$variable.$catin.$variable.$sensin;
                        }
                        $ligne=<INN>;
                        if ($ligne!~/EQ_LINKS((.|\n)*)/ &&
$ligne!~/@(\d+?)@\sWORD_MEANING/ && $ligne!~/INTERNAL_LINKS((.|\n)*)/ &&
$ligne!~/RELATION\s+?\"(.*?)\"/ && $ligne !~/FEATURES/ && $ligne
!~/REVERSED/){
                            $f=1;     #car on trouve des synsets sans offsets (qu'une ligne
vide)
                        }
                    }
                    if ($ligne =~/FEATURES/){
                        $ligne=<INN>;
                        if ($ligne =~/VARIANT_TO_VARIANT/){
                            $ligne=<INN>;
                        }
                        if ($ligne =~/SOURCE_VARIANT/){
                            $ligne=<INN>;
                        }
                        if ($ligne =~/TARGET_VARIANT/){
                            $ligne=<INN>;
                        }
                    }
                    if ($ligne =~/REVERSED/){
                        $ligne=<INN>;
                    }
                }
            }
        }
    }

    }else{
        #----------L'incrémentation du N° ($n) se fait juste quand on passe à
un nouveau synset
```

```
            #----------Enregistrer l'identifiant du synset dans un hash (%idsynsets) N°->idsynset
                $n++;
                $idsynset=$1;
                $idsynsets{$n}=$idsynset;
                $idsynsetid=$idsynsets{$n};
            }
        }
    close(INN);
    #----------Lecture de fichier clique
    foreach $k(keys %hash2){

      if ($k=~/(.+?)\#(.+?)\#(.+?)/ && $mots{$mot}==0){
        my $litteral=$1;
        my $categorie=$2;
        my $NSens=$3;   #on ne s'interesse pas au sens dans cette partie !!!!!
        if(($litteral eq $u) && ($categorie eq $c)){      #Chercher La langue et la catgorie des unités de clique dans le dico
            $v=$hash2{$k};
              #----------Cherche le synsets dans le dico afin de trouver son offset (ILI)
            foreach $ke(keys %offsettts){
              if ($ke=$v){
                $v1=$offsettts{$ke};
                $ofset{$l.$categorie.$litteral.$NSens}=$v1; #pour avoir les offsets liés à chaque sens (eviter d'écraser une clé répététive)
              }
            }
            #----------Extraire les relations sémantiques "INTERNAL_LINKS" trouvées dans WN pour chaque unité de cliques:
              #--------  Extraire les concepts entretiennent des relations sémantiques (autres que la synonymie) avec chaque unité de clique :
            foreach $key2(keys %RechercheSynsets){
              if ($key2 = $v){
                $trouve=$RechercheSynsets{$key2};
                $trouves{$l.$categorie.$litteral.$NSens}=$trouve;
              }
            }
            #--------Extraire les synonymes "variants"de chaque unité de clique:
            foreach $key3(keys %hash){
              if ($key3 = $v){
                $v2=$hash{$key3};
                $synonymes{$l.$categorie.$litteral.$NSens}=$v2;
              }
            }
          }
        }
      }
    }
```

```
    }
    $h++;
}
close (IN);
#-----------------------relier les cliques avec les synsets de chaque WN en basant sur les ILI
my $vallll;
foreach  $clee(keys %ofset){
    $vallll=$ofset{$clee};
    #Cherche le Gloss qui relie les unités de clique (si c'est le cas)
    if (exists($ILI{$vallll})){
        my $valllllll=$ILI{$vallll};
        $hasssh{$clee}=$valllllll;
    }
}
#--------------------------------tester s'il existe des unités partageant le même sens
my $i=0;
my $clef;
my $tt;
my $m=0;
my $rep;
my $p=0;
my $ppp; my %answer;
my %ts;
my %answers; my @vls=();
my @cls=();
my @list=();

#------------------
if (%hasssh){
    @cls=keys %hasssh;
    @vls=values %hasssh; #ts les sens de chaque unités;

}
#------------------------
my $count=$#vls+1;
my $nomb=$#vls;
my $vlu;

while ($i<=$nomb){
    $m=0;
    $rep=0;
    while ($m<=$nomb) {
        if (exists ($list[$m])){
            if ( $vls[$i] eq $list[$m] ){
                %ts=reverse (%answer);
                foreach $t(keys %ts){
                    $tt=$ts{$t};
                    if ($t eq $vls[$i]){
```

```
                    if (exists ($answers{$t})){
                       $answer{$cls[$i]} = $vls[$i];
                       $ppp=$answers{$t};
                       $answers{$t}=$ppp.$cls[$i]; #tous les sens communs entre les unités de clique;
                       $cls[$i]="";      #pour eviter la repetition des unités ayant du sens commun
                    }else{
                       $vlu=$tt;
                       $clef=$cls[$i].$vlu;
                       $vlu=$clef;
                       $answers{$t}=$vlu;    #Relier les unités de clique ayant des sens communs;

                    }
                 }
              }
           }
        }else{
           $rep=1;
        }
        $m++;
     }
     if (($m=$count) && ($rep=1)){
           $answer{$cls[$i]} = $vls[$i];
           $list[$p]= $vls[$i];
           $p++;
     }
     $i++;
}
#----------------------Impression de résultats
open(OUT,">","Résultats".$file2."txt");
my %x;
my $e;
my %xx;
my $ee;
my %tr;
my $trvl;
my %tr1;
my $trvl1;

foreach $e (sort keys %answer){ #--la fonction sort est utilisée pour traiter langue par langue & unité par unité
   $x=$answer{$e};
   foreach $tr(keys %trouves){
      $trvl=$trouves{$tr};
      foreach $tr1(keys %synonymes){
         $trvl1=$synonymes{$tr1};
         if (($e eq $tr) && ($tr eq $tr1)){
```

```
            print OUT "Tous les sens trouvés dans WN pour chaque unité de clique:\n";
            print OUT $e."      a comme Gloss       ".$x."\n";
            print OUT "Les relations sémantiques \"INTERNAL_LINKS\" trouvées dans WN pour chaque unité de cliques:\nExtraire les concepts entretiennent des relations sémantiques (autres que la synonymie) avec chaque unité de clique\n";
            print OUT $tr."---->".$trvl."\n\n";
            print OUT "Les synonymes \"variants\"trouvés dans WN pour chaque unité de clique: \n";
            print OUT $tr1."    est synonymes avec  ".$trvl1."\n";
            print OUT "\t\t\t----------------------------------------\n\n";
          }
        }
    }
}
print OUT "\n\n";
print OUT "\t\t\t\t Les unités de clique partageant les mêmes relations ILI:\n\n\n";
foreach $ee(keys %answers){
    $xx=$answers{$ee};
    print OUT $xx."      a comme Gloss       ".$ee."\n";
}
close (OUT);
```

Références

Abderrahim M. E. (2009) .Vers une interface pour l'enrichissement des requêtes en arabe dans un système de recherche d'information. *CIIA*, Vol. 547CEUR-WS.org.

Abouenour L., K. Boueoubaa, P. Rosso (2008). Improving Q/A Using Arabic Wordnet. *Proceedings The 2008 International Arab Conference on Information Technology* (ACIT'2008), Tunisia, December.

Agirre E., G. Rigau (1996). Word Sense Disambiguation Using Conceptual Density. Departament de Lengoaia eta Sistema Informatikoak saila. p.k. 649, 200800 Donostia. Spain.

Allen F. J., Alan M. Frisch (nd). What's in a Semantic Network?. Computer Science Department. The University of Rochester. Rochester, NY 14627.

Artale A., B. Margnini et C. Strapparava (1997). Lexical Discrimination with the italian Version of WordNet. *Proceedings of ACL Workshop Automatic Information Extraction and Building of Lexical Semantic Resources*. Madrid. Spain.

Atserias J., S. Climent, X. Farreres, G. Rigau, H. Rodrïguez (1997). Combining Multiple Methods for the Automatic Construction of Multilingual WordNets. Departament de Llenguatges i Sistemes Informatics. Universitat Politecnica de Catalunya. Carrer Jordi Girona Salgado, 1-3. 08034 Barcelona, Catalonia.

Atwell E., L. Al-Sulaiti, S. Al-Osaimi, B. Abu Shawar (2004). Review of Arabic Corpus Analysis Tools Un Examen d'Outils pour l'Analyse de Corpus Arabes. *JEP-TALN 2004*, Arabic Language Processing, Fez, 19-22 April 2004.

Audibert L. (2002). Etude des critères de désambiguïsation sémantique automatique : Présentation et premiers résultats sur les cooccurrences. Jeune équipe DELIC – *RÉCITAL 2002*, Nancy, 24-27 juin 2002. Université de Provence, 29 Avenue Robert SCHUMAN, 13621 Aix-en-Provence Cedex 1.

Ayewah N., R. Mihalcea, V. Nastase, D. Tatar (2004). RSDNET: a web-based collaborative framework for building multilingual semantic networks. Studia univ. babes–bolyai, informatica, volume xlix, number 1.

Bachimont B. (2000). Engagement sémantique et engagement ontologique: conception et réalisation d'ontologies en ingénierie des connaissances. *Ingénierie des connaissances Evolutions récentes et nouveaux défis*. Jean Charlet, Manuel Zacklad, Gilles Kassel, Didier Bourigault, Eyrolles, ISBN 2-212-09110-9.

BalKanet (2003). *Tracing Of The Common Base Concepts*. (ID*)* IST-2000-29388. Version 1

Baneyx A. (2007). Construire une ontologie de la pneumologie aspects théoriques, modèles et expérimentations. Université Pierre ET Marie Curie -Paris 6

Barbut, M., Monjardet, B., (1970). Ordre et classification, Algebre et combinatoire, Tome 2, Hachette.

Baziz M., M. Boughanem, N. Nassr (2003). La recherche d¿information multilingue : désambiguïsation et expansion de requêtes basées sur WordNet. *International Symposium On Programming and Systems (ISPS 2003)*, Alger, 05/05/2003-07/05/2003, International Symposium on Programming and Systems , p. 175-186, mai 2003.

Baziz M. (2006). Représentation/recherche de documents dans les masses de données textuelles. Équipe SIG – IRIT (Systèmes d'Information Généralisés). [Diaporama électronique PowerPoint].

Ben Youssef A. (2008). *Méthodes Mixtes pour la Traduction Automatique Statistique*. Mémoire de Master2.Université Stendhal Grenoble3. 1 juillet 2008.

Benzécri J.-P. (1980). L'analyse des données : l'analyse des correspondances. Bordas, Paris.

Berkley L, B. Bargmeyer (2005)**.** Metadata Registries – Next Edition. National Laboratory University of California. *SC 32 Tutorial Session*.

Bernhard D. (2006). Apprentissage de connaissances morphologiques pour l'acquisition automatique de ressources lexicales. Université Joseph Fourier – Grenoble I

Birkhoff, G., (1940). *Lattice Theory*. First Edition, Amer. Math. Soc. Pub. 25, Providence, R. I.

Bottius E. (nd). Construction d'une interface de gestion de documents textuels a des fins de production de dictionnaires et de lexique. Laboratoire Informatique et Mathématiques Appliquées**.** Site Enseeiht de l'IRIT-UMR CNRS 5505, 2, rue Charles Camichel, BP 7122 - 31071 Toulouse cedex 7. [Diaporama électronique PowerPoint].

Brachman R. (1977). What's in a concept: structural foundations for semantic networks. *Int. Journal of Man-Machine Studies.* 9. Pp. 127-152.

Brachman R. (1979). On the Epistemological Status of Semantic Networks. Associative Networks: Representation and Use of Knowledge by Computers. *Academic Press*. Pp. 3-50.

Brill E. (1992). A simple rule-based part of speech tagger. *Proceedings of the Third Conference on Applied Computational Language (ACL) Processing*, Trento.

BrillL E. (1995). Transformation-based error-driven learning and naturallanguage processing : A case study in part-of-speech tagging. *Computational Linguistics*, 21(4), 543-565.

Bruce R. and L. Guthrie (1992). Genus disambiguation: A study in weigthed preference. *In proceedings of the 14th International Conference on Computational Linguistics* (COLING'92). Nante, France.

Charlet, J. (2002). *L'Ingénierie des connaissances : développements, résultats et perspectives pour la gestion des connaissances médicales*. Habilitation à diriger des recherches, Université Paris 6.

Casagrande, J.B. and K.L. Hale (1967). *Semantic Relationships in Papago Folk-Definitions.* In D.H. Hymes and W.E. Bittle, eds., Studies in Southwestern Ethnolinguistics, Mouton, The Hague, 165-193.

Chen B., H. Meriam, K. Olivier (2005). Contextes multilingues alignés pour la désambigüisation sémantique: une étude expérimentale. *Actes de TALN-RECITAL 2005*, 6-10 juin 2005, Dourdan, vol. 1,pp. 415-420.

Cleuziou G., Viviane C., Lionel M. (2003). Une méthode de regroupement de mots fondée sur la recherche de cliques dans un graphe de cooccurrences. *Conférence TIA-2003*, Strasbourg, 31 mars et 1 avril 2003.

Crestan E., C. de Loupy, L. Manigot (nd). Analyses sémantiques pour la navigation textuelle. Sinequa, 51-54, rue Ledru-Rollin, 92400 Ivry-sur-Seine, France.

Cristea D., C. Butnariu (nd). Hierarchical XML Layers Representation for Heavily Annotated Corpora. University "Al. I. Cuza" of Iasi. Faculty of Computer Science and Institute for Theoretical Computer Science Romanian Academy – the Iasi Branch.

CRUSE, D.A. (1986). Lexical Semantics. *University Press*. Cambridge, London, New York, Cambridge.

Curtoni P., L. Dini, V. Di Tomaso, L. Mommers, Wim Peters, P. Quaresma, E. Schweighofer, D. Tiscornia (nd) **.** Semantic access to multilingual legal information.

Daelemans W., K. De Smedt and G. Gazdar (1992). Inheritance in Natural Language Processing. *Computational Linguistics*, 18.2 1992.

Dilekh T. (2011). *Implémentation d'un outil d'indexation et de recherche des textes en arabe*. Mémoire de Master soutenue le 28 septembre 2011. Université Hadj Lakhdar – Batna.

Diab T. M. (nd). Feasibility of Bootstrapping an Arabic WordNet Leveraging Parallel Corpora and an English WordNet. Linguistics Department, Margaret Jacks Hall, Stanford University Stanford, CA 94305, USA.

DiabM., K. Hacioglu, D. Jurafsky (2004). Automatic Tagging of Arabic Text: From Raw Text to Base Phrase Chunks, *HLT-NAACL 2004*. PP 149-152.

Elkateb S. (2005). *Design and implementation of an English Arabic dictionary/editor*. PhD thesis, The University of Manchester, United Kingdom.

Elkateb S., W. Black, P. Vossen, A. Pease, C. Fellbaum, H. Rodreiguez, M. Alkhalifa (nd). Introducing the Arabic WordNet Project.

Elkateb S., W. Black, H. Rodriguez, M. Alkhalifa, P. Vossen, A. Pease, and C. Fellbaum, (2006). Building a WordNet for Arabic. *Proceedings of The fifth international conference on Language Resources and Evaluation* (LREC 2006).

Elkateb S., W. J. Black (2001). Towards the design of English-Arabic terminological and lexical knowledge base. Dept of Computation. UMIST, PO Box 88, Manchester M60 1QD, UK.

Enguehard C (nd). Acquisition de terminologie à partir de gros corpus. Institut de Recherche en Informatique de Nantes, IUT, 3, rue du Maréchal Joffre, 44041 Nantes Cedex 01 – France.

Enguehard C. (1992). ANA, Apprentissage Naturel Automatique d'un Réseau Sémantique. Contrôle des systèmes. Université de Technologie de Compiègne.

Fahlman, S. E. (1979). NETL: A system for representing and using real world knowledge. Cambridge. Mass.: *MIT. Press*, 1979.

Farreres X., G. Rigau, H. Rodríguez (1998)**.** Using WordNet for Building WordNets. Departament de Llenguatges i Sistemes Informàtics.. Universitat Politècnica de Catalunya. Barcelona. Spain.

Fellbaum C., W. Black, S. Elkateb, A. Marti, A. Pease, H. Rodriguez, P. Vossen, Irion (nd) Constructing Arabic WordNet in Parallel with an Ontology. Funded by the REFLEX Program, DOI. [Diaporama électronique PowerPoint].

Fellbaum C. (1998). WordNet : An Electronic Lexical Database. Chapter A semantic network of English verbs. *MIT Press*, Cambridge, MA.

Festinger L (1949). *The analysis of sociograms using matrix algebra*. Human Relations 10:153-58.

François J. (nd). L'espace sémantique comparé des deux verbes français tenir et allemand halten : noyau et extensions. CRISCO, FRE 2805, Université de Caen.

Franz J. O., N. Hermann (2003). A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, volume 29, number 1, pp. 19-51 March 2003

Gale W., K. W. Church (1991). A program for aligning sentences in bilingual corpora. *Proceedings of the 29th Annual Meeting of the ACL*, Berkeley, CA, pp. 177-184.

GALE W., K. W. Church et D. Yarowsky (1992). Estimating Upper and Lower Bounds on the Performance of Word Sense Disambiguation. *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics*, ACL, 1992.

Gaume B., F. Venant, B. Victorri (2006). Hierarchy in lexical organisation of natural languages. *Hierarchie in Natural and social sciences*, methods series, vol 3, springer, 2006-Draft.

Godbert E. (1991). Contribution a l'étude des réseaux sémantiques. Élaboration d'un réseau sémantique pour la modélisation de la connaissance lexicale. L'université d'Aix-Marseille ii. Informatique et mathématiques. (Intelligence artificielle).

Godin R., G. Mineau, R. Missaoui., (1995). Méthodes de classification conceptuelle basées sur les treillis de Galois et applications. *Revue d'intelligence*, vol. 9, n°2, P.105-137.

Gonzalo J., F. Verdejo, C. Peters , N. Calzolari (1998). *Applying EuroWordNet to Cross-Language Text Retrieval.* Kluwer Academic Publishers. Printed in the Netherlands. Computers and the Humanities 32: 185–207, 1998.

Gonzalo J., I. Chugur, and F. Verdejo (2000). Sense clustering for information retrieval: evidence from Semcor and the EWN InterLingual Index. *Proceedings of the ACL'00 Workshop on Word Senses and Multilinguality*.

Greene B.B et G.M. Rubin (1971). Automatic grammatical tagging of English. Technical report, Department of linguistics, Brown University, Providence, Rhode Island.

Grefenstette G. (1994). Explorations in Automatic Thesaurus Discovery. Dordrecht, Kluwer.

Grigoriadou M., H. Kornilakis, E. Galiotou, S. Stamou, E. papakitsos (2004). The software infrastructure for the development and validation of the greek wordnet. *Romanian journal of information*, science and technology, volume 7, numbers 1-2, 2004, 89-105.

GRUBER, Thomas R. (1993). A translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*, vol. 5, n° 2, p.199-220

Guarino N. and Welty C. (2000). Towards a methodology for ontology-based model engineering. *In Proceedings of ECOOP-2000 Workshop on Model Engineering*. Cannes, France.

Haddara M., Olivier K. (2005)**.** Etude de contextes multilingues alignés en vue de la désambiguïsation sémantique, *Actes des 4èmes Journées de la Linguistique de Corpus,* Lorient, 15-17 septembre 2005.

Hamp B. et H. Feldweg (1997). GermaNet - a Lexical-semantic Net for German. *Proceedings of ACL workshop Automatic Information Extraction and Building of Lexical Semantic Resources*. Madrid. Spain.

HATON S., (2006). *Analyse et modélisation de la polysémie verbale dans une perspective multilingue : le dictionnaire bilingue vu dans un miroir*. Thèse de doctorat soutenue le 25 novembre 2006. Université de Nancy 2.

Haton S., N.Bouzid, F. Charpillet, M-C. Haton, B. Lâasri, H. Lâasri, P.Marquis, T. Mondot et A. Napoli (1991). Le raisonnement en intelligence artificielle. *InterEditions*, 1991.

HAWKINS, P. et D. NETTLETON (2000). Large scale wsd using learning applied to senseval. *Computer and the Humanities*, 34(1–2):135–140.

Hendrix, Gary G. (1979). *Encoding knowledge in partitioned networks*. Findler (1979) pp. 51-92.

Hindle D. (1990). Noun classification from predicate-argument structures, *ACL'83*, Berkeley, 268-275.

Huang C-R., I-Ju E. Tseng, D. B.S. Tsai (2002). Translating Lexical Semantic Relations: The First Step Towards Multilingual Wordnets. Institute of Linguistics, Preparatory Office, Academia Sinica 128 Sec.2 Academy Rd., Nangkang, Taipei, 115, Taiwan, R.O.C.

Huang C.R., I. Su, J. Hong, X.B. Li (nd). Cross-lingual Conversion of Lexical Semantic Relations: Building Parallel Wordnets. Institute of Linguistics - Institute of Information Science Academia Sinica, No.128 Academic Sinica Road, SEC.2 Nankang, Taipei 115, Taiwan.

Ion R., D. Tufis. (2004). Multilingual Word Sense Disambiguation Using Aligned Wordnets. *Romanian journal of information science and technology*. Volume 7, Numbers 1-2, 2004, 183-200. Research Institute for Artificial Intelligence, Romanian Academy.

Jacquemin B. et S. Ploux (2008). Du corpus au dictionnaire. Réalisation automatique d'un outil de gestion de l'information multilingue. *Cahiers de linguistique*, 33 (1), pp. 63-84.

Jacquet G., J.-L Manguin; F.Venant, B. Victorri (2010). Construction dynamique du sens: application à la prédication verbale. *Actes des Rencontres interdisciplinaires sur les systèmes complexes naturels et artificiels*, Rochebrune-2010.

Jacquin C., E. Desmontils., et L. Monceaux (2007). French eurowordnet lexical database improvements. *Proceedings of CICLing'07 (LNCS 4394),* Mexico City, Mexico.

Ji H., S. Ploux. (2003). Automatic contexonym organizing model (ACOM). *In Proceedings of the 25th Annual Conference of the Cognitive Science Society*, 622-627.

Ji H., S. Ploux, E. Wehrli (2003). Lexical knowledge representation with contexonyms. *In Proceedings of the 9th MT summit*, 194-201.

Karlsson F. A. Voutilainen, J. Heikki¨a, et A. Anttila (1995). *Constraint Grammar: a Language-Independent System for Parsing Unrestricted Text*. volume 4 of *Natural Language Processing*. Mouton de Gruyter, Berlin and New York.

Kayser, D. (1987). *Une sémantique qui n'a pas de sens. Langages*. Sémantique et intelligence artificielle – F. Rastier (ed)(87). P. 33–45.

Kayser D. (1988). *What kind of thing is a concept?*. Computational Intelligence n°4(2). P. 158-165.

Kilgarriff A. (1997). Sample the lexion. *Technical Report ITRI-97-01*, ITRI, University of Brighton.

Klavans J. et E. Tzoukermann (1995) "*Combining Corpus and Machinereadable Dictionary Data for Building Bilingual Lexicons*". Machine Translation, 10(3).

Kleiber G. (2000). Problèmes de sémantique, la polysémie en question. Villeneuve d'Ascq, Septentrion.

Koeva S., T. Tinchev, S. Mihov (2004). Bulgarian Wordnet - structure and validation. *Romanian Journal on Information Science and Technology*. Vol. 7, 61-79.

Kraif O. (2001). *Exploitation des cognats dans les systèmes d'alignement bi-textuel: architecture et évaluation*. TAL 42: 3, ATALA, Paris, pp.833-867

Kraif O. *(1999)*. Identification des cognats et alignement bi-textuel: une étude empirique. *Conférence TALN 1999*, Cargèse, 12-17 juillet 1999. LILLA, Université de Nice Sophia Antipolis, 98 Bd. E. Herriot BP 369 06007 Nice Cedex.

Kraif O. (2003)**.** Repérage de traduction et commutation interlingue : Intérêt et méthodes. *TALN 2003,* Batz-sur-Mer, 11-14 juin 2003 .LIDILEM. Laboratoire de Linguistique et de Didactique des Langues Etrangères et Maternelles. Université Stendhal - Grenoble 3, 38400 Saint-Martin d'Hères. France.

Le Grand B., M. Soto. *(nd).* Visualisation exploratoire, généricité, exhaustivité et facteur d'échelle. Laboratoire d'Informatique de Paris 68, rue du Capitaine Scott 75015 Paris.

Levrat B. and G. SABAH (1990). "*Sorte de", une façon de rendre compte de la relation d'hyponymie/ hyperonymie dans les réseaux sémantiques*. Dans *Langages,* n° 98, p. 87-102.

Ligozat A.L., B. Grau, I. Robba, A. Vilnat. (nd). Question-Réponse multilingue : évaluation et amélioration des stratégies de changement de langue. LIMSI-CNRS, BP 133, 91403 ORSAY Cedex.

Lin D., (1998). An Information-Theoretic Definition of Similarity. *Proceedings of the Fifteenth International Conference on Machine Learning*, p.296-304.

Luce, R. D. et A. Perry (1949). *A method of matrix analysis of group structure*. Psychometrika 14, 94-116.

Madani N. (1994). Etude de l'héritage des propriétés dans les réseaux sémantiques. Notion de réseau d'héritage légal. Laboratoire d'informatique. Université Paris XIII – Institut Galilée. Avenue J.B Clément 93430 Villtanteuse.

Mallak I. (2011). *De nouveaux facteurs pour l'exploitation de la sémantique d'un texte en Recherche d'Information*. Thèse de doctorat à l'Université Toulouse III - Paul Sabatier.11 juillet 2011.

Mangeot-Lerebours M. (nd). *Frameworks, Implementation and Open Problems for the Collaborative Building of a Multilingual Lexical Database*.Software Research Division, NII Hitotsubashi, 2-1-2-1913 Chiyoda-ku. 101-8430 Tokyo, Japan. Gilles Sérasset GETA-CLIPS-IMAG.185, rue de la bibliothèque, BP 53. F-38041 Grenoble cedex 9, France.

MANGUIN J.L. (2004). Transitivité partielle de la synonymie : application aux dictionnaires de synonymes. CRISCO – CNRS et Université de Caen.

Marrafa P., S. Mendes (2007). Using WordNet.PT for translation: disambiguation and lexical selection decisions. *International Journal of Translation*, Vol. 19, ISSN 0940-9819, Bahri Publications.

Mourad M., G. ANTONIADIS, M. ZRIGUI (2008). Nouvelles ressources et nouvelles pratiques pédagogiques avec les outils TAL. *TICEMED 08*, Journal Information Sciences for Decision Making (Journal ISDM), ISDM32, N°571, Avril 2008.

Mellal N. (2007). *Réalisation de l'interopérabilité sémantique des systèmes, basée sur les ontologies et les flux d'information*. Thèse de doctorat à Polytech'Savoie.19 Déc. 2007.

Ménard E. (2004). La désambiguïsation en recherche d'information multilingue (RIML). (Disponible à l'adresse : http://www.esi.umontreal.ca/~p0336101/RIML/desambi.html). Consulté en mai 2007.

Miller G., R. Beckwith, C. Fellbaum, D. Gross, K.K. Miller (1990). Five papers on WordNet. *Special Issue in International Journal of Lexicography*, vol.3, no4.

Miller G., R. Beckwith, C. Fellbaum, D. Gross & K. Miller (1993). *Introduction to WordNet: An on-line lexical database*. Disponible par ftp to clarity. Princeton. Edu.

Minh T., L. Romary, Luong vu X. (2003). Une étude de cas pour l'étiquetage morpho-syntaxique de textes vietnamiens. *TALN 2003*, Batz-sur-Mer, 11-14 juin 2003

Mounin G. (1963). *Les Problèmes théoriques de la traduction.* Paris, Gallimard, 296 p.

Mouton C. et G. Chalendar (2010). JAWS: Just AnotherWordNet Subset. *TALN 2010*, Montréal, 19–23 juillet 2010.

Nakache J.P. et J. Confais (2004). *Approche Pragmatique de la Classification*. TECHNIP, Paris.

Navarro B., M. Palomar, P. Martinez-Barco (2003). Automatic extraction of syntactic semantic patterns for multilingual resources . Departamento de lenguajes y Sistemas Informàticos, Universidad de Alicante, Spain.

Navarro B., M. Palomar, P. Martinez-Barco (2003). A General Proposal of Multilingual Information Access based on syntactic-semantic patterns. Departamento de Lenguajes y Sistemas Informáticos. Universidad de Alicante. [Diaporama électronique powerpoint].

Niles I., A pease (2001). *Linking Lexicons and Ontologies: Mapping WordNet to the Suggested Upper Merged Ontology.* 1800 Embarcadero Rd. Palo Alto CA 94303.

Niles I. (nd). *Mapping WordNet to the SUMO Ontology*. Teknowledge Corporation. 1810 Embarcadero Road, Palo Alto, CA 94303.

Ordan N., S. Wintner (2005). Representing natural gender in multilingual databases. Department of Computer Science, University of Haifa.

PALA K., P. SMR (2004). Building Czech Wordnet. *Romanian Journal of Information Science and Technology*. Volume 7, Numbers 1-2, 2004, 79-88. Faculty of Informatics, Masaryk University Brno, Botanick¶a 68a, 602 00 Brno, Czech Republic.

Paroubek P., M. Rajman (2000). *Etiquetage morpho-syntaxique*. *Ingénierie des langues* (p. 131-150) Paris, HERMES Science Europe.

Patry A. (2005). Survol du standard XCES. *RALI*. Département d'informatique et de recherche opérationnelle Université de Montréal.

Paul C., L.Viennot (1997). Quelques algorithmes linéaires de reconnaissances autour de lex-bfs ,*CAAP'97*, Springer, Berlin.

Pease A. (nd). *Global WordNet and the Suggested Upper Merged Ontology (SUMO): Ontologies, lexicons and their Relationships*. Articulate software. Spease at articulatesoftware dot com. Présenté en PANL1On. [Diaporama électronique PowerPoint].

Pedersen, B.S. et S. Nimb. (2008). Event Hierarchies in DanNet. A. Tanács, D. Csendes, V. Vincze, C. Fellbaum, et P. Vossen (eds). *Proceedings of Global WordNet Conference, Szeged*, Hungary, pp. 339--349.

Ploux S., Victorri B. (1998) . *Construction d'espaces sémantiques à l'aide de dictionnaires de synonymes*. TAL, Vol 39/1, pp. 161-182.

Ploux S., Hyungsuk A. (2003). *Model for matching semantic maps between languages (French / English, English / French)*. *Computational Linguistics* 29 (2): 155-178.

Ploux S. (1997). *Modélisation et traitement informatique de la synonymie*. Linguisticae Investigationes, vol. 21, no. 1, pp. 1–28.

Poibeau T., D. Dutoit, S. Bizouard (2002). Évaluer l'acquisition semi-automatique de classes sémantiques. *TALN 2002*, Nancy, 24-27 juin 2002. Thales et LIPN Domaine de Corbeville, 91404 Orsay, France.

Pouliquen B., R. Steinberger, C. Ignat, T. Oellinger (2006). *Building and Displaying Name Relations using Automatic Unsupervised Analysis of Newspaper Articles*.European Commission, Joint Research Centre 21020 Ispra (VA), Italy.

Quillian, M. Ross, (1966, 1968). Semantic memory*, dans Minsky, M. (réd.). Semantic information processing*. MIT Press, Cambridge, Massachusetts, pp. 227–270.

Quillian, M.R. (1985). *Word Concepts: a Theory and Simulation of Some Basic Semantic Capabilities.* in R.J. Brachman and H.J. Levesque, (eds), Readings in Knowledge Representation. Los Altos, CA: Morgan Kaufman.

Ranieri M., E. Pianta, L. Bentivogli (2002). Browsing Multilingual Information with the MultiSemCor Web Interface. *ITC-irst Via Sommarive* 18, 38050 Povo (Trento) - Italy

Rapaport, William J. (2000). How to Pass a Turing Test: Syntactic Semantics, Natural-Language Understanding, and First-Person Cognition. *Journal of Logic, Language, and Information*, 9(4): 467–490; reprinted in James H. Moor (ed.), The Turing Test: The Elusive Standard of Artificial Intelligence (Dordrecht: Kluwer, 2003): 161–184.

Rastier, F. (1991). *Sémantique et recherches cognitives*. Paris, PUF.

Resnik P. (1997). Selectional preference and sense disambiguation. *Proceedings of ACL SIGLEX Workshop on Tagging Text with Lexical Semantics*, ACL, Washington, D.C., 1997, p. 52–57.

Richardson S. (1997). *Determining Similarity and Inferring Relations in a lexical Knowledge Base*. PH.D. Thesis, the City University of New York, New York.

Rigau G., Atserias J. and Agirre E. (1997). Combining Unsupervised Lexical Knowledge Methods for Word Sens Disambiguisation. *Proceedings of the 35th Annual Meeting of the ACL (ACL'97)*. Madrid, Spain.

Saadani L., S. Bertrand-Gastaldy (2000). Cartes Conceptuelles et Thésaurus : Essai de Comparaison Entre Deux Modèles de Représentation Issus de Différentes Traditions Disciplinaires. *ACSI*. Les dimensions d'une science de l'information globale. Association canadienne des sciences de l'information. Travaux du 28e congrès annuel. Université de Montréal

SAGOT B. and FISER D. (2008). Building a Free FrenchWordNet fromMultilingual Resources. *Proceeding of Ontolex*, Marrakech, Maroc.

Schmid H. (1995). Improvements in Part-of-Speech Tagging with an Application to German. *Proceedings of the ACL SIGDAT-Workshop*. Dublin, Ireland.

Shapiro, C., S. Kandefer, M. (2005). *A SNePS approach to the wumpus world agent or Cassie meets the wumpus*. In Morgenstern, L. and Pagnucco, M., editors, IJCAI-05 Workshop on Nonmonotonic Reasoning, Action, and Change (NRAC'05): Working Notes, pages 96–103. IJCAII, Edinburgh, Scotland.

Shapiro C., S. Wiley (1992). *Semantic Networks, John F. Sowa.* Encyclopedia of Artificial Intelligence, second edition, 1992.

Shapiro, Stuart C. (2000). *An Introduction to SNePS 3*", in Bernhard Ganter & GuyW. Mineau (eds.), Conceptual Structures: Logical, Linguistic, and Computational Issues, Lecture Notes in Artificial Intelligence 1867 (Berlin: Springer-Verlag): 510–524.

Sidhom S.(nd).*Réseaux sémantiques*.MCF. Université Nancy 2.Équipe de recherche SITE – LORIA

SCHANK RC (1973). *Identification of conceptualizations underlying natural language*. [Schank, Colby eds], Freeman, San Francisco

Shapiro, Stuart C.; Rapaport, William J.; Cho, Sung-Hye; Choi, Joongmin; Feit, Elissa; Haller, Susan; Kankiewicz, Jason; & Kumar, Deepak (1996). A Dictionary of SNePS Case Frames. [http://www.cse.buffalo.edu/sneps/Manuals/dictionary.ps].

Sofia S., O. Kemal, P.Karel, C. Dimitris, C. Dan, T. Dan, K. Svetla, T. George, D. Dominique, G. Maria (nd). BALKANET: A Multilingual Semantic Network for Balkan Languages.

Soutet O. dir (2005). *La polysémie*. PU Paris-Sorbonne.

Sowa J.F. (1991). *Principles of Semantic Networks*. Explorations in the Représentation of Knowledge (San Mateo, CA: Morgan Kaufmann Publishers).

Sowa, John F. (2002). *Semantic Networks*. [http://www.jfsowa.com/pubs/semnet.htm].

Sidhom S. (nd). Représentation des connaissances : approches de représentation des connaissances: classification, thésaurus, ontologie. (Disponible à l'adresse : *www.loria.fr/~ssidhom/...L2.../cours_2_ue404a(L2-doc-0607).ppt).* Consulté en avril 2012.

Tanguy L., S. Armstrong, D.Walker (1999). Isotopies sémantiques pour la vérification de traduction. *Conférence TALN 1999, Cargése*, 12-17 juillet 1999. ISSCO - Université de Genève, 54 Route des Acacias - 1227 Genève – Suisse.

Touretzky D. S., (1986). *The Mathematics of Inheritance Systems.* Pitman, London.

Tufiş D., R. Ion, E. Barbu, V. Barbu (2003). Cross-Lingual Validation of Multilingual Wordnets. *Institute for Artificial Intelligence*, 13, Calea 13 Septembrie, 050711,Bucharest 5, Romania.

TURENNE N. (2000). *Apprentissage statistique pour l'extraction de concepts à partir de textes. Application au filtrage d'informations textuelles.* Thèse de doctorat en sciences, spécialité informatique, Université Louis Pasteur, Strasbourg.

Van der Plas L., J. Tiedemann, and J-L Manguin (2008)**.** Extraction de Synonymes à Partir d'un Corpus Multilingue Aligné. *Proceedings of the 5èmes Journées de Linguistique de Corpus.*

Vasilescu F.(nd). *La désambiguïsation de corpus monolingues par des approches de type Lesk*. DIRO Université de Montréal.

Venant F.(2004). Géométriser le sens. *RECITAL 2004*, Fès, 19- 22 avril 2004.

Venant F. (2007). Utiliser des classes de sélection distributionnelle pour désambiguïser les adjectifs. *Actes de la 14ème conférence de Traitement automatique des langues naturelles - TALN'07*, Toulouse : France (2007).

Veronis J., Ide N. (1990). Word Sense Disambiguation with very large neural networks extracted from machine-readable dictionaries, *COLING'90*, Helsinki, 389-394

Villavicencio A., T. Baldwin, B. Waldron (2004). A Multilingual Database of Idioms**.** *Proceedings of the Fourth International Conference on Language Resources and Evaluation* (LREC 2004), Lisbon, Portugal, pp. 1127-30.University of Cambridge Computer Laboratory, William Gates Building, JJ Thomson Avenue, Cambridge, CB3 0FD, UK

Vossen P. (nd). *Building wordnets*. Irion Technologies. [Diaporama électronique PowerPoint].

Vossen, P. (1996). *Right or Wrong. Combining lexical resources in the EuroWordNet project*. M. Gellerstam, J. Jarborg, S. Malmgren, K. Noren, L. Rogstrom, C.R. Papmehl, Proceedings of Euralex-96, Goetheborg, 1996, 715-728

Vossen P. (1998). *EuroWordNet: A Multilingual Database with Lexical SemanticNetworks.* Computational Linguistics Volume 25, Number 4. (University of Amsterdam)[Reprinted from *Computers and the Humanities,* 32(2-3), 1998], Dordrecht: Kluwer Academic, Publishers, 1998, 179 pp; hardbound.

Voutilainen A., J. Heikkilä (1994). *An English constraint grammar (ENGCG): a surface syntactic parser of English*. Fries, U., Tottie, G., Scheider, P. (Eds.). Creating and using English corpora. Amsterdam : Rodopi.

Warnesson I. (1992). *Lexicographie et informatique, vers une nouvelle generation de dictionnaires*. Publications scientifiques et techniques d'IBM France, décembre 1992, Paris, 107-157.

Weaver W. (1949/1955). *Translation*. Dans Machine Translation of Languages, Locke William N. et A. Donald BOOthe, dir., MIT Press, Cambridge, MA, 1949/1955, p. 15–23.

Woods, W. (1991). *Understanding subsumption and taxonomy: A framework for progress*. In Sowa, J. F., editor 1991, *Principles of Semantic Networks.* -Morgan Kaufmann. 45-94.

Woods, William A. (1975). *What's in a Link: Foundations for Semantic Networks*. In Daniel G. Bobrow & Allan M. Collins (eds.), *Representation and Understanding* (NewYork: Academic Press): 35–82; réimprimé à Brachman & Levesque 1985: 217–241.

Yarowsky D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pages 189–196.

Zouaghi A., M. zrigui, M. Ben Ahmad (2005). Un étiqueteur sémantique des énoncés en langue arabe. *Récital 2005*, Dourdan, 6-10 juin 2005. Laboratoire RIADI- Université du Centre, Faculté des sciences de Monastir-Tunisie.