Approche hybride - lexicale et thématique - pour la modélisation, la détection et l'exploitation des fonctions lexicales en vue de l'analyse sémantique de texte

Didier Schwab

these : *version du mardi 21 mars 2006 à 14 h 25*

Numéro d'identification :

Académie de Montpellier

Université Montpellier II
— **Sciences et Techniques du Languedoc** —

Thèse

présentée à l'Université des Sciences et Techniques du Languedoc
pour obtenir le diplôme de DOCTORAT

Spécialité	: **Informatique**
Formation Doctorale	: **Informatique**
École Doctorale	: **Information, Structures, Systèmes**

Approche hybride - lexicale et thématique - pour la modélisation, la détection et l'exploitation des fonctions lexicales en vue de l'analyse sémantique de texte

par

Didier Schwab

Soutenue le 7 décembre 2005 devant le Jury composé de :

Christian Boitet, Professeur, Université Joseph Fourier (Grenoble 1), GETA, CLIPS ..... Rapporteur
Gérard Sabah, Directeur de Recherche CNRS, LIMSI, Orsay ..... Rapporteur
Roland Ducournau, Professeur, Université Montpellier II ..... Président du jury
Christophe Lecerf, Professeur, École des Mines d'Alès ..... Examinateur
Violaine Prince, Professeur, Université Montpellier II ..... Directrice de thèse
Mathieu Lafourcade, Maître de conférence, Université Montpellier II ..... Co-directeur de thèse

these : *version du mardi 21 mars 2006 à 14 h 25*

Numéro d'identification :

Académie de Montpellier

Université Montpellier II
— **Sciences et Techniques du Languedoc** —

Thèse

présentée à l'Université des Sciences et Techniques du Languedoc
pour obtenir le diplôme de DOCTORAT

Spécialité	: **Informatique**
Formation Doctorale	: **Informatique**
École Doctorale	: **Information, Structures, Systèmes**

Approche hybride - lexicale et thématique - pour la modélisation, la détection et l'exploitation des fonctions lexicales en vue de l'analyse sémantique de texte

par

Didier Schwab

Soutenue le 7 décembre 2005 devant le Jury composé de :

Christian Boitet, Professeur, Université Joseph Fourier (Grenoble 1), GETA, CLIPS ..... Rapporteur
Gérard Sabah, Directeur de Recherche CNRS, LIMSI, Orsay ..... Rapporteur
Roland Ducournau, Professeur, Université Montpellier II ..... Président du jury
Christophe Lecerf, Professeur, École des Mines d'Alès ..... Examinateur
Violaine Prince, Professeur, Université Montpellier II ..... Directrice de thèse
Mathieu Lafourcade, Maître de conférence, Université Montpellier II ..... Co-directeur de thèse

Cette thèse est dédiée à l'ensemble des instituteurs et professeurs que j'ai eu au cours de ma scolarité ainsi qu'aux hommes et aux femmes qui ont permis l'école gratuite et obligatoire et qui m'ont, par là, autorisé à faire des études.

Remerciements

Une thèse est une aventure tant professionnelle que personnelle. Elle représente plusieurs années de travail. Défricher un domaine, essayer d'en assimiler les problématiques importantes et enfin comprendre ce qu'on pourrait essayer de lui apporter est une entreprise de presque chaque instant.

Pour m'avoir aidé dans cette exploration par leurs conseils et leurs encouragements, je voudrais remercier ici, mes directeurs *Violaine Prince* et *Mathieu Lafourcade*.

Pour avoir toujours critiqué mes travaux avec justesse, pour m'avoir encouragé et conseillé à plusieurs reprises et enfin pour avoir accepté d'être rapporteur de cette thèse, je tiens particulièrement à remercier *Christian Boitet*.

Toute ma gratitude va également à *Gérard Sabah* qui à la fois par ses écrits et ses interrogation a su éveiller chez moi certains questionnements sur la représentation du sens. Je le remercie de m'avoir fait l'honneur de rapporter mes travaux.

L'importance de la notion de double-boucle dans l'apprentissage a été à la fois un résultat et un objectif central de cette thèse. Son inventeur, *Christophe Lecerf*, a accepté d'être membre de mon jury et je l'en suis reconnaissant.

Lorsque je suis arrivé en licence d'informatique, *Roland Ducournau* était, avec d'autres, chargé du stage d'introduction sur UNIX. Par la suite, je l'ai cotoyé lorsqu'il était directeur du département informatique et que j'étais représentant des doctorants au département. Je le remercie d'avoir fermé la boucle en acceptant de présider mon jury et pour sa relecture attentive du mémoire.

Il va sans dire que cette thèse s'inscrit au sein d'une équipe, l'équipe traitement algorithmique du langage, dont je voudrais remercier ici les membres et particulièrement *Jacques Chauché* pour les fondements des vecteurs d'idées et pour SYGMART, *Alain Joubert* pour sa relecture soigneuse de certaines parties de la thèse et *Mehdi Yousfi-Monod* pour son enthousiasme, sa bonne humeur et nos discussions sur nos thèses respectives.

Je veux remercier l'ensemble des membres du LIRMM qui directement ou indirectement par leurs encouragements, ont participé à ce travail par l'intermédiaire des deux directeurs qui se sont succédés ici pendant sa réalisation *Michel Habib* et *Michel Robert*.

Je remercie les membres du service administratif dont le travail a largement aidé la réalisation de cette thèse et en particulier *Nicole Olivet* pour sa sympathie et sa gentilesse quotidienne.

Les expériences menées doivent beaucoup à l'appui offert par le Service Informatique et Technique en particulier *Jean-Luc Oms* et *Michel Jacquot* qui m'ont aidé en accueillant servlets, agents et données sur les serveurs du laboratoire.

Je tiens particulièrement à remercier ici quelques-uns des compagnons qui ont traversé, traversent ou traverseront les mêmes turpidudes que moi : *Xavier Baril*, *Nicolas Vidot*, *Pierre-Alain Laur*, *Lylia Abrouk*, *Abdelkader Gouaich*, *Adorjan Kiss*, *Simon Jaillet*, *Jérôme Chapelle*, *Alexis Criscuolo*, *Denis Bertrand*, *Fabien Jourdan*, *Laurent Brehelin*, *Sèverine Bérard*, *Mehdi Yousfi-Monod*, *John Tranier*, *Clément Jonquet*, *Cécile Bonnard*, *Jean Privat*, *Luc Frabresse*, *Christophe Crespelle*, *Céline Fiot*, *Leila Aouati*, *Fabien Michel*, *Fabien Jalabert*, *Mathias Paulin*, *François Boutin*.

Je voudrais remercier *Héloïse Reynaud* et *Séverine Lacroix* pour les instants partagés autour de thé, de narguilés, haltères ou vin, soirées qui m'ont permis de m'évader pendant les moments difficiles. Elles savent combien elles comptent pour moi et combien je serai heureux de les retrouver où qu'elles se trouvent dans le monde.

Merci à *Philippe Boulet* pour son amitié depuis vingt ans.

Merci à *Céline Chalbos* de m'avoir soutenu pour achever la version finale de cette thèse.

Que soient aussi remerciés pour simplement avoir été là *Charlotte Peis*, *Marion Groschène*, *Vincent Nabat*, *Gaël Pages*, *Fabien Lydoire*, *Élodie Zamora*, *Agnès et Xavier Pera*, *Céline Durand*, *Patricia Durand*, *Shiva*, *Alexandra Estève*, *Jean-michel et Eugénie Delorme*, *Christophe et Magali Palermo*, *Anny Castonguay*, mes cousins *Régis et Nathalie Lussan*.

Je voudrais aussi avoir une pensée pour tous les étudiants à qui j'ai enseigné ou que j'ai encadré pendant quatre années à l'université Montpellier II. Je leur souhaite de réussir en particulier ceux qui se dirigent vers la recherche scientifique. Je souhaite aussi remercier les personnes avec lesquelles j'ai enseigné *Éhoud Ahronovitz*, *Yolande Aronovithz* et *Mathieu Lafourcade*.

Je remercie mon oncle et ma tante *Maryse* et *Jean-Jacques Lussan* pour nous avoir aidé le jour de la soutenance, en particulier mon oncle pour avoir évité que mes camarades thésards « *aient trop longtemps leur verre vide* ».

Je voudrais finir en remerciant mes parents *Gilberte et Christian Schwab* pour l'aide qu'ils m'apportent depuis tant d'années.

Sommaire

Table des figures

Notations

CONCEPTS : notation des concepts. Exemple : *VIE*, *MORT*.

‹***item lexical***› : notation des items lexicaux. Exemple : ‹*vie*›, ‹*mort*›, ‹*vivre*›.

‹‹**mot**›› : notation des mots (formes fléchies). Exemple : ‹‹vies››, ‹‹vie››, ‹‹vivent››.

item*/annotation** : notation d'une acception particulière d'un item. À lire « *L'item lexical* ‹item› *dans son acception de* ***annotation. ». Par exemple, *souris*/**mammifère** « ‹souris› *dans son acception de* ***mammifère.*** » .

$V(x)$: Vecteur de x (x peut-être un *CONCEPT*, un ‹*item lexical*›, une acception, un ‹‹mot››, un « *segment textuel* », ...)

‹$\boldsymbol{x}$› ||| ‹$\boldsymbol{y}$› : les objets x et y sont antonymes.

‹$\boldsymbol{x}$› $|||_c$ ‹$\boldsymbol{y}$› : les objets x et y sont antonymes complémentaires.

‹$\boldsymbol{x}$› $|||_s$ ‹$\boldsymbol{y}$› : les objets x et y sont antonymes scalaires.

‹$\boldsymbol{x}$› $|||_d$ ‹$\boldsymbol{y}$› : les objets x et y sont antonymes duals.

V_i : i-ème composante du vecteur V.

ϑ : ensemble des vecteurs d'idées (conceptuels et/ou sémantiques suivant le cas).

ω : ensemble des ITEMS LEXICAUX.

κ : ensemble des ACCEPTIONS.

ι : ensemble des LEXIES.

σ : ensemble des objets lexicaux (ITEMS LEXICAUX, ACCEPTIONS, LEXIES).

$\mathcal{F}$: ensemble des Fonctions Lexicales d'Evaluation.

m : ensemble des morphologies.

$m(x)$: morphologie de l'objet lexical x.

MORPHOLOGIE : notation de la morphologie. Exemples : NOM, VERBE, PLUR.

freq(x) : Fréquence de l'objet lexical x.

Introduction

ÉCRITE ou parlée, la langue est ressentie, à juste titre, comme le moyen de communication le plus commun entre les personnes. Depuis son origine, aux alentours de deux millions d'années, elle a permis les échanges et surtout le maintien des relations sociales des individus avec leur entourage (tribu, famille, ...)[1]. Sans elle, difficile d'expliquer l'évolution rapide de l'être humain ces dernières centaines de milliers d'années : amélioration des techniques de construction de huttes, des techniques de chasse et surtout amélioration de la survie en raccourcissant le temps pour avertir d'un danger immédiat (« *Attention au loup !* »).

Malheureusement, du fait de leur diversité et de leur nombre, les langues agissent aussi comme une barrière entre les peuples. Elles ont longtemps été différentes d'une tribu à une autre, puis d'une région à une autre. En France par exemple, il faut attendre le XIX^{e} siècle et l'école obligatoire pour voir s'amorcer une véritable unification linguistique du pays. Même si les Hommes n'ont pas besoin de ça pour s'entretuer, les richesses et l'expansion étant souvent un argument suffisant, les différences linguistiques entraînent fréquemment l'incompréhension mutuelle qui est le début du rejet de l'autre. Les langues qui sont pourtant la richesse culturelle d'une société peuvent de la sorte entraîner la xénophobie et l'hostilité des peuples.

La connaissance de langues étrangères est ainsi un moyen fort de compréhension des autres, particulièrement dans le monde globalisé où nous vivons actuellement. Cependant, on considère qu'il existe aujourd'hui plus de 6500 langues, les États-Unis d'Amérique en répertorient 169 importantes sur le plan politique et l'on estime à une cinquantaine les langues économiquement fondamentales[2]. Dans ces conditions, il est bien difficile pour un individu d'échanger alors que la plupart des personnes ne sont capables de comprendre, au mieux, que deux ou trois langues. Il est ainsi bien naturel que l'Homme ait essayé de confier cette tâche à des machines, des automates dans un premier temps puis, à leur naissance, aux ordinateurs[3]. Depuis une soixantaine d'années, les applications se sont diversifiées, de nouveaux paradigmes sont apparus et les équipes de recherches sur ces thématiques se sont multipliées.

[1] Le lecteur trouvera un intéressant dossier sur les origines du langage à l'adresse http://www.hominides.com/html/dossiers/langage.html

[2] http://www.eurologos.be/htm/Pages/page24fr.asp

[3] Un aperçu de l'Histoire des "machines parlantes" pourra être trouvé dans http://www.up.univ-mrs.fr/~veronis/cours/INFZ18/veronis-INFZ18.pdf.

L'équipe de Traitement Algorithmique du Langage (TAL) du Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier (LIRMM) se consacre en particulier au langage écrit. Elle vise à améliorer par des techniques et des modèles innovants des applications hétérogènes qui vont du résumé automatique à la catégorisation de textes, en passant par la recherche d'informations et la traduction automatique.

Toutes ces applications nécessitent en amont une analyse sémantique des textes. Il s'agit en particulier de trouver quel est le sens des différents mots utilisés et le sens global du texte. En traduction automatique, il faudra ainsi trouver l'équivalent des termes les plus appropriés dans le contexte ; en résumé automatique, on pourra choisir de privilégier une partie du texte qui représente mieux les idées principales du discours général ; en catégorisation, choisir de regrouper les textes selon leur sens ; en recherche documentaire, choisir d'extraire d'une base de données les documents textuels les plus proches de la requête posée.

Pour trouver le sens d'un mot dans un énoncé, il s'agit tout d'abord de savoir quels sont les différents sens qu'il peut prendre et d'avoir un modèle calculatoire sur lequel repose cette caractéristique. Le modèle utilisé au cours de cette thèse est celui des vecteurs d'idées et plus précisément le modèle des vecteurs conceptuels. Il est basé sur la projection de la notion linguistique de champ sémantique dans le modèle mathématique d'espace vectoriel. À partir d'un ensemble de notions élémentaires, les concepts, représentés sous forme de vecteurs (dits *vecteurs génératifs*), on peut construire de nouveaux vecteurs d'idées et les associer à tout segment textuel (mots, phrases, textes, ...). Il est ainsi possible d'effectuer des manipulations formellement bien fondées auxquelles nous pouvons attacher des interprétations linguistiques raisonnables. L'hypothèse principale sur laquelle repose ce modèle est que les vecteurs génératifs constituent un espace générateur pour l'ensemble des mots de la langue.

Dans un premier temps, on affecte manuellement des vecteurs conceptuels aux sens des mots les plus fréquents puis, dans un deuxième temps, on en construit de nouveaux à partir de dictionnaires à usage humain comme le Larousse ou le Robert. Cet apprentissage est conçu pour réviser de manière permanente les vecteurs afin de permettre l'affinement des données et surtout le croisement de diverses sources. Il s'agit ainsi de réaliser l'analyse sémantique de la définition d'un sens du dictionnaire puis de considérer que ce sens est défini par le vecteur ainsi calculé. L'analyse sémantique est, de ce fait, au centre de nos problématiques. L'amélioration qualitative du processus d'analyse entraîne celle des vecteurs. En retour, cette meilleure pertinence a un effet positif sur l'analyse. Il s'agit ainsi d'utiliser la notion centrale d'analyse sémantique pour entrer dans un cercle vertueux d'amélioration de la représentation du sens.

Parmi les différentes voies à explorer pour améliorer l'analyse sémantique, l'une des plus intéressantes et des plus centrales semble être la découverte puis l'exploitation des relations lexicales entre les mots du texte. Ainsi, la détection de la relation d'antonymie dans la structure négative de la définition du terme ‹*inexistant*› : « *Qui n'existe pas.* » peut permettre par la construction d'un vecteur opposé à celui d'‹*exister*› de donner à ‹*inexistant*›, un vecteur plus pertinent que si la relation n'avait pas été détectée puis exploitée. Il en est de même pour la détection des références (dans la phrase « *L'homme creusait avec la pelle quand l'outil se cassa.* » ‹*pelle*› et ‹*outil*› font référence à la même entité) ou pour celle des collocations, ces combinaisons de termes qui prévalent sur d'autres sans qu'il ne semble y avoir de motif logique (on parle de « *dormir profondément* » et non pas de *« *dormir intensément* »).

L'ensemble de ces relations est modélisable sous la forme de fonctions lexicales. Énoncées essentiellement dans un cadre de production par Igor Mel'čuk, nous cherchons à les adapter, ici, à un cadre d'analyse.

Ce mémoire est composé de trois parties :

Dans la première, nous précisons le contexte de notre étude, l'état de l'art et les premières expériences menées sur la modélisation des fonctions lexicales.

Dans le premier chapitre, nous montrons en quoi la question du sens se pose dans de nombreuses applications du TALN. Nous exposons ainsi les différents niveaux de traitements que comportent de telles applications et les difficultés posées dans chacun d'eux. Nous nous concentrons en particulier sur les niveaux sémantique et pragmatique où est traitée plus spécifiquement cette question. Nous présentons ensuite plusieurs approches de la modélisation du sens qui ont été abordées en informatique : les approches issues de la linguistique distributionnelle comme les vecteurs saltoniens et les techniques de type LSA ainsi que les approches symboliques connexionnistes comme les réseaux sémantiques ou les bases d'acceptions. Nous abordons alors les relations lexicales et les fonctions lexicales de productions. Enfin, nous présentons la linguistique componentielle, les modèles informatiques parents et en particulier les travaux effectués par Jacques Chauché au début des années 1990. Ces travaux sont à la base du modèle des vecteurs d'idées, modèle central de notre thèse.

Dans le deuxième chapitre, nous décrivons le modèle des vecteurs d'idées. Contrairement aux représentations vectorielles classiques dont les dimensions correspondent à des éléments textuels, les dimensions correspondent ici à des idées. Ainsi, les segments textuels sont définis à partir d'un ensemble de notions, les concepts, censés pouvoir générer l'ensemble des idées exprimables en langue. Nous exposons le modèle général et les différentes opérations définies puis nous en présentons les deux grandes familles : les vecteurs sémantiques et les vecteurs conceptuels. Leurs principales différences sont essentiellement dues à leur mode de construction. Tandis que les premiers sont construits avant toute application, les seconds, autour desquels s'articule plus particulièrement cette thèse, sont en constant apprentissage.

Dans le troisième chapitre, qui est aussi le dernier de cette partie, nous montrons comment les fonctions lexicales peuvent nous aider à améliorer l'analyse des textes en général et des définitions en particulier. Les fonctions d'antonymie peuvent permettre de gérer certaines tournures négatives, les fonctions d'hyperonymie les cas de définitions hyperonymiques, les fonctions de synonymie le paraphrasage. Nous identifions deux types de fonctions lexicales : les *fonctions lexicales de construction* et les *fonctions lexicales d'évaluation*. Les premières permettent de construire un vecteur ce qui est utile, par exemple, dans certains cas de négation ou pour l'analyse des dictionnaires de synonymes tandis que les secondes évaluent la pertinence entre deux items d'une fonction lexicale. Nous introduisons les deux premières modélisations de fonctions lexicales qui ont été développées, celles qui concernent les fonctions symétriques. Nous exposerons en particulier les travaux réalisés sur la synonymie avant mon arrivée dans l'équipe ainsi que les améliorations auxquelles j'ai participé. Nous présentons ensuite les travaux sur l'antonymie réalisés au cours de mon DEA, revus et complétés durant ma thèse. De par leur modélisation, ces fonctions sont utilisables à la fois pour les vecteurs sémantiques et les vecteurs conceptuels. Ce chapitre se termine par les effets constatés sur la base et une réflexion qui porte en particulier sur les limites du modèle purement conceptuel.

La conclusion de la première partie mettra l'accent sur la difficulté à se limiter aux vecteurs d'idées pour représenter le sens des termes et les fonctions lexicales. Dans la seconde partie, nous revoyons le concept de base lexicale sémantique en introduisant dans les objets lexicaux des informations sur les relations entre les termes, à la fois pour la représentation du sens des termes et pour la modélisation des fonctions lexicales d'analyse.

Dans le quatrième chapitre, nous cherchons ainsi à introduire des informations purement lexicales dans les fonctions symétriques. Cette introduction se fait par l'utilisation de la méthode de contextualisation forte pour les deux relations à laquelle s'ajoute, dans le cas de l'antonymie, celle des oppositions déjà rencontrées. La deuxième partie de ce chapitre concerne la représentation des informations que nous donnent les fonctions lexicales dans la base lexicale sémantique. Nous explorons des méthodes pour regrouper les synonymes et les antonymes en fonction de leur sens afin de fabriquer des LEXIES. Le manque de source pour les antonymes pendant une grande partie de ces travaux a été un grand problème. Nous l'avons, dans un premier temps, partiellement réglé en construisant, de façon semi supervisée, une liste de couples d'antonymes en nous basant sur les oppositions de morphologie qui peuvent exister entre eux. Nous concluons par l'apport de ces informations dans la construction de la base lexicale sémantique.

Dans le cinquième chapitre, nous tirons le bilan de l'expérience acquise dans les précédents pour revoir en partie le modèle de Base Lexicale Sémantique, c'est-à-dire le modèle d'une base de données permettant la représentation et l'exploitation du sens des items lexicaux. Nous présentons les six hypothèses de départ qu'il nous paraît nécessaire d'adopter dans le but de construire une telle base. Ces hypothèses nous ont conduit à choisir une architecture à trois niveaux d'objets lexicaux (LEXIE, ACCEPTION, ITEM LEXICAL).Nous montrons comment ces hypothèses, les applications hétérogènes visées ainsi que des caractéristiques techniques nous ont amené à adopter une architecture multi-agent dont nous présentons les caractéristiques conceptuelles et techniques. Le système multi-agent proposé, appelé Blexisma, a pour but d'intégrer tout agent pouvant permettre de créer, d'améliorer et/ou d'exploiter une ou plusieurs Bases Lexicales Sémantiques. Nous exposons enfin les différents agents déjà implémentés ainsi qu'un exemple de leur interaction dans le cadre de l'acquisition d'informations sémantiques et de leur exploitation pour fabriquer des objets lexicaux.

Dans le sixième chapitre, nous cherchons à établir quelles relations il serait utile de modéliser dans la base lexicale sémantique (BLS) dans le cadre de l'analyse d'un énoncé. Nous présentons la liste des Fonctions Lexicales d'Analyse (FLA) que nous avons établie afin de bénéficier de connaissances linguistiques et de connaissances du monde. Nous étudions ensuite en détail la question de l'hyperonymie et de l'hyponymie en montrant le rôle fondamental qu'elles jouent dans le cadre de la construction des vecteurs conceptuels. Nous présentons les caractéristiques du grand réseau lexical induit par la structure de la BLS et nous évoquons les pistes à suivre pour le créer le plus automatiquement possible. Nous faisons enfin le bilan des FLA afin de comprendre lesquelles sont à caractère purement lexical (basées sur des relations entre objets lexicaux) et lesquelles sont à caractère à la fois lexical et thématique. Nous montrons alors comment le réseau peut aider à modéliser des fonctions lexicales de construction et des fonctions lexicales d'évaluation pour chacune des FLA.

La dernière partie de cette thèse est consacrée à l'étude de l'utilisation de toutes ces informations pour améliorer l'analyse sémantique, la collaboration entre bases ainsi qu'à l'une des applications importantes, la traduction automatique.

Dans le septième chapitre, nous revenons sur l'analyse sémantique en essayant de mettre en relief les différents problèmes d'ambiguïté qu'il est important de résoudre dans le cadre des différentes applications visées par l'équipe : le problème de l'ambiguïté lexicale, les problèmes de référence, les rattachements prépositionnels, les chemins interprétatifs possibles et enfin le dernier, qui nous intéresse plus particulièrement dans cette thèse, celui de l'instanciation des fonctions lexicales. Nous montrons comment ces dernières peuvent aider, non seulement à résoudre les autres ambiguïtés mais surtout en quoi leur instanciation peut être utilisée dans

le cadre de la traduction automatique, de la recherche d'informations ou du résumé automatique. Nous montrons que l'analyse sémantique en remontée-descente présentée précédemment est clairement insuffisante pour un tel objectif et nous présentons différents modèles d'analyse sémantique basés sur des algorithmes à fourmis. Le dernier, mis au point au cours de cette thèse, montre l'efficacité de cette méthode pour résoudre les problèmes posés dans une analyse sémantique, en particulier celui qui concerne l'instanciation des fonctions lexicales.

Dans le huitième et dernier chapitre, nous cherchons à mettre l'accent non plus sur les doubles boucles uniquement internes à la base mais sur celles qui sont en partie externes. Nous étudions, à cette fin, la mise en collaboration de plusieurs bases de vecteurs. Dans un premier temps, nous présentons la création d'une base lexicale sémantique monolingue à partir d'une base déjà existante pour une autre langue. Dans l'expérience menée, il s'est agi de construire une base pour l'anglais à partir de la base du français dont la constitution a été présentée dans les chapitres précédents et à l'aide de dictionnaires bilingues. Dans une seconde partie, nous montrons comment ces deux bases peuvent collaborer en s'échangeant des informations mutuelles et ainsi améliorer leurs représentations respectives. Nous élargissons enfin cette collaboration à des bases d'architecture éventuellement différentes en présentant comme exemple la collaboration entre les deux bases de vecteurs conceptuels de l'équipe et nous montrons que cette collaboration met en jeu une double boucle.

Sans trop anticiper sur la conclusion de notre thèse, nous pouvons d'ores et déjà pointer quelques points fondamentaux que nous relèverons. Ainsi, on estime que deux grands types d'informations sont nécessaires à la représentation du sens : les informations d'ordre thématique (vecteurs conceptuels) et les informations de type relationnel entre objets du lexique. Ces relations doivent sortir du cadre strict de l'ontologie et être les plus diverses possible : synonymie, antonymie, méronymie, hyperonymie, intensification, bonification, ... Il s'agit ainsi d'acquérir un nombre important de données qui ne peut se faire que grâce à un apprentissage. Ce faisant, il y a clairement une boucle entre le processus d'analyse et le processus d'apprentissage. Ce sont ces hypothèses sur la représentation et l'exploitation du sens que nous énonçons et validons expérimentalement ici.

Première partie

Contexte, état de l'art et premières expériences

1

La représentation du Sens en Informatique Linguistique

DANS ce chapitre, nous montrons en quoi la question du sens se pose dans de nombreuses applications du TALN. Nous exposons ainsi les différents niveaux de traitement que comportent de telles applications et les difficultés posées dans chacun d'eux. Nous nous concentrons en particulier sur les niveaux sémantiques et pragmatiques où est traitée plus spécifiquement cette question. Nous présentons ensuite plusieurs approches de la modélisation du sens qui ont été abordées en informatique : les approches issues de la linguistique distributionnelle comme les vecteurs saltoniens et les techniques de type LSA ainsi que les approches symboliques connexionnistes comme les réseaux sémantiques ou les bases d'acceptions. Nous abordons alors les relations lexicales et les fonctions lexicales de production. Enfin, nous présentons la linguistique componentielle, les modèles informatiques parents et en particulier les travaux effectués par Jacques Chauché au début des années 1990. Ces travaux sont à la base du modèle des vecteurs d'idées, modèle central de notre thèse.

Sommaire

À la fin de la deuxième guerre mondiale, deux blocs se font face : d'un côté, les États-Unis d'Amérique et leur modèle libéral, de l'autre le bloc soviétique mené par l'URSS. La guerre froide s'installe alors pour presque 50 ans. Cette même période voit l'avènement des premiers ordinateurs et de leur puissance de calcul jusqu'alors insoupçonnable. Les militaires américains ne tardent pas à voir dans ces machines un outil capable de traduire les milliers de pages en russe récupérées tous les jours par les services de renseignements. Des millions[4] de dollars sont alors investis dans de vastes programmes afin d'aboutir rapidement à la mise au point de traducteurs automatiques. L'euphorie des premières années retombe vite face à l'une des principales caractéristiques des langues humaines, l'ambiguïté.

Au cours des années, d'autres applications concernant les langues humaines ont été considérées. La discipline les étudiant plus particulièrement a ainsi pris le nom de Traitement Automatique du Langage Naturel (TALN). Les chercheurs de ce domaine se heurtent toujours au problème du sens en général et de la polysémie en particulier, problème resté central et ainsi très souvent incontournable.

Dans ce chapitre, nous présentons succinctement le domaine du TALN en montrant quelques-unes des applications quotidiennes issues de ces recherches. Nous abordons les quatre niveaux de traitements que l'on retrouve habituellement pour analyser un texte écrit : niveau *morphologique*, niveau *syntaxique*, niveau *sémantique* et niveau *pragmatique*. Nous nous concentrons en particulier sur les deux derniers qui concernent la question du sens. Nous présentons plusieurs approches sur la modélisation du sens qui ont été abordées en informatique. Nous exposons ainsi des approches issues de la linguistique distributionnelle d'Harris : les vecteurs saltoniens et les techniques de type LSA (*Latent Semantic Analysis*). Nous continuons par les approches symboliques connexionnistes (réseaux sémantiques, bases d'acceptions) avant, pour finir, de présenter la linguistique componentielle et les modèles informatiques parents. Nous nous attardons en particulier sur les travaux effectués par Jacques Chauché au début des années 1990. Ces travaux sont à la base du modèle des vecteurs d'idées qui constituent le modèle sémantique central de notre thèse.

1.1 Le Traitement Automatique du Langage Naturel

Depuis une cinquantaine d'années, le domaine que l'on nomme *Traitement Automatique du Langage Naturel* (TALN), *Traitement automatique des langues* (TAL) ou parfois *informatique linguistique* n'a cessé d'évoluer et de se diversifier au gré des progrès techniques et des besoins économiques, scientifiques ou militaires. S'il ne nous appartient pas dans cette thèse de revenir sur la constitution et l'histoire parfois houleuse du domaine pour lesquelles le lecteur trouvera une très intéressante présentation dans [Cori & Léon, 2002], il nous semble indispensable de le décrire brièvement ici tel qu'il nous apparaît aujourd'hui.

[4] C'est-à-dire l'équivalent de milliards d'aujourd'hui.

1.1.1 Qu'est-ce que le TALN ?

Commençons par définir ce que nous entendons par *langage naturel*[5]. Il s'agit du langage tel qu'il est parlé quotidiennement par les êtres humains et qu'ils ont créé de façon émergente (comme le français, l'anglais, le chinois ou le malais) par opposition aux langages artificiels construits de façon consciente par l'être humain et utilisés en logique, mathématiques ou informatique. Ainsi, à la vue de nos lectures et de notre expérience, une définition acceptable du TALN serait de le considérer comme :

le domaine d'étude des techniques automatiques
d'analyse (compréhension) et de génération (production)
d'énoncés oraux ou écrits.

La langue est la manière la plus naturelle de communiquer pour l'Homme. Il n'est donc pas très surprenant que, depuis le début de l'informatique et en particulier depuis que les ordinateurs ont commencé à rentrer de façon significative dans la vie des gens, le TALN ait connu un essor très important. Ainsi, un certain nombre d'outils issus des recherches en TALN font aujourd'hui partie du quotidien du grand public :

- ***Reconnaissance et synthèse de la parole***
  - *transcription de la parole* : après un temps d'apprentissage nécessaire au logiciel pour s'habituer aux spécificités d'accentuation et d'intonation d'un utilisateur ou pour acquérir un vocabulaire spécifique, des logiciels comme *Dragon Naturally Speaking*[6] ou *IBM ViaVoice*[7] transcrivent très correctement sous la dictée des textes en différentes langues ;
  - *serveurs vocaux* : pour consulter sa messagerie téléphonique ou obtenir des renseignements, il est maintenant possible de prononcer un mot-commande qui va orienter la suite des opérations au lieu de taper tel ou tel numéro. Certain de ces serveurs vocaux (météo, SNCF, ...) répondent aux utilisateurs en synthétisant une voix à partir des données qui leur sont accessibles ;
  - *téléphones portables* : il semble aujourd'hui tout à fait naturel pour un certain nombre de personnes de dire le nom de quelqu'un à son téléphone portable pour que celui-ci l'appelle ;
  - *aide pour personnes handicapées* : les personnes ne pouvant pas manipuler les claviers et les souris peuvent utiliser la voix pour naviguer sur le net ou contrôler des applications. Pour les malvoyants en particulier, des logiciels spécialisés sont capables de "lire" les textes écrits[8] ;
  - *outils d'enseignement assisté par ordinateur* : en apprentissage des langues, certains logiciels comme *Tell me more*[9] ou *Reflex'English*[10] aident à corriger la prononciation. En apprentissage de la lecture, d'autres logiciels permettent de vérifier la bonne compréhension de l'enfant[11]. Souvent, les textes présentés sont "lus" pour servir d'exemple à l'apprenant[12] [13] ;
  - jeux : de nombreux jeux, depuis déjà plus de quinze ans, utilisent la synthèse vocale : le *Manoir de Mortevielle*[14](1987), *Maupiti Island*[15] (1990), dans les jeux actuels citons

[5] Un glossaire des principaux termes utilisés dans cette thèse se trouve en annexe I
[6] http://www.scansoft.fr/naturallyspeaking/
[7] http://www-4.ibm.com/software/speech/
[8] http://www.handialog.com/we/index.htm
[9] http://www.auralog.com/fr/tellmemore_7.html
[10] http://www.reflex-deutsch.com/ref-english.htm
[11] www.tipataflam.com
[12] http://www.auralog.com/fr/tellmemore_7.html
[13] http://www.reflex-deutsch.com/ref-english.htm
[14] http://www.lankhor.net/jeux/jeux.php3?jeu=15&menu=presentation
[15] http://www.lankhor.net/jeux/jeux.php3?jeu=16&menu=presentation

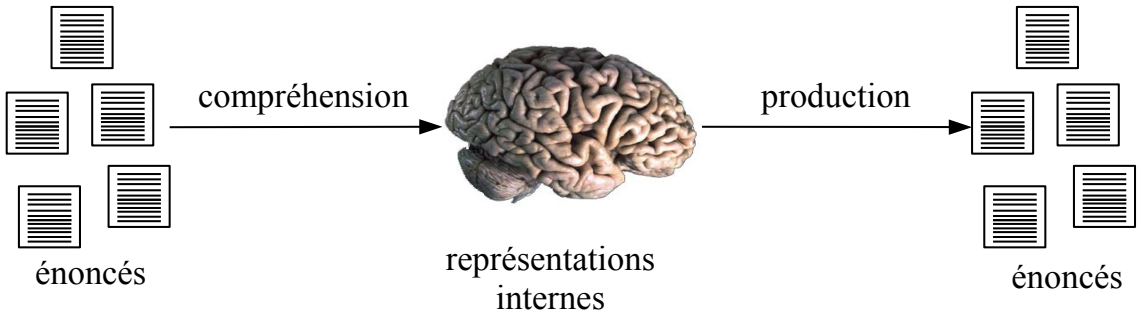
Neverball[16] qui utilise *NeoSpeech SAPI5 Voices Kate and Paul*[17]. Beaucoup plus récemment, sont apparus les premiers jeux utilisant la reconnaissance vocale. On citera en particulier *Socom II : U.S. Navy Seals*[18]où il s'agit de diriger un commando en donnant oralement ses ordres aux autres membres.

- ***Traduction automatique*** : la version Web de Systran[19] est souvent utilisée par les utilisateurs du net pour les aider à comprendre le contenu de certaines pages ;
- ***Correction orthographique, grammaticale*** : openOffice.org[20] et Ispell[21] peuvent corriger les fautes d'orthographe des fichiers textes, Microsoft Word[22] propose, en plus, une correction grammaticale relativement correcte. À noter les premiers efforts du libre dans cette catégorie d'outils avec GRAC[23] (GRAmmar Checker) écrit en Python et distribué sous licence GPL qui se base sur un apprentissage à partir de textes annotés et sans faute pour déduire des règles de grammaire. Il est théoriquement fonctionnel quelle que soit la langue ;
- ***Indexation documentaire, recherche d'informations*** : les moteurs de recherche Web comme Google[24] ou Yahoo[25] utilisent, en partie, des techniques issues du TALN. Les sites du *Monde*[26] et *Leroy Merlin*[27] ont choisi le moteur de recherche et de navigation "sémantique" *intuition* de la société *Sinequa*[28] ;
- ***Filtrage d'informations*** : il s'agit ici de traiter un grand nombre d'informations pour en extraire celles qui paraissent les plus pertinentes pour une clientèle donnée. Pour le grand public, il s'agira plutôt d'informations politiques, culturelles ou sportives comme le réalise Google news[29], pour les entreprises plutôt de veille technologique et pour les militaires, de veille stratégique (espionnage) ;
- ***Résumés, synthèses automatiques*** : le filtrage d'informations conserve en général de grandes masses de données que des synthèses automatiques permettent de largement réduire. La société *Pertinence Mining* propose un résumeur en ligne multilingue gratuit pendant un mois[30].

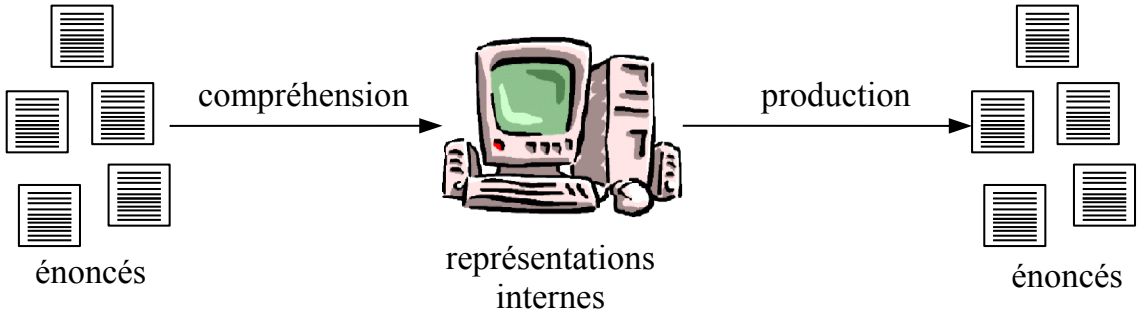
On pourrait sans doute continuer à rajouter des applications issues du TALN à cette liste mais son contenu aura déjà certainement convaincu le lecteur de la présence du TALN dans sa vie quotidienne.

Dans cette thèse, nous ne nous intéressons qu'aux énoncés écrits, aux textes, mais le traitement des énoncés oraux se déroule suivant des processus sensiblement identiques, en particulier on peut toujours décomposer une application de TALN en une phase d'analyse et une phase de production.

[16] http://icculus.org/neverball/
[17] http://www.sharewareorder.com/NeoSpeech-SAPI5-Voices-Kate-and-Paul-download-38174.htm
[18] http://us.playstation.com/Content/OGS/SCUS-97275/site/main.html?confirm=1
[19] http://www.systransoft.com/
[20] http://fr.openoffice.org/
[21] http://www.gnu.org/software/ispell/ispell.html
[22] http://office.microsoft.com/fr-fr/default.aspx
[23] http://grac.sourceforge.net/
[24] http://www.google.fr/
[25] http://fr.yahoo.com/
[26] http://www.lemonde.fr/
[27] http://www.leroymerlin.fr/
[28] http://www.sinequa.com/
[29] http://news.google.fr/nwshp?hl=fr&gl=fr
[30] http://www.pertinence.net/ps/index.html



(a) Le cerveau humain, dans une phase de compréhension, d'analyse des énoncés (dialogues, textes, ...), fabrique sa propre représentation interne qu'il peut ensuite utiliser pour produire, générer d'autres énoncés : dialogues, traductions, catégorisations, résumés,...



(b) Un système de TALN est conçu dans le but d'imiter le plus possible les résultats des activités langagières qu'un être humain serait capable de réaliser : dans une phase d'analyse des énoncés (dialogue, textes, ...), il fabrique sa propre représentation interne tandis que dans une phase de production, il l'utilise pour produire, générer d'autres énoncés : dialogue, traductions, catégorisations, résumés, ...

FIG. 1.1 – Activité langagière du cerveau et activité d'un système de TALN.

1.1.2 Analyse et production d'énoncés

Si on se place dans une perspective cognitiviste, on peut considérer que les énoncés ne sont rien de plus que les résultats d'une activité cognitive exercée par le cerveau humain. Le TALN cherche à développer des outils permettant de produire des résultats aussi proches que possible de ceux que pourrait créer cette activité cognitive. Le cerveau humain, dans une phase de compréhension, d'analyse des énoncés (dialogues, textes, ...), fabrique sa propre représentation interne qu'il peut ensuite utiliser pour produire, générer d'autres énoncés : dialogues, traductions, catégorisations, résumés,... Un système de TALN est conçu dans le but d'imiter le plus possible les résultats des activités langagières qu'un être humain serait capable de réaliser : dans une phase d'analyse des énoncés (dialogue, textes, ...), il fabrique sa propre représentation interne tandis que dans une phase de production, il l'utilise pour produire, générer d'autres énoncés : dialogue, traductions, catégorisations, résumés, ... Il convient de noter que certaines applications issues du TALN, tout comme certaines des activités du cerveau, ne nécessitent pas forcément une phase d'acquisition (synthèse vocale de la météo à partir des données brutes par exemple) ou une phase de génération (commandes vocales).

De nombreuses recherches sur les activités langagières de l'être humain ont été réalisées depuis le XIXe siècle tant dans le domaine des neurosciences [Damasio & Damasio, 1992, Damasio, 1995] que dans celui de la psychologie [Quillian, 1968, Le Ny, 1979]. Même si ces travaux ont apporté des avancées considérables sur les processus d'analyse, de production ainsi que sur la représentation interne des informations tant langagières que non langagières du cerveau, une grande

partie de ces phénomènes reste méconnue voire hypothétique. Devrait-on pour autant abandonner pour le moment l'idée de concevoir des applications TALN performantes ? Il est clair que non, les applications présentées précédemment, même dans leurs limites, en sont de bons exemples.

Il est fondamental de comprendre avant de se lancer dans l'exploration de ce domaine qu'est le TALN que, de même que les avions imitent les oiseaux par le vol mais pas dans la manière de voler, le TALN cherche à imiter les résultats des activités langagières mais pas forcément ces activités elles-mêmes. De la sorte, si dans les deux cas, il y a une phase d'analyse, une phase de production et une représentation interne des informations, celles-ci ne doivent (ne peuvent ?) pas être identiques.

Pour résumer, nous pouvons dire que construire une application TALN, c'est chercher à reproduire (imiter) des activités langagières pouvant être réalisées par un être humain sans nécessairement utiliser les mêmes représentations internes ni les mêmes processus de traitement.

1.1.3 Mot, item lexical, terme

Une définition communément admise considère qu'un mot est une suite de caractères graphiques formant une unité sémantique et pouvant être distinguée par un séparateur (généralement un blanc typographique ou un signe de ponctuation[31]) [Larousse, 2004] [Robert, 2000]. Toutefois cette définition reste relativement floue et souvent inadéquate aux problèmes qui nous sont posés puisqu'elle ne tient pas compte, par exemple, des locutions non-connexes comme ‹*mettre les pieds dans le plat*›. Non-connexe puisqu'on peut trouver une phrase comme « *Il met souvent les pieds dans le plat.* » où un mot s'intercale dans la locution.

Nous préférerons donc une définition plus "lexicale" et, en toute rigueur, *item lexical* plutôt que *mot*. Nous définissons un item lexical comme « *une suite de caractères formant une unité sémantique et pouvant constituer une entrée de dictionnaire* ». Ainsi, ‹*voiture*›, ‹*clair*›, ‹*être*› tout comme ‹*pomme de terre*›, ‹*moulin à vent*›, des locutions verbales comme ‹*tirer le diable par la queue*›, des mots ayant un sens différent au singulier et au pluriel comme ‹*orgue*› et ‹*orgues*›[32] et même des termes techniques comme ‹*pompe bivalve à échappement central*› sont des items lexicaux.

Dans cette thèse, nous réservons *item lexical*, *lemme* ou *terme* pour la forme canonique (cf. morphologie 1.1.4.1) tandis que nous utilisons plutôt *mot* pour les formes fléchies. Habituellement, *terme* est, comme son nom l'indique, plutôt utilisé dans un cadre terminologique ; toutefois nous le prenons dans cette thèse dans une acception plus large comme désignant une entrée d'un lexique général. Ainsi, nous considérons *terme* et *lemme* comme parfaitement synonymes d'*item lexical*. Nous insistons sur *item lexical* pour bien mettre l'accent sur l'idée que ce qui nous importe ici est que ce soit une entrée de dictionnaire.

En ce qui concerne les notations, les items lexicaux sont notés en petits caractères, en italique et entre apostrophes (‹*vie*›), les mots en petits caractères et entre guillemets (‹‹vie››).

[31] Nous nous restreignons volontairement ici aux langues indo-européennes écrites. En linguistique historique, l'indo-européen est la langue préhistorique dont les langues indo-européennes actuelles seraient issues. Ces langues sont environ un millier et possèdent trois milliards de locuteurs. Plusieurs groupes de langues indo-européennes coexistent, parmi lesquels le groupe *indo-iranien* (sanskrit, népalais, persan, kurde, ...), le groupe *celte* (irlandais, gaélique, ...), germanique (néerlandais, allemand, anglais, ...) et le groupe *roman* qui réunit les langues issues du latin (portugais, castillan, français, ...)

[32] Un ‹*orgue*› est un instrument de musique tandis que des ‹*orgues*› (au féminin) sont « *en géologie, des prismes d'une grande régularité formés lors du refroidissement d'une coulée de lave, basaltique le plus souvent, perpendiculaire à la surface.* » [Larousse, 2004]

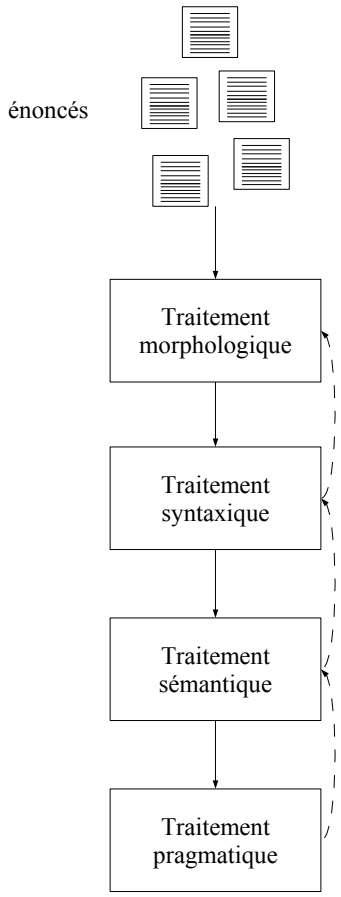


FIG. 1.2 – Schéma général d'analyse de textes.

1.1.4 Niveaux de traitement linguistique

Compréhension et production de textes peuvent se décomposer en une suite de traitements symétriques. La compréhension, comme le présente la figure 1.2, est constituée d'un *traitement morphologique* qui consiste à identifier les items lexicaux possibles à partir d'une forme fléchie donnée, puis d'un *traitement syntaxique* qui cherche à identifier les relations qui existent entre les mots d'une phrase (sujets, verbes, compléments, ...), d'un *traitement sémantique* qui cherche à capturer le "sens" des phrases et enfin d'un *traitement pragmatique* qui consiste à trouver la signification complète d'un texte liée en grande partie à la présupposition. Une phase de production est décomposable en une séquence inverse. Cette décomposition reste théorique et un certain nombre de cas d'ambiguïtés nécessite des informations issues du niveau suivant.

Il serait illusoire de penser que l'analyse d'un énoncé consiste pour chaque niveau à faire uniquement un choix parmi les possibilités proposées au niveau précédent, à *filtrer* des informations. En effet, la complexité de la langue tant au niveau de la construction des mots, des phrases, des idées qu'au niveau de son évolution (parfois apparition de nouvelles tournures mais très couramment de nouveaux termes souvent de façon éphémère) laisse difficile voire impossible une couverture totale des possibilités de chaque niveau. Ce modèle par couches permet éventuellement à un niveau supérieur de compléter les informations d'un niveau inférieur. Ainsi, dans le cas où le niveau morphologique ignorerait l'existence de ‹*orgues*› (au pluriel) le niveau sémantique pourra, s'il connaît cette forme, la lui indiquer.

Tandis que traitement morphologique et traitement syntaxique sont relativement bien connus et étudiés en linguistique, les deux suivants sont encore très méconnus et l'objet de bien des

théories. Ce sont ces deux niveaux, plus particulièrement, qui font l'objet de cette thèse.

1.1.4.1 Niveau morphologique

Niveau morphologique en linguistique En linguistique, la *morphologie* est l'étude de la façon dont sont formés les mots. On appelle *morphèmes* les unités minimales significatives qui constituent les mots. Par exemple, le mot ‹‹fleurs›› est constitué de deux morphèmes : le radical (ou base) correspondant à l'item ‹*fleur*› et le suffixe marquant le pluriel *s*. Il existe deux catégories de morphèmes :

- les *morphèmes lexicaux* qui correspondent aux items lexicaux ou à une légère variante ;
- les *morphèmes grammaticaux*, autrement appelés *affixes*. Situé avant le radical, un affixe est dit *préfixe*, après le radical, il est dit *suffixe* et dans le radical, *infixe*.

On peut distinguer deux types de formations morphologiques[33] :

- *flexion* : Les mots dits *fléchis* comportent un radical et une ou plusieurs désinences. Les désinences sont les morphèmes porteurs des indications de nombre et de genre pour les noms, adjectifs et déterminants, de temps, de personnes et de mode pour les verbes. Ainsi, ‹‹lisions›› est constitué du radical *lis-* issu de l'item ‹*lire*›, de la désinence temporelle *-i-* et de la désinence personnelle *-ons* ([Lehmann & Martin-Berthet, 1998], p. 132) tandis que ‹‹rattes›› est lui formé par *rat* (radical) + *te* (féminin) + s (pluriel). En aucun cas, la flexion ne modifie donc la catégorie syntaxique ;
- *dérivation* : On parle de dérivation lorsqu'un mot est formé à partir d'un autre en y adjoignant un ou plusieurs affixes porteurs de sens. Ainsi le sens du radical se trouve modifié et contrairement aux flexions, les dérivations peuvent amener à une nouvelle catégorie syntaxique. Si on prend pour exemple l'adjectif ‹‹inacceptable››, il est formé de *in* (affixe de contraire) + *accept* (le radical, le verbe ‹*accepter*›) + *able* (affixe de possibilité).

Dans la suite, lorsque nous parlerons de la *morphologie* pour un item lexical ou un mot, il s'agira d'un abus de langage qui désigne les informations que nous pouvons déduire de la morphologie de cet item ou de ce mot. Ainsi, nous aurons les *catégories grammaticales* (*nom, pronom, adjectif, adverbe, etc.*), le *genre* (*masculin, féminin, neutre*), le nombre (*singulier, pluriel*), le mode (*transitif, intransitif*), ...

La *forme canonique* d'un mot est la forme de ce mot telle qu'on peut la trouver comme entrée d'un dictionnaire par opposition à la forme fléchie. Par définition, un item lexical est donc toujours dans une forme canonique. Traditionnellement, suivant la nature de l'item, une forme particulière est choisie :

- *verbe* : à l'infinitif ;
- *nom* : au singulier (s'il existe) ;
- *adjectif* : au masculin singulier

On peut remarquer que, pour les mots invariables, formes fléchie et canonique sont identiques.

Niveau morphologique en TALN En TALN, dans une phase d'analyse, la partie dérivation de la morphologie est rarement utilisée. À notre connaissance, à part dans un module de notre système qui extrait les affixes des mots et qui est ainsi utilisé pour essayer de capturer le sens des mots inconnus, elle n'est jamais exploitée. En revanche, en ce qui concerne une phase de production, [Grabar & Zweigenbaum, 1999] utilise, par exemple, les propriétés dérivationnelles des mots pour compléter des listes de termes médicaux en créant les adjectifs correspondant aux noms déjà présents.

Habituellement, dans la phase d'analyse, il s'agit plutôt de reconnaître les items lexicaux des textes. Il s'agira ainsi de retrouver à partir d'un mot les possibilités de radical et d'affixe

[33] Il s'agit bien ici de types et non de catégories puisque ces formations ne sont pas absolument délimitées et l'on ne peut donc pas toujours véritablement les catégoriser.

ainsi que leurs caractéristiques grammaticales. Par exemple, ‹*charges*› peut être le nom féminin ‹*charge*› au pluriel comme le verbe ‹*charger*› à l'indicatif ou au subjonctif présent. Cette opération est aussi appelée *lemmatisation*. Une partie des ambiguïtés peut être levée au niveau syntaxique du processus d'analyse.

1.1.4.2 Niveau syntaxique

Niveau syntaxique en linguistique La syntaxe étudie la manière dont les mots se combinent pour former des phrases. D'un point de vue purement grammatical, l'étude de la syntaxe concerne trois types d'objets :

- Le *mot*, qui constitue la limite inférieure de l'étude syntaxique et qui en est donc le constituant de base ;
- La *phrase*, qui constitue la limite supérieure de l'étude syntaxique ;
- Le *syntagme* (ou *groupe*), qui en est l'unité intermédiaire.

Les mots et les syntagmes sont appelés *constituants* de la phrase. Des règles précises régissent les combinaisons entre les mots pour former des syntagmes et les combinaisons entre syntagmes pour former des phrases. Ainsi, si le syntagme nominal « *le chat* » est correct, il n'en est pas de même pour *« *chat le* » [34] ni pour la phrase *« *Riche il et beau est.* » dont l'expression conforme à la grammaire française serait plutôt « *Il est beau et riche.* ».

Si les mots ont des catégories grammaticales, les syntagmes ont aussi leurs propres catégories (syntagme verbal, syntagme nominal, syntagme prépositionnel, ...). Les règles syntaxiques permettent de décrire chacun des constituants de la phrase :

- leur *nature* : morphologie pour les mots, verbal, nominal, adjectival pour les syntagmes ;
- leur *structure hiérarchique* : la manière dont les mots se regroupent pour former des syntagmes et la manière dont les syntagmes se regroupent pour former la phrase ;
- leur *fonction syntaxique* : le rôle qu'ils tiennent à leur niveau de la hiérarchie : sujet, verbe, complément, ...

Dans chaque syntagme, il y a un constituant qui a un rôle particulier, il a la fonction syntaxique de *gouverneur*. Il s'agit du constituant principal du syntagme. Dans un syntagme nominal, il s'agit du nom, dans un syntagme verbal du verbe, dans une phrase, du syntagme sujet, etc.

Habituellement, on présente l'analyse syntaxique d'une phrase sous la forme d'un *arbre syntaxique*. Par exemple, l'arbre syntaxique de la phrase « *Le petit chat boit du lait.* » est présenté en figure 1.3.

Niveau syntaxique en TALN Après la phase d'analyse morphologique, un certain nombre de solutions sont envisageables pour les mots d'une phrase. Une analyse syntaxique permet grâce aux règles de ne conserver que les solutions qui sont possibles. Par exemple, prenons la phrase « *Des charges supplémentaires seront retenues contre l'accusé.* ». Le mot ‹‹charges››, comme nous l'avons vu dans la partie consacrée à la morphologie, peut être le *nom féminin* ‹*charge*› au pluriel comme le verbe ‹*charger*› à l'indicatif ou au subjonctif présent. La morphologie possible des mots constituant le groupe nominal sujet « *des charges supplémentaires* » (pour ‹‹des››, *déterminant pluriel*, pour ‹‹supplémentaires››, *adjectif masculin ou féminin pluriel*) rendent ici la seule solution possible pour ‹‹charges›› *nom féminin pluriel*.

La recherche pour mettre au point des analyseurs fiables est encore florissante. La tâche est complexe puisqu'il n'existe pas à l'heure actuelle de règles grammaticales pouvant couvrir l'ensemble des phrases correctes dans aucune des langues existantes. Ainsi, deux grandes familles d'analyseurs coexistent ([Bangalore, 1997], p. 5) :

[34] En linguistique, il est traditionnel de faire précéder d'un astérisque une construction erronée en langue.

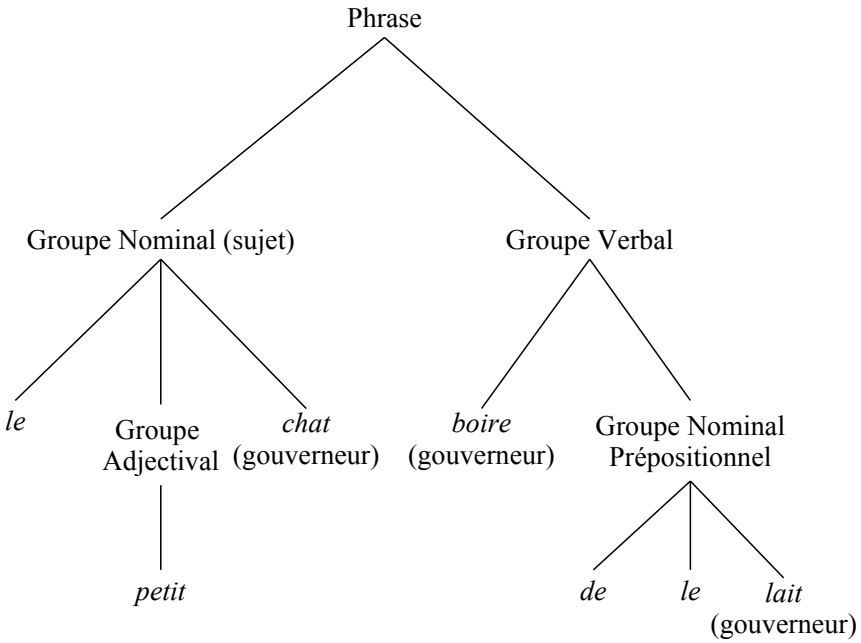


FIG. 1.3 – Arbre syntaxique de la phrase « *Le petit chat boit du lait.* ».

- *approche symbolique* : ces analyseurs se basent sur des règles grammaticales et nécessitent donc une recherche et une implémentation de ces règles. On peut citer dans cette catégorie, ARIANE [Boitet, 2000], l'analyseur du GREYC [Vergne & Giguet, 1998], IPF [Wehrli, 1992], SYGMART [Chauché, 1984] ;
- *approche statistique* : ces analyseurs se basent sur des méthodes d'apprentissage à partir de corpus annotés manuellement ou automatiquement pour produire des règles pondérées [Church, 1988, Collins, 1997, Muñoz *et al.*, 2000].

Quelle que soit la technique utilisée, les analyseurs syntaxiques ne renvoient pas tous un arbre syntaxique complet comme celui présenté en 1.3. Ainsi, deux autres types de résultats sont possibles : renvoyer les relations entre les mots de la phrase ou produire une segmentation en syntagmes.

Une analyse syntaxique ne peut pas toujours lever toutes les ambiguïtés. Ainsi, certaines phrases comme « *La petite brise la glace.* » ne peuvent être totalement désambiguïsées à ce niveau de traitement comme on peut le constater sur la figure 2.4. Deux interprétations syntaxiques sont ici possibles. Dans la première, “petite” et “glace” sont des noms, “brise” est la troisième personne du présent de l'indicatif du verbe ‹*briser*› (ie. une petite fille casse un miroir) tandis que dans la deuxième, “petite” correspond à l'adjectif ‹*petit*›, “glace” au verbe ‹*glacer*› et “brise” à l'item lexical ‹*brise*› (ie. un léger vent donne froid à quelqu'un ou quelque chose de féminin). Si syntaxiquement, il est absolument impossible de lever l'ambiguïté, des informations de nature sémantique et pragmatique sur cette phrase peuvent permettre d'émettre des préférences.

Niveau syntaxique dans le cadre de cette thèse : SYGFRAN Les membres de l'équipe TALN du LIRMM utilisent dans leurs expérimentations sur le français l'analyseur morpho-syntaxique SYGFRAN. Il s'agit d'un programme développé grâce à SYGMART (SYstème Grammatical de Manipulation Algorithmique et Récursive de Texte), un outil qui permet de construire des applications traitant des notions de morphologie, de syntaxe et/ou de sémantique [Chauché, 1984]. SYGMART est un système transformationnel d'arborescences fondé sur les algorithmes de Markov étendus aux arbres. Il permet d'analyser tout langage dont la grammaire

pourrait être écrite sous forme de transducteurs d'arbres. L'application pour le français, SYGFRAN comporte environ 11000 règles[35] inspirées des travaux du linguiste Jürgen Weissenborn. SYGMART et son application SYGFRAN sont développées par Jacques Chauché[36]. SYGFRAN renvoie, pour un texte quelconque en français, l'arbre morpho-syntaxique correspondant. Un exemple d'analyse rendue par SYGFRAN est celui de la phrase « *Le petit chat boit du lait.* » présenté dans la figure 1.3.

L'utilisation de SYGFRAN offre deux principaux avantages :

- *Analyse robuste* : SYGFRAN propose une analyse robuste. En effet, même lorsque certains termes sont inconnus, même si la phrase est agrammaticale, il en propose une analyse partielle et indique ses incertitudes ;
- *Rapidité* : Un deuxième grand avantage de SYGFRAN est sa rapidité. En effet, sa complexité théorique est relativement faible : $O(m \times n \times log_2(n))$ où n est la taille de la donnée et m le nombre de règles. En pratique, les résultats sont aussi très intéressants. L'analyseur vient de participer à la campagne EASY (Évaluation des Analyseurs Syntaxiques du Français). Si nous n'avons pas, à l'heure où nous écrivons ces lignes, les résultats, nous pouvons présenter les temps qu'il a fallu pour annoter le corpus d'évaluation de l'ELDA (Evaluation and Language Resources Distribution Agency) utilisé pour la campagne.

Corpus	taille	temps	vitesse
Le Monde	471 Ko	42 min	11.2 Ko.min^{-1}
littéraire	439 Ko	48 min	9.1 Ko.min^{-1}
médical	119 Ko	9 min	13.2 Ko.min^{-1}
global	4.19 Go	7H13	9.67 Ko.min^{-1}

FIG. 1.4 – Extrait des temps réalisés par SYGFRAN pour effectuer l'analyse syntaxique des textes du corpus de l'ELDA avec un Pentium IV 2,4Ghz (4734 Bigomips) 1Go Ram

Comme nous pouvons le constater, SYGFRAN est très rapide. Sa vitesse d'analyse varie, suivant le type de corpus, entre $9.1 Ko.min^{-1}$ pour un corpus littéraire et $13.2 Ko.min^{-1}$ pour un corpus médical. Cette différence s'explique par la longueur des phrases nettement plus importante pour le corpus littéraire. Quoi qu'il en soit, pour des textes de longueur moyenne, la durée d'une analyse est de l'ordre de 10 secondes.

1.1.4.3 Niveaux sémantique et pragmatique : un découpage difficile à réaliser

Niveaux sémantique et pragmatique en linguistique Les niveaux sémantique et pragmatique sont encore beaucoup plus complexes à décrire et à formaliser que les niveaux de traitement précédents. En effet, ils touchent à l'étude du sens de la phrase dans le contexte général dans lequel elle s'inscrit. Plus que les autres niveaux de traitement, sémantique et pragmatique sont extrêmement liées. Ce lien est dû à la question commune que se posent les deux disciplines « *Qu'est-ce que le sens ?* ».

Sémantique La sémantique est l'étude du sens des énoncés. Cette science qui, bien que fort ancienne puisque déjà étudiée par les philosophes de l'Antiquité, fait encore l'objet de bien des recherches car non seulement le sens est indispensable dans une phase de compréhension de textes mais aussi car aucun moyen de le décrire complètement ne fait aujourd'hui l'unanimité.

Nombreux sont les ouvrages traitant de sémantique, mais fort rares sont ceux qui se risquent à donner ne serait ce qu'une esquisse de définition du terme ‹*sens*›. En effet, le sens est quelque

[35] En Janvier 2005.
[36] http://www.lirmm.fr/~chauche/Pr%E9sentationSygmart.html

chose de difficile à décrire car intuitif et souvent considéré dans ces livres comme déjà acquis par le lecteur. ([Polguère, 2003], p. 98) déroge à cette règle en reconnaissant explicitement le caractère intuitif du sens et en présente une approche dont nous nous inspirons largement ici.

Il n'est pas rare, même pour un locuteur du français, de rencontrer des mots qui lui sont inconnus. Ainsi, nous avons tous vécu un dialogue tel que celui que nous pouvons imaginer entre un maître et son élève :

– Que veut dire « *prendre la poudre d'escampette* » ?
– Cela signifie « *s'enfuir* » ou « *se sauver à toutes jambes* ».

Un bon moyen de faire comprendre ce que signifie un mot est donc d'utiliser une expression équivalente, une paraphrase. Sur cette idée, Alain Polguère propose comme définition du sens :

Le sens d'une expression linguistique est la propriété qu'elle partage avec toutes ses paraphrases.

On le voit, cette définition repose sur la notion d'équivalence entre phrases, les paraphrases. Ces équivalences sont loin d'être rares en langue, c'est même une des caractéristiques essentielles des langues naturelles par rapport aux langages artificiels. Ainsi, pour Polguère, la notion de paraphrase est reconnue comme un concept primitif possédé par un locuteur qui permet de définir la notion de sens.

Le sens d'un énoncé est régi par le *principe de compositionnalité sémantique* pour lequel « *le tout est calculable à partir du sens de ses parties* ». Ainsi, un énoncé est directement calculable (dans sa composition lexicale et sa structure syntaxique) à partir de la combinaison du sens de chacun de ses constituants ([Polguère, 2003], p. 134). Par exemple, le sens d'une phrase comme « *L'enfant voit la mer.* » est calculable à partir :

– des items lexicaux ‹*le*›, ‹*enfant*›, ‹*voir*›, ‹*la*›, ‹*mer*› ;
– des règles syntaxiques et morphologiques du français utilisées dans la phrase.

Il est souvent spécifié dans la littérature que les locutions transgressent, au moins en partie, le principe de compositionnalité sémantique. Dans notre approche où un mot est défini comme une des formes fléchies d'un *item lexical* (notion qui englobe les locutions, cf. 1.1.3), le problème ne se pose pas.

De nombreuses théories sur la sémantique ont été élaborées comme la *sémantique du prototype* [Kleiber, 1990], la *sémantique distributionnelle* ou la *sémantique structurale*. Nous présentons les deux dernières dans la suite de ce chapitre.

Pragmatique La pragmatique considère le sens du point de vue du récepteur et s'intéresse ainsi aux mécanismes de l'interprétation des énoncés. La question du *sens d'un énoncé* est alors posée de façon différente. De la sorte, un énoncé veut dire [Kerbrat-Orecchioni, 1986] :

1. ce que ses récepteurs estiment qu'il veut dire ;
2. ce que ses récepteurs croient que l'émetteur a voulu dire dans/par cet énoncé ;
3. ce que ses récepteurs estiment (à tort ou à raison, de façon réelle ou feinte, de bonne ou de mauvaise foi) être la prétention et l'intention sémantico-pragmatiques du locuteur dans cet énoncé.

Un exemple classique et quasi-quotidien de la pragmatique est la phrase « *Peux-tu me donner le sel ?* ». Il est clair que le locuteur n'attend pas la réponse « *Oui.* » à la question littérale qu'il a posée mais attend qu'on lui donne le sel.

La pragmatique est donc l'étude du sens des énoncés en contexte, c'est-à-dire l'ensemble des significations que peut lui donner un être humain. Le niveau pragmatique de la compréhension

de textes consiste ainsi à découvrir le sens correct d'un énoncé en fonction des conditions situationnelles et contextuelles dans lesquelles il apparaît. La pragmatique s'occupe en particulier des problèmes de référence, d'anaphore, de subjectivité.

Niveaux sémantique et pragmatique en TALN En traitement automatique du langage naturel, il s'agira de trouver le sens d'un texte et pour cela, de désambiguïser le sens des mots qui le composent. Prenons l'exemple de la phrase « *La souris est reliée à l'ordinateur.* ». Les traitements morphologique et syntaxique permettront de savoir que le mot “souris” correspond à l'item ‹*souris*›. Considérons (pour simplifier) que ce terme a deux sens le premier correspondant à l'animal, le deuxième à la souris d'ordinateur. Le traitement sémantique permettra de trouver un *sens préférentiel* (dans notre exemple, la souris d'ordinateur). Le traitement pragmatique, lui, choisira le ”bon sens” en fonction du contexte général. On peut imaginer que nous sommes dans un texte où l'on parle d'une petite souris (l'animal) qui se promène et qui se coince la queue dans le tiroir du lecteur de DVD ; alors le sens préférentiel du traitement sémantique ne sera pas celui choisi au cours du traitement pragmatique.

Pour simplifier, le traitement sémantique propose des *sens préférentiels* que le traitement pragmatique choisira en fonction de connaissances extérieures à la phrase (les autres phrases, le monde).

Niveaux sémantique et pragmatique dans le cadre de cette thèse Dans cette thèse, nous cherchons à améliorer un certain nombre de tâches en y apportant des informations d'ordre sémantico-pragmatiques. Nous allons alors être confronté aux trois questions principales posées habituellement par ces problèmes :

- *Comment représenter informatiquement le sens ?*
- *Comment alors désambiguïser les mots d'un texte ?*
- *Comment calculer le sens d'un texte ?*

Ce sont, en partie, ces questions que nous étudions dans cette thèse. Lorsque nous envisagerons une désambiguïsation, la sémantique, dont nous parlerons alors, considérera toujours le niveau pragmatique même si pour simplifier le discours nous ne le préciserons pas.

1.2 Représentations d'origine distributionnaliste

1.2.1 Approche distributionnelle

La linguistique distributionnelle [Harris *et al.*, 1989] est le nom donné aux recherches menées aux États-Unis par Zelig Sabbatai Harris (*1909 - 1992*) à partir des années 1950 et qui poursuivaient celles de son maître, Léonard Bloomfield (*1887 - 1949*). L'analyse distributionnelle cherche à décrire les objets linguistiques en fonction du pouvoir d'associativité qu'ils possèdent ou ne possèdent pas entre eux. Ainsi, l'objectif premier de cette branche de la linguistique est d'examiner les distributions des unités linguistiques (phonèmes, morphèmes, mots) dans un corpus donné.

La linguistique distributionnelle considère que le sens d'un mot peut être défini à partir de l'ensemble des contextes dans lequel il apparaît, en d'autres termes, par l'ensemble des termes qui lui sont cooccurrents dans un corpus. Par exemple, considérons ces quelques phrases extraites du Web :

- « *Seuls les chatons et pas les chats peuvent boire du lait de vache.* »
- « *Le pédiatre a diagnostiqué une allergie au lait de vache.* »
- « *Dis papa, c'est quoi cette bouteille de lait ?* »
- « *À partir du lait, le fermier fait des fromages et des yaourts.* »

Selon la linguistique distributionnelle, la sémantique de l'item ‹*lait*› peut ainsi être décrite grâce aux termes ‹*vache*›, ‹*bouteille*›, ‹*fromage*›, ‹*yaourt*›, ‹*allergie*›, ‹*chat*›, ‹*chaton*›, ...

On pourra dire que deux mots ont un sens proche s'ils sont employés dans des contextes très voisins. Ce sont ces idées qui ont permis la mise au point des vecteurs saltoniens et de leurs dérivés.

1.2.2 Représentations saltoniennes et dérivées

Ces méthodes sont basées sur les théories de la linguistique distributionnelle pour la représentation des contenus textuels. Le sens d'un texte est donné par un vecteur dont les composantes correspondent directement (modèle vectoriel standard) ou indirectement (LSA) aux items lexicaux constituant le texte.

1.2.2.1 Représentations saltoniennes

À partir de la fin des années 1960, Gerard Salton[37] (*1927 - 1995*) professeur à la *Cornell University*[38] met au point ce que l'on appelle aujourd'hui le *modèle vectoriel standard* (VSM pour *Vector Space Model*). Son application la plus connue est le système de recherche documentaire SMART[39] [Salton, 1971, Salton & McGill, 1983, Salton, 1991]. Suivant des idées issues de la linguistique distributionnelle, les dimensions de l'espace vectoriel sont associées à des *termes d'indexation*, c'est-à-dire aux termes considérés comme les plus discriminants dans le corpus de recherche.

Indexation des documents : fabrication des vecteurs Si t est le nombre de termes d'indexation, chaque document (et chaque requête) est représenté par un vecteur à t dimensions tel que :

$$D_i = (p_{i_1}, p_{i_2}, \ldots, p_{i_t}) \tag{1.1}$$

où p_{i_k} est la k-ième composante de D_i et a pour valeur le poids du terme T_k dans le document D_i. Le poids est souvent calculé par une formule de type $tf * idf$ (*term frequency* $*$ *inverse document frequency*). Par cette formule, il s'agit de prendre en compte deux critères :

- *l'importance du terme dans le document* : on appelle fréquence d'un terme (*term frequency*) le nombre de fois où ce terme apparaît, on parle aussi du *nombre d'occurrences* ou de la *fréquence d'occurrence*. Ce critère doit permettre de prendre en compte le fait que, plus le terme est présent, plus il a une importance dans le texte ;
- *le pouvoir discriminant du terme* : les mots fréquents dans un texte ne sont pas forcément les plus discriminants par rapport au corpus entier. Par exemple, identifier un grand nombre d'occurrences du terme ‹*lait*› dans un corpus dont le sujet central est justement le lait ne va pas permettre de différencier les divers documents. C'est pour contrebalancer ces cas que la prise en compte de la fréquence inverse en document est nécessaire. Il s'agit d'une évaluation de l'importance du terme dans l'ensemble du corpus. Plus le terme est présent, moindre sera l'idf.

Pour ces deux critères, plusieurs heuristiques peuvent être choisies. Ces dernières sont généralement basées sur la fréquence du terme t dans le document d, notée $f(t,d)$ ainsi que sur le nombre d'occurrences du terme le plus fréquent de d $Max(f(t,d))$. Par exemple, pour tf on peut trouver :

[37] http://www.cs.cornell.edu/Info/Department/Annual95/Faculty/Salton.html
[38] Ithica, État de New York, États-Unis d'Amérique.
[39] Une version gratuite est accessible gratuitement pour la recherche à l'adresse ftp://ftp.cs.cornell.edu/pub/smart/

- $tf = f(t,d)$ si on considère que l'importance de l'item n'est donnée que par le nombre d'occurrences dans le texte ;
- $tf = log(f(t,d)+1)$: la fonction logarithme augmente fortement dans les petites valeurs (<100) et puis augmente de moins en moins vite. Cette formule du *tf* est donc à choisir si on considère que l'on doit distinguer de façon moindre deux items ayant un nombre d'occurrences proches si leur fréquence dans le texte est importante et de façon plus importante dans le cas contraire ;
- $tf = \frac{f(t,d)}{Max(f(t,d))}$ si on considère que l'importance d'un terme est relative à celle du terme le plus présent dans le document. Notons que cette formule offre aussi l'avantage d'effectuer une certaine normalisation sur les vecteurs produits puisque le poids des composantes n'est pas influencé par la taille du document.

Pour *idf*, les heuristiques sont moins nombreuses, on utilise en général $log(\frac{N}{n})$ où N est le nombre total de documents du corpus et n le nombre de documents du corpus où le terme apparaît au moins une fois.

$tf * idf$ est donc la multiplication des valeurs de ces deux critères. Ainsi on pourra choisir comme formule :

$$\frac{f(t,d)}{Max(f(t,d))} \times log(\frac{N}{n}) \tag{1.2}$$

Exploitation des vecteurs La similarité entre deux documents D_a et D_b (ou entre un document et une requête dans le cas de SMART) est donnée par la formule (parfois dite *du cosinus* ; cf. A.39) :

$$\mathbb{R}^t \times \mathbb{R}^t \rightarrow [0,1] : \quad sim(D_a, D_b) = \frac{\sum_{k=1}^{t} p_{a_k} \times p_{b_k}}{\sum_{k=1}^{t} p_a 2 \times \sum_{k=1}^{t} p_b 2} \tag{1.3}$$

Les document les plus proches du document (ou de la requête) sont ceux qui maximisent la similarité (l'angle entre les vecteurs est alors le plus petit). On peut ainsi obtenir une liste ordonnée des documents les plus proches d'un autre document ou, dans un cas de recherche d'informations, de la requête.

Problèmes posés par la méthode Le premier problème du modèle vectoriel standard est aussi posé à l'ensemble des représentations vectorielles : la mise à jour de la base ne peut pas se faire de façon incrémentale. En effet, l'utilisation de méthodes basées sur le critère *idf* entraîne obligatoirement le recalcul de l'ensemble des vecteurs lors de l'ajout du moindre document au corpus.

Le second problème concerne le choix des termes d'indexation qui entraîne trois conséquences notables :

- plus le nombre de termes retenus est important, plus fines sont les représentations ;
- plus le nombre de termes retenus est important, plus longues sont les opérations à réaliser (tant en indexation qu'en exploitation) et plus grande est la taille des données à stocker ;
- plus le nombre de termes retenus est important, moins la différence entre les documents les plus proches et les documents les plus éloignés d'un document donné est faible.

Suivant les corpus, le nombre d'item lexicaux différents peut être relativement important. De la sorte, si la méthode utilisée pour choisir les termes d'indexation ne fait que sélectionner ces items, les vecteurs obtenus seront de très grande taille. L'approche doit donc être menée d'une manière plus fine. Elle peut être basée sur un antidictionnaire pour éliminer certains termes inadéquats, sur une stemmatisation pour extraire la racine des termes (tous les mots ayant la même racine seront alors considérés par la même composante), ou sur une lemmatisation.

1.2.2.2 Une approche psycholinguistique : LSA

Le modèle LSA (*Latent Semantic analysis*), appelé souvent aussi LSI pour *Latent Semantic indexing*, a été créé dans un objectif de psycholinguistique pour simuler l'acquisition de connaissances d'un être humain à partir de grands corpus de textes. Techniquement, LSA est une variante du modèle vectoriel standard qui cherche à la fois à réduire le nombre de dimensions des vecteurs et à améliorer la représentation en rajoutant des informations sur la structure sémantique implicite des unités linguistiques représentées par leurs dépendances cachées [Deerwester *et al.*, 1990].

En effet, les auteurs considèrent que le co-texte d'un item I n'apporte pas suffisamment d'informations sur le sens puisqu'on ne sait rien des liens sémantiques qu'entretiennent les mots de ce co-texte avec les items qui n'apparaissent pas conjointement à I. Par exemple, le co-texte de ‹*chaise*› peut être donné par {‹*s'asseoir*›, ‹*repos*›, ‹*bureau*›, ‹*siège*›, ‹*cuisine*›, ... } mais si un item comme ‹*fauteuil*› n'apparaît pas dans les co-textes de ‹*chaise*›, aucune information sur les rapports sémantiques entre les deux termes ne sera disponible. L'idée est donc de croiser les informations de cooccurrence de chaque item, c'est-à-dire ce que l'on appelle les *affinités de second ordre* [Grefenstette, 1994]. Dans LSA, le sens des termes est donc engendré par les enchaînements de cooccurrences, à savoir les liens implicites. Pour résumer, dans LSA, deux items sont similaires si leurs co-textes sont similaires. Deux co-textes sont similaires s'ils comportent des termes similaires [Lemaire & Dessus, 2003].

Dans un premier temps, la technique LSA consiste à construire à la manière du modèle vectoriel standard des vecteurs correspondant aux mots (dans ce cas l'unité du co-texte utilisée est généralement le paragraphe) ou aux documents. Dans un deuxième temps, il s'agit de regrouper les vecteurs dans une matrice et d'effectuer une décomposition en valeurs propres. Seuls les k premiers vecteurs propres sont pris en compte, l'espace de représentation est donc réduit à k dimensions. Une composante ne correspond pas à un terme particulier, ce qui empêche toute interprétation directe et ne rend possible que les comparaisons entre les vecteurs. La valeur de k ne doit pas être trop importante pour éviter le bruit et doit être suffisamment faible pour éviter les trop grandes pertes d'information. La valeur optimale de k a été estimée empiriquement pour l'anglais autour de 300 [Deerwester *et al.*, 1990].

LSA utilise deux mesures. La première, identique à celle utilisée dans le modèle vectoriel standard, permet d'estimer la similarité entre deux mots ou deux groupes de mots, à partir du cosinus entre les angles des vecteurs correspondants. La seconde mesure caractérise la connaissance que LSA a sur un mot ou sur un groupe de mots, à partir de la longueur du vecteur associé. Cette mesure, beaucoup moins utilisée dans la littérature, dépend de la fréquence des mots et de la diversité des contextes dans lesquels ils apparaissent.

Outre la recherche documentaire, la technique LSA a été utilisée dans plusieurs applications comme l'extraction de métaphores [Kintsch, 2000] ou pour la segmentation automatique des textes [Bestgen, 2004].

1.3 Représentations symboliques connexionnistes

Ces représentations peuvent être dessinées grâce à des graphes dont les sommets correspondent à des objets lexicaux (item, acceptions) et les arêtes à des relations sémantiques.

1.3.1 Relations sémantiques et fonctions lexicales

1.3.1.1 Relations sémantiques lexicales (ou relations sémantiques externes)

Nous avons déjà vu l'importance des paraphrases puisque ce sont elles qui nous ont permis de présenter ce que nous appelons le sens (cf. 1.1.4.3). À travers ce petit dialogue, nous pouvons

voir que communiquer c'est donc aussi pouvoir comprendre l'équivalence. Mais c'est également pouvoir comprendre les différences entre les phrases, savoir reformuler, développer, condenser ou améliorer son expression. Il semblerait donc que les relations sémantiques lexicales soient nécessaires à la compréhension linguistique des individus. Elles nous permettent une certaine maîtrise du lexique.

Les relations sémantiques structurent le lexique sur le plan paradigmatique[40]. Nous présentons succinctement ici les six principales pour en donner une première idée nécessaire avant d'appréhender les fonctions lexicales et les réseaux sémantiques. Ces relations seront approfondies dans cette thèse lorsqu'il s'agira pour nous de les formaliser et de les modéliser à l'aide des vecteurs d'idées. Ces six relations sont de deux types, les *relations de hiérarchie* (*hyperonymie/hyponymie*, *holonymie/méronymie*) et les *relations symétriques* (*synonymie/antonymie*).

Relations de hiérarchie Ces relations sont unidirectionnelles et transitives. Elles structurent ainsi le lexique de façon hiérarchique. Il s'agit des deux paires hyponymie/hyperonymie et méronymie/holonymie. Si $\mathcal{R}$ est une relation hiérarchique entre deux items A et B, alors il existe une relation symétrique $\overline{\mathcal{R}}$ telle que :

$$\mathcal{R}(A,B) \equiv \overline{\mathcal{R}}(B,A)$$

Il existe deux paires de relations hiérarchiques : *hyponymie/hyperonymie* et *méronymie/holonymie*.

Hyponymie et hyperonymie La relation d'hyponymie est la relation hiérarchique qui lie un hyponyme à un item plus général, l'hyperonyme. La relation d'hyperonymie est la relation inverse. Parmi les exemples d'hyponymie, on peut trouver : ‹*chat*›\‹*animal*›, ‹*voilier*›\‹*bateau*›, ‹*bateau*›\‹*véhicule*›, ‹*rose*›\‹*fleur*›.

La relation tout-partie, méronymie et holonymie La relation de méronymie est la relation hiérarchique qui lie la partie au tout. Un des éléments de la relation est une partie de l'autre élément. Les deux relations sont symétriques c'est-à-dire que le tout est l'holonyme de la partie tandis que la partie est le méronyme du tout. Parmi les exemples de méronymie, on peut trouver : ‹*voile*›\‹*bateau*› (‹*voile*› est méronyme de ‹*bateau*›), ‹*page*›\‹*livre*›, ‹*mur*›\‹*maison*›, ‹*jour*›\‹*mois*›.

Relations symétriques Les relations symétriques sont la synonymie et l'antonymie. Comme leur nom l'indique, ces relations vérifient la propriété de symétrie. Ainsi, si $\mathcal{R}$ est une relation symétrique entre deux items A et B, alors :

$$\mathcal{R}(A,B) \equiv \mathcal{R}(B,A) \tag{1.4}$$

En d'autres termes, si A est synonyme (respectivement antonyme) de B, alors B est synonyme (respectivement antonyme) de A.

Synonymie La synonymie est la relation sémantique qu'il existe entre deux items lexicaux qui diffèrent par leur forme mais expriment le même sens ou un sens très proche. Nous aborderons plus en détail cette relation à la section 3.2.1.

‹*avion*›\‹*aéroplane*›, ‹*écrivain*›\‹*auteur*›, ‹*livre*›\‹*bouquin*›, ‹*chat*›\‹*matou*›.

[40] Le plan paradigmatique est le plan dans lequel les termes sont unis par leur sens à l'intérieur du lexique.

acception	autoriser.1		
propriétés grammaticales	*verbe*		
formule sémantique	donner le droit de : X autorise Y à Z.		
régime 1	1=X	2=Y	3=Z
	1. N	2. N *obligatoire*	3. *à* N
	« *Le docteur autorise l'avion à Pierre.* »		
régime 2	1=X	2=Z	3=Y
	1. N	2. N *obligatoire*	3. *à* V *obligatoire*
	« *Le docteur autorise Pierre à embarquer.* »		
Fonctions lexicales	$Syn_{\cap}$	permettre.I laisser.2, tolérer	
	Anti	interdire I, ne pas autoriser I.1	
	$Anti_{\subset}$	proscrire	
	$Anti_{\cap}$	défendre, s'opposer, empêcher,...	
	S_0	autorisation	
	Magn	formellement, expressément	

FIG. 1.5 – Entrée résumée du DiCo pour l'item ‹*autoriser*› [Mel'čuk, 1988]

Antonymie L'antonymie est la relation sémantique qui existe entre deux items lexicaux dont les sens s'opposent. On le verra en 3.3.1 , il existe trois types d'antonymie qui sont caractérisés par l'application ou non d'une propriété (‹*vie*›\‹*mort*›, ‹*certitude*›\‹*incertitude*›), une propriété étalonnable (‹*riche*›\‹*pauvre*›, ‹*chaud*›\‹*froid*›) ou un usage (‹*père*›\‹*fils*›, ‹*lune*›\‹*soleil*›).

1.3.1.2 Fonctions lexicales de production

Les fonctions lexicales ont été créées dans le cadre de la théorie linguistique *sens-texte* (*TST*) élaborée par Igor Mel'čuk à Moscou avec des applications au russe jusqu'à la fin des années 1970 puis ensuite à Montréal pour le français. L'un des outils mis au point est le *Dictionnaire explicatif et combinatoire du français contemporain* (*DEC*) dont l'objectif est de décrire, de façon systématique et rigoureuse, les informations permettant à un locuteur de construire toutes les expressions linguistiques correctes de n'importe quelle pensée et ce, dans n'importe quel contexte : c'est un dictionnaire de production. Cette lexicographie est très détaillée et très organisée, ce qui lui permet d'être exploitable pour la fois à un usage humain et pour un usage "machinal" en TALN. Les items décrits dans le DEC le sont donc sous toutes leurs facettes : sémantique, syntaxique, lexico-combinatoire, morphologique, ... Le tableau de la figure 1.5 présente l'entrée résumée du DiCo pour un des sens de l'item ‹*autoriser*› [Mel'čuk, 1988].

Les fonctions lexicales ont été définies dans le cadre de la TST pour décrire les relations sémantiques lexicales au moyen d'un outil formel conçu sur le modèle des fonctions mathématiques ([Polguère, 2003], p. 131), ([Mel'čuk *et al.*, 1995], p. 127).

Une fonction lexicale f décrit une relation existant entre un item lexical I (l'*argument* de f) et un ensemble d'items lexicaux $\{I_1, I_2, \ldots, I_n\}$ appelé la *valeur* de l'application de f à l'item I. La fonction lexicale f est telle que :

1. l'expression $f(I)$ représente l'application de f à l'item I : $f(I) = \{I_1, I_2, \ldots, I_n\}$
2. chaque élément de la valeur de $f(I)$ est lié à I de la même façon et remplit (à peu près) le même rôle : $\frac{f(I_1)}{I_1} \approx \frac{f(I_2)}{I_2}$

Ainsi, pour illustrer, la fonction lexicale de synonymie *Syn* modélise le rapport qu'entretiennent entre eux les termes ‹*livre*› et ‹*bouquin*› d'une part et ‹*chat*› et ‹*matou*› d'autre part. Mel'čuk note :

$$\frac{\text{‹}livre\text{›}}{\text{‹}bouquin\text{›}} \approx \frac{\text{‹}chat\text{›}}{\text{‹}matou\text{›}} \approx \frac{\text{‹}matou\text{›}}{\text{‹}chat\text{›}}$$

De même, la fonction lexicale d'antonymie *Anti* modélise le rapport qu'entretiennent entre eux les termes ‹*certitude*› et ‹*incertitude*› d'une part et ‹*vie*› et ‹*mort*› d'autre part.

$$\frac{\text{‹}certitude\text{›}}{\text{‹}incertitude\text{›}} \approx \frac{\text{‹}vie\text{›}}{\text{‹}mort\text{›}} \approx \frac{\text{‹}mort\text{›}}{\text{‹}vie\text{›}}$$

Il existe autant de fonctions lexicales qu'il existe de liens lexicaux et chaque fonction lexicale est identifiée par un nom particulier. Deux classes de fonctions lexicales sont identifiées : les *fonctions lexicales paradigmatiques* et les *fonctions lexicales syntagmatiques*.

Fonctions lexicales paradigmatiques Comme leur nom l'indique, les fonctions lexicales paradigmatiques formalisent les relations sémantiques. On distingue par exemple :

- *Synonymie* (*Syn*) :
  Syn(‹*avion*›) = ‹*aéronef*›, ‹*aéroplane*›, ‹*coucou*›,...
- *Antonymie* (*Anti*) :
  Anti(certitude) =‹*incertitude*›, ‹*doute*›, ‹*scepticisme*›, ...
- *Générique* (*Gener*) : Les génériques sont les équivalents des hyperonymes.
  Gener(‹*rose*›) = ‹*fleur*› ; *Gener*(‹*chat*›) =‹*animal*›, ...
- *Dérivés syntaxiques* : Ces fonctions associent à un item sa contrepartie nominale (Substantification S_0), verbale (verbalisation V_0), adjectivale (adjectivisation A_0) ou adverbiale (Adv_0) :
  S_0 (‹*tuer*›) = ‹*meurtre*›, S_0 (‹*vivre*›) = ‹*vie*›, V_0(‹*serment*›) = ‹*jurer*›, S_0(‹*jurer*›)=‹*serment*›

Les fonctions lexicales peuvent être indicées par des opérateurs ensemblistes pour indiquer des nuances de sens. Ainsi, on trouvera :

- $\subset$ pour indiquer une inclusion du sens de l'argument dans la valeur de la fonction. On trouve ce cas avec les rapports d'hyperonymie : $Syn_{\subset}$(‹*pigeon*›) =‹*oiseau*› ;
- $\supset$ pour indiquer une inclusion du sens de la valeur dans l'argument de la fonction : $Syn_{\supset}$(‹*oiseau*›) =‹*pigeon*› ;
- $\cap$ pour indiquer une intersection de sens. Ainsi $Syn_{\cap}$(‹*jouer*›) =‹*s'amuser*› puisqu'on peut jouer sans s'amuser et s'amuser sans jouer.

Fonctions lexicales syntagmatiques : collocations Dans toutes les langues, certaines combinaisons d'items lexicaux prévalent sur d'autres sans qu'il ne semble n'y avoir de motif logique. Par exemple, on parle de « *dormir profondément* » plutôt que de *« *dormir intensément* » ou *« *dormir totalement* » pourtant aucune raison (du moins en synchronie) ne semble expliquer cette préférence. On parle, dans ces cas, de phénomène de *collocation*.

L'énoncé *AB* (ou *BA*) formé des items lexicaux *A* et *B* est une collocation si, pour produire cette expression, le locuteur sélectionne *A* librement d'après son sens alors qu'il sélectionne *B* pour exprimer un autre sens en fonction de *A* ([Polguère, 2003], p. 134). On appelle *A* *base de la collocation* et *B* *collocatif*. On peut citer comme exemples de collocations en français : ‹*tir*›$_{[=A]}$ ‹*nourri*›$_{[=B]}$, ‹*peur*›$_{[=A]}$ ‹*bleue*›$_{[=B]}$, ‹*forte*›$_{[=B]}$ ‹*fièvre*›$_{[=A]}$,‹*dormir*›$_{[=A]}$ ‹*profondément*›$_{[=B]}$.

Les fonctions lexicales paradigmatiques ont été créées pour rendre compte des collocations non seulement dans le rôle syntaxique que joue le collocatif auprès de la base mais aussi par le sens qu'il exprime. Parmi les fonctions lexicales syntagmatiques, on peut citer :

- *Bon* qui marque une évaluation positive : *Bon*(‹*choix*›) = ‹*bon*› ;

– *Magn* qui marque l'intensification : *Magn*(‹*majorité*›) =‹*forte*›, ‹*écrasante*›.
ou leurs opposés :
– *AntiBon* qui marque une évaluation négative : *AntiBon*(‹*choix*›) = ‹*mauvais*› ;
– *AntiMagn* qui marque une modification inverse à l'intensification : *AntiMagn*(‹*majorité*›) =‹*courte*›, ‹*faible*›.

1.3.2 Réseaux sémantiques

Les réseaux sémantiques tirent leur origine de la psychologie expérimentale et plus particulièrement des travaux concernant l'organisation mentale des concepts.

1.3.2.1 Origines

À la fin des années 1960, Alan Collins et Ross Quillian observent que, pour un être humain, le temps d'estimation de la validité d'une phrase comme « *un chien est un mammifère* » est plus long que celui d'une phrase comme « *un chien est un animal* » [Collins & Quillian, 1969] [Collins & Quillian, 1970]. Cette différence dans le délai de réaction semble liée au nombre d'individus de la classe [Juola & Atkinson, 1971] [Landauer & Freedman, 1968] comme le montre l'expérience présentée dans la figure 1.6.

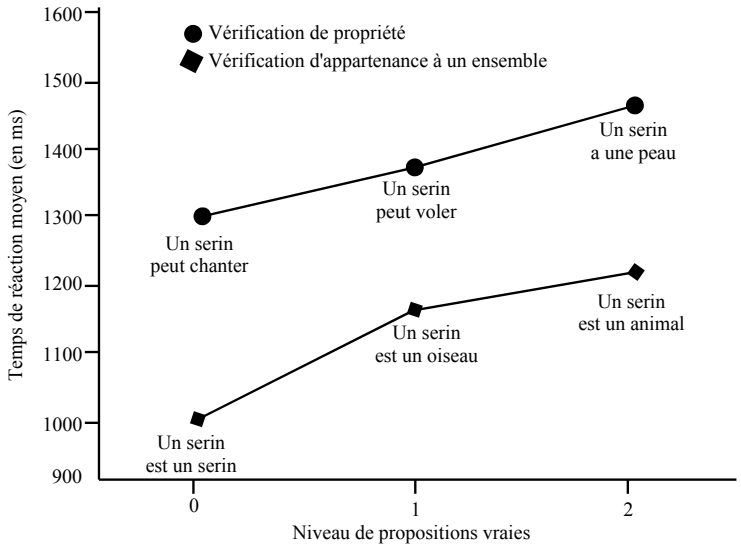


FIG. 1.6 – Vérification d'énoncés sémantiques : résultats expérimentaux [Collins & Quillian, 1969]

Ces expériences semblent montrer que les informations associées à certains concepts sont transmissibles aux concepts hyponymes et ne sont pas mémorisées directement avec ceux-ci. En d'autres termes, les concepts spécialisés héritent des propriétés et des attributs des concepts plus généraux auxquels ils sont associés et y adjoignent leurs propres propriétés et attributs. Ainsi, il s'agit d'un *principe d'économie* reposant sur la mise en facteur des connaissances communes à plusieurs sortes d'objets [Bernard, 2000] [Sabah, 1988]. Par exemple, ‹*mammifère*› met en facteur les propriétés et les attributs communs aux différentes espèces qui le composent (‹*Homme*›, ‹*chien*›, ‹*lapin*›,...) et qui en retour en héritent. Le figure 1.7 présente l'automate de Quillian qui est un exemple de hiérarchie centrée autour d'‹*animal*›.

Plusieurs expériences dont celle d'Harold Goodglass (*1920 - 2002*) et Errol Baker concernant l'aphasie[41] corroborent cette thèse [Goodglass & Baker, 1976]. Pour les sujets de l'expérience,

[41] L'aphasie est une perte totale ou partielle du langage consécutive à une atteinte cérébrale. Les personnes

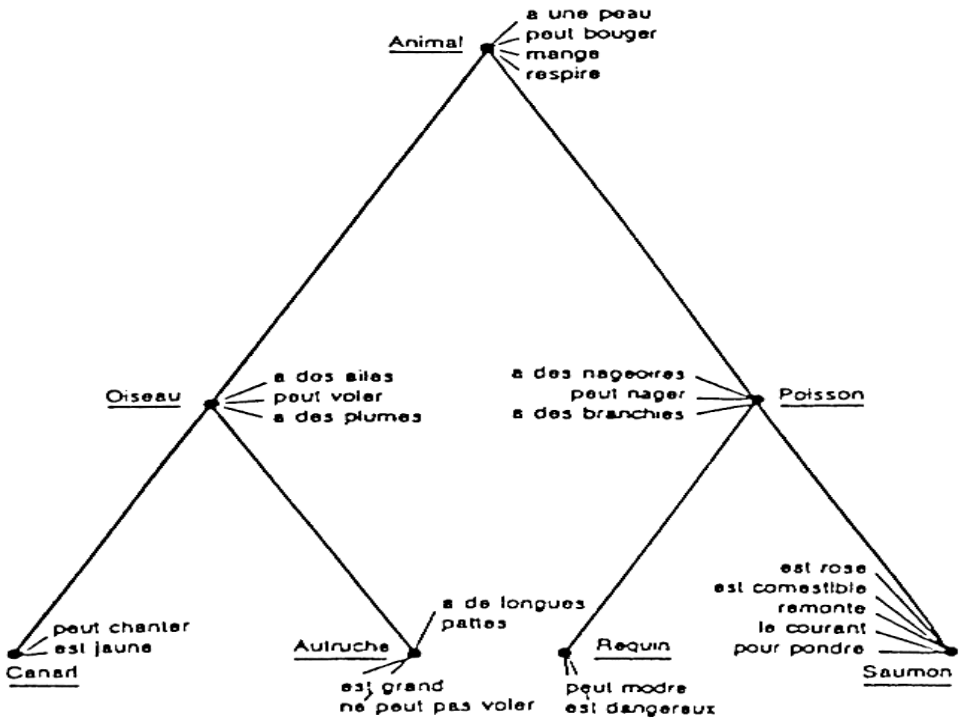


FIG. 1.7 – L'automate de Quillian : hiérarchie centrée autour d'‹*animal*›.

il s'agit d'associer ou non à un terme cible des termes cités oralement. Goodglass et Baker constatent que suivant les relations existant entre les termes proposés (similarité, hyponymie, pragmatique) et plus particulièrement le nombre et la nature des attributs sémantiques que partagent les termes, les temps de réponse diffèrent.

Les travaux de Ross Quillian [Quillian, 1968] proposent les réseaux sémantiques comme un modèle de cette mémoire associative.

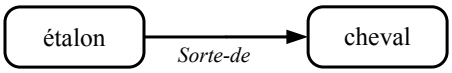
1.3.2.2 Modèle

Notions de base Un réseau sémantique est une représentation des connaissances sous forme de graphe orienté étiqueté. Comme le dit François Rastier, « *La valeur de connaissance d'un réseau ne réside ni dans ses nœuds, ni dans ses liens, mais dans l'interrelation de ses constituants.* » [Rastier, 2004]. Les nœuds correspondent ainsi aux concepts et les arcs, aux relations entre ces concepts. La relation typique, celle de taxonomie, est la relation *Sorte-de*. Par exemple, pour représenter l'existence d'un *étalon* nommé *Tornado*, on ajoute simplement un nœud au réseau (cf. 1.8).

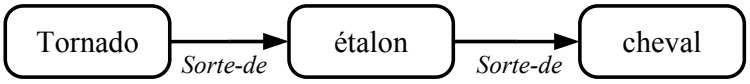
On constate sur l'exemple 1.8 que la représentation des deux premières informations (« *Un étalon est un cheval* » et « *Tornado est un étalon* ») permet de déduire facilement par transitivité que « *Tornado est un cheval* ».

Composition de relations L'exemple précédent présente une composition entre deux relations. Il est possible de retrouver les informations contenues dans le graphe par simple *héritage des propriétés* en suivant les arcs *Sorte-de*. Cette composition des relations est la transposition

aphasiques ne peuvent plus (ou avec difficulté) parler, comprendre, lire et écrire. http://membres.lycos.fr/aphasie/



(a) Ce réseau représente « *un étalon est un cheval.* ».



(b) On rajoute un nœud au réseau pour indiquer que Tornado est un étalon

FIG. 1.8 – Réseaux sémantiques élémentaires.

dans les réseaux sémantiques du principe d'économie abordé à la section 1.3.2, il permet de réaliser une économie d'espace mémoire pour la représentation du réseau mais ralentit le temps de traitement. Pour Quillian, cognitivement, la longueur du chemin reliant deux nœuds doit refléter le temps mis par des humains pour associer les concepts correspondant à ces nœuds et leurs propriétés. Le réseau de la figure 1.9 permet de retrouver que les *étalons* en général et *Tornado* en particulier possèdent des *sabots*.

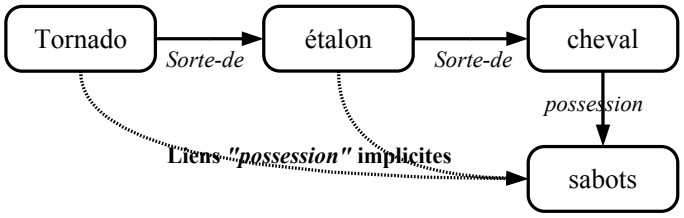
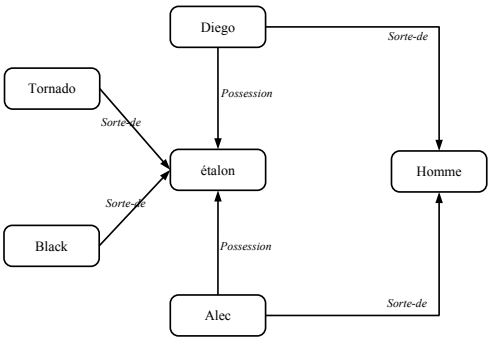
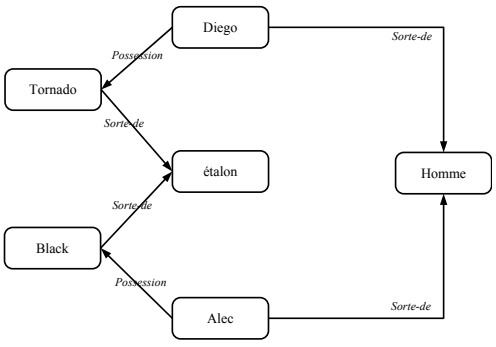


FIG. 1.9 – Héritage de propriétés

Tous les raisonnements sur le réseau peuvent être modélisés par une *table de composition des relations* qui contient l'ensemble des compositions autorisées dans le réseau et leurs relations résultantes respectives. Par exemple, le réseau de la figure 1.9 contient la propriété $Sorte\text{-}de \circ possession = possession$.



(a) Représentation erronée des informations dans un réseau



(b) Représentation correcte des informations de la figure

FIG. 1.10 – Importance de la différenciation des types de nœuds dans un réseau sémantique

Types de nœud Représenter une information comme « *Diego possède un étalon* » peut poser le problème de l'arc à introduire dans le réseau. Il ne faut pas comme sur le réseau de la

figure 1.10(a) mettre un arc de *Possession* entre les nœuds *Diego* et *cheval* puisque cela signifierait que l'ensemble des éléments de la catégorie *cheval* appartient à *Diego*. La représentation correcte verrait un arc entre un propriétaire et une instance de la classe *cheval*, dans l'exemple 1.10(b) pour *Diego*, *Tornado* et pour *Alec*, *Black*. Il est donc nécessaire de différencier les nœuds d'instance des nœuds de classes.

On veut aussi pouvoir représenter d'autres informations comme des informations temporelles (« *Alec possède Black d'octobre 1941 à août 1950* ») auquel cas l'état n'est plus simplement codé par un lien mais aussi par un nœud. Ce nœud sera alors une spécialisation de la notion d'*appartenance* comme le montre la figure 1.11

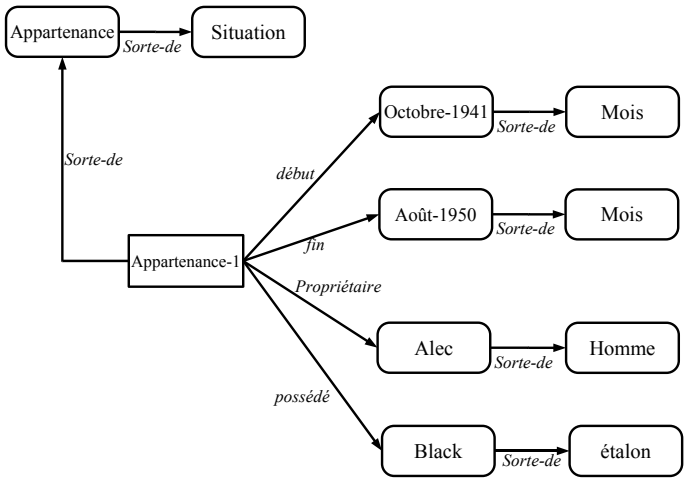


FIG. 1.11 – Représentation de l'information « *Alec possède Black d'octobre 1941 à août 1950* »

Dans la famille des réseaux sémantiques, il convient de citer les *graphes conceptuels* (GC) [Sowa, 1984] [Sowa, 2000][42]. Un GC est un graphe biparti étiqueté dont les deux classes de sommets correspondent à des *concepts* et des *relations conceptuelles* entre ces concepts. La figure 1.12 présente un exemple de graphe conceptuel avec la phrase « *John va à Boston en bus.* ». L'avantage principal qu'offrent les GC est que le modèle est muni d'une sémantique en logique du premier ordre qui est adéquate et complète par rapport à la déduction. Les applications des graphes conceptuels concernent, entre autres, la génération automatique de langage [Nogier, 1991] ou la recherche d'informations [Genest, 2000].

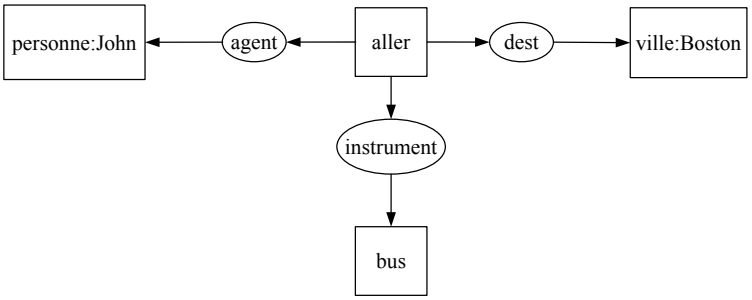


FIG. 1.12 – Exemple de graphe conceptuel : « *John va à Boston en bus.* » [Sowa, 2000]

[42]Une pré-version de cet ouvrage est disponible en ligne à l'adresse http://www.jfsowa.com/krbook/index.htm

1.3.2.3 Les réseaux d'aujourd'hui : WordNet

Contrairement aux réseaux sémantiques classiques qui cherchent à la fois à représenter des phrases et à ordonner les connaissances sur le monde (le côté ontologique des réseaux), le projet WordNet est uniquement concentré sur cette deuxième tâche. Libre ensuite aux développeurs de l'utiliser pour d'autres applications. WordNet est une base de données lexicale pour l'anglais développée sous la direction de George Armitage Miller (*né en 1920*) par le *Cognitive Science Laboratory* de l'université de Princeton (États-Unis d'Amérique). Il se veut représentatif du fonctionnement de l'accès au lexique mental humain.

WordNet est organisé en ensembles de synonymes appelés synsets. À chaque synset correspond un concept. Le sens des termes est décrit dans WordNet par trois moyens :

- leur *définition*
- le *synset* auquel ce sens est rattaché.
- les *relations lexicales* qui unissent entre eux les synsets. Ces relations sont ici l'hyperonymie, la méronymie ainsi que l'antonymie.

La version 2 de WordNet compte 152059 termes ce qui constitue une couverture relativement large de la langue anglaise. Les relations lexicales présentes dans WordNet ne connectent que les termes de même morphologie (il n'y a pas de relations comme la substantification S_0 cf. 1.3.1.2). Il y a donc une hiérarchie pour les noms, une pour les adjectifs, une pour les verbes et enfin une dernière pour les adverbes. Un extrait de la hiérarchie des noms est présenté dans la figure 1.13.

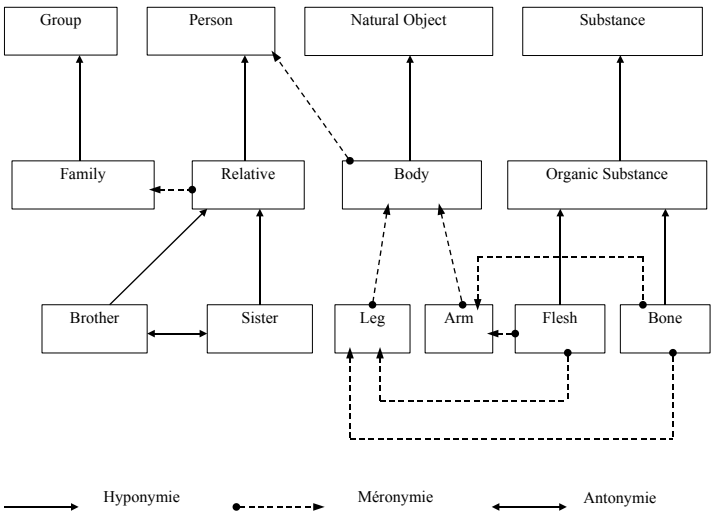


FIG. 1.13 – Extrait de la hiérarchie des noms

1.3.2.4 Limites des réseaux sémantiques

L'une des principales critiques adressées aux réseaux sémantiques concerne le modèle de la mémoire associative dont ils sont issus. En effet, diverses études ont montré que certains hyponymes sont plus caractéristiques de leur catégorie que d'autres. Par exemple, une ‹*pomme*› ou une ‹*orange*› sont plus considérées comme appartenant au genre ‹*fruit*› qu'une ‹*noix*› ou une ‹*olive*› si on se réfère aux temps de réaction des sujets. Par ailleurs, Carol Conrad [Conrad, 1972] a montré que les temps de réaction des sujets à des énoncés n'étaient pas seulement fonction du nombre d'individus de la classe et du parcours d'un réseau d'hyponymes, mais aussi de la fréquence des énoncés (cf. figure 1.14).

Ces remarques sont à la base de la *sémantique du prototype* dont les tenants considèrent que chaque classe contient un prototype, c'est-à-dire un élément plus "typique" que les autres (la

niveau	fréquent	rare
1	« *Un requin peut bouger* »	« *un saumon a une bouche* »
2	« *Un oiseau peut bouger* »	« *Un poisson a des yeux* »
3	« *Un animal peut bouger* »	« *Un animal a une peau* »

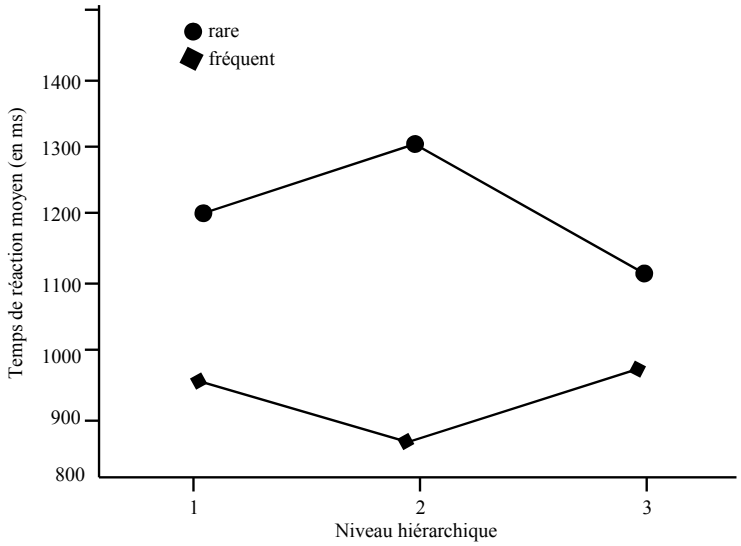


FIG. 1.14 – Expérience de Conrad [Conrad, 1972]

‹*pomme*› pour les fruits par exemple) ; en d'autres termes, qu'il présente dans sa catégorie « *à la fois un maximum de points communs avec les autres éléments de la catégorie et un minimum de points communs avec les éléments de catégories opposées* » [Nyckees, 1998].

Comme nous l'avons dit dans la partie 1.1.2, l'adéquation avec le modèle cognitif ne suffit pas à justifier que ce modèle informatique est insuffisant. Celui-ci est aussi l'objet de critiques de fond, en particulier de la part de François Rastier [Rastier, 2004]. Il lui reproche de n'être qu'une vision du monde. « *Ce qu'on appelle le mobilier ontologique du monde, ce qui est présenté comme naturel et dit par toutes les langues appartient en réalité à un certain type de civilisation* ». Certaines langues distinguent ‹*pied*› et ‹*jambe*›, ce que les langues slaves ou malaises ne font pas.

Les réseaux sont utilisés comme si les sens étaient prédéterminés or ce n'est pas le cas. Il y a des coutumes d'usage mais les langues évoluent et ses coutumes aussi. Certains termes sont créés (néologismes), d'autres prennent de nouvelles acceptions, certaines de ces formations devenant plus fréquentes se verront lexicaliser et finalement figurer dans un dictionnaire. Dans le cas des réseaux, « *on fixe (la langue), on la bloque dans un moment, dans une forme de civilisation ou du moins dans une forme de système économico-technique et on dit voilà ce que c'est que l'esprit humain. Ça s'appelle une prise de pouvoir.* » [43]

1.3.3 Bases d'acceptions

Le modèle de base d'acceptions a été développé à Grenoble depuis le début des années 1990 par Gilles Sérasset et Christian Boitet . Aujourd'hui, sa principale implémentation concerne le projet Papillon dont elle constitue la macrostructure. Ce projet, mené depuis 2000, vise à la constitution d'une base lexicale multilingue linguistiquement riche. Du côté organisationnel, son principal atout est de se baser sur un principe collaboratif qui permet à n'importe quelle

[43] Extrait de l'émission *Tire ta langue* du 18 Février 2003 sur *France Culture* http://www.radiofrance.fr/chaines/france-culture2/emissions/tire_langue/fiche.php?diffusion_id=11608

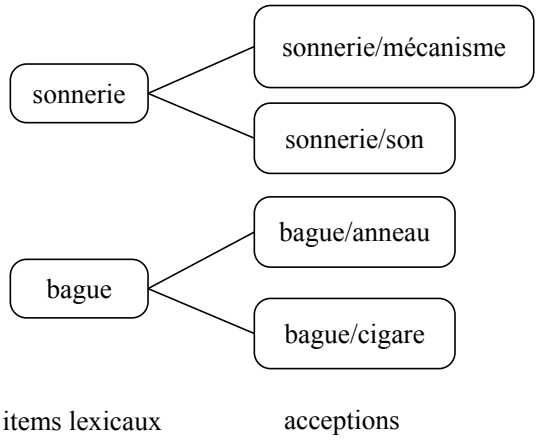


FIG. 1.15 – Items lexicaux et acceptions de ‹*bague*› et ‹*sonnerie*›

personne qui le souhaite de l'enrichir [Mangeot-Lerebours *et al.*, 2003]. La base comprend entre autres l'anglais, le français, le japonais, le malais, le lao, le thaï, le vietnamien et le chinois.

1.3.3.1 Acceptions

Une acception est un sens particulier d'un mot, admis et reconnu par l'usage. Il s'agit d'une unité sémantique propre à une langue donnée [Sérasset & Mangeot, 2001]. Par exemple, en français, l'item lexical ‹*bague*› a, au moins, deux acceptions, l'**anneau**, ou la collerette de papier entourant le cigare (annotée **cigare**) tandis que l'item lexical ‹*sonnerie*› en a deux, le **son** et le **mécanisme** qui l'émet. Une acception est en fait ce qu'on appelle communément « *un sens d'un mot* ». Ainsi, pour revenir à l'une de nos préoccupations principales, désambiguïser, c'est trouver quelle est l'acception d'un mot qui semble concorder le mieux avec les autres mots de la phrase.

1.3.3.2 Base d'acceptions

Une base d'acceptions monolingue contient des entités ITEM LEXICAL et des entités ACCEPTION qui regroupent les informations sur les différents sens que peut prendre l'item. Dans le cas du dictionnaire Papillon, par exemple, cette microstructure est basée sur la Lexicographie Explicative et Combinatoire, partie de la Théorie Sens-Texte (cf 1.3.1.2).

La figure 1.15 présente l'architecture d'une base par acception avec ‹*bague*› et ‹*sonnerie*› dans un cadre monolingue. Les acceptions ont été annotées pour faciliter la compréhension.

Dans un cadre multilingue [Mangeot-Lerebours *et al.*, 2003], une base contient plusieurs bases d'acceptions monolingues dont les acceptions sont reliées par des acceptions interlingues appelées *axies*. Un exemple d'une telle architecture est présenté en 1.16. D'un côté les acceptions monolingues du français, de l'autre celles de l'anglais reliées entre elles par des axies.

Ce système permet de représenter les raffinements de sens de certaines langues. L'exemple de ‹*fleuve*› et ‹*rivière*› est caractéristique. Le français différencie, en effet, « *un cours d'eau qui se jette dans la mer ou l'océan.* » (‹*fleuve*›) et « *cours d'eau qui se jette dans un autre cours d'eau* » (‹*rivière*›) tandis que ni l'anglais ni l'espagnol ne font cette distinction (cf.figure 1.17).

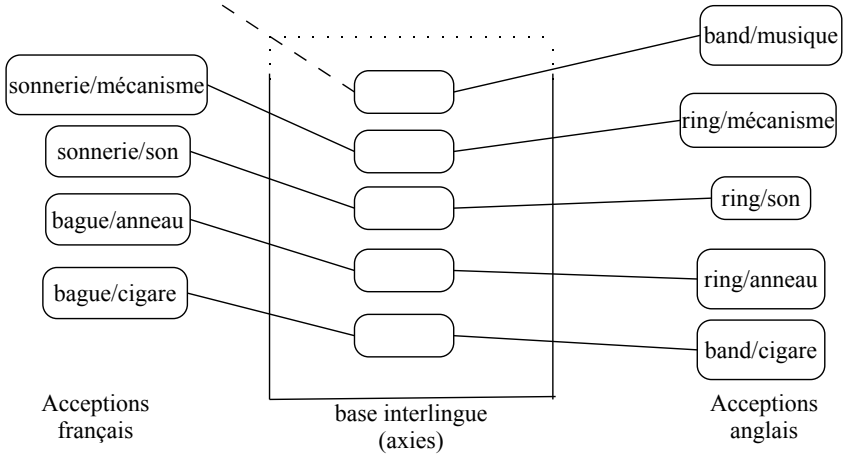


FIG. 1.16 – Acceptions et axies en multilingue

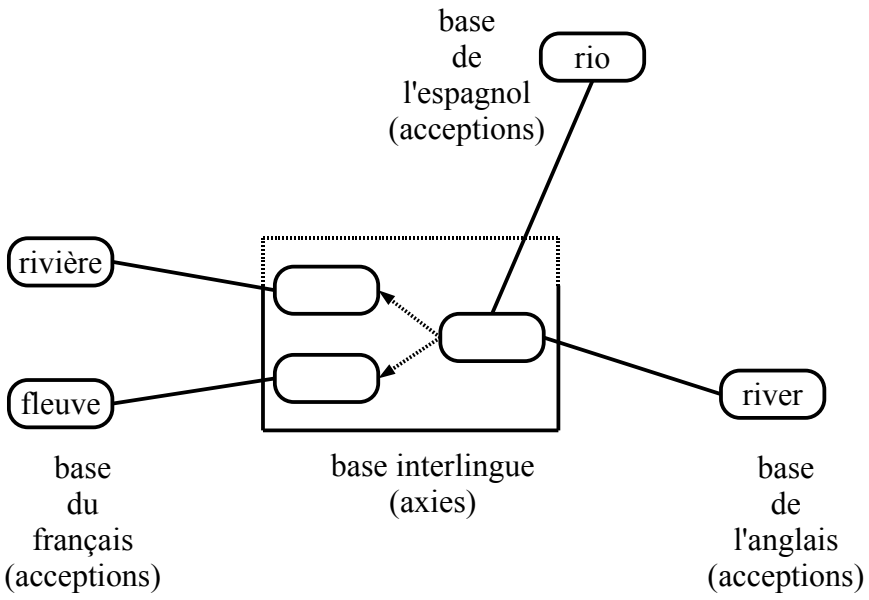


FIG. 1.17 – Exemple de raffinement de sens.

1.4 Approche componentielle (ou sémique)

1.4.1 Le sens vu comme la composition de primitives

La linguistique componentielle postule que le sens d'un terme peut être défini par un ensemble de primitives de base. Cette idée est le prolongement des réflexions de Leibniz (*1646 - 1716*) qui a passé une partie de sa vie à la recherche d'un *alphabet des pensées*. Si on pense qu'il peut exister un tel alphabet, il doit en exister nécessairement un qui permettrait de représenter les mots qui ne sont, après tout, que des étiquettes accolées à certaines pensées. Les structuralistes, en particulier les linguistes héritiers de Leibniz comme Hjelmslev, Pottier, Greimas ou Rastier, s'inspirent à

la fois de ces idées et des théories de la phonologie pour mettre au point l'analyse sémique et la théorie des primitives sémantiques qui en est une conséquence directe. Prenant la suite de Leibniz, la linguiste Wierbicka a étudié de nombreuses langues à la recherche de ces primitives, les informaticiens Wilks et Schank ont essayé de construire de manière moins universelle un ensemble d'atomes de sens qui permettrait à un système informatique de représenter les sens exprimables en langue.

1.4.1.1 Origine de l'approche componentielle : l'analyse sémique

Au début des années 1940, l'étude de la sémantique a pris énormément de retard sur les autres branches de la linguistique ([Nyckees, 1998], p. 206). Les significations des mots sont encore expliquées uniquement par le rapport au monde que chacune d'entre elles entretient. La phonologie, en revanche, est descendue bien en dessous du mot en le décomposant en unités phoniques plus petites, les *phonèmes* eux-mêmes descriptibles suivant la zone de la bouche ou le mode d'articulation qu'il faut utiliser pour les produire. L'étude de la structure des phonèmes a permis de montrer qu'ils pouvaient être considérés comme un faisceau de *traits distinctifs*. Ainsi, si on compare /p/ /b/ et /m/ , il se rejoignent sur le trait de la bilabialité (utilisation des deux lèvres) mais se distinguent sur le trait sonore (vibration des cordes vocales), et le trait nasal (un + dénote l'existence du trait, un - sa non-existence) ([Nyckees, 1998], p. 207) :

	sonore	nasal
/p/	-	-
/b/	+	-
/m/	+	+

Certains sémanticiens se demandent alors si on ne peut pas extraire des structures similaires pour décrire le sens des items lexicaux. En d'autre termes, peut-on faire une description fine du sens des items lexicaux grâce à une décomposition en unités sémantiques minimales ?

La linguistique componentielle suppose donc l'existence d'une atomisation de la signification ([Le Ny, 1979], p. 122), c'est-à-dire que le sens d'un terme n'est plus considéré comme primitif, mais peut être décomposé en éléments de sens plus petits appelés suivant les diverses écoles : sèmes[44], noèmes, traits sémantiques, atomes de sens, primitives, ...

Contrairement aux phonèmes qui ne sont que quelques dizaines[45], il y a des centaines de milliers voire des millions d'items lexicaux dans une langue. Les sémanticiens n'ont donc pas cherché à décrire l'ensemble du lexique, mais se sont restreints à décrire des termes qui ont un trait sémantique commun fort.

La linguistique componentielle tire son origine des années 1940 et des travaux de Hjelmslev sur l'analyse en composants sémantiques.

Hjelmslev et l'analyse en composants sémantiques Le danois Louis Hjelmslev (*1899 - 1965*) prône dès 1943 la comparaison des termes en fonction des sèmes qui les composent (analyse en composants sémantiques) [Hjelmlev, 1968]. Selon cette théorie, les termes ‹*garçon*›, ‹*fille*›, ‹*homme*› et ‹*femme*› peuvent êtres analysés grâce aux traits sémantiques *ANIMÉ*, *HUMAIN*, *MÂLE*, *FEMELLE*, *ADULTE* ([Éco, 1988], p. 136).

[44] Greimas appelle sèmes ces éléments. Comme bien souvent dans d'autres disciplines, la terminologie utilisée par les auteurs est loin d'être unifiée. Nous le constaterons à plusieurs reprises dans cette thèse. Le terme ‹*sème*› en est un bon exemple et le lecteur, sous peine de confusion dans cette partie, doit en avoir bien conscience. En effet, un sème est un atome de sens chez Greimas mais chez Pottier, et c'est sa définition qui est le plus souvent utilisée, un ‹*sème*› est plutôt une composition de ‹*noèmes*›, le nom des primitives de sens pour Pottier.

[45] En français, on compte, par exemple, 37 phonèmes.

	sur terre	sur rail	deux roues	individuel	payant	4 à 6 personnes	intra-urbain	transport d'objets	transport de personnes
voiture	+	-	-	+	-	+	~	~	+
taxi	+	-	-	~	+	+	~	~	+
autobus	+	-	-	-	+	-	+	~	+
autocar	+	-	-	-	+	-	-	~	+
métro	+	+	-	-	+	-	+	~	+
train	+	+	-	-	+	-	-	~	+
avion	-	-	-	~	+	~	-	~	+
moto	+	-	+	+	-	-	~	~	+
bicyclette	+	-	+	+	-	-	~	~	+

FIG. 1.18 – Analyse sémique des véhicules selon Pottier

‹*garçon*› : *ANIMÉ* + *HUMAIN* + *MÂLE* - *ADULTE*
‹*fille*› : *ANIMÉ* + *HUMAIN* + *FEMELLE* - *ADULTE*
‹*homme*› : *ANIMÉ* + *HUMAIN* + *MÂLE* + *ADULTE*
‹*femme*› : *ANIMÉ* + *HUMAIN* + *FEMELLE* + *ADULTE*

Certains traits sémantiques, comme ici *ANIMÉ*, sont considérés afin de justifier la concordance des termes avec certains verbes. On estime que les phrases « *L'homme respire.* » ou « *Le chien respire.* » sont grammaticalement justes parce que ‹*respirer*› s'adapte positivement au trait *ANIMÉ* et indifféremment à *HUMAIN* ou *ANIMAL*. Par contre, on considère qu'il est incorrect de dire *« *Le rocher respire* » (si on exclut les sens métaphoriques) parce que ‹*respirer*› ne peut concerner que des *ANIMÉS*. Ces valences combinatoires du verbe sont dites *restrictions sélectives*. Bien qu'elle ait donné des résultats intéressants, l'analyse en traits sémantiques sert davantage à expliquer les concordances grammaticales qu'à expliquer les concordances sémantiques.

Éco (*né en 1932*) formule deux objections à cette démarche ([Éco, 1988], p. 137).

1. Tandis qu'il est possible d'organiser en système les catégories grammaticales du fait de leur nombre restreint, il est difficile d'organiser les catégories sémantiques qui sont bien plus nombreuses.
2. Le grand nombre de ces catégories fait que s'il est aisé de définir ‹*homme*› par rapport à ‹*femme*›, il l'est moins pour ‹*vache*› en fonction de ‹*brebis*›. Il s'agit dans les deux cas d'animés, d'animaux et de femelles et pourtant il ne s'agit pas de la même chose.

L'analyse sémique de Pottier Parmi les analyses sémiques les plus connues figurent celles effectuées par Bernard Pottier [Pottier, 1964]. La figure 1.18 présente l'analyse sémique de certains véhicules. Les signes + et - marquent la présence ou l'absence du trait tandis que ~ spécifie que le trait est indifférent.

Dans cet exemple, chaque ligne représente un sémème, c'est-à-dire l'ensemble des sèmes que le mot comporte. Le seul sème qui appartienne à tous les mots est celui de la dernière colonne, *TRANSPORT DE PERSONNES*. Il constitue aussi, à lui seul, le sémème de ‹*véhicule*›, item lexical qui peut s'appliquer à tous les objets dénommés par les autres termes de la liste et qui est donc par rapport à eux, l'hyperonyme le plus proche. On l'appelle *archisémène*.

Tout comme pour une analyse phonologique, il s'agit ici de chercher des traits distinctifs entre les items. Ceux sont uniquement eux qui sont notés dans le tableau (à l'exception de l'archisème). Les sèmes constituent donc l'ensemble minimal permettant de différencier les items étudiés entre eux.

Par la suite, l'analyse sémique a été reprise, en psycholinguistique, par Jean-François Le Ny [Le Ny, 1979] et, en linguistique par François Rastier [Rastier, 1989] et Algirdas Julien Grei-

mas [Greimas, 1986] mais, pour ce dernier, avec une définition du terme ‹*sème*› plus proche d'‹*atome de sens*› que de ‹*trait distinctif*›.

1.4.1.2 Les primitives de sens

L'analyse sémique s'attache à identifier pour un certain nombre de termes leur sémène, c'est-à-dire l'ensemble des sèmes qu'ils comportent. Même si ces sèmes ne sont pas des atomes de sens[46], mais des traits distinctifs, ils supposent l'existence dans l'esprit humain de primitives de sens, les sèmes n'en étant alors que des compositions s'opposant entre elles. Ainsi, contrairement à la distributionnalité présentée en 1.2.1 qui est une théorie purement linguistique et qui donc ne repose pas sur un postulat cognitif, il s'agit ici de comprendre comment les mots coexistent dans notre esprit ([Nyckees, 1998], p. 216).

Les primitives de sens doivent permettre d'exprimer la signification de tout énoncé quelle que soit la langue dans laquelle il est exprimé. Ainsi, dans la théorie atomiste, les primitives sémantiques sont nécessairement universelles et surtout elles sont à la fois *indépendantes* et *antérieures* au langage. Ainsi, elles devraient nécessairement se retrouver présentes dans toutes les langues.

La recherche des primitives

Chez les linguistes À la suite d'études approfondies sur les langues les plus diverses durant presque trente ans, Anna Wierbicka a proposé en 1992 une liste de 35 primitifs universaux présentée par la figure 1.19.

‹*je*›, ‹*tu*›, ‹*quelqu'un*›, ‹*quelque chose*›, ‹*on*› (ou ‹*les gens*›),
‹*penser*›, ‹*savoir*›, ‹*dire*›, ‹*éprouver*›, ‹*vouloir*›,
‹*ceci*›, ‹*le même*›, ‹*autre*›, ‹*un*›, ‹*deux*›, ‹*tous*›, ‹*beaucoup*›,
‹*faire*›, ‹*arriver à/dans*›,
‹*ne pas vouloir*› (ou ‹*non !*›), ‹*si*›, ‹*pouvoir*› (ou ‹*peut-être*›), ‹*comme*›, ‹*à cause de*›, ‹*près*› (temps et lieu), ‹*quand*› (ou ‹*temps*›), ‹*où*› (ou ‹*endroit*›), ‹*après*› (ou ‹*avant*›), ‹*au-dessous de*› (ou ‹*au-dessus de*›),
‹*avoir*› (des parties), ‹*différentes espèces*›,
‹*bon*›, ‹*mauvais*›, ‹*grand*›, ‹*petit*›

FIG. 1.19 – Les 35 primitifs sémantiques d'Anna Wierzbicka [Wierzbicka, 1993]

Ces primitifs, selon elle, « *(. . .) sont censés être des universaux d'ordre lexical ; il est présumé qu'ils sont lexicalisés dans toutes les langues du monde. Les recherches en sémantique transculturelle semblent indiquer qu'il y a un nombre plutôt restreint de concepts qui sont bel et bien universels (et probablement innés). L'hypothèse qu'il n'y en aurait qu'une douzaine s'est révélée incorrecte ; il faut multiplier ce chiffre par trois. De toute façon, le nombre de primitifs ne s'élève pas dans les milliers, ni même dans les centaines.* »

Le principal problème posé par cette liste est que, là où on simplifie les choses en présentant un nombre fort restreint de primitives, on augmente singulièrement la difficulté de représenter le sens d'un terme. En effet, comment représenter avec ces primitifs ‹*voiture*›, ‹*lit*› ou ‹*dormir*› ? Dans l'hypothèse où ces primitifs seraient réellement à la base de toute la pensée humaine, il est difficile (impossible ?) de retracer le cheminement parcouru pour arriver au sens de ces termes.

Un deuxième problème concernant cette vision de Wierzbicka est son caractère utopique. En effet, la notion ne rentre dans les primitifs que si elle est universelle donc si elle est présente dans

[46] Toujours en adoptant la terminologie de Pottier mais pas celle de Greimas.

classe	nombre	exemples
entités	19	*HUMANITÉ*, *SUBSTANCE*, *OBJET PHYSIQUE*
actions	34	*CAUSER*, *COULER*, *FRAPPER*, *ÊTRE*
cas	19	*VERS*, *DANS*, *AGENT*, *LIEU*
qualificatifs	16	*BON*, *CONTENANT*
indicateurs de type	2	*QUALITÉ*, *MANIÈRE*

FIG. 1.20 – Quelques-uns des éléments primitifs de Wilks

l'ensemble des langues du monde. Or du fait de leur grand nombre, rappelons-le, plus de 6000, cette étude semble particulièrement difficile à réaliser. Pour cette raison purement pratique, on ne pourra donc jamais être totalement certain de l'universalité d'un concept.

Chez les informaticiens Les créateurs des systèmes informatiques des années 1970 comme Schank [Schank, 1972] et Wilks sont directement héritiers de l'analyse sémique et à ce titre, ils cherchent principalement quelles primitives permettraient de représenter l'ensemble des sens en langue. Ainsi, Yorick Wilks énonce quelques critères très généraux pour fabriquer un ensemble de primitives [Wilks, 1977] :

1. *finitude* : l'ensemble de primitives doit être fini et de relativement faible dimension. En particulier, cette dernière doit être très largement inférieure au nombre de sens à décrire ;
2. *étendue* : les primitives doivent couvrir l'ensemble de l'intervalle des sens à exprimer ;
3. *complétude* : toutes les informations sur le sens d'une entité doivent pouvoir être décrites grâce à l'ensemble de primitives ;
4. *canonicité* : la description d'une entité doit être unique et non-ambiguë ;
5. *indépendance* : aucune primitive ne doit pouvoir être décomposable en un ensemble d'autres ;
6. *non-réductibilité* : l'ensemble de primitives ne peut être remplacé par un ensemble plus petit.

Ces recherches ont fait l'objet de nombreuses discussions tout au long des années 1970 [Winograd, 1978] et jusqu'aux années 1980. Les systèmes basés sur ces primitives étaient lourds et les résultats loin d'être satisfaisants. Les critères proposés pour construire ces listes sont souvent jugés trop généraux pour être utiles mais comme le note [Sabah, 1996], « *les tentatives de réfutation n'ont pas apporté d'idées beaucoup plus constructives en tout cas pour les mises en oeuvre informatiques* ».

1.4.1.3 Le problème de l'antériorité et de l'indépendance au langage

Les tenants de l'approche componentielle considèrent donc que tous les locuteurs humains partagent un ensemble d'atomes de sens et donc que ceux-ci sont alors forcément antérieurs au langage. Pottier présente plusieurs arguments qui sont, selon lui, favorables à ces idées et qui recoupent ceux que nous avons en partie déjà constatés dans les parties précédentes :

- *traductions* : il semble possible d'effectuer des traductions entre tout couple de langues, du français au chinois, du chinois à l'égyptien,... Il doit ainsi exister un espace conceptuel hors langage commun à l'ensemble de l'humanité qui permet le passage d'une langue à une autre ;
- *acquisition des informations* : en France pour la plupart des gens l'année *1515* évoque la bataille de Marignan tout comme *1789* évoque le début de la Révolution française. Pourtant, qui se souvient exactement où, quand et comment il a appris ces dates ? Les a-t-on lues, entendues ? Au mieux, on croit se souvenir l'avoir appris à l'école mais rien

n'est vraiment sûr. Pourtant, on a retenu ces notions. Il semble donc exister un niveau conceptuel indépendant du langage.

L'argument de la traduction est toutefois très contestable. En effet, la traduction est en grande partie le fruit de compromis des traducteurs. Certains concepts issus de la culture et de l'environnement des locuteurs sont présents dans des langues et ne le sont pas dans d'autres. Il s'agit donc pour un traducteur de chercher dans l'autre langue comment exprimer le mieux possible les idées d'un énoncé.

L'antériorité et surtout l'indépendance sont aussi largement contestables. L'évolution culturelle de l'Homme s'est fortement accélérée lorsque celui-ci a acquis le langage. Il a été plus à même de transmettre aux générations suivantes comment couper la viande, ce qui était bon, ce qui était dangereux. Les peuples ont acquis des savoirs, acquis des croyances. Ainsi, la plupart des concepts humains se sont trouvés à la fois qualitativement et quantitativement modifiés par l'apparition des langues, et considérablement réorganisés par les échanges entre les êtres humains ([Nyckees, 1998], p. 220).

1.4.2 Le Dictionnaire Intégral

Le *dictionnaire intégral* est un réseau sémantique développé par la société Memodata[47] depuis plus d'une quinzaine d'années. Il allie à la fois des connaissances relatives à l'approche componentielle et des fonctions lexicales à la Mel'čuk ; sa couverture lexicale est très étendue (comparable à celle de WordNet) ; enfin, les applications visées sont hétérogènes et s'étendent du résumé automatique au filtrage d'information en passant par la comparaison textuelle et la traduction automatique. C'est pour ces caractéristiques, très proches de celles visées par notre équipe, qu'il a paru intéressant de le présenter ici.Dans sa thèse [Dutoit, 2000], Dominique Dutoit co-fondateur de Memodata, présente ce dictionnaire intégral (DI), sa construction et les diverses applications mises en œuvre.

1.4.2.1 Architecture du dictionnaire intégral

Dans ce modèle, la description de chaque sens de mots (appelé *mot-sens*) se fait selon trois points de vue considérés comme complémentaires : la sémantique componentielle, les fonctions lexicales sémantiques et les propositions courantes.

Sémantique componentielle La brique de base du dictionnaire intégral est le concept. Chaque concept est désigné par un identifiant qui commence par un '\' et une majuscule comme, par exemple, *FLEUR* ou *VENDRE*. Les concepts sont organisés en hiérarchie dont les feuilles sont les mots-sens. Selon ses concepteurs [Dutoit, 2000], un concept du dictionnaire intégral est artificiel et peut être conçu selon de très nombreux besoins différents. Ainsi, un concept sert à renseigner de façon non-ambiguë, d'un côté un utilisateur humain sur son contenu (les concepts qui en découlent) et sur son contenant (les concepts dont il découle) d'un autre, la machine sur la façon de l'utiliser (possibilité ou impossibilité de passer d'un concept à un autre).

Il existe deux sortes de concepts :

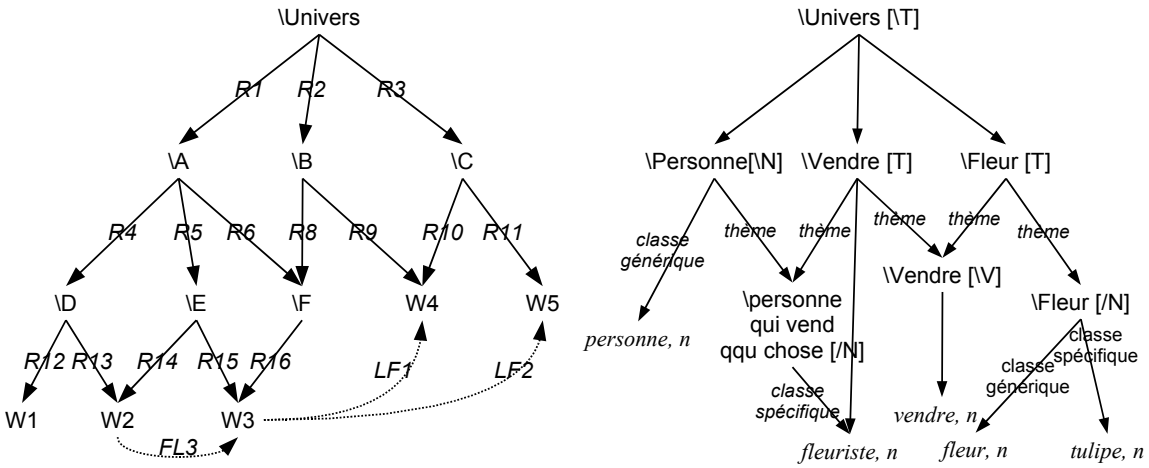
- les *classes* sont situées dans la partie basse de la hiérarchie et correspondent des sous-hiérarchies de concepts de même morphologie (verbes [/V], noms [/N], adjectifs [/A], adverbes) ;
- les *thèmes*, notés ([T]), qui sont situés dans la partie haute de la hiérarchie générale et correspondent au champ sémantique commun à tous les thème et classes successeurs dans la hiérarchie.

[47] http://www.memodata.com/

Propositions courantes Les propositions courantes correspondent aux relations syntaxiques qu'il peut exister entre mots-sens. Ils reprennent les idées adoptées par le DEC en ce qui concerne leur régime (cf. 1.3.1.2) qu'ils adaptent à leur architecture pour éviter en particulier la redondance d'informations. Ainsi alors qu'une relation est notée dans tous les articles du DEC, elle sera mentionnée à part dans le DI.

Fonctions lexicales sémantiques Le DI relie les mots-sens entre eux grâce à des fonctions lexicales sémantiques. Il y en a 66 différentes pour 96 000 liens décrits [Dutoit & Nugues, 2002]. La liste de ces fonctions a été établie à partir de celles de Mel'čuk (cf. 1.3.1.2) mais chacune a été réétudié en fonction des spécificités du dictionnaire intégral :

- *spécificités syntaxiques* : Certaines fonctions lexicales rendent compte de phénomènes sémantiques décrits en partie par la syntaxe. C'est le cas par exemple des compléments circonstanciels. Ainsi, les dérivés sémantiques nominaux circonstanciels (S_{instr}, S_{loc}, S_{med}, S_{mod}, S_{res}) ne sont pas introduits dans le DI comme des relations de nature sémantique mais comme des relations syntaxiques. Ainsi, la fonction S_{instr} n'existe pas dans le DI mais peut être trouvée grâce au complément circonstanciel de moyen ;
- *spécificités sémantiques* : les concepteurs du DI considère que le sens des mots-sens est donné par les sèmes qu'il met en jeu. Sur cette idée, ils n'ont pas de fonction lexicale sémantique de synonymie telle que celle de Mel'čuk qui sont remplacées par le réseau de concepts.



(a) Le graphe de concepts, mots, relations et fonctions lexicales du dictionnaire intégral (extrait de [Dutoit & Nugues, 2002])

(b) Une partie du dictionnaire intégral

FIG. 1.21 – Architecture du dictionnaire intégral.

1.4.2.2 Construction

Cette base est construite manuellement depuis une quinzaine d'années par une équipe de trois personnes[48]. Il contenait en 2002, 16 000 thèmes, 25 000 classes et pour le français 190 000 mots-sens. Il continue actuellement à être construit en particulier vers d'autres langues.

On peut s'interroger sur cette construction manuelle qui certes a le mérite d'offrir une précision manifeste mais est coûteuse en temps. En effet, cette tâche est en grande partie similaire à celle effectuée par les lexicographes pour réaliser un dictionnaire papier. Or, cette tâche leur

[48] Communication personnelle.

a pris des dizaines d'années et doit constamment être renouvelée puisque des termes et des sens apparaissent tandis que d'autres disparaissent. Ainsi n'y avait-il pas quelques possibilité pour fabriquer (au moins en partie) le DI à partir de tels dictionnaires ? Nous verrons dans la suite de la thèse que c'est pour ces raisons que nous avons adopté pour les vecteurs conceptuels un apprentissage permanent à partir de dictionnaires à usage humain préexistants à nos expériences.

1.4.2.3 Applications

La société Memodata développe à partir de ce réseau de nombreuses applications commerciales. L'une des plus anciennes est le dicologique qui est la version du dictionnaire intégral de 1992[49]. Il permet de trouver définitions, synonymes, analogies, rimes, mots-croisés, anagrammes, conjugaisons, féminins et pluriels [Dutoit, 1992]. Ils développent aussi des outils directs (puisque ces données sont stockées telles quelles dans le DI) tel qu'un dictionnaire de synonymes ou un conjugueur ou des outils qui nécessitent des opérations à l'aide du DI comme :

- le *résumé automatique* qui est ici un outil qui relève les mots importants d'un texte c'est-à-dire les mots qui possèdent dans le DI un grand nombre de relations avec les autres ;
- la *classification automatique de documents* et en particulier le *routage de courriels* qui consiste à acheminer les courriers électroniques vers des boîtes prédéfinies en fonction de critères sémantiques ;
- la *comparaison textuelle* qui consiste à comparer les sèmes et les mots-sens contenus dans deux documents ;
- la *traduction automatique* grâce à des liens "*se traduit par*" entre mots-sens issus de langues différentes.

L'ensemble de ces applications est basé sur deux opérations ensemblistes réalisées sur les sèmes. La première, l'*activation*, permet d'évaluer la ressemblance de deux mots-sens c'est-à-dire les sèmes qu'ils ont en commun tandis que la seconde, le *calcul de proximité* permet d'évaluer, elle, à la fois les ressemblances et les différences.

On peut regretter que les informations relationnelles ne soient pas utilisées dans les tâches de désambiguïsation. Nous le verrons plus loin dans ce mémoire, l'utilisation d'informations de nature relationnelle permet de résoudre des ambiguïtés que l'usage d'informations mutuelles telles que le sont l'activation ou la proximité sémantique (ou dans le cas des vecteurs d'idées que nous aborderons dans le chapitre suivant, la distance thématique) soit ne peuvent résoudre soit peuvent largement aider à résoudre. Il s'agit ici d'un des résultats les plus importants de nos travaux.

1.4.3 Une première expérience utilisant des listes préétablies : les proto-vecteurs d'idées.

Au début des années 1990, Jacques Chauché, dans le but de réaliser un système de Traduction Automatique, propose de représenter le "sens"[50] des items lexicaux grâce à un espace vectoriel dont les axes seraient associés à un ensemble de concepts définis *a priori*. Dans une telle expérience, le choix de cet ensemble définit l'espace vectoriel et est donc, par conséquent, très important. Jacques Chauché considère que ce choix doit être « *assez général pour permettre le codage d'un mot quelconque et ne doit pas être construit pour l'expérience afin d'éviter une prédétermination des sens.* ». Il préfère ainsi utiliser une liste de 416 concepts déjà définie par les rédacteurs de l'encyclopédie Universalis pour leur *organum* [Universalis, 1968].

Le sens d'un item est défini comme un vecteur de cet espace. Pour construire un vecteur, il associe au sens à définir un ensemble de concepts proches sémantiquement. Quatre types d'asso-

[49] http://www.memodata.com/2004/fr/dicologique/index.shtml
[50] Dans [Chauché, 1990] Jacques Chauché met lui-même sens entre guillemets.

ciations sont définies : associations fortes, associations faibles et leurs contraires. Ces derniers ont été introduits en prévision d'un traitement futur de l'antonymie, mais finalement ne semblent jamais avoir été employés. Ainsi, si le concept A est opposé au concept B et si la liste des associations fortes positives contient A, celle des associations fortes négatives contiendra B. Les poids choisis sont de 1 pour les associations fortes et de $0,5$ pour les associations faibles. Par exemple, l'item ‹*valeur*› dans son sens de **prix (sens commercial)** noté *valeur*/**prix**, est associé aux concepts suivants :

association forte	:	*prix, commerce*
association faible	:	*monnaie*

Le vecteur de ‹*valeur*› calculé à partir de ces associations est donc $(1;1;0,5)$ dans l'espace vectoriel à trois dimensions qui a pour axes *(prix, commerce, monnaie)*. On voit ici une différence majeure avec la théorie componentielle classique. Alors que celle-ci considère les concepts comme des primitives, des atomes et les utilise donc de manière booléenne (le concept est présent ou non), les proto-vecteurs d'idées ne considèrent pas les concepts comme des atomes et donc permettent de quantifier l'importance du concept, de l'idée, dans le terme.

Dans l'expérience présentée [Chauché, 1990], les associations ont été définies par 6 personnes différentes ce qui a amené à des différences notables.

Ainsi, pour le terme ‹*bilan*›, un premier codeur a choisi :

- *Association forte* : ACCUMULATION, CAPITAL, CONVERGENCE, DÉNOMBREMENT, GESTION ;
- *Association faible* : ASSOCIATION, CONNAISSANCE, HISTOIRE, INDUCTION, INFORMATION, INTÉGRATION-DES-SENS-DATA.

Tandis qu'un second associe lui :

- *Association forte* : AVOIR, CONNAISSANCE, BIEN, DESCRIPTION, INFORMATION, QUANTIFICATION, REPRÉSENTATION ;
- *Associations faible* : CAPITAL, ACCUMULATION, ACQUIS, APPROXIMATION, CRÉDIT, MESSAGE, OBSERVATION, PROPRIÉTÉ, POSSESSION, PREUVE, REFLET, SIGNAL, SOURCE.

Pour comparer les sens entre eux, la distance utilisée est la distance euclidienne. Lors d'une désambiguïsation, le sens choisi sera celui dont le vecteur sera le plus proche des termes de référence. Ainsi, si on compare le sens de ‹*bilan*› présenté ci-dessus avec les différents sens possibles de l'item ‹*cours*›, on obtient :

1. *cours*/**monnaie** : $94,0$
2. *cours*/**durée** : $104,5$
3. *cours*/**déplacement** : $105,5$
4. *cours*/**polycopié** : $106,25$
5. *cours*/**niveau** : $107,5$
6. *cours*/**enseignement** : $108,25$
7. *cours*/**rue** : $108,5$

Le sens choisi dans ce cas est le premier, *cours*/**monnaie**.

Ce modèle vectoriel est le précurseur de celui des vecteurs d'idées que nous utilisons dans cette thèse (d'où le nom de *proto-vecteurs d'idées*). Ces vecteurs d'idées sont basés, eux, sur le thésaurus Larousse, présenté dans la partie 1.4.5, qui offre le double avantage de présenter une liste de concepts présentés comme pouvant décrire l'ensemble des idées contenues en langue ainsi qu'une description des idées contenues dans quelques milliers d'items lexicaux.

1.4.4 Notre vision

Comme nous l'avons vu précédemment (cf. 1.4.1.3), les tenants de l'approche componentielle, considèrent que tous les locuteurs humains partagent un ensemble d'atomes de sens et donc qu'ils sont forcément antérieurs au langage. Toutefois, deux objections peuvent être soulevées. La première, d'ordre cognitif, considère que l'évolution de l'homme et sa différenciation avec les autres espèces animales s'est réellement accélérée du fait de l'invention du langage. La seconde, d'ordre plus pragmatique, concerne la difficulté à trouver une combinaison de primitives de base permettant de représenter le sens d'un terme lorsqu'elles sont trop peu nombreuses et ainsi trop abstraites.

Il ne s'agit pas, pour nous, de formuler des hypothèses sur l'organisation des concepts chez l'humain mais plutôt de chercher à représenter le sens par une méthode à la fois calculable et efficace. Nous préférons ainsi considérer un ensemble de concepts qui ne seraient pas forcément indépendants les uns des autres mais grâce auxquels il serait relativement aisé de définir les sens des termes. Les travaux de Jacques Chauché (cf. 1.4.3) ont montré la faisabilité d'une telle approche.

Ces concepts ne sont pas alors à envisager comme des concepts correspondant à un être humain en particulier mais plutôt comme les concepts fondamentaux d'une société humaine particulière dont les membres partagent un certain nombre de faits culturels. Ils évoluent au cours de l'histoire et de l'acquisition de nouvelles techniques. Des concepts comme ceux concernant le feu ou les outils sont apparus durant la préhistoire, ceux qui concernent les téléphones portables ou les ordinateurs peuvent aujourd'hui être considérés alors qu'ils ne l'auraient pas été il y a cent ou cinquante ans. C'est pour cette raison que nous considérerons qu'un ensemble de primitives de sens ne devrait et ne pourrait être choisi que pour une certaine société humaine à une certaine époque. Il s'agit de considérer un ensemble de concepts permettant de représenter l'ensemble des idées exprimables pour une langue à une époque donnée.

Dans le cadre de ses recherches, l'équipe TALN du LIRMM utilise le formalisme des vecteurs d'idées. Ce modèle, proche dans sa conception de la théorie componentielle, est l'héritier direct de celui des proto-vecteurs d'idées présenté en 1.4.3. Ainsi, il postule que le sens des termes peut être calculé à partir d'un ensemble de concepts. Il va de soi que le problème posé par la recherche d'un ensemble de primitives sémantiques n'est pas l'objectif des recherches menées actuellement par l'équipe au sein du LIRMM. Nous préférons nous reposer sur des thésaurus généraux. Ces thésaurus ont pour but d'organiser les termes du lexique en fonction des idées qu'ils véhiculent. Ainsi, pour un certain nombre de langues, il existe des thésaurus qui décrivent le lexique en fonction d'une classification mise au point pour cet exercice.

1.4.5 Les thésaurus : un exemple, le Larousse

Un thésaurus comporte un ensemble de concepts censés permettre de décrire l'ensemble des idées exprimables en langue (*hypothèse du thésaurus*). Un thésaurus n'est pas à proprement parlé d'inspiration componentielle, puisque les premiers sont antérieurs de près d'un siècle à cette théorie, mais s'en rapprochent fortement. Le but d'un thésaurus est, selon les auteurs de [Larousse, 1992], « *d'explorer à partir d'une idée l'univers des mots qui s'y rattachent et de trouver des idées à partir des mots liées à une notion* ».

Un thésaurus est le résultat d'un long processus de tri des items lexicaux d'une langue donnée. Ce tri conduit à la constitution d'une hiérarchie qui diffère donc suivant les idées importantes dans le vocabulaire de telle ou telle société (donc les idées importantes dans telle ou telle société). Ainsi, le thésaurus du français se différencie de celui de l'anglais beaucoup plus raffiné, par exemple, sur des notions comme celle qui touchent au fait religieux.

De même, les thésaurus s'adaptent aux évolutions de la société. Ainsi, comme les rédacteurs

du thésaurus Roget le notent dans la préface de la version datant de 1987 [Kirkpatrick, 1987] « *Cette version a été rendue nécessaire par l'extension sans précédent du vocabulaire de l'anglais durant les années 1980, qui reflète les principaux changements d'ordre scientifique, culturel ou social. Les découvertes et les inventions dans le monde des sciences, de la médecine et des technologies ont fait apparaître des termes comme acid rain, AIDS, genetic fingerprinting, nuclear winter, ...* [51] ».

Le thésaurus Larousse, que nous allons présenter plus particulièrement ici, est inspiré du thésaurus de Roget paru en Grande-Bretagne au milieu du XIXe siècle [Roget, 1852]. Il est constitué de trois parties : (1) *organisation des idées* qui constitue la hiérarchie du thésaurus ; (2) la partie *thésaurus* proprement dite qui permet à partir d'idées de trouver des mots dans le même thème et (3) la partie *index* qui permet de trouver les idées associées aux mots.

1.4.5.1 La partie *Organisation des idées* : la hiérarchie Larousse

Ce thésaurus est basé sur une classification organisée selon une structure hiérarchique d'arbre qui comporte 5 niveaux :

- niveau 0 : 1 concept (*C0 :OMEGA*), la racine de l'arbre. Il faut noter que ce concept n'existe pas explicitement dans la hiérarchie, nous l'avons rajouté pour disposer d'un véritable arbre hiérarchique.
- niveau 1 : 3 concepts (*C1 :LE MONDE*, *C1 :L'HOMME*, *C1 :LA SOCIÉTÉ*)
- niveau 2 : 26 concepts
- niveau 3 : 95 concepts
- niveau 4 : 873 concepts, les feuilles de l'arbre.

Afin de les distinguer suivant leur niveau de hiérarchie, nous notons ici les concepts par un *c* concaténé au numéro de niveau du concept, à deux points puis enfin au nom du concept. Par exemple, le concept de niveau 0 oméga est noté *C0 :OMEGA*, le concept de niveau 4 existence est noté *C4 :EXISTENCE*. Pour des raisons de clarté, nous omettrons souvent cette convention d'écriture en ce qui concerne les niveau 4 de la hiérarchie (*C4 :EXISTENCE* sera alors noté *EXISTENCE*).

Les concepts de niveau 4 se succèdent, quand cela s'y prête, en fonction des domaines auxquels ils appartiennent par paires de notions proches, corrélatives ou opposées. Nous avons donc des contraires comme *EXISTENCE* (1) et *INEXISTENCE* (2), *HONNEUR* (641) et *DISCRÉDIT* (642), qui se suivent ainsi que des termes proches thématiquement comme *AMOUR* (600) ou *CARESSE* (601). La figure 1.22 présente un extrait de cette hiérarchie. La hiérarchie complète de [Larousse, 1992] se trouve en annexe B.

Selon les auteurs, la hiérarchie du thésaurus « *(couvre) méthodiquement l'ensemble des champs notionnels possibles (de la langue)* ». Ainsi, l'ensemble des concepts de niveau 4 permettrait de définir la globalité des termes du lexique. C'est sur cette hypothèse, que nous appelons *hypothèse du thésaurus*, que reposent nos expérimentations.

1.4.5.2 La partie *Thésaurus* : des idées aux mots

Cette section est constituée pour permettre au lecteur de trouver des mots en fonctions d'idées. Cette partie du thésaurus comporte 873 articles qui correspondent à chacun des concepts de niveau 4. Les notions traitées sont elles-mêmes divisées en paragraphes ordonnés selon les catégories grammaticales. Chacun de ces paragraphes regroupe des mots proches sémantiquement qu'il est possible de parcourir grâce à des renvois vers des notions communes permettant ainsi de faciliter les associations d'idées ou la recherche d'expressions les plus pertinentes possible.

[51] pluies acides, SIDA, empreintes génétiques, hiver nucléaire, ...

0 OMEGA
1 MONDE
...
2 ESPACE
2 TEMPS
2 TEMPS ET DURÉE
3 DATE ET CHRONOLOGIE
4 PASSÉ
4 PRÉSENT
4 FUTUR
...
3 ÉVOLUTION ET HISTOIRE
...
2 MATIÈRE
2 VIE
...
1 HOMME
2 ÊTRE HUMAIN
2 CORPS ET VIE
3 CORPS
4 TÊTE
4 MEMBRES
4 MAIN
4 PIED
3 FONCTIONS VITALES
...
2 CORPS ET PERCEPTIONS
2 ESPRIT
...
1 SOCIÉTÉ
...

FIG. 1.22 – Extrait de la hiérarchie du thésaurus Larousse [Larousse, 1992]

fable (*nom fem*) : MORALE, MENSONGE, REPRÉSENTATION, RÉCIT
million (*nom masc*) : MULTITUDE, MILLE
échelle (*nom fem*) : MUSIQUE, MONTÉE et MESURE
pêcher (*verbe*) : PÊCHE

FIG. 1.23 – Exemple de quelques termes extraits du thésaurus Larousse [Larousse, 1992]

1.4.5.3 La partie *Index* : des mots aux idées

Cette dernière partie est sans nul doute la plus utilisée dans le cadre des vecteurs d'idées puisqu'elle permet de retrouver à partir d'un mot les idées, les thèmes qui lui sont associés. On y trouve, par exemple, que ‹*échelle*› est un *nom commun* et que ses concepts associés sont MUSIQUE, MONTÉE et MESURE. On voit donc que cette partie du thésaurus permet facilement de construire une base de données de vecteurs, une fois les concepts eux-mêmes munis d'un vecteur (vecteurs génératifs cf. 2.1.2). On peut toutefois regretter que les distinctions entre les sens ne soit pas bien marquées. Si on reprend l'exemple d'‹*échelle*› les sens *échelle*/**escalier** et *échelle*/**musique** sont, par exemple, clairement fusionnés.

1.5 Conclusions du chapitre

Au cours de ce premier chapitre, nous avons montré en quoi la question du sens se pose dans de nombreuses applications du TALN. Nous avons présenté les quatre niveaux de traitement que comportent de telles applications concernant l'écrit : niveau morphologique, niveau syntaxique, niveau sémantique et niveau pragmatique. Chacun de ces niveaux présente des difficultés et des méthodes ont été proposées par la communauté scientifique pour les résoudre.

Nous nous sommes concentrés en particulier sur les problèmes issus des niveaux sémantique et pragmatique où est traitée plus spécifiquement la question du sens. Ainsi, nous avons présenté plusieurs approches informatiques sur sa modélisation.

La sémantique distributionnelle d'Harris qui considère que le sens d'un mot peut être défini à partir de l'ensemble des contextes linguistiques dans lequel il apparaît a ainsi donné naissance au modèle vectoriel le plus connu : le modèle vectoriel standard de Salton. Les tenants d'un courant issu de la psycholinguistique l'ont prolongé pour créer les techniques de type LSA (*Latent Semantic Analysis*).

La recherche en psychologie expérimentale a permis, elle, de mettre au point les réseaux sémantiques. Ces réseaux sont des graphes orientés qui relient des termes par des arcs étiquetés représentant la relation qui les unit. Ces réseaux font partie, avec les bases d'acceptions, de ce que nous avons appelé les approches symboliques connexionnistes. Ces dernières cherchent à décrire finement chaque acception des items lexicaux qu'elle comprend.

Nous avons présenté la linguistique componentielle qui considère que le sens d'un terme est donné par des unités atomiques de sens. Nous avons exposé en particulier les travaux de Hjelmslev et Greimas puis discuté de la pertinence cognitive de cette théorie. Assez peu de modèles informatiques parents ont été développés mais nous avons évoqué les recherches réalisées par la société Memodata ainsi que celles de Schank et Wilks et en particulier les travaux de ce dernier sur les primitives de sens.

Pour conclure, nous avons présenté les travaux effectués par Jacques Chauché au début des années 1990. Il s'agit là de construire des vecteurs dont les composantes correspondent aux idées contenues dans les items lexicaux. Ces idées se réfèrent à une liste pré-établie par des linguistes (l'*organum* de l'encyclopédie universalis). Ces travaux sont à la base du modèle des vecteurs d'idées dont la construction s'appuie, elle, sur des thésaurus qui comportent un ensemble de concepts censés permettre de décrire l'ensemble des idées exprimables en langue (*hypothèse du thésaurus)*. Les vecteurs d'idées sont le modèle central de notre thèse, modèle que nous allons présenter dans le chapitre suivant.

2

Vecteurs d'idées

DANS ce chapitre, nous présentons le modèle des vecteurs d'idées. Contrairement aux représentations vectorielles classiques dont les dimensions correspondent à des éléments textuels, les dimensions correspondent ici à des idées. Ainsi les segments textuels sont définis à partir d'un ensemble de notions, les concepts, censés pouvoir générer l'ensemble des idées exprimables en langue. Nous exposons le modèle général et les différentes opérations définies sur les vecteurs d'idées puis nous en présentons les deux grandes familles : les vecteurs sémantiques et les vecteurs conceptuels. Leurs principales différences tiennent essentiellement en leur mode de construction. Tandis que les premiers sont construits avant toute application, les seconds, autour desquels s'articule plus particulièrement cette thèse, sont en constant apprentissage.

Sommaire

Si l'utilisation du modèle vectoriel n'est pas récente dans le domaine du Traitement Automatique du Langage Naturel (TALN) puisqu'il a été introduit dès la fin des années 1960 par Salton en recherche documentaire [Salton, 1968], sa réhabilitation dans les recherches en TALN est, en revanche, relativement récente. Elle a été essentiellement motivée par la mise à disposition des chercheurs de grandes bases de textes, en particulier le Web, alors que précédemment, ces recherches passaient par des phases ardues de constitution de corpus d'expériences.

La plupart des approches vectorielles actuelles sont inspirées linguistiquement de la sémantique distributionnelle [Hirschman, 1986, Harris *et al.*, 1989] et informatiquement du modèle vectoriel classique [Salton & McGill, 1983] dont l'implémentation la plus connue est SMART [Salton, 1971]. L'hypothèse principale sur laquelle reposent ces formalismes est que la sémantique d'un item lexical dépend de l'ensemble des contextes dans lequel il apparaît ([Besançon, 2001], p. 55). Par exemple, la sémantique de l'item ‹*lait*› peut être décrite grâce à la liste {‹*vache*›, ‹*bouteille*›, ‹*fromage*›, ‹*yaourt*›, ... }. Dans le modèle classique, les dimensions de l'espace vectoriel correspondent donc aux éléments textuels considérés comme les plus discriminants, préalablement extraits du jeu d'apprentissage : les termes d'indexation.

Le modèle des vecteurs d'idées se rapproche, lui, de la notion de linguistique componentielle développée entre autres par Hjelmslev, Le Ny et Pottier (cf. 1.4). Il s'agit de la projection de la notion linguistique de champ sémantique dans le modèle mathématique d'espace vectoriel. Cette approche vectorielle est fondée sur des propriétés mathématiques bien connues sur lesquelles il est possible d'effectuer des manipulations formellement pertinentes auxquelles sont attachées des interprétations linguistiques raisonnables. Chaque segment textuel peut ainsi se voir attribuer un vecteur représentant les idées qu'il véhicule et une distance basée sur l'angle entre deux vecteurs peut alors être introduite pour pouvoir estimer la proximité thématique entre deux segments. À partir d'une base de données contenant l'indexation des termes d'une langue, une opération d'*analyse sémantique* permet de calculer un vecteur pour un texte donné. Cette opération est à la base des applications principales des vecteurs d'idées : traduction, recherche d'informations, résumé automatique, ...

Nous présentons dans ce chapitre deux variantes des vecteurs d'idées. Si leur nombre de composantes ainsi que l'ensemble de concepts de base (issus du thésaurus Larousse [Larousse, 1992]) sont les mêmes, elles diffèrent à la fois par leur mode de construction et par leur mode d'exploitation. Pour la première, les *vecteurs sémantiques* sont générés automatiquement à partir d'une version électronique du thésaurus Larousse avant toute application. Pour la seconde, celle qui constitue la représentation vectorielle autour de laquelle cette thèse est centrée, les *vecteurs conceptuels* sont construits grâce à un apprentissage à partir de dictionnaires à usage humain présentés sous forme électronique.

2.1 Modèle des vecteurs d'idées

Le modèle des vecteurs d'idées est basé sur la projection de la notion linguistique de champ sémantique dans le modèle mathématique d'espace vectoriel[52]. À partir d'un ensemble de notions élémentaires, les concepts, représentés sous forme de vecteurs (dits *vecteurs génératifs*), on peut construire de nouveaux vecteurs d'idées et les associer à tout segment textuel (items lexicaux, phrases, textes, ...). Il est ainsi possible d'effectuer des manipulations formellement bien fondées auxquelles nous pouvons attacher des interprétations linguistiques raisonnables. L'hypothèse principale sur laquelle repose ce modèle est que les vecteurs génératifs constituent un espace générateur pour l'ensemble des mots de la langue.

2.1.1 Un peu d'histoire

Au sein de l'équipe TALN, plusieurs expérimentations sur les vecteurs d'idées sont en cours. Toutes se font parallèlement et s'influencent largement les unes les autres. Nous nous intéressons plus particulièrement dans ce chapitre à trois expériences. La première qui concerne les vecteurs sémantiques est implémentée par Jacques Chauché. Les deux autres sont basées sur les vecteurs conceptuels. L'une est implémentée depuis environ cinq ans par Mathieu Lafourcade tandis que l'autre est implémentée par moi-même depuis le début de ma thèse.

Historiquement, les vecteurs d'idées sont le prolongement des travaux effectués par Jacques Chauché au début des années 1990 que nous avons présentés au chapitre précédent (cf. proto-vecteurs d'idées section 1.4.3). À partir de l'arrivée dans l'équipe de Mathieu Lafourcade en 1997, les deux expériences ont commencé à diverger sur un certain nombre de points (granularité de la représentation du sens, base de données vectorielle fixe ou en apprentissage, ...) mais un socle commun reste et continue à être développé parallèlement. En témoignent, par exemple, les travaux sur la structuration des textes [Yousfi-Monod & Prince, 2005b] effectués par Mehdi Yousfi-Monod et Violaine Prince, dont les résultats peuvent être utilisés dans l'analyse des textes quelle que soit l'expérience, ou bien les travaux de Violaine Prince sur la génération à partir de l'arbre d'analyse d'une phrase en français, d'une traduction grâce à des règles de transformation qui fournissent la structure syntaxique en anglais[53]. De même, les chercheurs participent à tel ou tel point particulier dans telle ou telle expérimentation. Violaine Prince, par exemple, a participé aux travaux de thèse de Simon Jaillet [Jaillet, 2005] sur la catégorisation de textes grâce aux vecteurs sémantiques et travaille avec moi sur la représentation et l'utilisation des fonctions lexicales pour l'amélioration des représentations basées sur les vecteurs conceptuels.

Un des problèmes auxquels nous avons été confrontés au cours de cette thèse a été d'étudier en quoi les différentes expériences divergeaient, en quoi elles convergeaient, pour quelles raisons et quelles hypothèses de départ expliquaient ces différences. Pour parler des points de convergence, nous introduisons dans cette thèse le terme *"vecteurs d'idées"* qui n'a pas été utilisé dans les différents articles de l'équipe qui traitent indifféremment suivant les auteurs de vecteurs sémantiques ou de vecteurs conceptuels.

2.1.2 Vecteurs génératifs et espace des vecteurs d'idées

Le principe de base des vecteurs d'idées est semblable à celui de la linguistique componentielle. Il suppose l'existence d'une atomisation de la signification, c'est-à-dire que le sens d'un terme peut être décomposé en éléments de sens plus petits (cf. 1.4). Dans notre modèle, nous considérons que ces éléments de sens plus petits, que nous appelons *concepts*, peuvent être représentés par des vecteurs de $\mathbb{R}^{+n}$ et qu'ils sont susceptibles de générer l'ensemble des vecteurs

[52] Un rappel des principales notions concernant les espaces vectoriels est présenté en annexe A.
[53] http://www.lirmm.fr/~prince/ExempleAnlTal.html

d'idées. Les vecteurs des concepts sont appelés *vecteurs génératifs*.

Nous notons l'ensemble des vecteurs d'idées ϑ, celui correspondant aux items lexicaux est noté ω. Les concepts sont notés (à la Mel'čuk) en majuscules (*VIE*), les items en italique et entre guillemets (‹*vie*›). Enfin, $V(x)$ correspond au vecteur d'idées d'un élément x quel que soit cet élément, item lexical, concept ou segment textuel[54].

2.1.2.1 Vecteurs d'idées, première approximation

Nous pouvons considérer, en première approximation, que les vecteurs d'idées sont construits grâce à des combinaisons linéaires de vecteurs génératifs. Soit $\mathcal{C} = \{c_1, c_2, \ldots, c_n\}$ un ensemble fini de n concepts, soit $\mathcal{F} = \{V(c_1), V(c_2), \ldots, V(c_n)\}$ la famille de vecteurs correspondants, soit l un item lexical et soit $\alpha_i \in \mathbb{R}^+$ l'intensité de c_i dans l, alors nous posons que :

$$V(l) = \frac{V}{\| V \|} \quad \text{où } V = \sum_{i=1}^{n} \alpha_i V(c_i) \tag{2.1}$$

Un vecteur d'idées est le vecteur normé d'une combinaison linéaire des vecteurs génératifs. Par exemple, si nous considérons que ‹*Ferrari*› peut être construit à partir des idées *VOITURE*, *ROUGE*, *RAPIDE*, le vecteur d'idée associé est :

$$V(\text{‹}Ferrari\text{›}) = \frac{\alpha_{VOITURE} V(VOITURE) + \alpha_{RAPIDITÉ} V(RAPIDITÉ) + \alpha_{ROUGE} V(ROUGE)}{\| \alpha_{VOITURE} V(VOITURE) + \alpha_{RAPIDITÉ} V(RAPIDITÉ) + \alpha_{ROUGE} V(ROUGE) \|} \tag{2.2}$$

La valeur de $\alpha_{VOITURE}$ (respectivement $\alpha_{RAPIDITÉ}$, α_{ROUGE}) est alors fonction de l'importance de l'idée dans l'item lexical ‹*Ferrari*›.

2.1.2.2 Espace des vecteurs d'idées et interprétation linguistique

Les vecteurs d'idées sont des vecteurs de $\mathbb{R}^n$ où n correspond au nombre de vecteurs génératifs. lorsqu'ils sont associés à des segments textuels, ils sont normés à 1 et forment donc une hypersphère de rayon 1. En pratique, plus n est grand, plus fines sont les descriptions de sens offertes par les vecteurs, mais plus leur manipulation informatique est lourde. Le choix de n doit donc être un compromis entre une meilleure finesse de la représentation et des contraintes matérielles.

En pratique, par construction[55], les composantes des vecteurs d'idées sont positives. On peut donc dire que ce sont des vecteurs de $\mathbb{R}^{+n}$. En toute généralité, et pour respecter les axiomes des espaces vectoriels normés, nous n'oublierons pas que les vecteurs d'idées sont des vecteurs de $\mathbb{R}^n$. Toutefois, par souci de simplification, lorsqu'il s'agira de définir les opérations sur les vecteurs, nous prendrons en compte la positivité des composantes des vecteurs, en particulier en ce qui concerne les domaines de définition.

Profitons de ce point pour essayer d'éviter une erreur parfois commise. L'espace des vecteurs d'idées associés à des segments textuels ϑ est-il un sous-espace vectoriel de $\mathbb{R}$? Pour être un sous-espace vectoriel, il faut que ϑ soit stable par combinaison linéaire (cf. A.4.2) c'est-à-dire que nous devons avoir :

$$\forall u, v \in \vartheta, \forall \alpha, \beta \in \mathbb{R}, \alpha u + \beta v \in \vartheta$$

Or cette propriété n'est pas vérifiée à cause de la normalisation des vecteurs. Ainsi, **l'espace des vecteurs d'idées n'est pas un sous-espace vectoriel dont les vecteurs génératifs seraient une base ou une famille génératrice.**

[54] Le lecteur trouvera en début de manuscrit, une présentation des notations utilisées dans cette thèse

[55] En tout cas dans l'état actuel de nos recherches.

Formellement, nous avons $\vartheta \equiv \mathbb{R}^n$. Toutes les opérations vectorielles qu'il est possible d'effectuer dans un espace vectoriel normé (cf. Annexe A.5) sont donc réalisables dans ϑ. La différence est que nous cherchons à donner à ces vecteurs une interprétation linguistique voire psycholinguistique. Ils représentent des idées qui peuvent faire référence éventuellement à un item lexical, un terme de la langue. Les opérations réalisables sur les vecteurs d'idées peuvent donc elles aussi avoir des interprétations linguistiques qu'il convient d'analyser. Nous nous appliquerons dans la suite de cette thèse à donner, autant que faire se peut, des interprétations linguistiques aux opérations présentées.

2.1.2.3 Vecteurs normés

Dans notre modèle, la norme n'est pas considérée comme une information qualitative. En effet, nous considérons que les idées ne prennent sens que si elles sont appréciées les unes par rapport aux autres. Cette affirmation est vraie tant à l'intérieur des vecteurs qu'entre les vecteurs. Ainsi, il est plus pertinent de comparer les proportions des différentes idées à l'intérieur d'un terme, d'un vecteur, plutôt que de les analyser de façon absolue.

Il en est de même entre deux vecteurs. Que signifierait la comparaison de deux vecteurs qui n'ont pas la même norme ? Si on prend l'exemple du calcul du vecteur d'idées d'un texte quelconque par une méthode d'analyse sémantique (cf. 2.1.6), le principe est, en schématisant beaucoup, de faire une somme pondérée des vecteurs. Si les vecteurs n'étaient pas normés, la norme d'un vecteur serait un simple indicateur de la longueur du texte à partir duquel il a été construit.

Si, au cours des opérations que nous présenterons par la suite (cf. 2.1.4), la norme d'un vecteur n'est pas toujours égale à l'unité, en particulier dans le cas de l'utilisation du produit terme à terme, il n'en est jamais de même lorsqu'il s'agit d'un vecteur correspondant à un segment textuel. Par exemple, les vecteurs d'idées stockés dans une base de données vectorielle ont tous une norme égale à 1.

2.1.3 Distance et voisinage thématique

Un des premiers outils utilisés pour vérifier la cohérence d'une base de vecteurs d'idées a été la fonction de voisinage thématique. Elle permet de connaître les items dont le vecteur est le plus proche du vecteur d'un item donné. Cette fonction de voisinage est basée sur la notion de distance thématique.

2.1.3.1 Distance thématique

Il est souvent souhaitable de pouvoir mesurer la proximité entre deux items, c'est-à-dire une distance (ou au moins une mesure) entre leurs vecteurs d'idées. Ces opérations peuvent permettre non seulement d'estimer la cohérence d'une base de vecteurs mais aussi s'avèrent souvent déterminantes pour connaître le sens d'un terme. Il existe de nombreuses distances possibles entre vecteurs, la thèse de Romaric Besançon ([Besançon, 2001], section 2.2.5) en fait une bonne synthèse. Dans le cadre des vecteurs d'idées, la distance angulaire est utilisée pour les opérations de base. En effet, elle permet une meilleure discrimination pour les faibles angles et offre d'intéressantes interprétations géométriques.

Par abus de langage, dans la suite de cet exposé, nous parlerons parfois de distance entre deux segments textuels au lieu de parler de distance entre les vecteurs associés à ces deux segments.

Similarité et distance angulaire Soit $Sim(X,Y)$ une des mesures de *similarité* entre deux vecteurs X et Y, utilisée habituellement en recherche d'informations ([Salton & McGill, 1983], p. 121). Cette valeur est le cosinus de l'angle entre les deux vecteurs (cf. A.39).

$$\vartheta^2 \rightarrow [0,1] : \quad Sim(X,Y) = \cos(\widehat{X,Y}) = \frac{X \cdot Y}{\|X\| \times \|Y\|} \tag{2.3}$$

Nous définissons la fonction de *distance thématique* D_A entre deux vecteurs X et Y comme la distance angulaire entre les deux vecteurs (cf. A.40).

$$\vartheta^2 \rightarrow [0,\frac{\pi}{2}] : \quad D_A(X,Y) = \arccos(Sim(X,Y)) \tag{2.4}$$

Par définition, nous posons :

$$D_A(\vec{\mathbf{0}},\vec{\mathbf{0}}) = 0 \quad et \quad D_A(X,\vec{\mathbf{0}}) = \frac{\pi}{2} \text{ si } X \neq \vec{\mathbf{0}} \tag{2.5}$$

avec $\vec{\mathbf{0}}$ dénotant le vecteur nul. Précisons que ce vecteur n'a sans doute pas de représentation en langue. Il s'agirait d'un mot qui n'active aucun concept, l'idée vide.

La distance angulaire est une vraie distance puisqu'elle vérifie les propriétés de réflexivité, symétrie et inégalité triangulaire.

$$réflexivité \ : D_A(X,X) = 0 \tag{2.6}$$

$$symétrie \ : D_A(X,Y) = D_A(Y,X) \tag{2.7}$$

$$inégalité\ triangulaire \ : D_A(A,B) + D_A(B,C) \geq D_A(A,C) \tag{2.8}$$

Pourquoi choisir l'angle entre les deux vecteurs ? Observons la figure 2.1. L'arc cosinus est une fonction décroissante par rapport à la similarité. Plus cette dernière est grande, plus l'angle entre les deux vecteurs est important. Cette fonction est intéressante dans notre problématique car elle est fortement non linéaire par rapport à la similarité pour les faibles valeurs de l'angle (en dessous de $\frac{\pi}{4}$, là où les comparaisons à effectuer sont les plus fines) tandis qu'elle est pratiquement linéaire au-delà de $\frac{\pi}{4}$.

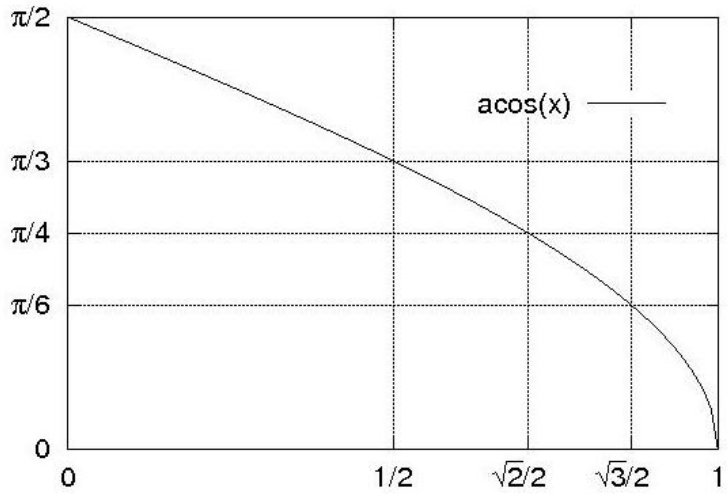


FIG. 2.1 – fonction arc cosinus

En effet, un des objectifs des vecteurs d'idées est de participer à des tâches de désambiguïsation sémantique. Il peut arriver que les différences entre les sens soient relativement faibles. Par exemple, l'éloignement des sens *agneau*/**viande** et *agneau*/**animal** semble assez peu important. C'est une des propriétés qui a entraîné le choix de l'angle entre les vecteurs comme l'un de nos principaux outils. La deuxième, que nous allons voir plus en détail maintenant, concerne les relativement bonnes interprétations géométriques qu'il est possible d'effectuer.

Exemples Le tableau 2.2 présente les distances thématiques (en radians) entre les vecteurs de plusieurs termes. Ces exemples sont réalisés à partir de la base de vecteurs conceptuels dont l'architecture est présentée en 2.3.

D_A	destinée	destin	vie	existence	mort	automobile	train	action	inaction	réaction
destinée	0	0,51	0,82	0,7	0,99	1,29	1,38	1,31	1,14	1,2
destin		0	0,83	0,75	0,99	1,3	1,38	1,25	1,07	1,16
vie			0	0,61	0,89	1,28	1,35	1,3	1,1	1,2
existence				0	0,98	1,37	1,43	1,37	1,25	1,3
mort					0	1,33	1,4	1,32	1,15	1,26
automobile						0	0,88	1,4	1,22	1,29
train							0	1,43	1,3	1,39
action								0	1,01	0,67
inaction									0	0,9
réaction										0

FIG. 2.2 – Exemples de résultats de la distance thématique D_A(X, Y).

Le tableau est symétrique (symétrie de D_A(X, Y)) et la diagonale est toujours égale à 0 (réflexivité de D_A(X, Y)). On remarquera qu'une valeur prend toute sa signification relativement à une autre. En particulier, il est satisfaisant d'avoir :

- D_A(‹*destin*›, ‹*destinée*›) $\leq$ D_A(‹*destinée*›, ‹*vie*›) et D_A(‹*existence*›, ‹*vie*›) $\leq$ D_A(‹*destinée*›, ‹*vie*›) ce qui correspond bien au fait que ‹*destin*› et ‹*destinée*› d'une part, et ‹*existence*› et ‹*vie*› sont plus proches que ‹*destinée*› et ‹*vie*›.
- D_A(‹*vie*›, ‹*mort*›) $> \frac{\pi}{4}$ (0,78) ce qui dénote un certain éloignement des idées.
- D_A(‹*vie*›, ‹*automobile*›) $> \frac{\pi}{3}$ (1,04) ce qui relève d'un éloignement important.
- D_A(‹*action*›, ‹*réaction*›) est la plus petite valeur de D_A(‹*action*›, Y) car les deux concepts *ACTION* et *RÉACTION* sont relativement proches et que ‹*action*› partage beaucoup moins d'idées avec les autres termes.

Interprétations On remarquera qu'habituellement, les comparaisons entre les valeurs sont plus significatives que les valeurs elles-mêmes, toutefois, on estime empiriquement que :

- si $D_A(X,Y) \leq \frac{\pi}{4}$, X et Y partagent des concepts et sont considérés comme sémantiquement proche.
- si $D_A(X,Y) \geq \frac{\pi}{4}$, la proximité sémantique de A et B est considérée comme faible.
- Aux alentours de $\frac{\pi}{2}$, les sens sont sans rapport.

Intuitivement, cette fonction constitue une évaluation possible de la *proximité thématique*. La métaphore de la nuit étoilée peut aider à appréhender cette idée de distance angulaire pour calculer la proximité thématique. Nous pouvons nous représenter l'espace des sens comme un ciel rempli d'étoiles. Les étoiles sont les items lexicaux. Les mots, tout comme les étoiles, forment des constellations. Certaines parties de l'espace sont très densément peuplées tandis que d'autres sont quasi-désertes. Un sens est une direction de l'espace et non un point. Un observateur ne peut connaître exactement la distance entre une étoile et lui-même mais il connaît la direction de l'astre. Dans le ciel, la distance entre deux étoiles est la distance apparente, l'angle entre les deux. Il en est de même, dans notre espace, avec les items lexicaux.

2.1.3.2 Voisinage thématique

La fonction de voisinage thématique permet de connaître les items lexicaux voisins d'un item lexical donné. On définit $\mathcal{V}$ la fonction de proximité thématique qui renvoie les k items les plus proches en termes de distance angulaire d'un texte Z dans une base vectorielle. Soit :

$$\left|\mathcal{V}(Z)\right| = k \qquad \forall X \in \mathcal{V}(Z), \quad \forall Y \notin \mathcal{V}(Z), \quad D_A(\mathrm{X},\,\mathrm{Y}) \leq D_A(\mathrm{Y},\,\mathrm{Z}) \tag{2.9}$$

Par exemple, les termes proches et ordonnés par distance thématique croissante des mots ‹*vie*›, ‹*ranger*› et ‹*couper*› pourraient être :

$\mathcal{V}$(‹*vie*›) = ‹*vie quotidienne*›, *VIE*, ‹*s'animer*›, ‹*demi-vie*›, ‹*survivant*›, ‹*avoir la vie devant soi*›, ‹*naissance*›, ‹*viabilité*›, ‹*vital*›, ‹*naître*›, ‹*vivant*›, ‹*assurance-vie*›,...

$\mathcal{V}$(‹*ranger*›) = ‹*trier*›, ‹*cataloguer*›, ‹*sélectionner*›, ‹*classer*›, ‹*distribuer*›, ‹*grouper*›, ‹*ordonner*›, ‹*répartir*›, ‹*aligner*›, ‹*caser*›, ‹*arranger*›, ‹*nettoyer*›, ‹*distribuer*›, ‹*démêler*›, ‹*ajuster*›, ...

$\mathcal{V}$(‹*couper*›) = ‹*cisailler*›, ‹*émincer*›, ‹*scier*›, ‹*tronçonner*› , ‹*ébarber*›, ‹*entrecouper*›, ‹*baptiser*›, ‹*recouper*›, ‹*sectionner*›, ‹*bêcher*›, ‹*hongrer*›, ‹*essoriller*›, ‹*rogner*›, ‹*égorger*›, ‹*écimer*›, ...

2.1.4 Opérations classiques

Cette section présente les opérations définies pour les vecteurs d'idées que nous utiliserons dans la suite de cette thèse.

2.1.4.1 Somme vectorielle

Soient X et Y deux vecteurs, leur *somme vectorielle* V est définie par :

$$\vartheta^2 \to \vartheta \;:V = X + Y \quad | \quad V_i = X_i + Y_i \tag{2.10}$$

où V_i (resp X_i, Y_i) représente la i-ème composante du vecteur V (resp. X, Y).

Soient X et Y deux vecteurs, leur *somme vectorielle normée* V est définie par :

$$\vartheta^2 \to \vartheta \;:V = X \oplus Y \quad | \quad V_i = \frac{X_i + Y_i}{\|X + Y\|} \tag{2.11}$$

L'opérateur $\oplus$ est idempotent et nous avons $X \oplus X = X$. Le vecteur nul $\vec{\mathbf{0}}$ est l'élément neutre de la somme vectorielle et, par définition, on pose,

$$\vec{\mathbf{0}} \oplus \vec{\mathbf{0}} = \vec{\mathbf{0}}. \tag{2.12}$$

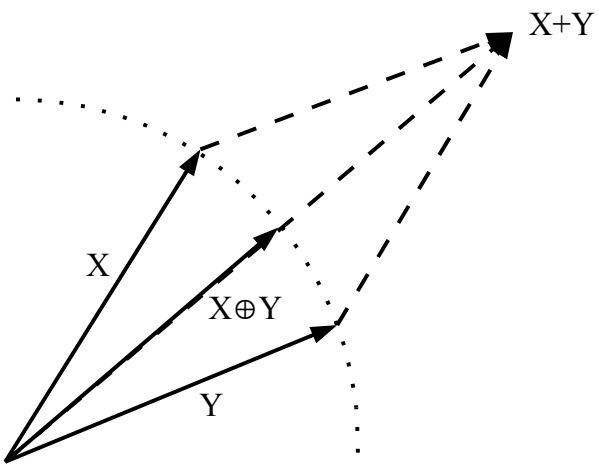


FIG. 2.3 – Somme vectorielle normée

De ce qui précède, on peut facilement déduire les propriétés de rapprochement (local et généralisé) :

$$D_A(X \oplus X, Y \oplus X) = D_A(X, Y \oplus X) \leq D_A(X,Y) \tag{2.13}$$

$$D_A(X \oplus Z, Y \oplus Z) \leq D_A(X,Y) \tag{2.14}$$

La somme vectorielle est généralisée à n'importe quel nombre de vecteurs par :

$$\vartheta^n \rightarrow \vartheta \ : V = \sum_{i=1}^{n} V(x_i) \quad | \quad V_j = \sum_{j=1}^{n} V(x_i)_j \tag{2.15}$$

où $V(x_i)$ représente le vecteur d'idée de l'objet x_i, V_j et $V(x_i)_j$ la j-ème composante des vecteurs V et $V(x_i)$. La somme vectorielle normée est généralisée à n'importe quel nombre de vecteurs par :

$$\vartheta^n \rightarrow \vartheta \ : V = \bigoplus_{i=1}^{n} V(x_i) \quad | \quad V_j = \frac{\sum_{j=1}^{n} V(x_i)_j}{\| \sum_{i=1}^{n} V(x_i) \|} \tag{2.16}$$

Précision importante sur la notation

La somme vectorielle normée binaire n'est pas associative. Toutefois, pour pouvoir parfois simplifier notre discours, nous écrirons $\bigoplus_{i=1}^{n} V(x_i)$ au lieu de $V(x_1) \oplus V(x_2) \oplus \ldots \oplus V(x_n)$.

2.1.4.2 Interprétation

La somme vectorielle normée de deux vecteurs donne un vecteur équidistant en termes d'angle des deux premiers vecteurs. Il s'agit en fait d'une moyenne des vecteurs sommés. En tant qu'opération sur les vecteurs d'idées, on peut donc voir la somme vectorielle normée comme l'union des idées contenues dans les termes.

2.1.4.3 Produit terme à terme

Soient X et Y deux vecteurs, leur *produit terme à terme* V est défini par :

$$\vartheta^2 \rightarrow \vartheta \ : V = X \odot Y \quad | \quad v_i = x_i y_i \tag{2.17}$$

Soient X et Y deux vecteurs, leur *produit terme à terme normalisé* V est défini par :

$$\vartheta^2 \rightarrow \vartheta \ : V = X \otimes Y \quad | \quad v_i = \sqrt{x_i y_i} \tag{2.18}$$

Cet opérateur est idempotent ($X \otimes X = X$) et $\vec{\mathbf{0}}$ est absorbant ($X \otimes \vec{\mathbf{0}} = \vec{\mathbf{0}}$). Il peut être généralisé à n'importe quel nombre de vecteurs par :

$$\vartheta^n \rightarrow \vartheta \ : V = \bigotimes_{i=1}^{n} V(x_i) \quad | \quad V_j = \sqrt[n]{\prod_{j=1}^{n} V(x_i)_j} \tag{2.19}$$

L'opérateur $\otimes$ peut être interprété comme un opérateur d'intersection entre vecteurs. Si l'intersection entre deux vecteurs est le vecteur nul, alors ils n'ont rien en commun. On a, de plus, la propriété suivante :

$$\forall X \neq \vec{\mathbf{0}} \quad \forall Y \neq \vec{\mathbf{0}} \quad X \otimes Y = \vec{\mathbf{0}} \Leftrightarrow D_A(X,Y) = \frac{\pi}{2} \tag{2.20}$$

2.1.4.4 Interprétation

Comme nous venons de le dire, l'opérateur $\otimes$ peut être vu comme une intersection des vecteurs. Du point de vue des vecteurs d'idées, cette opération permet donc de sélectionner les idées communes à un ensemble de termes. Il est utilisé en particulier dans l'opération de contextualisation faible.

2.1.4.5 Contextualisation faible

Lorsque que deux termes sont en présence, pour chacun d'eux, certaines idées se trouvent sélectionnées par le contexte que constitue l'autre terme. Ce phénomène de *contextualisation* consiste à augmenter chaque vecteur de ce qu'il a de commun avec l'autre. Comme nous venons de le voir, les idées communes à deux termes sont données par le produit terme à terme. Ainsi, nous pouvons définir la contextualisation faible $\gamma(X,Y)$ des vecteurs X par Y par :

$$\vartheta^2 \rightarrow \vartheta \ : \gamma(X,Y) = X \oplus (X \odot Y) \tag{2.21}$$

Cette fonction n'est pas symétrique. L'opérateur γ est idempotent ($\gamma(X,X) = X$) et le vecteur nul est un élément neutre ($\gamma(X,\vec{\mathbf{0}}) = X \oplus \vec{\mathbf{0}} = X$).

La propriété de *rapprochement* suivante peut être tirée :

$$D_A(\gamma(X,Y),\gamma(Y,X)) \leq D_A(\gamma(X,Y),Y) \leq D_A(X,Y) \tag{2.22}$$

$$D_A(\gamma(X,Y),\gamma(Y,X)) \leq D_A(X,\gamma(Y,X)) \leq D_A(X,Y) \tag{2.23}$$

La contextualisation $\gamma(X,Y)$ rapproche les vecteurs X de Y proportionnellement à leur intersection.

2.1.5 Statistiques

Le but de cette section est de rappeler quelques formules de statistiques que nous pouvons appliquer sur les composantes V_i d'un vecteur d'idée quelconque V. Nous présentons autant que possible les interprétations que nous pouvons faire de l'application de ces formules sur des vecteurs :

2.1.5.1 Moyenne

La moyenne arithmétique μ de N valeurs est :

$$\mu = \frac{\sum_{i=1}^{N} x_i}{N} \tag{2.24}$$

2.1.5.2 Variance

La variance v est donnée par la formule :

$$v = \frac{\sum_{i=1}^{N} (x_i - \mu)^2}{N} \tag{2.25}$$

2.1.5.3 Écart type

L'écart type e est défini par :

$$e = \sqrt{v} \tag{2.26}$$

2.1.5.4 Coefficient de variation

Le coefficient de variation c est donné par :

$$c = \frac{e}{\mu} \tag{2.27}$$

Le coefficient de variation n'est défini que lorsque $\mu \neq 0$. Toutefois, il peut être arbitrairement étendu pour tenir compte du vecteur nul :

$$c(\vec{0}) = 0 \tag{2.28}$$

Dans le cadre des vecteurs d'idées, on peut voir le coefficient de variation c comme une mesure statistique normalisée (sans unité) de la "conceptualité" du vecteur V. Il est d'autant plus important que les composantes du vecteur sont contrastées et vaut 0 si elles ont toutes la même valeur $\frac{\sqrt{n}}{n}$.

2.1.6 Analyse sémantique de textes : l'algorithme de remontée-redescente

L'objectif des vecteurs d'idées est d'améliorer la plupart des applications du TALN où la sémantique peut jouer un rôle. Citons quelques exemples. En *recherche documentaire*, on peut dans une phase de préparation des données affecter un vecteur à chaque texte et dans une phase d'exploitation renvoyer les plus proches du vecteur d'une requête ; en *traduction automatique* il peut s'agir de trouver le vecteur correspondant à l'équivalent le plus proche dans une langue cible (cf. chapitre 8) ; en *résumé automatique de textes* on peut choisir de privilégier une partie du texte qui représente mieux les idées principales du discours général plutôt qu'une autre ; en *catégorisation* on peut regrouper les textes les plus proches suivant une méthode basée sur la distance angulaire, ... L'idée sous-jacente est donc de pouvoir affecter un vecteur d'idées à tout segment textuel et c'est dans cette perspective qu'a été définie l'analyse sémantique de textes. Cette analyse est différente suivant le type de vecteurs utilisés (sémantique ou conceptuel) cependant le principe général est le même.

2.1.6.1 Principe

L'analyse sémantique de textes permet de calculer le vecteur d'idées d'un texte quelconque. Son principe général est de se baser sur l'hypothèse de compositionnalité de la sémantique linguistique c'est-à-dire que « *le tout est calculable à partir du sens de ses parties* » (cf. 1.1.4.3). Le vecteur d'un texte est donc calculé de façon générale par une fonction ayant pour paramètres l'ensemble des vecteurs d'idées des items du texte. En pratique, il s'agit d'une somme pondérée des vecteurs contextualisés des termes de ce texte.

L'idée originale de cette opération est de se baser sur une analyse morpho-syntaxique préliminaire telle que celle présentée en 1.1.4.2. En effet, il a été montré que l'apport d'informations syntaxiques pour la construction de vecteurs de type saltonien donne de meilleures performances, entre autres, dans le domaine de la recherche d'informations [Besançon, 2001]. Ce constat peut être renouvelé avec les vecteurs d'idées. L'analyse morpho-syntaxique réalisée permet de pondérer par un scalaire le vecteur de chaque mot ou groupe de mots en fonction de son rôle syntaxique [Chauché *et al.*, 2003]. Ainsi, dans le segment « *voile de bateau* », ‹*voile*› est gouverneur syntaxique, son vecteur aura donc un poids plus important que ‹*bateau*›. En revanche, l'inverse sera appliqué pour ‹*bateau à voile*›.

La méthode utilisée pour calculer les vecteurs sémantiques (cf. 2.2.3) et celle utilisée pour calculer les vecteurs conceptuels (cf. 2.3.7) se différencient sur la manière d'utiliser le contexte ainsi que sur l'affectation des vecteurs aux feuilles.

2.1.6.2 Préformatage de textes

L'analyse sémantique des textes s'effectue donc en deux parties. Dans une première, on extrait la structure morpho-syntaxique que l'on utilise dans une deuxième partie pour calculer un vecteur d'idées. Il est clair que la première partie influence grandement la seconde. En effet, une mauvaise analyse morpho-syntaxique peut faire insister l'analyse sémantique sur des aspects du texte moins pertinents, par exemple si un gouverneur est mal identifié, voire erroné si l'arbre indique des morphologies ou des fonctions syntaxiques incorrectes. Ce type d'erreur est généré lorsque l'analyseur syntaxique est confronté à des termes qui lui sont inconnus. Un préformatage à partir des informations contenues dans la base de données vectorielle peut influencer bénéfiquement l'analyseur morpho-syntaxique en lui indiquant quelles morphologies sont possibles pour tel ou tel terme. Cette étape préliminaire peut aussi permettre de faciliter l'analyse sémantique si les textes contiennent des constructions de phrases particulières ou de préparer les textes en fonction d'une tâche spécifique. Le préformatage des textes vise donc à améliorer soit l'analyse morpho-syntaxique soit la partie analyse sémantique proprement dite.

Préformatage pour améliorer l'analyse morpho-syntaxique Deux principaux préformatages peuvent être effectués pour améliorer l'analyse morpho-syntaxique des textes : la *gestion des mots inconnus* et la *reconnaissance des locutions*.

Gestion des mots inconnus Les mots inconnus pour l'analyseur morpho-syntaxique entraînent souvent des erreurs dans l'arbre renvoyé : les constituants ne sont pas ou sont mal identifiés. Ce genre de problème est extrêmement fréquent. Bien qu'il ne soit pas rare de trouver des fautes d'orthographe quelle que soit la provenance des textes (extraits de journaux, dictionnaires, dépêches d'agences, etc.), la plupart des mots inconnus sont plutôt le fait de néologismes ou de noms propres que l'analyseur ne peut pas forcément posséder. Les fautes doivent être corrigées ou les informations complétées afin que l'analyse syntaxique soit la plus fiable possible. Des méthodes automatiques de correction existent. En pratique, le texte est envoyé à l'analyseur syntaxique qui attire l'attention sur certains mots qu'il n'a pas réussi à identifier. Des systèmes experts capables de trouver dans la base de données vectorielle quels sont les termes les plus proches en fonction des informations présentes permettent alors de proposer une correction en cas de fautes d'orthographe ou indiquent quelles informations morphologiques utiliser. Le texte est alors annoté par des balises spécifiques et resoumis à l'analyseur morpho-syntaxique. Par exemple, si nous prenons la phrase « *Napoléon a été sacré en 1804.* », si Napoléon n'est pas reconnu au cours de l'analyse morpho-syntaxique, la phrase est resoumise balisée « *@nom_propre_masculin_singulier@Napoléon a été sacré en 1804.* ». L'analyseur peut alors renvoyer un arbre correct. Notons que soumettre à nouveau la requête à SYGFRAN n'est pas un réel problème puisque le temps de calcul de l'arbre d'une définition standard est négligeable (souvent très inférieur à la seconde) par rapport à l'analyse sémantique globale (cf. 1.1.4.2).

Reconnaissance des locutions Un autre moyen d'aider l'analyseur morpho-syntaxique est de lui indiquer les *locutions semi-figées connexes*, c'est-à-dire les items dont la sémantique ne peut pas être calculée uniquement à partir de la somme de leurs parties. C'est le cas par exemple de termes comme ‹*moulin à vent*›, ‹*avion de chasse*› ou ‹*lampe de chevet*›. Il est nécessaire de les indiquer à l'analyseur morpho-syntaxique afin que celui-ci ne renvoie pas un sous-arbre pour ces mots.

Préformatage pour améliorer l'analyse sémantique Le préformatage permet aussi de préparer le texte en vue de l'analyse sémantique proprement dite. Il s'agit alors de préparer les

textes en vue d'une analyse spécifique comme c'est le cas avec le métalangage des définitions pour les vecteurs conceptuels (cf. 2.3.5.1).

2.2 Les vecteurs sémantiques

Les vecteurs sémantiques sont le prolongement direct des travaux de Jacques Chauché initiés au début des années 1990 (cf. 1.4.3). Il s'agit d'une formalisation de la projection de la notion linguistique de champ sémantique dans un espace vectoriel. Après la traduction automatique, une des premières idées d'application est alors d'améliorer les analyses syntaxiques réalisées par SYGFRAN grâce à des informations de nature sémantique. Par la suite, les applications sont devenues encore plus variées (catégorisation, résumé de textes). Les vecteurs génératifs de l'espace des vecteurs sémantiques sont constitués à partir de la hiérarchie issue de [Larousse, 1992] (cf. 1.4.5.1). La version électronique de la troisième partie du thésaurus (cf. 1.4.5.3) de ce thésaurus a permis d'indexer un ensemble important de termes généraux de la langue (environ 50000).

2.2.1 Objectifs visés

Une des premières applications des vecteurs sémantiques est d'améliorer les analyses syntaxiques réalisées par SYGFRAN. Il s'agit de pouvoir lever un certain nombre d'ambiguïtés relevées par l'analyseur. Par exemple, la phrase classique « *La petite brise la glace* » peut syntaxiquement avoir deux interprétations. Dans la première, ‹*petite*› et ‹*glace*› sont des noms, *brise* est la troisième personne du présent de l'indicatif du verbe ‹*briser*› tandis que dans la deuxième, *petite* correspond à l'adjectif ‹*petit*›, *glace* au verbe ‹*glacer*› et ‹*brise*› est un nom (cf. figure 2.4). Si syntaxiquement, il est absolument impossible de lever l'ambiguïté, des informations de nature sémantique sur le contexte de cette phrase peuvent permettre d'émettre des préférences.

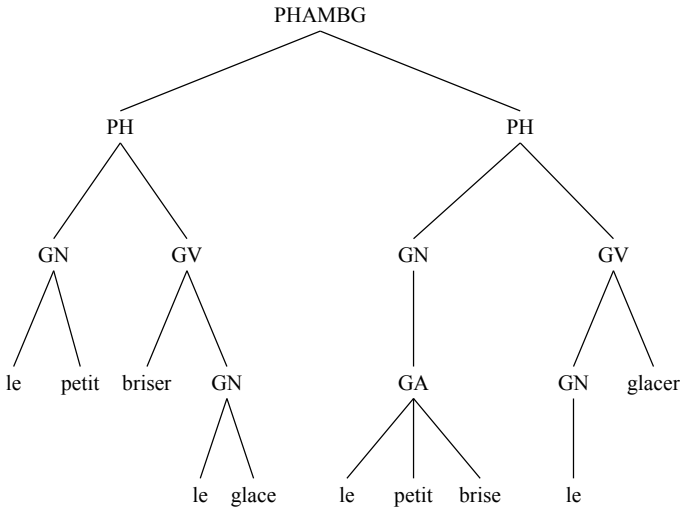


FIG. 2.4 – Analyse syntaxique de la phrase « *La petite brise la glace.* ».

Les travaux ont été poursuivis par des essais sur la traduction du français vers le Tahitien (Eugène Sandford, Jacques Chauché) [Sandford, 1998] et sont actuellement menés par l'équipe sur la traduction du français vers l'anglais[56] (Violaine Prince), la catégorisation (Simon Jaillet, Jacques Chauché, Violaine Prince) [Chauché *et al.*, 2003] [Jaillet, 2005] ainsi que sur le

[56] http://www.lirmm.fr/~prince/ExempleAnlTal.html

résumé automatique (Mehdi Yousfi-Monod, Violaine Prince) [Yousfi-Monod & Prince, 2005a] [Yousfi-Monod & Prince, 2005b] .

2.2.2 Architecture de la base

La fabrication des vecteurs sémantiques a évolué au cours des années. D'une utilisation de la partie *organum* de l'encyclopédie Universalis (cf. 1.4.3), on est passé à l'usage du thésaurus Larousse, outil présenté comme permettant de couvrir l'ensemble des idées de la langue (hypothèse du thésaurus cf. 1.4.5.1).

2.2.2.1 Vecteurs génératifs

Les vecteurs génératifs des vecteurs sémantiques sont fabriqués à partir de la partie index du thésaurus. Pour chaque entrée de même nom que le concept, cette partie présente les concepts associés (cf. 1.4.5.3). Le vecteur génératif de ce concept est le vecteur normé du vecteur qui comporte un 1 pour chacune des composantes correspondant aux concepts cités. La figure 2.5 présente l'exemple du concept *PAIX* défini par le thésaurus par *ÉQUILIBRE*, *SILENCE*, *ACCORD*, *CALME*, *REPOS*, *SÉCURITÉ* et *GUERRE*.

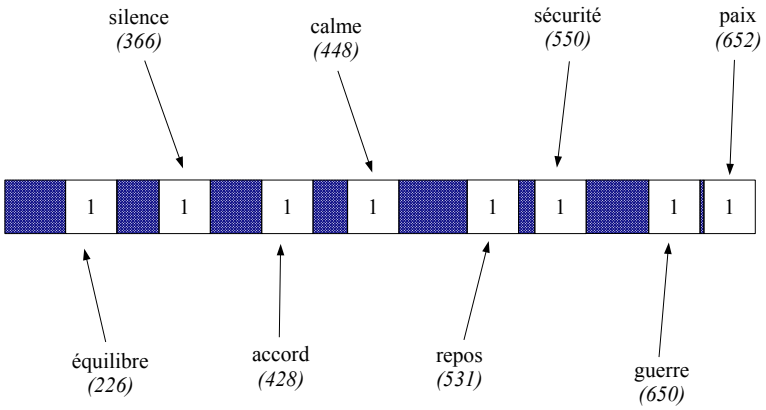


FIG. 2.5 – Vecteur sémantique génératif du concept *PAIX* avant normalisation

2.2.2.2 Objets lexicaux

La base lexicale sémantique ne comporte qu'un seul type d'objets lexicaux : le type ITEM LEXICAL. Les ITEMS LEXICAUX sont générés automatiquement à partir d'une version électronique de la partie index du thésaurus Larousse (cf. 1.4.5.3). Pour chaque entrée, on construit un ITEM LEXICAL dont les caractéristiques sont :

- **identifiant** : l'entrée du thésaurus ;
- **morphologie** : composée des *catégories grammaticales* (*nom, pronom, adjectif, adverbe, etc.*), du *genre* (*masculin, féminin, neutre*) et du nombre (*singulier, pluriel*) tels que les donne le thésaurus ;
- **vecteur sémantique** : la somme vectorielle normée des vecteurs génératifs correspondant aux concepts énoncés par le thésaurus pour l'entrée. Les vecteurs sémantiques fusionnent donc l'ensemble des sens des mots.

Ce choix de construction, un seul objet lexical par item, ne permet donc pas une réelle désambiguïsation, en tout cas dans le sens que l'on entend habituellement qui est un choix (ou

une préférence) parmi plusieurs solutions possibles. Nous allons le voir maintenant avec l'analyse sémantique, le sens global du texte émerge grâce aux vecteurs et à la contextualisation faible.

Si nous prenons l'exemple de ‹*botte*› dont la définition extraite du thésaurus est « *#nom commun# sports, agriculture, enseignement, herbes_et_fougères, pluie* », nous avons :

- **identifiant** : botte ;
- **morphologie** : nom commun ;
- **vecteur sémantique** : $V(\textit{SPORTS}) \oplus V(\textit{AGRICULTURE}) \oplus V(\textit{ENSEIGNEMENT}) \oplus V(\textit{HERBES ET FOUGÈRES}) \oplus V(\textit{PLUIE})$

Si nous prenons l'exemple de ‹*manger*› dont la définition extraite du thésaurus est « *#v# prodigalité gloutonnerie difficulté nutrition dents creux #nom commun# repas* », nous avons :

- **identifiant** : manger ;
- **morphologie** : v, nom commun ;
- **vecteur sémantique** : $V(\textit{PRODIGALITÉ}) \oplus V(\textit{GLOUTONNERIE}) \oplus V(\textit{DIFFICULTÉ}) \oplus V(\textit{NUTRITION}) \oplus V(\textit{DENTS}) \oplus V(\textit{CREUX}) \oplus V(\textit{REPAS})$

Ces deux exemples montrent bien que les vecteurs sémantiques fusionnent à la fois les morphologies et les idées des différents sens des mots.

2.2.3 Analyse sémantique de textes en remontée-redescente grâce aux vecteurs sémantiques

2.2.3.1 Algorithme

L'analyse sémantique des textes en remontée-redescente grâce aux vecteurs conceptuels est définie par les algorithmes 1 et 2.

Algorithme 1: analyse : algorithme d'analyse sémantique avec les vecteurs sémantiques

Entrée : vecteur sémantique $V_{contexte}$, A arbre morpho-syntaxique du texte, seuil s
Sortie : vecteur sémantique du texte
Vecteur V = analyse ($V_{contexte}$, A.racine)
répéter
  Vecteur $V_2 = V$
  V = analyse_VS($V_{contexte}$, A.racine)
jusqu'à $(D_A(V, V_2) < s)$;
retourner V

Cet algorithme 1 est aussi utilisé pour une analyse sémantique avec les vecteurs conceptuels (cf. 2.3.7) avec un appel à analyse_VC au lieu d'un appel à analyse_VS.

2.2.3.2 Principe

L'analyse sémantique des textes grâce aux vecteurs sémantiques est basée sur le principe suivant. Dans une première phase, les feuilles se voient affectées d'un vecteur d'idées correspondant à celui de l'item correspondant. Ce vecteur est contextualisé par un vecteur contexte. Dans le cas général, ce vecteur est le vecteur nul, toutefois si on possède des informations sur le thème du texte on peut utiliser un vecteur contexte plus approprié. Ensuite, une remontée vers le sommet de l'arbre est effectuée. Les vecteurs de chaque nœud sont calculés à partir des vecteurs de leurs fils et de pondérations fonction de leur rôle syntaxique. Le vecteur de chaque nœud est ainsi calculé récursivement jusqu'au sommet de l'arbre. Ce vecteur possède les idées contenues dans tout mots du textes. À ce moment du calcul, il n'y a eu, dans le cas général, aucune contextualisation. Le vecteur du sommet de l'arbre contient donc les idées pertinentes du textes mais aussi beaucoup de bruit. Il s'agit donc de faire passer aux feuilles de l'arbre le

Algorithme 2: analyse_VS : algorithme d'analyse sémantique avec les vecteurs sémantiques

Entrée : vecteur sémantique $V_{contexte}$, nœud N
Sortie : vecteur sémantique du sous-arbre de N
si *N est une feuille* **alors**
  si *N.item.estUnMotVide()* **alors**
    N.vecteur=$\vec{V_0}$
  sinon
    N.vecteur= vecteur(N.item)
  si *N.estGouverneur* **alors**
    N.vecteur = 2⊙N.vecteur
  retourner *N.vecteur*
sinon
  $V = \vec{0}$
  pour *chacun des fils* f_i *de N* **faire**
    $V = V + analyse(V_{contexte}, f_i)$
    N.vecteur=norm(V)

vecteur contexte global et ensuite de sélectionner les idées les plus pertinentes de chaque vecteur grâce à une contextualisation faible. On effectue ensuite une nouvelle remontée toujours avec les pondérations précédentes. Une analyse sémantique nécessite un certain nombre de remontées-redescentes tant que le vecteur conceptuel ne s'est pas relativement stabilisé c'est-à-dire tant que la distance angulaire entre deux vecteurs du sommet calculée n'est pas inférieure à un certain seuil s.

En général le nombre de remontées-descentes est inférieur à 3, cette méthode est ainsi très légère en temps de calcul. L'expérience sur la catégorisation que nous présentons en 2.2.4 et qui concernait 5222 dépêches d'agences a duré à peu près une demi-heure toutes opérations comprises (analyse morpho-syntaxique, analyse sémantique puis catégorisation des documents) sur un ordinateur standard (Pentium IV, 2.4 Ghz, 512 Mo RAM, 4771 bogomips).

2.2.3.3 Exemple

La figure 2.6 présente un petit exemple d'analyse sémantique grâce aux vecteurs sémantiques avec le segment textuel « *la souris d'ordinateur.* ». Les feuilles correspondant aux mots vides de sens (‹*le*›, ‹*de*›) se voient affectées d'un vecteur vide tandis que les autres se voient affectées du vecteur correspondant à l'item de la feuille dans la base. Le vecteur du nœud 4 est constitué de la somme vectorielle pondérée des vecteurs des nœuds 5 et 6. Le nœud 6 est gouverneur, il a donc un poids supérieur (2 dans l'exemple). De même, pour le calcul du nœud 1, le vecteur du nœud 3 aura un poids prépondérant sur les vecteurs des nœuds 2 et 4. Le vecteur général du texte, ici 1, est ensuite utilisé pour faire une contextualisation faible avec les vecteurs des feuilles de l'arbre. Le principe de la remontée est le même. On remarquera que le vecteur général du texte s'oriente vers une idée d'*INFORMATIQUE* dès la première remontée dans cet exemple et que la contextualisation ne pourra qu'accentuer le phénomène.

2.2.4 Exemple d'application des vecteurs sémantiques : catégorisation automatique de documents

Dans sa thèse [Jaillet, 2005], Simon Jaillet étudie diverses méthodes de *catégorisation de documents*. Deux voies sont suivies. Dans une première, il exploite des méthodes de fouille de

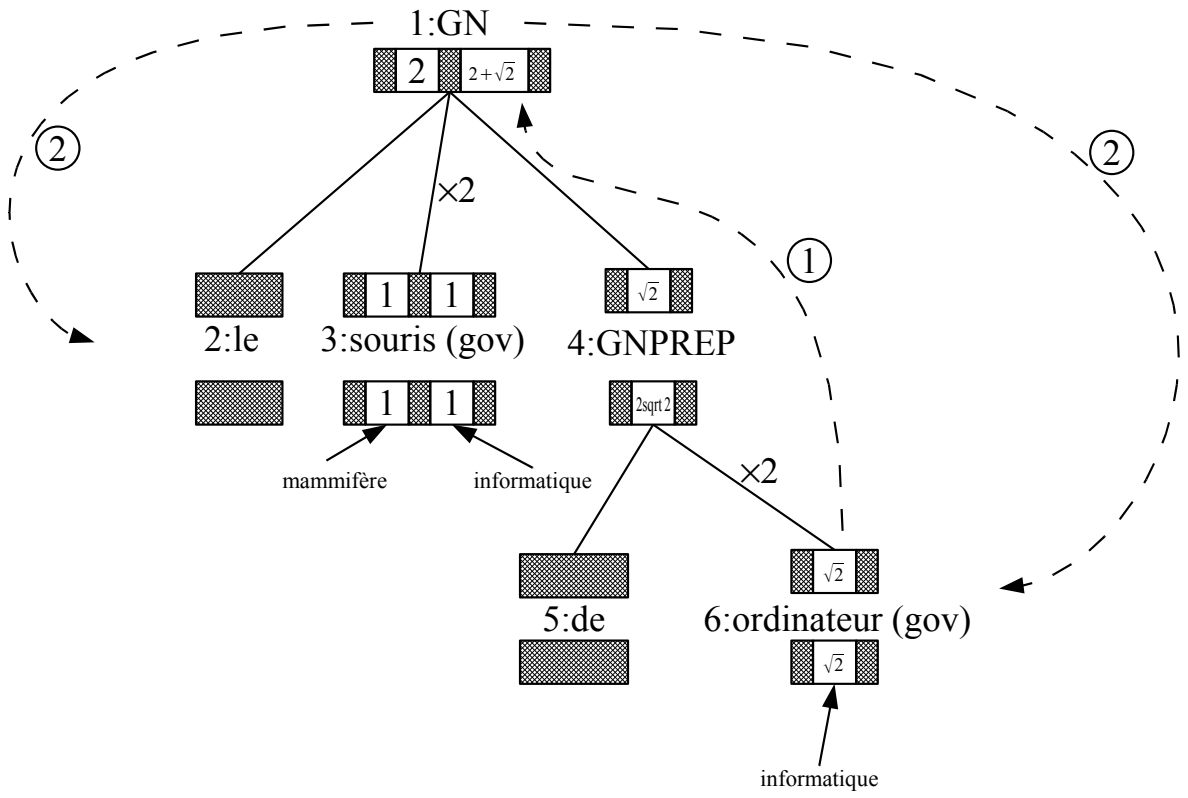


FIG. 2.6 – Exemple d'analyse sémantique grâce aux vecteurs sémantiques

données (motifs séquentiels) tandis que dans une seconde, il utilise des méthodes issues du TALN pour représenter les documents. On trouve ici l'une des applications les plus abouties utilisant les vecteurs sémantiques[57].

Ce type de vecteurs est particulièrement creux puisque pour un terme, on trouve au maximum une demi-douzaine de composantes non vides sur les 873 que comptent les vecteurs sémantiques. Cette caractéristique favorise la discrimination entre vecteurs et fait ainsi de la catégorisation de textes une application qui leur est particulièrement destinée.

Deux expériences sont menées avec les vecteurs sémantiques. Il s'agit dans les deux cas de retrouver automatiquement de quelles catégories sont issus les textes d'un corpus pré-catégorisé par des spécialistes. Ce corpus comprend 28 catégories qui vont des *nouvelles internationales* à l'*aéronautique* en passant par les *finances* ou les *télécoms*. 8022 dépêches d'agences de presse sont réparties entre ces catégories. Pour l'expérience, elles sont constituées en un jeu d'entraînement pour la méthode de catégorisation de 2800 articles et un jeu de test de 5222 articles. Dans la première expérience, un seul vecteur permet de représenter l'ensemble d'un texte tandis que dans la seconde dite *méthode des sous-textes*, il est représenté par trois vecteurs sémantiques : un premier pour l'introduction, un deuxième pour la conclusion et un dernier pour le corps. Quelle que soit la méthode de catégorisation utilisée (*méthode des deux écarts*, *Support Vector Machine*[58]), la méthode des sous-textes s'avère meilleure. En effet, pour des textes courts les résultats sont quasi-identiques (gain sur le F_{score} inférieur à 1%) tandis que sur les textes plus

[57]Simon Jaillet utilise tout au long de sa thèse le terme *vecteur conceptuel* mais il serait plus juste d'employer *vecteur sémantique*.

[58]Support Vector machine (ou SVM) [Boser *et al.*, 1992] est un algorithme d'apprentissage qui construit un hyperplan optimisant la frontière entre des cas positifs et des cas négatifs pour une tâche donnée.

longs, qui se prêtent donc mieux à cette méthode, le gain est net (de l'ordre de + 4%).

2.2.5 Limites de l'approche "vecteurs sémantiques" pure

Deux critiques semblent pouvoir être faites concernant les vecteurs sémantiques. La première concerne en particulier leur construction. Elle est basée sur une seule source, la version électronique du thésaurus Larousse, qui comme toutes les autres peut difficilement couvrir l'ensemble du lexique. En particulier, aucun nom propre n'y figure et bien entendu aucun néologisme ce qui peut entraîner des difficultés pour véritablement faire émerger la thématique associée à certains textes comme « *Renault a affiché des résultats record en 2004 et prévoit une dégradation de sa rentabilité en 2005 dans un environnement moins favorable, marqué notamment par un impact accru de la hausse des matières premières.* » qui ne contient aucune information autre que ‹*Renault*› sur le domaine *automobile*. Il faut bien comprendre ici que le problème ne vient pas de l'utilisation du thésaurus Larousse en particulier mais du fait qu'il n'y a qu'une seule source.

La deuxième critique que l'on pourrait faire des vecteurs sémantiques concerne ce qu'on appelle l'*hybridation vectorielle*. En effet, fusionner l'ensemble des acceptions d'un terme peut conduire à faire émerger des idées qui ne coexistent pas dans le même sens. Par exemple, si on considère que l'item ‹*souris*› a deux acceptions l'une **animal** caractérisée par des idées de ANIMAL, GRIS, FROMAGE et l'autre ‹*souris*› d'**ordinateur** caractérisée, elle, par INFORMATIQUE, dans une phrase comme « *Il posa son pain au fromage près de la souris de son ordinateur* », les idées émergentes pour le vecteur associé à ‹*souris*› seront INFORMATIQUE et GRIS, ce qui ne correspond pourtant pas à un sens de l'item.

2.2.6 La méthode mixte : une approche combinant des vecteurs sémantiques et distributionnels

Pour pallier ces limites, [Jaillet, 2005] introduit la méthode mixte. Il s'agit de tirer profit à la fois des avantages de la représentation sous forme de vecteurs saltoniens (cf. 1.2.2) et des avantages de la représentation sous forme de vecteurs sémantiques. Comme le note l'auteur, « *la représentation statistique permet de mettre en évidence le vocabulaire discriminant tandis que la représentation conceptuelle permet, quant à elle, d'obtenir une vision globale du texte en projetant ce dernier sur un ensemble de concepts.* »

En pratique, les vecteurs utilisés pour représenter les textes sont définis comme la concaténation du vecteur statistique et du vecteur sémantique. Dans le vecteur résultant, les composantes issues du vecteur distributionnel sont largement plus nombreuses que celles issues du vecteur sémantique. On peut estimer le rapport en fonction du nombre d'items dans le corpus comme pouvant aller de 1 à 10 voire de 1 à 100 dans un cas extrême (environ 1000 concepts pour la représentation sous forme de vecteurs sémantiques contre 100000 items en langue). Il faut donc utiliser une méthode de catégorisation qui tienne compte de ce phénomène afin que les composantes "idées" ne soient pas complètement noyées dans les composantes statistiques. Des méthodes comme les SVM ou les réseaux de neurones donnent à chacune des dimensions une importance qui lui est propre.

Grâce à la méthode mixte, les termes inconnus de la base de données vectorielle se voient affectés d'un vecteur dont les composantes ne sont pas toutes vides et ils participent ainsi à la représentation globale du texte.

Sur le corpus en français présenté dans la partie 2.2.4, les résultats obtenus par la méthode mixte ($F_{score} = 0.51$) se voient largement améliorés, que ce soit par rapport aux résultats obtenus avec les vecteurs sémantiques purs ($F_{score} = 0,404$; +26%) ou par rapport à ceux obtenus en utilisant les vecteurs saltoniens purs ($F_{score} = 0,486$; +4,9%). Ces expériences montrent que l'utilisation de représentations conceptuelles permet d'améliorer la représentation textuelle puis-

qu'elle atténue les effets de la polysémie du vocabulaire des textes, contrairement aux vecteurs saltoniens purs qui n'envisagent que les formes de surface.

Il convient de noter que les résultats en termes qualitatifs peuvent paraître modestes avec un $F_{score} = 0.51$ mais nous n'avons présenté ici que l'expérience menée sur le français. Dans cette expérience, les catégories se sont avérées souvent très générales et leurs intersections non vides. Ainsi, il n'est pas rare de trouver un texte dans une catégorie alors qu'il aurait sans doute pu appartenir à une ou plusieurs autres. En revanche, dans l'expérience menée sur l'anglais, même sans analyse morpho-syntaxique préalable puisque l'équipe n'a pas actuellement à sa disposition un équivalent à SYGFRAN pour cette langue, les catégories sont bien mieux définies, ce qui se ressent fortement sur les résultats obtenus avec les trois méthodes. L'expérience a été menée sur le corpus en anglais *Reuters-21578*. Il comporte 12902 dépêches qui ont été réparties en 9603 destinées au corpus d'entraînement et 3299 au corpus de test. 9624 items lexicaux ont été extraits de ce premier corpus pour constituer l'espace vectoriel saltonien. Les vecteurs sémantiques anglais ont été obtenus grâce à la version électronique du thésaurus Roget disponible en ligne[59].

Ici encore, les résultats obtenus par la méthode mixte ($F_{score} = 0,86$) se voient largement améliorés, que ce soit par rapport aux résultats obtenus avec les vecteurs sémantiques ($F_{score} = 0,771$; $+11,54\%$) ou avec les vecteurs saltoniens ($F_{score} = 0,84$; $+2,38\%$).

2.3 Les vecteurs conceptuels

Nous venons de le voir, les vecteurs sémantiques permettent de représenter une thématique globale des segments textuels. Un de leurs grands avantages est leur rapidité dans le traitement des opérations ainsi qu'une relativement faible importance de la taille de la base de données. Toutefois, ils peuvent s'avérer limités pour des traitements plus évolués à cause de leurs limites en ce qui concerne la couverture lexicale ou l'hybridation de vecteurs. L'équipe a ainsi commencé à mettre au point les vecteurs conceptuels.

Les vecteurs conceptuels visent à une représentation ainsi qu'à des traitements plus fins du sens des segments textuels. Ainsi, nous ne considérons pas, comme dans les vecteurs sémantiques, un seul objet lexical pour représenter le sens d'un terme mais plusieurs : les LEXIES. L'objet lexical ITEM LEXICAL correspond au terme et fusionne les informations des différentes lexies. Ces lexies sont constituées à partir d'un apprentissage permanent utilisant des dictionnaires à usage humain. Ce mode de construction de la base de données vectorielles permet d'assurer une certaine couverture lexicale.

2.3.1 Objectifs visés

Les objectifs des vecteurs conceptuels sont hétérogènes. Ils visent à mettre au point ou améliorer toute application dans laquelle une représentation fine du sens pourrait s'avérer utile. On peut ainsi citer la *recherche d'informations* pour laquelle préciser un sens particulier pour un terme peut permettre d'affiner les résultats, la *traduction automatique* où elle peut permettre de trouver le meilleur équivalent possible pour la langue cible. Dans la phrase « *l'élève a fait le devoir que lui avait donné la maîtresse.* », le substantif ‹*devoir*› devrait être traduit en anglais par ‹*homework*› et non par ‹*duty*› qui a plutôt un sens d'‹*obligation morale*›.

2.3.2 Vecteurs génératifs : origine et interdépendance des concepts

Les vecteurs génératifs sont les seuls vecteurs conceptuels à être construits manuellement. Leur construction est basée sur l'hypothèse forte que les concepts ne sont pas indépendants les uns des autres.

[59] ftp://ibiblio.org/pub/docs/books/gutenberg/etext91/roget13.zip

Pour la construction de nos vecteurs génératifs, nous avons choisi un ensemble de concepts issus du thésaurus Larousse [Larousse, 1992] que nous avons présenté en 1.4.5. L'ensemble de concepts considéré correspond à celui des 873 concepts de niveau 4 de cette hiérarchie. L'interdépendance des concepts se situe à deux niveaux dans la construction des vecteurs génératifs : au niveau hiérarchique, c'est-à-dire en tenant compte de la place que le concept a dans la hiérarchie (*vecteurs génératifs hiérarchiquement augmentés*) et à un niveau transversal à cette hiérarchie (*vecteurs génératifs transversalement augmentés*).

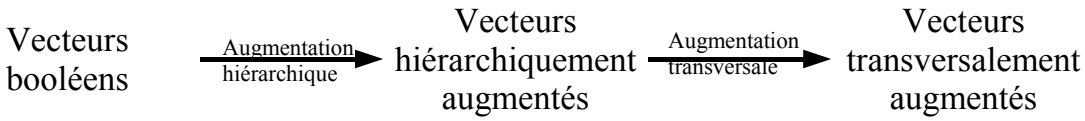


FIG. 2.7 – Séquence d'opérations pour la construction des vecteurs génératifs

2.3.2.1 Interdépendance hiérarchique : vecteurs génératifs hiérarchiquement augmentés

Le point de départ de cette construction est un vecteur booléen. Le vecteur booléen du concept i est le vecteur dont tous les éléments sont à 0 sauf la composante i qui elle est à 1. Cette construction est simple et elle obtient souvent de bons résultats mais semble inadéquate dans plusieurs cas. Il paraît curieux que deux concepts proches comme le sont *GUERRE* et *PAIX* partagent quantitativement autant d'idées que *PAIX* et *CHAMPIGNON*. Nous l'avons déjà dit, les concepts ne sont clairement pas indépendants et leurs vecteurs respectifs doivent en tenir compte. L'ensemble des concepts défini selon [Larousse, 1992] est hiérarchiquement ordonné selon un arbre (cf. 1.4.5). La construction des vecteurs génératifs est basée sur cette structure et plus particulièrement sur la distance ultramétrique entre deux concepts. Il s'agit de la longueur du chemin minimal à parcourir dans l'arbre des concepts pour aller d'un concept à l'autre. Cette distance est définie par :

$$D_u(C,C) = 0 \tag{2.29}$$

$$D_u(C_1,C_2) = \min \begin{bmatrix} D_u(Sup(C_1),C_2)+1 \\ D_u(C_1,Sup(C_2))+1 \end{bmatrix} \tag{2.30}$$

où $Sup(X)$ est le père du concept X. Par définition, on a $Sup(racine(arbre)) = racine(arbre)$. Si nous nous référons à la figure 1.22, nous avons $D_u(tête, membre) = 2$. Tous les concepts frères de tête sont à une distance ultramétrique égale à 2. Nous avons également $D_u(tête, corps) = 1$, $D_u(tête, fonctionsvitales) = 3$ et $D_u(tête, présent) = 8$. La valeur 8 est d'ailleurs la plus grande possible entre deux concepts de cette hiérarchie puisque leur ancêtre commun le plus éloigné est la racine, située au maximum à quatre niveaux au-dessus.

Grâce à cette distance ultramétrique, nous construisons les *vecteurs génératifs hiérarchiquement augmentés*. Appelons X_i le vecteur booléen correspondant au i-ème concept de la base $\mathcal{C}$. Y_i est le vecteur conceptuel hiérarchiquement augmenté défini par :

$$Y_i = X_i \oplus \bigoplus_{j=0}^{\dim(C)} \frac{1}{2^{D_u(C_i,C_j)}} \times X_j \tag{2.31}$$

Le vecteur original X_i est ajouté afin que, de tous les vecteurs Y_j, le plus proche de X_i soit toujours Y_i. La figure 2.8 montre, pour *PAIX*, le vecteur booléen et le vecteur hiérarchiquement augmenté.

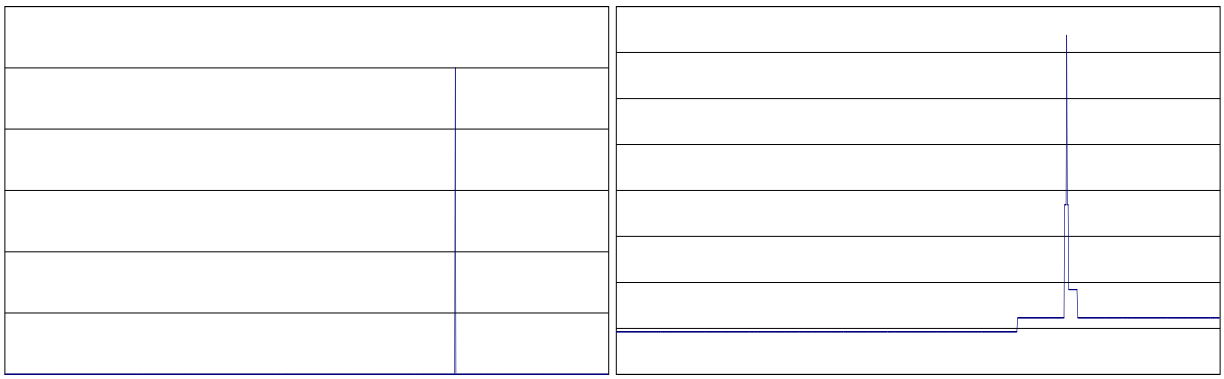


FIG. 2.8 – Vecteur du concept *PAIX* et vecteur hiérarchiquement augmenté du concept *PAIX*

2.3.2.2 Interdépendance transversales : vecteurs génératifs transversalement augmentés

Bien qu'augmenter un vecteur par ses voisins améliore sa qualité, il faut admettre que la hiérarchie des concepts n'est qu'une vue particulière de la façon selon laquelle ils peuvent être organisés. D'autres liens spécifiques peuvent être exhibés. C'est le cas entre *CHAMPIGNON* et *TOXICITÉ* ou *GASTRONOMIE* par exemple. L'augmentation transversale d'un concept C est une opération manuelle réalisée une seule fois, à des ajustements près, qui consiste à énumérer les concepts relatifs à C qui ne sont pas représentés dans la hiérarchie. Le nouveau vecteur est appelé *vecteur transversalement augmenté*.

Par exemple, le concept *PAIX* a comme concepts transversalement associés les concepts *CONCORDE*, *GUERRE*, *CALME*, *SÉCURITÉ*, *REPOS*, *ÉQUILIBRE*. Ces concepts transversaux sont sélectionnés manuellement et peuvent être trouvés dans la partie index de [Larousse, 1992] (cf. 1.4.5.3).

Si Y_i est le vecteur hiérarchiquement augmenté du concept i défini sur C, nous pouvons calculer le i-ème vecteur augmenté transversalement Z_i en faisant la somme pondérée de tous les vecteurs Y_j avec Y_i. Cette construction assure que le vecteur Z_j le plus proche de Y_i demeure Z_i.

$$Z_i = \bigoplus_{j=0}^{dim(C)} \alpha_{ij}(Y_j \oplus Y_i) \tag{2.32}$$

où α_{ij} est la pondération du concept transversal j pour le concept i.

2.3.3 Pourquoi nos vecteurs sont-ils dits "conceptuels" ?

On pourrait penser que le terme ‹*conceptuel*› qui qualifie nos vecteurs provient du fait que leur construction est basée sur ces objets que nous appelons concepts. En pratique, ce nom a été introduit pour marquer le fait que ces vecteurs représentent des idées plus ou moins abstraites, plus ou moins générales.

2.3.4 Architecture et construction de la base

Nous présentons succinctement ici l'architecture de la base lexicale sémantique telle qu'elle est conçue au début de nos travaux. Un certain nombre de nos choix seront débattus plus avant

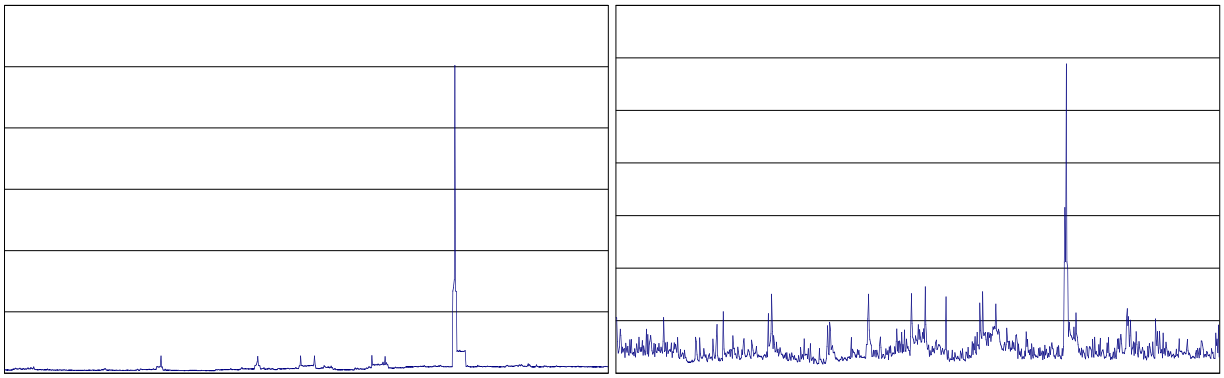


FIG. 2.9 – Vecteurs transversalement augmenté du concept *PAIX* et pour l'item ‹*paix*›

lorsqu'il sera temps, au vue des expériences que nous présentons dans cette partie de la thèse, de mettre en place une architecture de base lexicale sémantique plus pertinente (cf. chapitre 5). Cette architecture est centrée sur le stockage et l'exploitation de deux objets lexicaux l'objet ITEM LEXICAL et l'objet LEXIE.

2.3.4.1 Structure des objets lexicaux

Les objets lexicaux sont composés d'un certain nombre d'*informations linguistiques* :
- un **identifiant** ;
- la **morphologie** composé des *catégories grammaticales* (*nom, pronom, adjectif, adverbe, etc.*), du *genre* (*masculin, féminin, neutre*) et du nombre (*singulier, pluriel*) ;
- la **fréquence en usage** c'est-à-dire le nombre de fois (ou au moins une estimation) où l'objet a été rencontré ;
- un **vecteur conceptuel**.

2.3.4.2 Objets lexicaux

L'architecture est composée de deux sortes d'objets lexicaux, les ITEMS LEXICAUX et les LEXIES. Cette architecture provient essentiellement du mode de fabrication des vecteurs conceptuels c'est-à-dire à partir de dictionnaires à usage humain.

Lexies Les LEXIES constituent le socle de la base lexicale sémantique. À partir de définitions de dictionnaires à usage humain, il est possible d'extraire un certain nombre d'informations linguistiques et de calculer un vecteur conceptuel. Examinons, par exemple, les entrées de [Larousse, 2004] pour l'item ‹*botte*›.

1.botte : #nf# (néerl. bote, touffe de lin) Assemblage de végétaux de même nature liés ensemble : (Botte de paille. Botte de radis.).
2.botte : #nf# (#ethym-it# botta, coup) . Coup de pointe donné avec le fleuret ou l'épée.
3.botte : #nf# (p.-ê. de bot) . Chaussure à tige montante qui enferme le pied et la jambe généralement jusqu'au genou : (Bottes de cuir).

L'idée est pour chacune de ces définitions de fabriquer un objet LEXIE qui comportera :
- un **identifiant** : habituellement pour les lexies, il est constitué du nom du terme et d'un numéro (ex : *botte.1*, *botte.2*, *chat.1*, ...) ;
- la **morphologie** généralement facile à récupérer puisqu'elle est souvent assez bien délimitée ;

- la **fréquence en usage** qui est estimée par des heuristiques, que nous ne détaillerons pas ici, à partir de la fréquence de l'ITEM LEXICAL ;
- un **vecteur conceptuel** calculé à partir du texte de la définition grâce à une analyse sémantique telle que celle présentée en 2.3.7.

Items Lexicaux Les objets ITEMS LEXICAUX correspondant à un terme sont fabriqués à partir des LEXIES de ce terme. Leur structure est la suivante :

- un **identifiant** qui est généralement le nom du terme (ex : *botte*, *chat*) ;
- la **morphologie** qui rassemble l'ensemble des morphologies des lexies ;
- la **fréquence en usage** qui correspond au nombre de fois où le terme a été repéré dans les textes étudiés par l'analyse sémantique ;
- un **vecteur conceptuel** qui est la somme vectorielle normée des vecteurs conceptuels des lexies.

2.3.5 Apprentissage des objets lexicaux

Comme nous l'avons vu à propos des vecteurs sémantiques, le problème de la couverture lexicale s'avère important dès qu'il s'agit d'analyser des textes. La solution choisie pour pallier ce problème a été, pour les vecteurs sémantiques, de rajouter une représentation vectorielle de type saltonien (cf. méthode mixte section 2.2.6). Pour les vecteurs conceptuels, le choix s'est porté sur un apprentissage automatique. Ainsi, la fabrication des objets lexicaux se fait à partir de définitions extraites de dictionnaires à usage humain sous forme électronique. On crée ainsi une LEXIE par définition puis les objets ITEM LEXICAUX à partir de ces LEXIES.

On voit ici une autre des différences avec les vecteurs sémantiques. Alors que ces derniers sont construits une fois pour toutes et peuvent donc être utilisés tout de suite dans le cadre d'une application, l'apprentissage ne le permet pas dans le cas des vecteurs conceptuels. Nous reviendrons plus en détail au chapitre 5 sur les raisons qui nous ont poussés à faire ce choix.

2.3.5.1 Lexies : apprentissage à partir de définitions issues de dictionnaires classiques

L'apprentissage à partir de définitions n'est pas sans poser un certain nombre de problèmes. Il s'agit d'extraire d'un dictionnaire la ou les entrées correspondant à un terme. À cause de la polysémie ainsi que de l'homonymie, nous pouvons avoir plusieurs définitions pour une même entrée. Nous avons divisé en trois étapes successives le traitement d'une entrée d'un dictionnaire comme nous le présentons sur la figure 2.10 : (1) un *prétraitement* dont l'objectif est de préparer les données avant les 2 étapes suivantes. Il consiste à séparer les différentes définitions et à les unifier dans un même format prédéfini puis à préparer la définition en vue d'une analyse sémantique. (2) Une *extraction des informations lexicales* et en particulier de la morphologie et enfin (3) le *calcul d'un vecteur conceptuel* à partir de la définition formatée.

Prétraitement des données La première étape consiste à prétraiter les données. En effet, les définitions sont extraites de dictionnaires hétérogènes dont le formatage et les informations disponibles peuvent fortement différer à la fois pour des raisons purement techniques (codage utilisé, format des données : XML, HTML, ...) ainsi que pour des raisons formelles (séparation des sens, des homonymes, ...).

Cette opération d'unification du formatage est fondamentale car elle consiste à convertir le format des données vers un format conçu pour faciliter l'extraction des informations lexicales ainsi que le calcul du vecteur conceptuel des définitions. Cette partie de l'apprentissage est totalement *ad hoc*. Elle doit être conçue pour chaque dictionnaire afin que les deuxième et

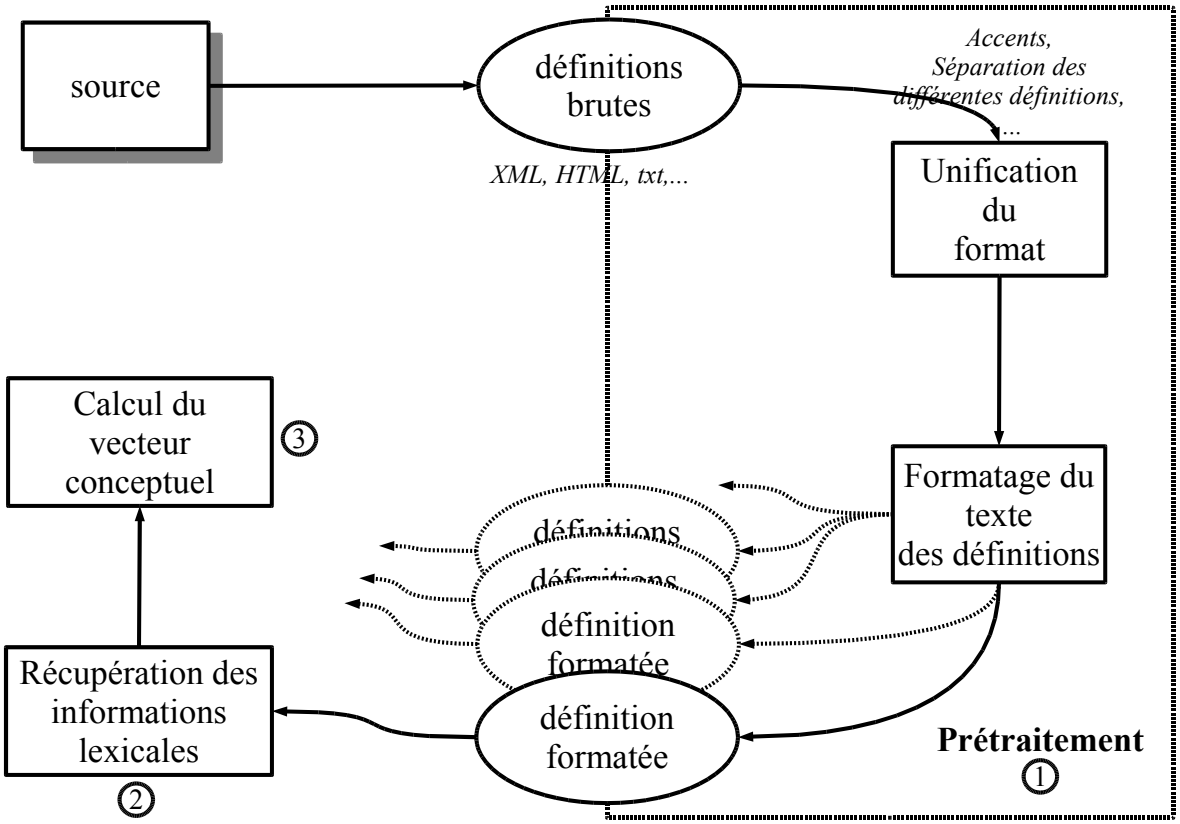


FIG. 2.10 – Séquence d'opérations pour l'apprentissage de vecteurs conceptuels à partir d'une source

troisième étapes soient les mêmes, quelle que soit la source. Le prétraitement des données effectue deux catégories de tâches : une *unification du format* et un *formatage du texte des définitions*. On appelle *définitions non formatées* les textes bruts récupérés depuis les sources et *définitions formatées* les textes obtenus après ce prétraitement.

Unification du format Les dictionnaires utilisés sont sous forme électronique. Les formats des données sont très hétérogènes. Nous pouvons avoir de simples textes comme ceux des dictionnaires papiers, du format HTML pour les dictionnaires disponibles en ligne ou bien du XML. Il est clair que suivant le format, il est plus ou moins simple de repérer les informations pertinentes : pour le XML, langage balisé généralement de façon assez claire, beaucoup plus simplement que pour le simple texte dont il faut chercher à repérer de façon souvent beaucoup plus heuristique les schémas permettant de détecter la morphologie et les séparations entre les différentes définitions. En revanche, un certain nombre de caractères spéciaux (lettres avec diacritiques, symboles, ...) devront être convertis dans le cas du HTML.

Formatage du texte des définitions Cette opération vise à préparer l'apprentissage des vecteurs conceptuels grâce aux textes des définitions. Le formatage est employé à plusieurs niveaux :

– *Gestion du métalangage* L'apprentissage va s'effectuer grâce à une méthode d'analyse sémantique telle que celle présentée en 2.1.6. Comme nous l'avons vu, il est parfois nécessaire de préformater les textes afin de simplifier l'apprentissage. L'étude des définitions pose en

particulier des problèmes essentiellement à cause du *métalangage*, c'est-à-dire le langage utilisé pour structurer le discours. Un certain nombre de tournures caractérisent ce métalangage et ne sont pas porteuses de sens : ‹*se dit de*›, ‹*relatif à*›, ‹*action de*›, ‹*nom usuel de*›, ‹*Abr. de*›, ...
Ce prétraitement consiste à rechercher ces tournures dans les définitions et à les remplacer par un symbole. L'analyseur sémantique lorsqu'il le rencontre ne lui attribue aucune sémantique. Ainsi, ces symboles ne seront absolument pas utilisés dans le calcul des vecteurs car considérés comme non porteurs de sens par l'analyseur sémantique. Nous avons répertorié à ce jour[60] un peu moins de 80 tournures. Ce chiffre est en constante augmentation au fur et à mesure de la découverte de nouveaux cas, phénomène de plus en plus rares toutefois.

– *balisage de la morphologie* Cette opération est assez simple, la morphologie étant généralement bien indiquée.
– *formatage des informations thématiques* Ce prétraitement exploite aussi des informations qui peuvent permettre par la suite d'aider l'analyse sémantique. C'est le cas en particulier du domaine qui est un très bon indice du champ sémantique des items constituant une définition. En pratique ceux-ci sont remplacés par des concepts (par exemple, le domaine *COMM.* sera remplacé par COMMERCE et *BOURSE* par BOURSE). De même, certains dictionnaires fournissent des résumés d'où on peut extraire certaines informations simples. Par exemple, un résumé indiquant *poète français* sera annoté par le concept POÉSIE tandis qu'un résumé indiquant *dramaturge* sera annoté par THÉÂTRE.

Extraction des informations lexicales La deuxième étape de l'apprentissage de lexies consiste à extraire les informations lexicales. Pour l'instant, cette partie de l'apprentissage ne consiste qu'en l'extraction des informations concernant la morphologie. Nous verrons dans les chapitres suivants l'ajout d'autres informations lexicales en particulier des informations concernant les relations lexicales.

Calcul des vecteurs conceptuels L'analyse des définitions se fait grâce à la méthode d'analyse sémantique présentée en 2.3.7. Les définitions formatées au cours du pré-traitement sont analysées et le vecteur résultant de l'analyse est affecté à la lexie.

2.3.5.2 Noyau

Afin d'amorcer le système d'apprentissage, une partie des termes est indexée de façon manuelle. Ce noyau est constitué d'un ensemble de termes choisis parmi les plus courants et/ou les plus polysémiques. Il s'agit donc de fabriquer manuellement les LEXIES de ces items lexicaux particuliers. L'identifiant et la morphologie sont remplis de manière triviale, la fréquence est, comme pour tous les objets lexicaux, nulle à la création. Le vecteur conceptuel associé à la lexie est fabriqué par la somme pondérée des vecteurs génératifs correspondant aux concepts présents dans cette lexie. Par exemple, nous avons pour l'item lexical ‹*paix*› les lexies manuellement indexées suivantes :

– **identifiant** : paix.1
– **morphologie** : [NOM]
– **fréquence en usage** : 0
– **vecteur conceptuel** : $V(\textsc{paix}) \oplus \frac{1}{2}V(\textsc{guerre}) \oplus V(\textsc{sécurité}) \oplus \frac{1}{2}V(\textsc{accord})$

[60] Mars 2005

- **identifiant** : paix.2
- **morphologie** : [NOM]
- **fréquence en usage** : 0
- **vecteur conceptuel** : $V(REPOS) \oplus V(CALME) \oplus V(SILENCE) \oplus \frac{1}{2}V(ÉQUILIBRE)$

Deux sens ont été indexés pour ‹*paix*›. Le premier se réfère à une *absence de guerre* et le deuxième à une *situation de calme*. L'énumération des concepts pondérés est une tâche difficile car subjective. Dans notre expérimentation, nous laissons cette tâche aux lexicographes qui ont créé le thésaurus Larousse [Larousse, 1992] (cf. 1.4.5) pour les concepts ainsi que les dictionnaires [Larousse, 2001b] et [Robert, 2000] pour les découpages de sens. Seuls les mots parmi les plus importants sont ainsi décrits dans le noyau. Nous nous reposons sur l'apprentissage automatique pour l'indexation en masse. Toutefois, les séparations délicates de sens nécessitent des ajustements manuels.

Les items lexicaux de ce noyau sont considérés comme pertinents. Cet ensemble constitue la base d'items lexicaux à partir de laquelle démarre l'apprentissage (cf. figure 2.11). Nous cherchons à mettre au point un apprentissage qui soit le plus cohérent possible afin d'obtenir une base augmentée pertinente. Dans ce chapitre, cet apprentissage se fait uniquement à partir de dictionnaires mais, dans les suivants, il sera basé, en partie, sur les relations sémantiques, en particulier symétriques, comme la synonymie et l'antonymie.

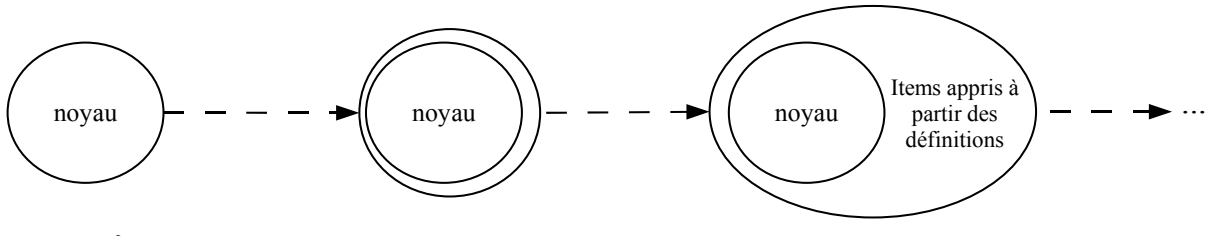


FIG. 2.11 – Augmentation de la couverture lexicale grâce à l'apprentissage

2.3.6 Contextualisation forte

2.3.6.1 Définition

Tandis que la méthode de contextualisation faible peut être utilisée avec n'importe quel vecteur (cf. 2.1.4.5), la méthode de contextualisation forte est utilisée, elle, avec les ITEMS LEXICAUX. L'idée est de considérer les vecteurs conceptuels associés à chacune des LEXIES de l'item en fonction des informations contextuelles disponibles. Ces informations peuvent être alors non seulement d'ordre vectoriel mais aussi d'ordre morphologique. Ainsi, lors d'une analyse sémantique, les informations morphologiques des feuilles peuvent être utilisées grâce à cette méthode. La méthode de contextualisation forte ne fait donc pas un choix parmi les divers vecteurs des LEXIES, mais en favorise certains aux détriments d'autres. La fonction de contextualisation forte est ainsi définie par :

$$\omega \times m \times \vartheta \rightarrow \vartheta \quad : \quad V = \Gamma(I, M_c, V_c)$$

où ω est l'ensemble des ITEMS LEXICAUX, m l'ensemble des morphologies et ϑ l'ensemble des vecteurs conceptuels.

Il s'agit ici d'un principe général et diverses méthodes ont été testées. Celle qui suit est actuellement utilisée dans l'expérimentation effectuée sur les vecteurs conceptuels au cours de

ma thèse dont l'implémentation est présentée en 5.4. D'autres expérimentations ont été menées par Mathieu Lafourcade sur sa propre base de vecteurs conceptuels.

$$\Gamma(I, M_c, V_c) = \bigoplus_{L_i}(P_i \odot V(L_i))$$

$$\text{où } P_i = P_{morpho}(L_i, M_c)^l \times P_{ang}(L_i, V_c)^m \times P_{freq}(L_i)^n \tag{2.33}$$

$$\text{et } l, m, n \in \{0, 1\}$$

où I représente l'item dont on cherche le vecteur contextualisé et qui est composé d'un ensemble de LEXIES $\{L_1, L_2, \ldots, L_i, \ldots, L_n\}$, P_{morpho} le poids morphologique, P_{ang} le poids angulaire et P_{freq} le poids de la fréquence, tous trois définis ci-après.

Le poids P_i est donc la moyenne géométrique pondérée des poids morphologique, angulaire et de fréquence des lexies. Suivant les utilisations de la méthode de contextualisation forte, l'une ou l'autre des informations contextuelles peut ainsi être favorisée ou au contraire être ignorée. Ainsi, dans le cas de l'apprentissage, les exposants l, m, n valent 1, tandis que dans le calcul du vecteur d'un ITEM LEXICAL, aucun des critères n'est considé ($l = m = n = 0$), ce qui correspond à la somme vectorielle normée des vecteurs des LEXIES (cf. 2.3.4.2).

Sauf indication contraire, nous utiliserons, dans la suite de cette thèse, une méthode de contextualisation forte où $l = m = n = 1$.

2.3.6.2 Poids angulaire

$$\iota \times \vartheta \rightarrow [0, k] \ : P_{ang}(L, X) = Min(k, cotan(D_A(\mathrm{V(L)}, \mathrm{X}))) \tag{2.34}$$

où *cotan* est la fonction cotangente, l'inverse de la fonction tangente ($\frac{1}{tan(x)}$). Expérimentalement, nous avons posé $k = 10$.

2.3.6.3 Poids de la fréquence

$$\iota \rightarrow [0, 1] \ : P_{freq} = \frac{freq(L)}{freq(item(L))} \tag{2.35}$$

où $item(L)$ renvoi l'ITEM LEXICAL correspondant à la LEXIE L et $freq(x)$ renvoie la fréquence de l'objet lexical x.

2.3.6.4 Poids et distance morphologique

Soit le poids morphologique P_{morpho} défini par :

$$m \times m \rightarrow [0, \frac{\pi}{2}] \ : P_{morpho}(M_1, M_2) = \frac{\pi}{2} - arctan(D_{morpho}(M_1, M_2)) \tag{2.36}$$

M_1 et M_2 sont des ensembles au sens mathématique du terme. Par exemple, la morphologie de ‹*botte*› peut être vue comme l'ensemble à deux éléments $\{nom, masculin\}$ et celle de ‹*orgues*› comme l'ensemble à trois éléments $\{nom, masculin, pluriel\}$.

Par définition, on pose :

$$P_{morpho}(M_1, M_2) = \frac{\pi}{2} \text{ si } M_1 = \varnothing \text{ ou } M_2 = \varnothing. \tag{2.37}$$

La mesure du poids est calculée comme suit :

$$D_{morpho}(M_1,M_2) = \sum_{x\in(M_1\cup M_2)-(M_1\cap M_2)} p(X) \text{ où } \begin{cases} p(X)=1 \text{ si } X \text{ est une catégorie grammaticale} \\ p(X)=0,5 \text{ sinon} \end{cases} \tag{2.38}$$

Cette distance est donc uniquement fonction de ce qui sépare les deux morphologies. Par exemple, la distance morphologique entre {*nom*, *masculin*} et {*nom*, *masculin*, *plur*} ne sera que fonction de *plur* et vaudra ainsi 0,5. De fait, nous avons $D_{morpho}(X,X)=0$. Voici quelques exemples de poids et de distances morphologiques :

$morpho_1$	$morpho_2$	D_{morpho}	P_{morpho}
{nom}	{nom}	0	$\frac{\pi}{2}$
{nom, masc}	{nom, masc}	0	$\frac{\pi}{2}$
{nom, masc}	{nom}	0,5	1,11
{nom, masc}	{fem}	2	0,47
{nom, masc}	{nom, fem}	1	0,79
{verbe}	{nom}	1,25	0,68
{verbe, intransitif}	{nom masc plur}	2,75	0,35

2.3.7 Analyse sémantique des textes en remontée-redescente grâce aux vecteurs conceptuels

2.3.7.1 Algorithmes

L'analyse sémantique des textes en remontée-redescente à l'aide des vecteurs conceptuels est permise par les algorithmes 3 (pratiquement identique à celui des vecteurs sémantiques : algo 1) et 4.

Algorithme 3: analyse : algorithme d'analyse sémantique avec les vecteurs conceptuels

Entrée : vecteur conceptuel $V_{contexte}$, A arbre morpho-syntaxique du texte, seuil s
Sortie : vecteur conceptuel du texte
Vecteur V = analyse ($V_{contexte}$, A.racine)
répéter
  Vecteur $V_2 = V$
  V = analyse(V, A.racine)
jusqu'à $(D_A(V, V_2)<s)$;
retourner V

2.3.7.2 Principe

Le principe est de faire descendre les informations du vecteur contexte du texte jusqu'aux feuilles de l'arbre en les enrichissant par les informations contenues dans les nœuds de l'arbre. Dans le cas général où nous n'avons aucune information thématique au début de l'analyse, le vecteur contexte utilisé lors de la première descente est le vecteur nul. Dans d'autres cas, l'analyse de définitions par exemple, si des informations du domaine sont spécifiées, le vecteur contexte utilisé sera celui de ce domaine.

Dans un arbre morpho-syntaxique, les feuilles contiennent les informations sur les items ainsi que leur morphologie dans le texte. Ces deux informations sont utilisées pour calculer les vecteurs correspondant aux contextualisations fortes de ces items (cf. 2.3.6) et les affecter à chacune des feuilles de l'arbre. Les feuilles qui correspondent à des mots vides de sens (déterminants, conjonction de coordination, préposition, ...) se voient affecter un vecteur vide.

Algorithme 4: algorithme d'analyse sémantique avec les vecteurs conceptuels : analyse

Entrée : vecteur conceptuel $V_{contexte}$, nœud N
Sortie : vecteur conceptuel du sous-arbre de N
si *N est une feuille* **alors**
  si *N.item.estUnMotVide()* **alors**
    N.vecteur=$\vec{V_0}$
  sinon
    N.vecteur=Γ(N.item, N.morpho, $V_{contexte}$)
  si *N.estGouverneur* **alors**
    N.vecteur = $2 \odot$N.vecteur
  retourner *N.vecteur*
sinon
  $V = \vec{0}$
  pour *chacun des fils* f_i *de N* **faire**
    $V = V + analyse(\gamma(N.vecteur, V_{contexte}), f_i)$
    N.vecteur=norm(V)

La remontée se fait alors de la même manière que celle de l'analyse avec les vecteurs sémantiques. Les vecteurs de chaque nœud sont calculés à partir des vecteurs de leurs fils et de pondérations calculées en fonction de leur rôle syntaxique. Le vecteur de chaque nœud est ainsi calculé récursivement jusqu'au sommet de l'arbre. Ce vecteur possède les idées contenues dans tout mot du texte. À ce moment du calcul, il n'y a eu, dans le cas général, aucune contextualisation. Le vecteur du sommet de l'arbre contient donc les idées pertinentes du texte mais aussi beaucoup de bruit. On effectue à nouveau une descente. On calcule la contextualisation faible du vecteur de chacun des nœuds en fonction de celui de son père. Ainsi, le vecteur contexte n'est pas le même pour tous les nœuds de l'arbre mais est plus directement fonction du sous-arbre dont il est ancêtre. Cette solution améliore largement l'analyse dans le cas d'une phrase comme « *La souris d'ordinateur est posée sur la table du vétérinaire.* ». En effet, si on n'utilise pas un tel mécanisme, le sens de souris serait autant influencé par l'idée d'*INFORMATIQUE* contenue dans ‹*ordinateur*› que par celle d'*ANIMAL* contenue dans ‹‹vétérinaire››, ce qui empêcherait la désambiguïsation du texte.

Au niveau des feuilles, on effectue une contextualisation forte puis une remontée. Ces opérations sont renouvelées un certain nombre de fois jusqu'à une relative stabilisation du vecteur général c'est-à-dire tant que la distance angulaire entre deux vecteurs du sommet calculés successivement n'est pas inférieure à un certain seuil s.

2.3.7.3 Exemple

Reprenons l'exemple « *La souris d'ordinateur.* » déjà utilisé pour présenter l'analyse sémantique de textes avec les vecteurs sémantiques dans la partie 2.2.3. La figure 2.12 présente son analyse sémantique grâce aux vecteurs conceptuels. Considérons le cas général où le vecteur contexte est nul. Nous le faisons redescendre jusqu'aux feuilles de l'arbre en faisant une contextualisation faible aux vecteurs de chaque nœud. Bien entendu, lors de cette première descente, tous ces vecteurs seront nuls. Les feuilles 2 et 5 qui correspondent respectivement à un déterminant et à une préposition et qui sont donc des mots vides de sens se voient affectées d'un vecteur nul. En revanche, le nœud 3 se voit affecté du vecteur conceptuel correspondant à la contextualisation forte de ‹*souris*› avec la morphologie *nom fem* et le vecteur contexte nul. La

même opération est réalisée sur le noeud 5 avec ‹*ordinateur*› et *nom masc*.

Lors de la remontée, le nœud 4 est affecté par le vecteur correspondant à la somme vectorielle pondérée entre ses deux nœuds fils 5 et 6. Le vecteur de la feuille est considéré avec un poids de 2 pour cette opération puisqu'il est gouverneur syntaxique du sous-arbre correspondant à « *d'ordinateur* ». Il en est de même pour le vecteur du nœud 3 (‹*souris*›) dans le calcul du vecteur global du texte.

L'opération de redescente-remontée se renouvelle ainsi de suite jusqu'à une stabilisation du vecteur global du texte.

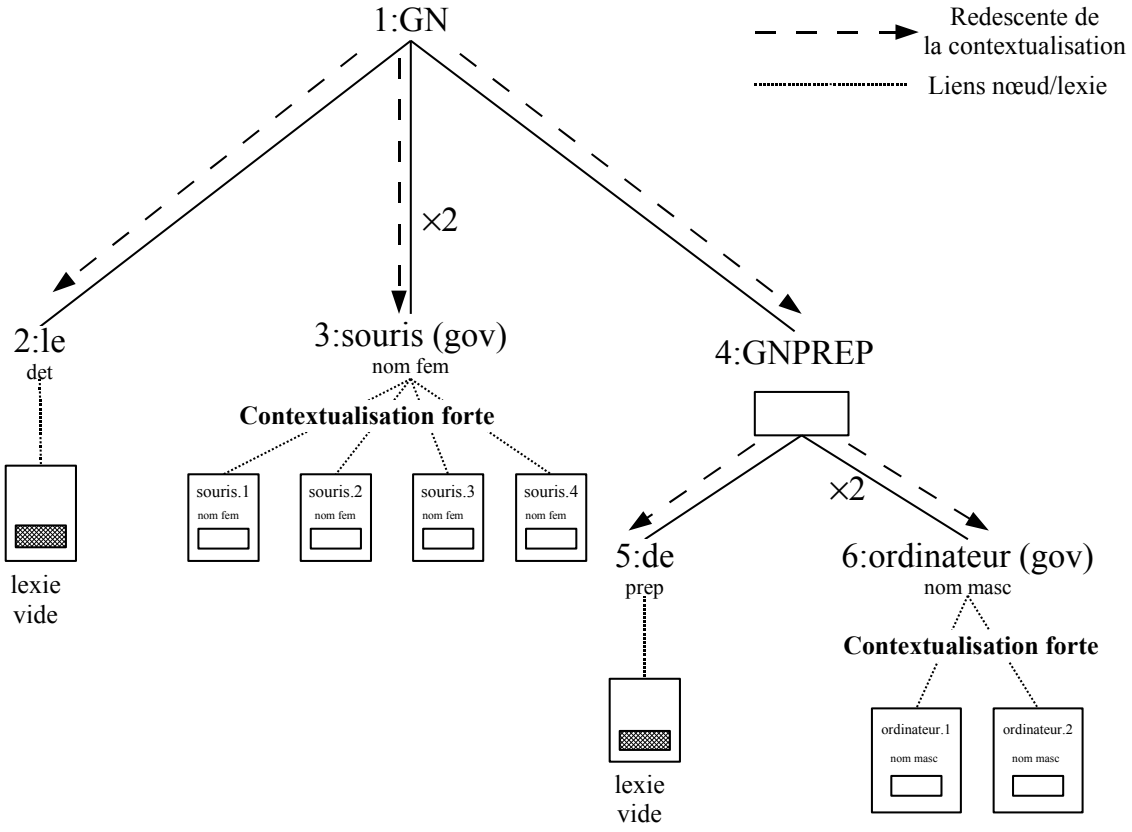


FIG. 2.12 – Exemple d'analyse sémantique grâce aux vecteurs conceptuels

2.3.8 Différences entre l'analyse sémantique avec les vecteurs conceptuels et les vecteurs sémantiques

Deux différences importantes peuvent être mises en évidence entre les deux méthodes d'analyse sémantique :

- La première est en relation directe avec la structure de la base de données vectorielles. Là où il n'y a qu'un seul objet ITEM LEXICAL pour les vecteurs sémantiques et donc l'affectation directe d'un vecteur aux feuilles, il y a plusieurs objets LEXIE pour les vecteurs conceptuels et l'utilisation de la méthode de contextualisation forte pour l'affectation les vecteurs aux feuilles.
- La seconde concerne la propagation du contexte. Dans le cas des vecteurs sémantiques, le vecteur contexte n'est pas modifié par les nœuds internes et donc le contexte utilisé est seulement celui calculé à l'itération précédente pour la racine. Pour les vecteurs concep-

tuels, en revanche, on effectue à chaque nœud interne une contextualisation faible du vecteur de ce nœud par rapport au vecteur de son père. Ainsi, les idées contenues dans le sous-arbre sont prépondérantes par rapport au contexte global.

Nous verrons au chapitre 7 d'autres méthodes d'analyse basées sur des algorithmes à fourmis et sur l'utilisation d'un vaste réseau lexical.

2.4 Bilan comparatif des deux approches

Le tableau de la figure 2.13 tire un bilan comparatif des deux approches.

		vecteurs sémantiques	vecteurs conceptuels
caractéristiques des vecteurs	vecteurs creux	oui	non
	interdépendance hiérarchique	non	oui
	interdépendance transversale	oui	oui
calcul des vecteurs	apprentissage	non	oui
base lexicale	couverture lexicale	non (oui avec la méthode mixte)	oui (grâce à l'apprentissage)
	granularité de représentation	1 (items lexicaux)	2 (items lexicaux, lexies)
	taille	restreinte	grande
	utilisation	directe	après une phase d'apprentissage
analyse sémantique	feuilles	contextualisation faible	contextualisation forte
	nœuds	rien	contextualisation faible

FIG. 2.13 – Tableau récapitulatif des deux approches

Certaines de ces différences sont d'ordre conceptuel, elles tiennent donc aux fondements du modèle et ne peuvent pas être réellement partagées. Le niveau de granularité de la représentation sémantique, un pour les vecteurs sémantiques et deux pour les vecteurs conceptuels, l'existence de l'opération de contextualisation forte chez les vecteurs sémantiques qui en provient directement mais aussi la construction des vecteurs en sont des exemples. En revanche, rien n'empêcherait d'utiliser une redescente du vecteur contexte comme celle utilisée pour les vecteurs conceptuels, ni même de construire des vecteurs sémantiques par apprentissage mais, dans ce cas, ils perdraient leur principal avantage, la rapidité d'exploitation.

2.5 Conclusions du chapitre

Dans ce chapitre, nous avons présenté le modèle de représentation sémantique développé au LIRMM depuis quelques années : le modèle des vecteurs d'idées. Ce modèle peut être relativement rapproché de la notion de linguistique componentielle que nous avions présenté dans le chapitre précédent. Il s'agit de la projection de la notion linguistique de champ sémantique dans le modèle mathématique d'espace vectoriel. Ce modèle repose sur des propriétés mathématiques

bien connues sur lesquelles il est possible de définir des opérations auxquelles sont attachées des interprétations linguistiquement raisonnables. Ainsi, nous avons présenté, entre autres, la distance angulaire qui peut être interprétée comme une distance thématique entre deux objets lexicaux, la somme vectorielle qui réalise l'union des idées entre des objets lexicaux, le produit terme à terme qui sélectionne les idées communes aux objets et l'opération de contextualisation faible qui renforce certaines idées d'un vecteur si elles existent dans un deuxième.

Nous avons aussi présenté la méthode d'analyse sémantique qui permet de calculer le vecteur d'idées d'un texte. Cette méthode est à la base de toutes les applications où peuvent intervenir les vecteurs d'idées. Elle utilise un arbre morpho-syntaxique et considère les vecteurs d'idées des termes du texte en particulier en fonction des informations syntaxiques de cet arbre.

Plusieurs expérimentations pour la construction et l'exploitation de vecteurs d'idées sont menées actuellement au sein du LIRMM. La première, implantée par Jacques Chauché, concerne les vecteurs dits sémantiques. Ceux-ci se caractérisent par la présence dans la base de données d'un seul objet lexical qui regroupe l'ensemble des informations morphologiques et vectorielles d'un item lexical ainsi que par le fait qu'ils sont très creux par construction. Cette dernière caractéristique les destine particulièrement à des tâches de discrimination comme la catégorisation de documents textuels. En revanche, leur construction est basée sur une source unique, ce qui ne leur permet pas une couverture lexicale globale. Cette limite peut être largement compensée par l'utilisation de la méthode mixte qui allie aux vecteurs sémantiques des vecteurs de nature saltonienne. Les expériences menées avec cette méthode ont aussi montré que l'utilisation de vecteurs basés sur les idées améliorait aussi les résultats obtenus uniquement avec des vecteurs basés sur la distributionnalité. Leur relative simplicité permet aux vecteurs sémantiques d'occuper une place restreinte sur disque ainsi que de raccourcir les temps de calcul.

Les deux autres expériences sont menées sur les vecteurs conceptuels. Une première est implantée par Mathieu Lafourcade, tandis que la deuxième a été réalisée par moi-même au cours de cette thèse. Les méthodes utilisées sont conceptuellement assez proches, bien que les implémentations diffèrent relativement. Leur développement a été mené parallèlement et de façon incrémentale. Cette première version de la base vectorielle, que nous avons présentée dans ce chapitre, se caractérise par un apprentissage automatique à partir de dictionnaires à usage humain et par l'existence de deux types d'objets lexicaux : ITEM LEXICAL et LEXIE. Ces objets regroupent les informations de type morphologique, fréquenciel et vectoriel. Il s'agit d'un principe fort qui permet de garantir une certaine couverture lexicale. Cet apprentissage s'effectue grâce à la méthode d'analyse sémantique à partir du texte des définitions. Un noyau de lexies indexées à la main permet d'amorcer le système. L'idée est de rendre possible une analyse cohérente à partir de ce premier ensemble pertinent qui permettra d'obtenir une base de données vectorielle pertinente.

Une des problématiques concernant les vecteurs conceptuels est donc l'apprentissage. Si l'utilisation de dictionnaires à usage humain donne déjà une certaine cohérence à la base, d'autres voies pour son amélioration sont envisagées et examinées au cours de cette thèse. Dans le chapitre suivant, nous commençons à étudier l'utilisation des fonctions lexicales pour l'amélioration de la pertinence de notre base en abordant les fonctions symétriques : synonymie et antonymie.

3

Enrichissement de la base à l'aide des fonctions lexicales symétriques

DANS ce chapitre, nous montrons comment les fonctions lexicales peuvent nous aider à améliorer l'analyse des textes en général et des définitions en particulier. Les fonctions d'antonymie peuvent permettre de gérer certaines tournures négatives, les fonctions d'hyperonymie les cas de définitions aristotéliciennes, les fonctions de synonymie le paraphrasage. Nous identifions deux types de fonctions lexicales : les *fonctions lexicales de construction* et les *fonctions lexicales d'évaluation*. Les premières permettent de construire un vecteur, ce qui est utile, par exemple, dans certains cas de négation ou pour l'analyse des dictionnaires de synonymes, tandis que les secondes évaluent la pertinence entre deux items d'une relation lexicale. Nous introduisons les deux premières modélisations de fonctions lexicales qui ont été développées, celles qui concernent les fonctions symétriques. Nous exposerons en particulier les travaux réalisés sur la synonymie avant mon arrivée dans l'équipe ainsi que les améliorations auxquelles j'ai participé. Nous présentons ensuite les travaux sur l'antonymie réalisés au cours de mon DEA, revus et complétés durant ma thèse. De part leur modélisation, ces fonctions sont utilisables à la fois pour les vecteurs sémantiques et les vecteurs conceptuels. Ce chapitre se termine par les effets constatés sur la base et une réflexion qui porte en particulier sur les limites du modèle purement conceptuel.

Sommaire

DAns le cadre du projet qui nous intéresse ici, nous cherchons à concevoir et à exploiter une représentation du sens des termes basée sur les vecteurs d'idées. Un vecteur rassemble l'ensemble des idées contenues dans un objet linguistique, qu'il s'agisse d'un objet lexical comme un ITEM LEXICAL ou une LEXIE ou bien d'un segment textuel (mot, syntagme, phrase, paragraphe, ...). L'exploitation principale qui en est faite est l'analyse de textes. Celle-ci est, non seulement une composante centrale des problématiques traitées par notre équipe, mais surtout la construction des vecteurs conceptuels s'opère grâce à elle sur des définitions issues de dictionnaires. Utilisée à la fois en exploitation et en construction, l'analyse des textes est donc le nœud central de notre projet. Son amélioration permet ainsi d'améliorer globalement les vecteurs conceptuels ce qui a une influence positive sur l'analyse des textes. C'est ce cercle vertueux que nous visons ici. Il s'agit d'une première manifestation du phénomène de double boucle qui sera, en quelque sorte, un des fils conducteurs de notre thèse.

Dans ce chapitre, nous montrons dans un premier temps comment les fonctions lexicales peuvent nous aider à améliorer l'analyse des textes en général et des définitions en particulier. Les fonctions d'antonymie peuvent permettre de gérer certaines tournures négatives, les fonctions d'hyperonymie les cas de définitions aristotéliciennes, les fonctions de synonymie le paraphrasage. Nous identifions deux types de fonctions lexicales : les *fonctions lexicales de construction* et les *fonctions lexicales d'évaluation*. Les premières permettent de construire un vecteur, ce qui est utile, par exemple, dans certains cas de négation ou pour l'analyse des dictionnaires de synonymes, tandis que les secondes évaluent la pertinence entre deux items d'une relation lexicale. Ces dernières peuvent ainsi participer à la sélection de sens dans le cas de fonctions syntagmatiques qui permettent de mettre en évidence les phénomènes de collocations (cf. 1.3.1.2) ou permettre d'évaluer la pertinence globale d'une base. Par exemple, il est raisonnable de penser que ‹*destin*› et ‹*destinée*› sont plus synonymes que ‹*destin*› et ‹*vie*›. Nous verrons en 3.2.3.1 que la modélisation de la fonction de synonymie vérifie cette intuition, puisque nous avons Syn_P(‹*destin*›, ‹*destinée*›) > Syn_P(‹*destin*›, ‹*vie*›).

Nous introduisons les deux premières modélisations de fonctions lexicales qui ont été développées, celles qui concernent les fonctions symétriques. Nous exposerons en particulier les travaux réalisés sur la synonymie avant mon arrivée dans l'équipe ainsi que les améliorations auxquelles j'ai participé. Nous présentons ensuite les travaux sur l'antonymie réalisés au cours de mon DEA, revus et complétés pendant ma thèse. De part leur modélisation, basée sur les ITEMS LEXICAUX, ces fonctions sont utilisables à la fois pour les vecteurs sémantiques et les vecteurs conceptuels. Ce chapitre se termine par une analyse des effets constatés sur la base et une réflexion sur les limites du modèle purement conceptuel ainsi constatées.

3.1 Fonctions lexicales pour l'analyse

Nous avons vu en 1.3.1 que les termes entretenaient entre eux des relations à la fois sur le plan paradigmatique, ou plan du sens, qui relie les termes entre eux à l'intérieur du lexique (synonymie, antonymie, hyperonymie, ...) et sur le plan syntagmatique, qui relie, lui, les termes à l'intérieur de la phrase (phénomènes de collocations : intensificateurs, verbes supports, ...). Les fonctions lexicales permettent de modéliser ces diverses relations. Dans le cadre de l'analyse des textes en général et celles des définitions en particulier, l'utilisation des fonctions lexicales peut se révéler particulièrement intéressante à deux niveaux : (1) pour permettre de construire des vecteurs conceptuels (*fonctions lexicales de construction*) et (2) pour évaluer la pertinence d'une relation entre deux termes (*fonctions lexicales ! d'évaluation*).

3.1.1 fonctions lexicales de construction

Lors d'une analyse sémantique de texte, l'une des opérations les plus délicates est d'affecter à chaque nœud de l'arbre morpho-syntaxique un vecteur conceptuel qui soit le plus pertinent possible (cf. 2.1.6). Dans de nombreux cas, une simple contextualisation est nécessaire. Dans une phrase comme « *La souris est reliée à l'ordinateur* », le co-texte (les informations données par les mots du texte considéré, ‹*ordinateur*› en l'occurrence) ainsi que le contexte (les informations connues sur le terme par ailleurs : *sens*, *fréquence*, *etc.*) permettent par une opération de contextualisation de calculer un vecteur conceptuel pertinent pour la feuille correspondant à ‹*souris*›. Dans d'autres cas pourtant, cette opération conduirait à des non-sens, à une mauvaise sélection des vecteurs conceptuels. Considérons, par exemple, une définition extraite de [Larousse, 2004] pour le terme ‹*inapproprié*› « *qui n'est pas approprié* ». Il est clair qu'il ne suffit pas de contextualiser l'adjectif ‹*approprié*› pour obtenir un vecteur conceptuel adéquat. Dans ce cas précis, une fonction lexicale de construction d'antonyme est nécessaire. Il s'agit de construire à partir de ‹*approprié*› un vecteur antonyme. De même, dans le cas de l'analyse d'un dictionnaire de synonymes, il s'agira de construire le vecteur d'un synonyme à partir d'une fonction lexicale de construction de synonymes.

3.1.2 fonctions lexicales d'évaluation

Les fonctions lexicales d'évaluation permettent de mesurer la pertinence d'une relation lexicale entre plusieurs termes. Ces fonctions lexicales ont une double utilité pour l'apprentissage :

- dans un *but d'évaluation* en donnant la possibilité d'évaluer la pertinence globale d'une base par la vérification de la correspondance entre les liens existant en langue par rapport à ceux existant dans la base.
- dans un *but de sélection de sens* en particulier pour les fonctions lexicales de type syntagmatiques, c'est-à-dire celles qui caractérisent les phénomènes de collocations.

Nous verrons au chapitre 6 l'utilisation des fonctions lexicales d'évaluation pour construire un vaste réseau lexical ainsi que leur utilisation dans son exploitation.

Nous différencierons les fonctions lexicales d'évaluation que nous noterons avec une majuscule (par exemple, *Anti*, *Syn*, *Hypo*, *Méro*) des fonctions lexicales de construction que nous noterons avec un *C* (par exemple, *Canti*, *Csyn*, *Chypo*, *Cméro*).

3.2 Relations d'équivalence : la synonymie

3.2.1 Définitions et caractérisation de la synonymie

3.2.1.1 Synonymie absolue et quasi-synonymie

La synonymie est « *la relation sémantique qui existe entre deux items lexicaux qui diffèrent sur leur forme mais expriment le même sens* ». Il s'agit d'une relation d'équivalence dont le critère de discrimination est la substitution en contexte ([Nyckees, 1998], p. 180). On peut dire que deux termes (ou segments de texte) sont synonymes si la substitution de l'un par l'autre dans un énoncé ne modifie pas son sens global [Sparck Jones, 1986]. Peu de mots sont parfaitement synonymes. Les exemples souvent cités de cette *synonymie absolue* sont des termes soit issus d'un niveau de langue différent, soit des noms courants comparés à des noms savants. On trouve alors un terme familier (‹*bouffer*›), un terme courant (‹*manger*›), et un terme soutenu (‹*se restaurer*›), ou encore un terme courant (‹*carotte*›) et un terme scientifique (‹*Daucus carota*›). Il faut tout de même reconnaître que dans ces cas, la substitution n'est pas aisée à cause du niveau de langue. Les linguistes considèrent donc souvent que la synonymie absolue n'existe pas. De fait, les dictionnaires de synonymes regroupent plutôt des quasi-synonymes, des termes qui ne sont synonymes que dans certains contextes. C'est le cas, par exemple, des verbes ‹*commander*› et ‹*demander*›. On peut « *demander un café* » ou « *commander un café* » à un serveur, on peut « *demander à voir quelqu'un* » mais on ne peut pas *« *commander à voir quelqu'un* ».

À cause de cette polysémie, la propriété de transitivité n'est pas vérifiée. En effet, on peut considérer que ‹*échouer*› est synonyme de ‹*sécher*› et ‹*sécher*› est synonyme de ‹*déshydrater*› mais on ne peut pas considérer ‹*échouer*› comme synonyme de ‹*déshydrater*›. En tant que relation lexicale, la synonymie n'a donc pas les propriétés des relations mathématiques d'équivalence.

Nous noterons, à la Polguère ([Polguère, 2003], p. 122), les synonymes absolus par le signe mathématique d'équivalence $\equiv$ (‹*carotte*›$\equiv$‹*Daucus Carotta*›) et les synonymes approximatifs par $\cong$ (‹*déshydrater*›$\cong$‹*sécher*›).

3.2.1.2 Notion de synonymie relative

La notion de *synonymie relative* a été introduite pour pouvoir exploiter, dans les nombreux cas où elles peuvent s'avérer utiles, les propriétés d'équivalence. La synonymie relative évalue la possibilité de substituer un item lexical (ou un segment textuel) à un autre, dans le contexte d'un troisième [Lafourcade & Prince, 2001a]. Cette relation est une pseudo-équivalence puisqu'elle vérifie une pseudo-transitivité [Prince, 1991]. En effet, dans un contexte de ‹*déshydratation*›, on a ‹*sécher*›$\cong$‹*dessécher*›, ‹*dessécher*›$\cong$‹*déshydrater*› mais aussi ‹*déshydrater*›$\cong$‹*sécher*›.

3.2.2 Fonctions lexicales de construction d'un vecteur synonyme

Dans l'objectif de l'apprentissage à partir de définitions, il est parfois nécessaire de générer un vecteur synonyme. C'est en particulier vrai pour les définitions issues de dictionnaires de synonymes. Par exemple, on trouve dans un dictionnaire des synonymes comme [Larousse, 2001a] « ‹avion› *:* ‹aéroplane›, ‹jet›, ‹zinc›, ‹coucou›, ‹taxi› ». La méthode d'apprentissage doit être capable de générer un vecteur conceptuel à partir de ces items connus comme synonymes. Nous introduisons donc ici les fonctions de construction de synonymie relative et partielle.

3.2.2.1 Fonction lexicale de construction de synonymie relative

Nous définissons la fonction lexicale de construction d'un vecteur synonyme $Csyn_R$ qui donne le vecteur synonyme aux vecteurs X et à Y dans un contexte C (figure 3.1).

$$\begin{aligned} \vartheta^3 \to \vartheta \quad &: \quad X,Y,C \to Z = Csyn_R(X,Y,C) \\ Csyn_R(X,Y,C) \quad &= \quad \gamma(X,C) \oplus \gamma(Y,C) \end{aligned}$$

FIG. 3.1 – Construction d'un vecteur synonyme : fonction $Csyn_R$

3.2.2.2 Généralisation de la fonction lexicale de construction de synonymie relative

Chaque entrée d'un dictionnaire de synonymes se présente sous la forme d'une ou de plusieurs listes d'items lexicaux. Dans le cas d'une liste unique, les synonymes représentent tous les sens du terme tandis que dans l'autre cas, chaque liste correspond à un de ses sens particulier. Dans certains cas, cette liste est annotée d'une *glose* c'est-à-dire d'une indication textuelle permettant de connaître le contexte dans lequel ces termes sont synonymes. Cette généralisation de la fonction lexicale de synonymie partielle peut servir dans les cas (rares) où l'on rencontre cette situation. Elle nous servira, en tout cas, à définir la fonction lexicale généralisée de synonymie partielle qui est elle, en revanche, utilisée dans l'apprentissage.

Nous définissons donc la fonction lexicale généralisée de construction d'un vecteur synonyme $Csyn_R$ comme la fonction qui donne le vecteur synonyme d'un ensemble de vecteurs $X_1, X_2, \ldots, X_n$ dans un contexte C (figure 3.2).

$$\begin{aligned} \vartheta^n \to \vartheta \quad &: \quad X_1,X_2,\ldots,X_n \to Z = Csyn_R(X_1,X_2,\ldots,X_n,C) \\ Csyn_R(X_1,X_2,\ldots,X_n,C) \quad &= \quad \bigoplus_{i=1}^{n} \gamma(X_i,C) \end{aligned}$$

FIG. 3.2 – Fonction $Csyn_R$ généralisée à n termes

3.2.2.3 Fonction lexicale de construction de synonymie partielle

Dans le cas de l'analyse d'un dictionnaire, il n'y a pas toujours indication du contexte particulier dans lequel les deux termes sont synonymes, il est donc nécessaire d'en trouver un satisfaisant. On peut considérer qu'un tel contexte serait constitué des idées communes aux deux items. À ce stade de notre discours, rappelons que les vecteurs d'idées des OBJETS LEXICAUX rassemblent les divers sens de chaque terme. Pour trouver le ou les sens partagés par les deux termes, il faut donc construire un vecteur correspondant au "dénominateur commun", aux idées communes aux deux termes synonymes. Mathématiquement, ce vecteur peut être le vecteur médian aux vecteurs faiblement contextualisés. Une bonne heuristique dans un cas d'analyse est donc de considérer que le contexte C vaut $\gamma(X, X \oplus Y) \oplus \gamma(Y, X \oplus Y)$.

Nous pouvons ainsi proposer la fonction lexicale de construction partielle d'un synonyme $Csyn_P$ qui calcule le vecteur synonyme des vecteurs X et Y dans un contexte C (figure 3.3).

$$\begin{aligned} \vartheta^2 \to \vartheta \quad &: \quad X,Y \to Z = Csyn_P(X,Y) \\ Csyn_P(X,Y) \quad &= \quad Csyn_R(X,Y,\gamma(X,X \oplus Y) \oplus \gamma(Y,X \oplus Y)) \end{aligned}$$

FIG. 3.3 – Construction d'un vecteur synonyme : fonction $Csyn_P$

3.2.2.4 Généralisation de la fonction lexicale de construction de synonymie partielle

L'idée principale de cette méthode est la même que pour la fonction partielle, il s'agit de trouver le ou les sens partagés par l'ensemble des termes synonymes. C'est-à-dire que, dans ce cas, le contexte est donné par $C = \bigoplus_{i=1}^{n} \gamma(X_i, X_1 \oplus X_2 \oplus \ldots \oplus X_i \oplus \ldots \oplus X_n)$ (figure 3.4).

$$\vartheta^n \rightarrow \vartheta \quad : \quad X_1, X_2, \ldots, X_n \rightarrow Z = Csyn_P(X_1, X_2, \ldots, X_n)$$

$$Csyn_P(X_1, X_2, \ldots, X_n) \quad = \quad Csyn_R(X_1, X_2, \ldots, X_n, \bigoplus_{i=1}^{n} \gamma(X_i, X_1 \oplus X_2 \oplus \ldots \oplus X_i \oplus \ldots \oplus X_n))$$

FIG. 3.4 – Fonction $Csyn_P$ généralisée à n termes

Cette généralisation de la fonction lexicale de synonymie partielle devrait être utilisée avec des termes qui sont synonymes dans le même contexte, c'est-à-dire dans le cas où le dictionnaire sépare les sens mais ne donne pas d'indication sur ce sens ou après classification des termes en fonction de leur sens.

Nous verrons en 3.4 un premier exemple d'utilisation de cette fonction qui ne tient pas compte de la remarque précédente et en 3.5 les effets de ces fonctions de construction sur l'apprentissage. Nous ne chercherons à séparer réellement les sens, et donc à mieux utiliser ces fonctions, qu'au chapitre suivant (section 4.2.1).

3.2.3 Fonctions lexicales d'évaluation de la synonymie

3.2.3.1 Fonction de synonymie relative Syn_R

Principes et définitions Une première fonction de synonymie relative a été proposée avant mon arrivée dans l'équipe par Mathieu Lafourcade et Violaine Prince [Lafourcade & Prince, 2001a] et [Lafourcade & Prince, 2001b]. Elle offre l'avantage d'être rapide à calculer. La fonction de synonymie relative Syn_R est la fonction qui évalue la synonymie entre deux vecteurs conceptuels par rapport à un troisième (figure 3.5).

$$\vartheta^3 \rightarrow [0, \tfrac{\pi}{2}] : \qquad X, Y, C \rightarrow D = Syn_R(X, Y, C)$$

$$Syn_R(X, Y, C) = D_A(\gamma(X, C),\ \gamma(Y, C)) = D_A(X \oplus (X \odot C),\ Y \oplus (Y \odot C))$$

FIG. 3.5 – Calcul de la fonction de synonymie relative Syn_R

Nous introduirons en 4.1.1.2 la fonction de synonymie relative adaptée à la méthode de contextualisation forte qui tient compte des informations vectorielles sur chacune des lexies des termes et aussi des informations lexicales, en particulier morphologiques.

Cette méthode de synonymie relative est difficilement utilisable dans la perspective d'une analyse. En effet, dans ce cas, elle est généralement utilisée à partir de deux termes ou vecteurs sans autre information en particulier sans information contextuelle. Ce problème est analysé et une solution, la méthode de synonymie partielle, est proposée en 4.1.1.2. Pour une évaluation de la base, la fonction de synonymie relative reste toutefois fort utile et nettement plus intéressante que la fonction de synonymie partielle.

Propriétés La fonction de synonymie relative est une distance [Lafourcade *et al.*, 2002]. Elle respecte :

1. la *réflexivité* : $Syn_R(X,X,C) = 0$ La réflexivité est héritée de celle de la distance angulaire (équation 2.6).
2. la *symétrie* : $Syn_R(X,Y,C) = Syn_R(Y,X,C)$ La symétrie pour les deux premiers arguments provient de celle de la distance angulaire (équation 2.7).
3. la *pseudo-transitivité* : Nous avons par héritage de l'inégalité triangulaire de D_A (équation 2.8) : $Syn_R(X,Y,C) + Syn_R(Y,Z,C) \geq Syn_R(X,Z,C)$. Cette propriété implique l'existence de la propriété de transitivité puisqu'elle est plus précise que cette dernière : elle permet de constater que, par rapport à C, la synonymie entre X et Z est au moins égale à la somme des mesures de la synonymie entre X et Y et la synonymie entre Y et Z.

Deux propriétés supplémentaires peuvent être mises en évidence :
– Le vecteur nul $\vec{0}$ ramène la synonymie relative à la distance angulaire. Nous avons donc $Syn_R(X,Y,\vec{0}) = D_A(X \oplus \vec{0},\ Y \oplus \vec{0}) = D_A(X,\ Y)$
– Par héritage du rapprochement de D_A, quel que soit le relatif, la synonymie relative ne peut que rapprocher X et Y. Nous avons donc $Syn_R(X,Y,C) \leq D_A(X,\ Y)$

Résultats Les tableaux de la figure 3.6 contiennent quelques exemples de résultats obtenus avec la fonction de synonymie relative. Dans le premier tableau, la partie supérieure rappelle les résultats sur la distance angulaire D_A(X, Y) présentés en 2.1.3.1 et la partie inférieure les valeurs de $Syn_R(X,Y,\text{‹}vie\text{›})$. Le deuxième tableau indique le rapprochement en pourcentage de la fonction de synonymie relative par rapport à la distance thématique. Toutes les valeurs sont indiquées en radians.

D_A / Syn_R	destinée	destin	vie	existence	mort	automobile	train	action	inaction	réaction
destinée	**0**	0,51	0,82	0,7	0,99	1,29	1,38	1,31	1,14	1,2
destin	**0,35**	**0**	0,83	0,75	0,99	1,3	1,38	1,25	1,07	1,16
vie	**0,63**	**0,64**	**0**	0,61	0,89	1,28	1,35	1,3	1,1	1,2
existence	**0,5**	**0,55**	**0,51**	**0**	0,98	1,37	1,43	1,37	1,25	1,3
mort	**0,76**	**0,76**	**0,69**	**0,78**	**0**	1,33	1,4	1,32	1,15	1,26
automobile	**1,08**	**1,08**	**1,09**	**1,18**	**1,12**	**0**	0,88	1,4	1,22	1,29
train	**1,18**	**1,17**	**1,16**	**1,26**	**1,19**	**0,7**	**0**	1,43	1,3	1,39
action	**1,08**	**1,02**	**1,07**	**1,17**	**1,07**	**1,17**	**1,22**	**0**	1,01	0,67
inaction	**0,93**	**0,87**	**0,9**	**1,06**	**0,9**	**1,0**	**1,08**	**0,8**	**0**	0,9
réaction	**0,97**	**0,93**	**1,03**	**1,1**	**1,02**	**1,06**	**1,17**	**0,54**	**0,72**	**0**

Rapprochement (%)	destinée	destin	vie	existence	mort	automobile	train	action	inaction	réaction
destinée	**0**									
destin	**31,3**	**0**								
vie	**23,1**	**22,9**	**0**							
existence	**28,6**	**26,7**	**16,4**	**0**						
mort	**23,2**	**23,2**	**22,5**	**20,4**	**0**					
automobile	**16,3**	**16,9**	**14,8**	**13,9**	**15,8**	**0**				
train	**14,5**	**15,2**	**14,0**	**11,8**	**15,0**	**20,5**	**0**			
action	**17,6**	**18,4**	**17,7**	**14,6**	**18,9**	**16,4**	**14,7**	**0**		
inaction	**18,4**	**18,7**	**18,2**	**15,2**	**21,7**	**18,0**	**16,9**	**20,8**	**0**	
réaction	**19,1**	**19,8**	**14,6**	**15,4**	**19,0**	**17,8**	**15,8**	**19,4**	**20,0**	**0**

FIG. 3.6 – Exemples de résultats de la fonction de synonymie relative : $Syn_R(X,Y,\text{‹}vie\text{›})$ comparée à la distance thématique D_A(X, Y) en pourcentage de rapprochement.

Ces résultats sont obtenus avant utilisation des informations de synonymie dans l'apprentissage, et, bien que certaines valeurs soient encore loin d'être satisfaisantes, on peut faire quelques

remarques. On peut constater que plusieurs termes ne sont absolument pas en situation de synonymie. C'est le cas par exemple de ‹*train*› et ‹*destinée*› (distance de 1.18 radians soit 68°) ou ‹*inaction*› et ‹*existence*› (distance de 1,06 radian soit 61°). À l'inverse, d'autres termes apparaissent clairement comme synonymes. On peut constater, par exemple, que Syn_R(‹*destin*›, ‹*destinée*›, ‹*vie*›) vaut 0,35 radian, soit environ 20°. Ce résultat indique une synonymie relative de ‹*destin*› et ‹*destinée*› par rapport à ‹*vie*› assez importante, ce que déjà la distance thématique pouvait laisser supposer (0,51 radian).

Comme on pouvait s'y attendre, la synonymie relative est un bon indicateur de polysémie. Syn_R(‹*vie*›, ‹*destinée*›, ‹*vie*›) vaut 0,63 radian (soit 36°) ce qui indique que ‹*destinée*› et ‹*vie*› sont beaucoup moins synonymes relativement à ‹*vie*› que ‹*vie*› et ‹*existence*› (on a Syn_R(‹*vie*›, ‹*destinée*›, ‹*vie*›) > Syn_R(‹*existence*›, ‹*vie*›, ‹*vie*›)). Ces résultats mettent bien en évidence la polysémie de ‹*vie*› et ‹*destinée*› qui, bien que synonymes tous deux de ‹*existence*›, relèvent donc majoritairement de champs sémantiques différents. Ce résultat est corroboré par le rapprochement effectué par la fonction de synonymie sur la distance thématique. Bien que le contexte choisi soit constitué du vecteur de ‹*vie*› il rapproche nettement moins ‹*vie*› et ‹*destinée*› que ‹*vie*› et ‹*existence*› (23,1% contre 16,4%).

Il convient de noter que les termes antonymes, puisqu'ils partagent un certain nombre de caractéristiques et ne s'opposent que sur d'autres (cf. 3.3.1), ne sont pas à une distance trop éloignée. On a, par exemple, ‹*vie*› et ‹*mort*› à une distance de 0,69 radian (40°).

3.2.3.2 Fonction de synonymie partielle Syn_P

Principes et définitions Comme nous l'avons vu dans la section précédente, la méthode de synonymie relative Syn_R est limitée dans les cas réels d'utilisation où il faut souvent évaluer la synonymie entre deux termes sans autre précision de contexte. Nous proposons ici la fonction de synonymie partielle Syn_P :

$$\vartheta^2 \rightarrow [0, \tfrac{\pi}{2}] : \qquad X, Y \rightarrow D = Syn_P(X, Y)$$

$$Syn_P(X, Y) = Syn_R(X, Y, \gamma(X, X \oplus Y) \oplus \gamma(Y, X \oplus Y))$$

FIG. 3.7 – Calcul de la fonction de synonymie partielle Syn_P

Il s'agit de la fonction de synonymie relative pour laquelle le contexte est donné par la somme vectorielle des vecteurs de la contextualisation de X et Y par leur somme vectorielle. Ainsi, cette fonction, à l'image de la fonction lexicale partielle de construction, renvoie la distance angulaire entre les deux vecteurs renforcés par ce qu'ils ont en commun.

Propriétés La fonction de synonymie partielle hérite des propriétés de réflexivité et de symétrie de la fonction de synonymie relative.

1. *réflexivité* : $Syn_P(X, X) = Syn_R(X, X, \gamma(X, X)) = 0$ La réflexivité est héritée de celle de la fonction de synonymie relative.
2. *symétrie* : $Syn_P(X, Y) = Syn_P(Y, X)$ La symétrie est héritée de celle de la synonymie relative. On a en effet $\forall C\ Syn_R(X, Y, C) = Syn_R(Y, X, C)$

En revanche, la relation de pseudo-transitivité n'est pas respectée. La fonction de synonymie partielle n'est donc pas une distance.

Résultats Les tableaux de la figure 3.7 présentent quelques résultats obtenus avec la fonction de synonymie partielle. Dans le premier tableau, la partie supérieure droite rappelle les résultats obtenus avec la distance thématique et la partie inférieure gauche présente les résultats de la fonction de synonymie partielle. Le deuxième tableau indique le rapprochement en pourcentage de la fonction de synonymie partielle par rapport à la distance thématique.

D_A / Syn_P	destinée	destin	vie	existence	mort	automobile	train	action	inaction	réaction
destinée	0	0,51	0,82	0,7	0,99	1,29	1,38	1,31	1,14	1,2
destin	**0,25**	0	0,83	0,75	0,99	1,3	1,38	1,25	1,07	1,16
vie	**0,57**	**0,58**	0	0,61	0,89	1,28	1,35	1,3	1,1	1,2
existence	**0,42**	**0,45**	**0,43**	0	0,98	1,37	1,43	1,37	1,25	1,3
mort	**0,74**	**0,74**	**0,68**	**0,76**	0	1,33	1,4	1,32	1,15	1,26
automobile	**1,01**	**1,02**	**1,06**	**1,14**	**1,1**	0	0,88	1,4	1,22	1,29
train	**1,15**	**1,14**	**1,12**	**1,23**	**1,17**	**0,62**	0	1,43	1,3	1,39
action	**1,06**	**1,01**	**1,05**	**1,15**	**1,06**	**1,16**	**1,2**	0	1,01	0,67
inaction	**0,91**	**0,86**	**0,88**	**1,04**	**0,88**	**0,97**	**1,08**	**0,75**	0	0,9
réaction	**0,96**	**0,9**	**0,99**	**0,98**	**1,0**	**1,04**	**1,15**	**0,47**	**0,68**	0

Rapprochement (%)	destinée	destin	vie	existence	mort	automobile	train	action	inaction	réaction
destinée	**0**									
destin	**51,0**	**0**								
vie	**30,5**	**30,1**	**0**							
existence	**40,0**	**40,0**	**29,5**	**0**						
mort	**25,3**	**25,3**	**23,6**	**22,4**	**0**					
automobile	**14,8**	**21,5**	**17,2**	**16,8**	**17,3**	**0**				
train	**16,7**	**17,4**	**17,0**	**14,0**	**16,4**	**29,5**	**0**			
action	**19,1**	**19,2**	**19,2**	**16,1**	**19,7**	**17,1**	**16,1**	**0**		
inaction	**20,2**	**19,6**	**20,0**	**16,8**	**23,5**	**20,5**	**16,9**	**25,8**	**0**	
réaction	**20,0**	**22,4**	**17,5**	**24,6**	**20,6**	**19,4**	**17,3**	**29,9**	**24,4**	**0**

FIG. 3.8 – Exemples de résultats de la fonction de synonymie partielle : $Syn_P(X,Y)$ comparé avec la distance thématique D_A(X, Y).

Les résultats que nous obtenons sont en tout point comparables à ceux de la fonction de synonymie relative. En effet, la comparaison de ces chiffres entre eux fait clairement apparaître les mêmes relations que nous avions constatées avec la synonymie relative.

Comparons maintenant les résultats obtenus en synonymie relative et ceux obtenus en synonymie partielle. On peut constater que les rapprochements sont tous supérieurs. Le rapprochement est d'autant plus important que les termes sont en situation de synonymie. Il atteint 51% pour ‹*destin*› et ‹*destinée*›, soit une augmentation de près de 20% sur la fonction de synonymie relative avec comme contexte ‹*vie*›. En effet, dans le cas de la synonymie partielle, les idées du contexte utilisé sont celles communes aux deux items. La synonymie relative utilise, par contre, comme référent, le vecteur d'un item qui n'a, en pratique, aucune chance de posséder plus d'idées en commun avec les deux items que le contexte "artificiel" fabriqué par la fonction de synonymie partielle. Effectivement, pour posséder plus d'idées en commun avec les deux vecteurs, il faudrait que ce vecteur ait une norme plus importante, ce qui n'est pas conforme à notre modèle.

3.2.3.3 Fonctions de voisinage synonymique

Nous introduisons dans cette section, deux outils de proximité basés sur les fonctions de synonymie.

Principes et définitions La *fonction de voisinage synonymique* permet de connaître les items lexicaux dont les vecteurs sont les plus synonymes d'un terme donné selon la fonction

de synonymie relative. Soit :

$$\left|\mathcal{V}_{Syn_{R_k}}(X,C)\right| = k \qquad \forall Y \in \mathcal{V}_{Syn_{R_k}}(X,C), \quad \forall Z \notin \mathcal{V}_{Syn_{R_k}}(X,C), \quad Syn_R(X,Y,C) \leq Syn_R(X,Z,C) \quad (3.1)$$

De même, la *fonction de voisinage synonymique partielle* permet de connaître les items lexicaux dont les vecteurs sont les plus synonymes d'un terme donné selon la fonction de synonymie partielle. Soit :

$$\left|\mathcal{V}_{Syn_{P_k}}(X)\right| = k \qquad \forall Y \in \mathcal{V}_{Syn_{P_k}}(X), \quad \forall Z \notin \mathcal{V}_{Syn_{P_k}}(X), \quad Syn_P(X,Y) \leq Syn_P(X,Z) \quad (3.2)$$

Le calcul des fonctions de synonymie se fait en temps linéaire, mais, à cause de la taille importante de la base (plus de 100 000 items), ces deux outils sont, à l'heure actuelle, difficiles d'utilisation puisqu'en pratique le temps de calcul d'un voisinage est supérieur à une minute sur un Sun à 8 processeurs 800 Mhz (contre une à deux secondes pour un calcul de voisinage thématique). Ces deux outils peuvent ainsi être utilisés dans le but de vérifier une certaine cohérence de la base et ils viennent s'ajouter à la méthode de voisinage thématique présentée en 2.1.3.2. En revanche, leur usage est plus délicat dans le cadre de réelles applications telles que la recherche d'informations.

Exemples Voici l'exemple du voisinage thématique de ‹*destin*› suivi du voisinage synomique de ‹*destin*› dans le contexte de ‹*vie*›.

$\mathcal{V}$(‹*destin*›) = (‹*destin*› ; 0) (‹*destinée*› ; 0,51) (‹*sort*› ; 0,56) (‹*détermination*› ; 0,57) (‹*déterminer*› ; 0,58) (‹*être*› ; 0,58) (‹*fatidique*› ; 0,61) (‹*fatalité*› ; 0,62) (‹*déterminisme*› ; 0,63) (‹*constant*› ; 0,63) (‹*provoquer*› ; 0,64)

$\mathcal{V}_{Syn_R}$(‹*destin*›, ‹*vie*›) = (‹*destin*› ; 0,0) (‹*destinée*› ; 0,25) (‹*sort*› ; 0,38) (‹*vivifier*› ; 0,4) (‹*détermination*› ; 0,41) (‹*accident*› ; 0,43) (‹*déterminer*› ; 0,43) (‹*vital*› ; 0,43) (‹*déterminisme*› ; 0,44) (‹*existence*› ; 0,45) (‹*fatidique*› ; 0,45) (‹*fatalité*› ; 0,46) ... (‹*être*› ; 0,53) ... (‹*provoquer*› ; 0,6)

Ces premiers résultats montrent bien une différence entre le voisinage thématique et le voisinage synonymique. Ces résultats semblent montrer dans une certaine mesure la différence entre la thématique et le sens. En effet, des termes comme ‹*être*› ou ‹*provoquer*› sont parmi les voisins thématiques alors qu'ils sont plus éloignés, en terme de rang, en ce qui concerne le voisinage synonymique.

3.3 Relations d'opposition : l'antonymie

Au cours de l'analyse sémantique d'un texte, en particulier dans le cas qui nous intéresse le plus ici, l'analyse de définitions, il est fréquent de trouver des tournures négatives. Par exemple, dans [Larousse, 2004], ‹*inexistant*› est défini par « *qui n'existe pas* ». Une analyse sémantique automatique telle que celle présentée en 2.1.6 utilisera le vecteur d'idées de l'item lexical ‹*exister*› plutôt que celui de l'item antonyme ‹*inexister*›. De même, les définitions qui contiennent des antonymes comme celle du terme ‹*action*› « *antonymes :* ‹réaction›, ‹inaction› » ne seront pas bien analysées puisque les vecteurs utilisés sont alors ceux de ‹*réaction*› et ‹*inaction*› plutôt que ceux de leurs opposés.

Ces raisons nous ont donc amenés à chercher à définir une fonction d'antonymie pour les vecteurs d'idées [Schwab, 2001]. Cette fonction, à partir d'un vecteur fabriquera son vecteur opposé. Ainsi, dans le cas de la définition de l'item ‹*inexistant*›, le vecteur conceptuel correspondant

à sa définition n'est plus directement calculé à partir du vecteur d'‹*existant*› mais à partir d'un vecteur dont les idées sont opposées à celles d'‹*existant*›.

Nous allons, dans un premier temps, exposer la définition de l'antonymie compatible avec notre modélisation vectorielle ainsi que les trois différents types d'antonymie (*complémentaire, scalaire, duale*) et les propriétés qu'elle induit. Cette étude théorique a été réalisée au cours de mon DEA [Schwab, 2001] et nous en reprenons ici et adaptons les principaux points indispensables à la compréhension de la modélisation. Nous présenterons ensuite la démarche que nous avons suivie pour définir une fonction lexicale par type d'antonymie, qui construit à partir d'un vecteur X et, éventuellement, de vecteurs contexte C et référent R, le vecteur opposé à X. Nous présenterons deux mesures : la *mesure de potentiel d'antonymie* qui permet d'estimer si un vecteur peut avoir un antonyme et la *mesure d'évaluation de l'antonymie* ou fonction lexicale d'évaluation de l'antonymie qui permet d'apprécier si deux vecteurs peuvent être antonymes l'un de l'autre.

3.3.1 Définitions et caractérisation de l'antonymie

La définition de l'antonymie fournie par [Larousse, 1991] est la suivante : « *Un antonyme est un mot qui a un sens opposé à celui d'un autre. Les antonymes, ou contraires, sont des mots appartenant obligatoirement à la même classe grammaticale (*‹grand› *est l'antonyme de* ‹petit› *et non celui de* ‹petitesse›*) et s'opposant par un ou plusieurs traits sémantiques, les autres étant communs. Par exemple,* ‹monter› *et* ‹descendre› *possèdent en commun le trait* DÉPLACEMENT VERTICAL *et s'opposent par les traits* VERS LE HAUT *et* VERS LE BAS*. L'antonymie peut donc se définir comme une relation d'incompatibilité entre deux termes. Elle est, à cet égard, l'exact opposé de la synonymie* ».

Dans le cadre de la modélisation vectorielle que nous utilisons, cette notion d'incompatibilité se prête difficilement à la construction d'un antonyme. En effet, si on veut caractériser la construction de l'opposé d'un concept, il est préférable d'utiliser la notion de *symétrie* plutôt que celle d'incompatibilité. Nous avons proposé dans [Schwab *et al.*, 2002c] la définition suivante :

Deux items lexicaux sont en relation d'antonymie si on peut exhiber une symétrie de leurs traits sémantiques par rapport à un axe.

La symétrie se décline alors de différentes manières, selon la nature de son support. On distingue, comme supports :

- une propriété affectant une valeur étalonnable (valeur élevée, valeur faible) : par exemple, ‹*chaud*›, ‹*froid*› sont des valeurs symétriques de température ;
- l'application d'une propriété (applicable/non applicable, présence/absence) : par exemple, ‹*informe*› est antonyme de tout ce qui a une forme, de même que ‹*insipide*›, ‹*incolore*›, ‹*inodore*›, etc. de tout ce qui pourrait avoir saveur, couleur, odeur, ...
- l'existence d'une propriété ou d'un élément considérés comme symétriques par l'usage (e.g. ‹*soleil*› ⫼ ‹*lune*›[61]), ou par des propriétés naturelles ou physiques des objets considérés (e.g. ‹*mâle*› ⫼ ‹*femelle*›, ‹*tête*› ⫼ ‹*pied*›, ...).

Notre idée est que les constructions d'antonymes sont dépendantes du type de support de symétrie. Il peut alors exister plusieurs types d'antonymes pour un même terme, comme il peut ne pas en exister d'évidents, si la symétrie n'est pas immédiatement décelable. En tant que fonction lexicale, comparée à la synonymie, on peut dire que si la synonymie est la recherche de la ressemblance avec comme test la substitution (*x est synonyme de y si x peut "remplacer"*

[61] Nous notons l'antonymie par un signe d'équivalence ayant subit une rotation de 90 degrés (‹*riche*› ⫼ ‹*pauvre*›). Ce signe rappelle à la fois le signe qui marque chez Polguère ([Polguère, 2003], p. 122) la relation considérée comme opposée à l'antonymie, la synonymie, et la symétrie axiale existant entre les deux termes antonymes.

y), l'antonymie est la recherche de la symétrie avec comme test la recherche du support de la symétrie (*x est antonyme de y s'il existe un support de symétrie t tel que x symétrique de y par rapport à t*).

De même que pour la synonymie [Lafourcade & Prince, 2001a], il n'existe pas d'antonymes absolus, c'est-à-dire deux mots qui seraient antonymes l'un de l'autre quel que soit le contexte. L'antonymie s'apprécie toujours en contexte. Par exemple, ‹*frais*› peut être le contraire de ‹*tiède*›, ‹*chaud*›, ‹*racorni*›, ‹*flétri*›, ‹*maladif*›, ‹*rassis*›, ‹*confit*›, ‹*sec*›, ‹*surgelé*›, ‹*pourri*›, ...

Les travaux de Franck Robert Palmer (*né en 1922*) [Palmer, 1976], Sir John Lyons (*né en 1932*) [Lyons, 1977] et Victoria Muehleisen [Muehleisen, 1997], nous ont permis de distinguer trois types de symétrie, chacune caractérisant une classe d'antonymes. Si les deux premières, l'antonymie complémentaire et l'antonymie scalaire, sont classiques en linguistique, nous en avons introduit une troisième, rarement considérée comme antonymie, celle des duals.

3.3.1.1 Antonymie et linguistique

En général, les linguistes précédemment cités considèrent deux différents types de relation entre antonymes, l'antonymie complémentaire et l'antonymie scalaire.

Antonymes complémentaires L'antonymie complémentaire concerne les couples tels que ‹*pair*› $\amalg_c$ ‹*impair*›, ‹*présence*› $\amalg_c$ ‹*absence*› ou ‹*existence*› $\amalg_c$ ‹*inexistence*›.

il est présent $\Rightarrow$ il n'est pas absent	il n'est pas absent $\Rightarrow$ il est présent
il est absent $\Rightarrow$ il n'est pas présent	il n'est pas présent $\Rightarrow$ il est absent

En termes de logique, nous avons :

$$(\forall x)\ [P(x) \Rightarrow \neg Q(x)] \qquad (\forall x)\ [\neg P(x) \Rightarrow Q(x)]$$
$$(\forall x)\ [Q(x) \Rightarrow \neg P(x)] \qquad (\forall x)\ [\neg Q(x) \Rightarrow P(x)]$$

Nous reconnaissons ici une relation de disjonction exclusive. Dans ce cadre, l'affirmation d'un des termes implique nécessairement la négation de l'autre. Sur le plan de la symétrie, l'antonymie complémentaire présente deux types de symétrie :

- une symétrie de valeurs dans un système à deux valeurs seulement, comme dans l'exemple précédent.
- une symétrie par rapport à l'application d'une propriété : le *noir* est l'absence de couleur, il est donc "opposé" à toute couleur, et à toute combinaison de couleurs.

Le mélange des deux types de symétrie peut parfois introduire des divergences entre la linguistique et une modélisation qui se voudrait cohérente avec le modèle vectoriel. Pour en donner un exemple, en linguistique, on considère aussi le couple ‹*vivant*›/‹*mort*› comme relevant de l'antonymie complémentaire, alors que sur un plan logique :

$$(\forall x)\ [\neg \mathit{vivant}(x) \Rightarrow \mathit{mort}(x)]$$

est falsifiable puisque ce qui est inanimé n'est ni vivant ni mort. Une bonne façon ne pas falsifier ces propriétés est de limiter l'application des propriétés aux termes pour lesquelles elles sont pertinentes, ce qui peut se faire par une conjonction :

$$(\forall x)\ [C(x) \wedge P(x) \Rightarrow \neg Q(x)]$$

où C est la condition préalable pour l'application de P, ce qui donne dans notre exemple :

$$(\forall x)\ [\mathit{animé}(x) \wedge \mathit{vivant}(x) \Rightarrow \neg \mathit{mort}(x)]$$

En pratique, l'inanimé est complémentaire du vivant et du mort, par non-application de la propriété de *vie*, alors que ‹*vivant*› $\shortmid\!\shortmid\!\shortmid_c$ ‹*mort*› est un couple de symétriques en valeur dans un système à deux valeurs. En rendre compte sur le plan logique nécessite des précautions qui, comme nous le montrerons, ne sont pas nécessaires dans la modélisation vectorielle.

Une modélisation logique peut aussi être remise en cause par la possibilité d'usage linguistique. Si la logique peut accepter ce qui est ni vivant ni mort (l'inanimé en l'occurrence) en cherchant à isoler le type de symétrie, la langue peut aussi accepter le paradoxe, qui défie la logique, et qu'elle atteste parfois par l'usage. C'est le cas des antonymes de notre exemple, puisque la forme *mort-vivant* existe pour exprimer ce qui possède les deux propriétés[62].

Antonymes scalaires Les antonymes scalaires (ou gradables[63]) concernent les systèmes échelonnés comme la taille (‹*grand*› $\shortmid\!\shortmid\!\shortmid_s$ ‹*petit*›) ou la température (‹*chaud*› $\shortmid\!\shortmid\!\shortmid_s$ ‹*froid*›). La symétrie se réalise par rapport à une valeur de référence du système qui n'est pas toujours représentée par un mot. Par exemple, pour ‹*grand*› $\shortmid\!\shortmid\!\shortmid_s$ ‹*petit*›, nous avons :

Cet homme est grand	$\Rightarrow$ Cet homme n'est pas petit
Cet homme est petit	$\Rightarrow$ Cet homme n'est pas grand
Cet homme n'est pas grand	$\Rightarrow$ Cet homme est petit $\vee$ cet homme est de taille moyenne
Cet homme n'est pas petit	$\Rightarrow$ Cet homme est grand $\vee$ cet homme est de taille moyenne

Cet homme est « *ni grand ni petit* » qui désigne en général la taille moyenne, mais qui ne signifie pas dans le cas présent (comme dans le cas de ‹*vivant*› $\shortmid\!\shortmid\!\shortmid_c$ ‹*mort*›) que la propriété ne s'applique pas. C'est simplement qu'il existe ici une "valeur neutre" à partir de laquelle les autres s'échelonnent. En logique classique, on pourrait l'exprimer par

$$(\forall x)\ \ [P(x) \Rightarrow \neg Q(x) \wedge R(x)]$$

si R est la propriété ayant la valeur de référence (neutre ou médiane)

$$(\forall x)\ \ [Q(x) \Rightarrow \neg P(x) \vee R(x)] \qquad (\forall x)\ \ [\neg Q(x) \not\Rightarrow P(x)]$$
$$(\forall x)\ \ [R(x) \Rightarrow \neg Q(x) \wedge \neg P(x)] \qquad (\forall x)\ \ [\neg P(x) \not\Rightarrow Q(x)]$$

La valeur de référence peut ne pas être la seule valeur possible, mais un des éléments remarquables de l'échelle (pour des propriétés multi-valuées par exemple). L'usage de termes gradables implique toujours une évaluation et donc une comparaison. Celle-ci peut être explicite : « *Jean est plus petit/grand que Pierre* », « *il avance/recule* » (le terme moyen étant *immobile*[64]). Elle peut aussi être implicite et renvoyer à des normes tacitement admises par l'individu ou la communauté à laquelle il appartient : « *il fait chaud* » dit par un habitant d'un pays équatorial ne renverra pas à la même idée de chaleur (donc à la même valeur de référence) qu'un habitant des fjords de Norvège.

L'usage de deux termes antonymes n'est pas toujours symétrique. On dit « *la colline est haute de 137 mètres* » (sous-entendu par rapport au niveau de la mer) mais on ne dit pas *« *la mer est basse de 137 mètres* » (sous-entendu par rapport à la colline). L'usage consacre celui des antonymes qui est en isotopie[65] avec le terme que l'on veut évaluer (ici la colline qui est une protubérance, et qui renforce le terme de hauteur).

[62] On appelle *oxymore* ou *oxymoron* le rapprochement de termes qui semblent contradictoires comme c'est le cas pour ‹*mort-vivant*› ou ‹*clair-obscur*›.

[63] Selon la terminologie de [Nyckees, 1998].

[64] On peut remarquer que le neutre d'un type d'antonymie peut être opposable dans une autre antonymie. Ici, par exemple, ‹*mobile*› $\shortmid\!\shortmid\!\shortmid_c$ ‹*immobile*›.

[65] Notion de récurrence ou de renforcement du ou des traits communs. En linguistique componentielle, elle a été introduite par Greimas [Greimas, 1986] et Pottier [Pottier, 1964], et développée par Rastier [Rastier, 1985].

Extensions de l'antonymie Nous considérons, dans notre étude, un type d'antonymes supplémentaire : les antonymes *duals*. Ils sont composés de deux sous-familles : les antonymes conversifs [Mel'čuk *et al.*, 1995] et les duals propres. Ils correspondent au troisième type de symétrie, celui que l'usage et la nature même des objets peuvent introduire.

Conversifs On appelle conversifs (ou réciproques) les couples comme ‹*acheter*› $|||_d$ ‹*vendre*›, ‹*prêter*› $|||_d$ ‹*emprunter*›, ‹*mari*› $|||_d$ ‹*femme*›, ‹*avant*› $|||_d$ ‹*après*›, ‹*père*› $|||_d$ ‹*fils*›. De nombreux linguistes comme Igor Mel'čuk ([Mel'čuk *et al.*, 1995], p. 130) ne les considèrent pas comme des antonymes. La fonction *anti* de son DEC désigne en réalité les antonymes complémentaires et les antonymes scalaires. Il dédie aux conversifs une autre fonction lexicale *conv*. Cependant, dans la mesure où pour nous, la modélisation de l'antonymie correspond à une étude complète des mécanismes de symétrie, nous avons considéré les conversifs comme un cas particulier de symétrie, et les avons naturellement associés à un processus antonymique "étendu".

Pierre est le père de Marc ⇔ Marc est le fils de Pierre.

Ce qui s'exprime, en termes de logique, comme :

$$(\forall x,y)\ [P(x,y) \Leftrightarrow Q(y,x)]$$

Dans le cas des conversifs, si on remplace dans une phrase un terme P par son réciproque Q, on peut systématiquement rétablir la synonymie entre les deux phrases à condition de permuter les arguments syntaxiques mis en relation par P comme le montre la formule. Ainsi, pour les conversifs, il y a symétrie par rapport à la place des arguments (P est réciproque de Q).

Duals Les duals propres sont une notion d'antonymie que nous introduisons pour rendre compte d'un effet particulier de mise en relation de termes où la symétrie porte cette fois-ci sur des fonctions culturelles (symétrie consacrée par l'usage) et spatio-temporelles (propriétés particulières de l'espace-temps). L'antonyme dual d'un mot est le pendant de celui-ci. Les duals sont des mots que la culture associe comme ‹*soleil*› $|||_d$ ‹*lune*›, ou qui ne vont pas, *a priori*, l'un sans l'autre comme ‹*question*› $|||_d$ ‹*réponse*› ou alors sont l'expression d'une antonymie temporelle i.e. qui exprime le passage d'un état à un autre comme ‹*naissance*› $|||_d$ ‹*décès*›. Dans ce troisième cas, on peut remarquer que ces deux événements marquent le passage entre deux antonymes complémentaires (‹*inexistence*› $|||_c$ ‹*existence*› dans le cas de ‹*naissance*› $|||_d$ ‹*décès*› ou bien ‹*présence*› $|||_c$ ‹*absence*› dans le cas de ‹*départ*› $|||_d$ ‹*arrivée*›). L'antonymie duale propre présente naturellement une symétrie qui n'est pas relevée dans l'échange des places d'argument puisqu'il s'agit de prédicats unaires. Elle exprime le fait que si l'un des deux prédicats est vrai, il existe une valeur pour lequel l'autre l'est aussi nécessairement. Pour la modéliser, on écrira :

$$(\exists x)\ [P(x)] \Leftrightarrow (\exists Q)\ [Q(x)]$$

avec Q dual de P qui modélise par exemple le fait que si x a un début, alors il existe aussi une fin à x ou :

$$(\exists x)\ [P(x)] \Leftrightarrow (\exists Q\, \exists y)\ [Q(y)]$$

avec Q dual de P qui exprime que si x est une question, il existe un objet y et il existe un prédicat réponse, tel que y est une réponse à x.

Cette nécessité du prédicat dual peut rendre compte de certains couples de description temporelle. Ainsi, ‹*avant*› $|||_d$ ‹*après*› en prédicats unaires, sont linguistiquement différenciés sur le plan de la catégorie grammaticale, comme dans « *il y a un avant et un après* » à ne pas confondre avec ‹*avant*› $|||_s$ ‹*après*› qui sont des scalaires avec comme valeur médiane ‹*pendant*›.

3.3.1.2 Propriété des points fixes

Une conséquence importante de cette définition basée sur la symétrie est que tout vecteur peut avoir un vecteur antonyme. En effet, pour un axe donné, tout vecteur a un symétrique. La linguistique classique considère que certains termes n'ont pas d'antonymes avérés. C'est le cas, par exemple, des objets matériels comme ‹*table*›, ‹*voiture*› ou ‹*porte*›. Les idées principales constituant ces termes, autrement dit, les idées, ne sont pas obligatoirement opposables. Dans un espace géométrique, un point qui n'a pas de symétrique, par rapport à un certain axe, se trouve sur cet axe. De même, dans notre formalisme, les idées qui ne sont pas opposables sont sur l'axe de symétrie et donc l'antonyme d'un item lexical qui ne possède pas d'antonyme avéré est l'item lexical lui-même. Nous appelons cette propriété *la propriété des points fixes.*

En pratique, les concepts possèdent plus facilement la propriété des points fixes que les items. Ces derniers, en se projetant sur plusieurs concepts, peuvent dans certains contextes, hériter de la capacité d'opposition de certains concepts. Ainsi, par exemple en antonymie scalaire, une ‹*Ferrari*›, bien que sorte d'*AUTOMOBILE*, concept sans antonyme, se projette aussi sur une notion de *RAPIDITÉ* qui, elle, est opposable. C'est pourquoi on peut très bien imaginer comme possible antonyme d'une ‹*Ferrari*› un item possédant conjointement les propriétés *AUTOMOBILE* et *LENTEUR* une ‹*deux chevaux*›, par exemple.

3.3.2 Fonctions d'antonymie : mise au point

La suite de ce chapitre reprend certaines des avancées réalisées au cours de mon DEA [Schwab, 2001], revues, affinées et complétées pendant ma thèse dans différents articles [Schwab *et al.*, 2002a, Schwab *et al.*, 2002c]. Les noms des fonctions ont été modifiés afin de les normaliser par rapport aux autres fonctions lexicales. En effet, la fonction de synonymie $Syn_R(X,Y)$ renvoie une évaluation de la synonymie entre X et Y (cf. 3.2.3.1). Dans mon DEA, ainsi que dans les articles précédemment cités, la fonction *Anti* est la méthode de construction d'un vecteur antonyme et celle d'évaluation de l'antonymie est $Manti_{Eval}$. Après normalisation, méthode de construction du vecteur d'antonymie devient $Canti_R$ et la méthode d'évaluation de l'antonymie devient $Anti_R$.

3.3.3 Construction d'un vecteur antonyme

3.3.3.1 Principes et définitions

Dans cette section, nous cherchons à construire à partir du vecteur d'un mot (ou d'un sens) un autre vecteur qui soit aussi proche que possible du vecteur d'un de ses opposés. Par exemple, la construction du vecteur antonyme du vecteur de l'item ‹*chaud*› devrait être proche de celui de l'item ‹*froid*›. Nous nous heurtons ici à une importante difficulté : quel type d'antonymie appliquer dans tel ou tel cas particulier et donc comment construire un vecteur adéquat pour une certaine phrase ? Une réponse est apportée avec la mesure de potentiel d'antonymie (cf. 3.3.3.3) qui nous permet de fabriquer un vecteur antonyme global (cf. 3.3.3.3) à partir des vecteurs antonymes de chaque type, dont nous allons présenter la construction.

3.3.3.2 Fonctions de construction d'un vecteur antonyme

Fonctions de voisinage anti-thématique : *AntiLex* Dans la perspective de tester les fonctions de construction des vecteurs antonymes, nous avons défini un ensemble de fonctions *AntiLex* qui, à partir d'items lexicaux, renvoient les N plus proches antonymes à la manière de la fonction de voisinage thématique déjà présentée en 2.1.3.2. Ces fonctions construisent ce qu'on pourrait appeler le voisinage anti-thématique. Elle peuvent ainsi servir dans la recherche d'une thématique opposée en génération de texte, par exemple pour le paraphrasage.

Ces fonctions existent, bien sûr, sous les formes complémentaire, scalaire et duale. Ces trois fonctions nécessitent toutes la même séquence d'opérations (cf. figure 3.9).

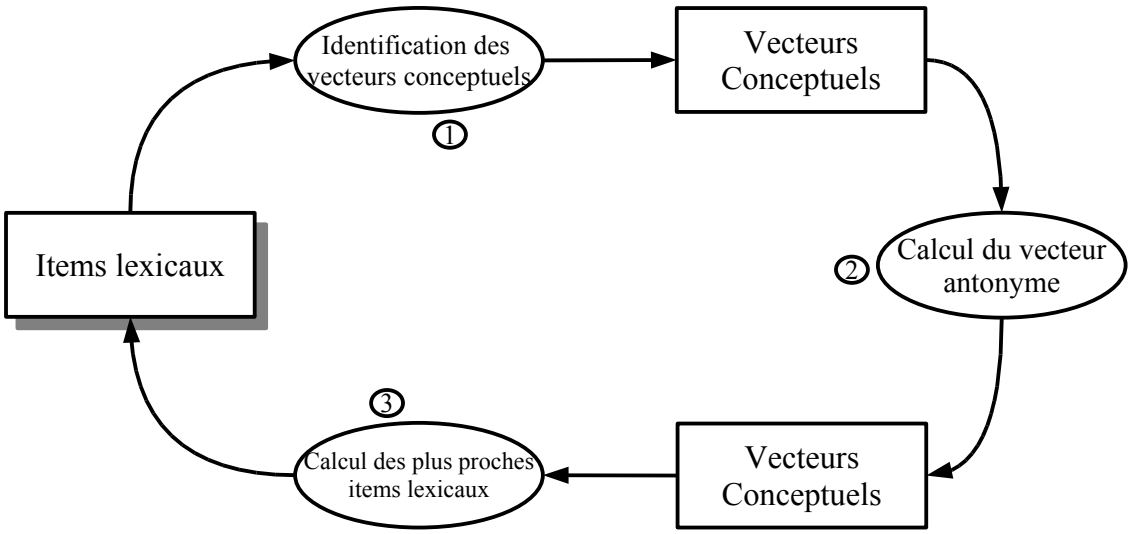


FIG. 3.9 – Séquence d'opérations des fonctions *AntiLex*

Cette séquence se déroule en trois étapes. Dans la première, il s'agit d'identifier les vecteurs d'idées à partir des items lexicaux correspondants. Cette étape est importante car elle détermine les vecteurs qui seront ensuite employés dans la deuxième étape avec l'utilisation de la fonction lexicale d'antonymie proprement dite que nous cherchons à construire. Une meilleure qualité des vecteurs facilitera sa fabrication. Il s'agit alors de recourir à une méthode de contextualisation. Celle-ci peut être faible (cf. 2.1.4.5) pour une utilisation avec les deux types de vecteurs d'idées ou forte (cf. 2.3.6) pour une utilisation basée sur les vecteurs conceptuels. Enfin, la dernière étape consiste à passer des vecteurs aux items lexicaux. Nous faisons usage ici de la fonction de voisinage définie en 2.1.3.2. Dans la suite, nous allons uniquement nous consacrer à l'étape 2, les deux autres étant définies et commentées par ailleurs.

Nous l'avons vu en 3.3.1, un antonyme s'apprécie toujours en contexte. Dans certains cas, ce contexte seul n'est pas suffisant pour déterminer un axe de symétrie pour l'antonymie. Prenons l'exemple de l'item lexical ‹*père*›. Dans le contexte ‹*famille*›, il peut être opposable à ‹*mère*› ou ‹*enfants*›. Il peut être pertinent, dans les cas où il ne sert pas d'axe de symétrie, d'affiner le contexte par un vecteur qui peut, lui, jouer ce rôle de référent. Dans notre exemple, il faudrait alors prendre comme référent ‹*filiation*›, ‹*mariage*› ou ‹*homme*›. C'est la raison pour laquelle nous avons défini la fonction de distance anti-thématique $\mathit{AntiLex}_R$ qui renvoie les n plus proches antonymes du mot X dans le contexte défini par le mot C en référence au mot R de la manière suivante :

$$\omega^3 * \mathbb{N} \rightarrow \omega^* : \quad X, C, R, n \rightarrow Z = \mathrm{AntiLex}_R(X, C, R, n) \tag{3.3}$$

où ω représente l'ensemble des items lexicaux.

Dans la plupart des cas, le contexte suffit pourtant à déterminer un axe de symétrie, et nous avons donc défini la fonction partielle $\mathit{AntiLex}_P$:

$$\omega^2 * \mathbb{N} \rightarrow \omega^* : \quad X, C, n \rightarrow Z = \mathrm{AntiLex}_P(X, C, n) = \mathrm{AntiLex}_R(X, C, C, n) \tag{3.4}$$

Enfin, nous définissons la fonction de distance anti-thématique absolue comme :

$$\omega * \mathbb{N} \rightarrow \omega^* : \quad X, n \rightarrow Z = \mathrm{AntiLex}_A(X, n) = \mathrm{AntiLex}_R(X, X, X, n) \tag{3.5}$$

Fonction de construction d'un vecteur antonyme : $Canti_P$ Nous appelons $Canti_P$ la fonction partielle de construction d'un vecteur antonyme du vecteur X par rapport à un contexte référent C :

$$\vartheta^2 \rightarrow \vartheta : \qquad X, R \rightarrow Z = Canti_P(X, C)$$

où ϑ représente l'ensemble de vecteurs utilisé.

Nous appelons $Canti_R$ la fonction de construction d'un vecteur antonyme du vecteur X dans un contexte C par rapport à un référent R :

$$\vartheta^3 \rightarrow \vartheta : \qquad X, C, R \rightarrow Z = Canti_R(X, C, R) = Canti_P(\gamma(X, C), \gamma(R, C))$$

L'idée est de construire grâce à l'opération de contextualisation faible le vecteur contextualisé du terme que nous cherchons à opposer et le vecteur du référent lui aussi contextualisé.

Enfin, $Canti_A$ la fonction absolue de construction d'un vecteur antonyme du vecteur X se définit comme :

$$\vartheta \rightarrow \vartheta : \qquad X \rightarrow Z = Canti_A(X) = Canti_R(X, X, X)$$

où ϑ représente l'ensemble de vecteurs utilisé.

Il faut noter que la fonction d'antonymie absolue $Canti_A$ est d'usage délicat dans les cas réels d'utilisation, les cas d'analyse sémantique. En effet, l'antonymie nécessite d'exhiber une symétrie, mais celle-ci ressortira difficilement pour une utilisation de $Canti_A$ à cause de la polysémie des termes.

Nous allons donc chercher à définir $Canti_P$ et à travers elle, les fonctions $Canti_R$ et $Canti_A$. Comme il existe trois types d'antonymie, nous avons défini trois fonctions :

- $Canti_{P_c}$ pour l'antonymie complémentaire.
- $Canti_{P_s}$ pour l'antonymie scalaire.
- $Canti_{P_d}$ pour l'antonymie duale.

Comme les fonctions de synonymie (cf. 3.2), les diverses fonctions $Canti_P$ sont dépendantes du contexte mais, contrairement à elles, elles ne peuvent pas être indépendantes de l'organisation des concepts. Le contexte est ici représenté par un vecteur qui sert de cadre de référence.

L'idée de base que nous avons suivie pour fabriquer les fonctions $Canti_P$ est que si les idées d'un terme peuvent être opposées, alors son antonyme doit posséder les idées inverses en proportion identique. Ces fonctions nécessitent donc d'identifier pour chaque concept, pour chaque contexte et pour chaque type d'antonymie un vecteur qui sera considéré comme son opposé. Il faut donc construire trois listes de triplets $\langle concept, contexte, vecteur \rangle$ que nous appellerons, dans la suite de notre exposé, *listes de vecteurs antonymes des concepts* (VAC).

Vecteur antonyme d'un concept Le vecteur antonyme d'un concept de base est noté :

$$CantiC_\alpha(C_i, V(contexte))$$

où $\alpha \in \{comp, scal, dual\}$, C_i désigne le i-ème concept de l'espace générateur et $V(contexte)$ le vecteur représentant le contexte. Afin de simplifier l'écriture, l'indice α sera omis lorsque nous parlons indifféremment de l'une ou l'autre des fonction $CantiC$.

Construction des listes des vecteurs antonymes aux concepts (VAC) Ces vecteurs antonymes sont construits manuellement uniquement à partir de vecteurs génératifs y compris le vecteur lui-même le cas échéant (*propriété des points fixes* 3.3.1.2). Ainsi, pour l'antonymie complémentaire, nous pouvons avoir par exemple :

$$
\begin{array}{lcll}
\mathit{CantiC}_c(\textsc{existence},V) & = & V(\textsc{inexistence}) & \forall V \\
\mathit{CantiC}_c(\textsc{inexistence},V) & = & V(\textsc{existence}) & \forall V \\
\mathit{CantiC}_c(\textsc{agitation},V) & = & V(\textsc{inertie}) \oplus V(\textsc{repos}) & \forall V \\
\mathit{CantiC}_c(\textsc{jouet},V) & = & V(\textsc{jouet}) & \forall V
\end{array}
$$

Comme les items lexicaux, les concepts peuvent avoir suivant le contexte un antonyme différent même si ils ne sont pas polysémiques. Ainsi, *DESTRUCTION* peut avoir comme antonyme *PRÉSERVATION, CONSTRUCTION, RÉPARATION* ou *PROTECTION*. Nous avons ainsi défini pour chacun d'eux un vecteur qui permettra la sélection de l'antonyme le mieux adapté à la situation (cf 3.3.3.2).

Ces *listes des vecteurs antonymes aux concepts* (VAC), contiennent en antonymie complémentaire par exemple :

concept	contexte	vecteurs constituant le vecteur antonyme.
EXISTENCE	$\forall V$	$V(\textsc{inexistence})$
INEXISTENCE	$\forall V$	$V(\textsc{existence})$
AMOUR	$\forall V$	$V(\textsc{désaccord}) \oplus V(\textsc{aversion}) \oplus V(\textsc{inimitié})$
DÉSORDRE	$V(\textsc{ordre}) \oplus V(\textsc{désordre})$	$V(\textsc{ordre})$
DÉSORDRE	$V(\textsc{ordre}) \oplus V(\textsc{classification})$	$V(\textsc{classification})$
DÉSORDRE	$V(\textsc{organisation}) \oplus V(\textsc{désorganisation})$	$V(\textsc{organisation})$
JOUET	$\forall V$	$V(\textsc{jouet})$

On peut constater que le concept *EXISTENCE* a pour antonyme le vecteur *INEXISTENCE* quel que soit le contexte ou que le concept *DÉSORDRE* a pour antonyme le vecteur *ORDRE* dans le contexte constitué des vecteurs *ORDRE* et *DÉSORDRE*. Le dernier, *JOUET*, est un exemple de point fixe. Dans le cas où plusieurs antonymes sont possibles, comme ici avec *DÉSORDRE*, nous considérons souvent que le contexte est formé des deux concepts. En effet, il est souvent difficile, parfois sans doute impossible de trouver un contexte à la fois fiable et monosémique mais si celui-ci existait, il posséderait au moins les mêmes idées que les concepts. Géométriquement, deux vecteurs sont bien symétriques par rapport à leur somme vectorielle et donc cette réflexion peut être renouvelée à l'identique avec tous vecteurs dont on connaît l'opposition.

La création de cette liste est un travail long et fastidieux. Pour un concept, deux personnes ne sont pas toujours d'accord sur la construction du vecteur antonyme. L'une peut considérer qu'il est formé de certains concepts tandis qu'une autre personne en considérera d'autres ou même n'en considérera aucun. Par exemple, l'antonyme de *HASARD* peut être *BUT* pour quelqu'un, *DÉTERMINISME* pour un autre et ne pas en avoir pour un troisième. La création de la liste est donc très subjective. C'est la partie délicate mais nécessaire de notre méthode car il s'agit de la base sur laquelle sera construit le vecteur antonyme (cf. 3.3.3.2). Il convient de préciser toutefois, qu'elle est modifiable à tout moment si des ajustements sont nécessaires par exemple si l'usage consacre de nouveaux antonymes.

Fonction *CantiC* La fonction *CantiC* basée sur la liste $L_{opposés}$ construite en 3.3.3.2 renvoie un concept, en fonction du type d'antonymie α et du contexte, son vecteur antonyme (cf. algorithme 5).

Algorithme 5: *CantiC*

Entrée : concept *CONCEPT*, vecteur *contexte*, liste $L_{opposés}$
Sortie : $V_{ant}(CONCEPT)$
$l \leftarrow L.chercher(concept)$ % la méthode *chercher* renvoie la liste des antonymes possibles
si *l.estVide()* **alors**
  retourner V(CONCEPT) % le vecteur du concept lui-même
sinon
  Vecteur $rep = \vec{0}$
  pour *chaque élément E de l* **faire**
    $rep = rep \oplus (\frac{\pi}{2} - D_A(\mathrm{V(contexte(E))}, \mathrm{V}(CONTEXTE))) \otimes V(opposé)$
  retourner *rep*

L'algorithme 5 parcourt la liste d'antonymes et renvoie le vecteur antonyme correspondant au concept et au contexte choisi. Dans le cas où le concept n'a pas d'antonyme, son vecteur est renvoyé. Dans le cas inverse, on fait la somme des vecteurs de chacun de ses possibles opposés pondérée par le complément à $\frac{\pi}{2}$ de la distance angulaire entre le contexte de E et le contexte général. Il s'agit du passage de l'intervalle $[0, \frac{\pi}{2}]$ à l'intervalle $[0,1]$ de la distance angulaire. Cette transformation inverse le domaine image de façon linéaire. Pour une transformation vers $[0,1]$, nous aurions pu utiliser le cosinus mais on se garde bien de le faire car on souhaite focaliser le pouvoir de discrimination dans les faibles valeurs de l'angle comme nous l'avons déjà remarqué en 2.1.3.1.

Muni de la fonction *CantiC*, nous allons maintenant pouvoir étudier la méthode de construction d'un vecteur antonyme.

Construction du vecteur antonyme Dans cette section, nous présentons la fonction $Canti_P$ en montrant comment à partir de deux vecteurs, un pour l'item lexical dont nous voulons l'antonyme (noté $V(item)$), l'autre pour le contexte, nous construisons le vecteur opposé (noté $V(contexte)$). L'idée de base de la construction est d'accentuer sur les notions saillantes à la fois dans $V(item)$ et $V(contexte)$. Si ces idées peuvent être opposées alors l'antonyme doit posséder les idées inverses en proportion identique. Nous en déduisons la formule :

$$Canti_P(V(item), V(contexte)) = \bigoplus_{i=1}^{N} P_i \times CantiC(C_i, V(contexte)) \tag{3.6}$$

où $\bigoplus$ représente la somme vectorielle normée (cf 2.1.4) et P_i le poids du i-ème concept, c'est-à-dire l'importance que possède ce concept à la fois dans $V(item)$ et $V(contexte)$. En première approximation, nous posons :

$$P_i = V(item)_i \times V(contexte)_i \tag{3.7}$$

Nous rencontrons cependant un problème. Avec la formule 3.7, la fonction $Canti_P$ est symétrique $Canti_P(X,C) = Canti_P(C,X)$ or elle ne doit pas l'être. L'antonyme de ‹*froid*› par rapport à ‹*température*› est ‹*chaud*›, mais l'antonyme de ‹*température*› par rapport à ‹*froid*› n'est certainement pas ‹*chaud*›. Cela vient du fait de l'importance du contexte dans la symétrisation. Le contexte est le pivot de la symétrie. Il faut donc désymétriser cette fonction et renforcer l'influence des composantes de $V(item)$ sur celles de $V(contexte)$. Multiplier $V(item)_i$ par une valeur β ne modifierait pas, bien sûr, ce problème de symétrie. Une solution non-linéaire intéressante est d'introduire

un exposant dans l'opération. En effet, il semble pertinent de donner d'avantage de poids à une composante selon son importance.

$$P_i = V(item)_i^{\beta} \times V(contexte)_i \tag{3.8}$$

avec $\beta \in \mathbb{R} \mid \beta > 1$.

La valeur β doit donc être supérieure à 1 afin de privilégier les concepts de *item* sur ceux du *contexte*. Nous avons cherché expérimentalement quelle valeur donner à β. Nous nous sommes vite aperçus que lui donner une valeur trop grande favorisait déraisonnablement la composante la plus importante. Cela est approprié pour un vecteur qui n'a qu'une composante majeure (les composantes du vecteur dont la valeur est supérieure à la moyenne de l'ensemble des composantes du vecteur sont considérées comme majeures) comme c'est le cas avec les vecteurs génératifs mais pour d'autres, l'effet est néfaste. Dans le cas d'un vecteur qui a plusieurs composantes importantes, un β important appuierait uniquement sur C_i la composante la plus forte et le vecteur antonyme serait extrêmement proche de $Canti_{P\alpha}(c_i)$. Par exemple, considérons que le vecteur de ‹*Ferrari*› est la combinaison linéaire des vecteurs de AUTOMOBILE, de ROUGE et de RAPIDITÉ avec la composante AUTOMOBILE relativement majoritaire. Dans le cas où β serait grand, on aurait $Canti_P(V(Ferrari)) = V(\text{AUTOMOBILE})$, si l'on considère que l'antonyme de AUTOMOBILE est AUTOMOBILE. Or, l'antonyme serait plus raisonnablement un vecteur dont les composantes importantes seraient plutôt AUTOMOBILE, VERT et LENTEUR ce qui pourrait correspondre, par exemple à une « *deux-chevaux verte* ».

La valeur de β pourrait donc être calculée en fonction des caractéristiques des composantes du vecteur. Par exemple, en utilisant le coefficient de variation de $V(item)$. Le coefficient de variation $CV(V)$ est une mesure statistique normalisée (sans unité) de la "conceptualité" du vecteur V. Il est d'autant plus important que les composantes du vecteur sont contrastées et vaut 0 si elles ont toutes la même valeur (cf. 2.1.5.4).

Puisque la valeur de β doit être supérieure à 1, nous posons :

$$\beta = 1 + CV(V(item)) \tag{3.9}$$

Nous obtenons donc, la formule suivante :

$$P_i = V(item)_i^{1+CV(V(item))} \times V(contexte)_i \tag{3.10}$$

Nous remarquerons que pour le vecteur $\vec{\mathbf{1}}$ dont toutes les composantes sont identiques, nous sommes ramenés à $\beta = 1$. Dans ce cas, le contexte prend autant d'importance que l'item. Il s'agit d'un cas totalement artificiel, ce vecteur n'étant certainement pas représentable dans une langue puisqu'il signifierait tout et son contraire. Le résultat du contraire de $\vec{\mathbf{1}}$ serait de renvoyer un vecteur égal à celui du contexte, ce qui n'est pas le résultat escompté. Il convient de préciser toutefois que dans ce cas, la mesure de potentiel d'antonymie, que nous détaillerons en 3.3.3.3, est nulle. Le vecteur rendu est donc inexploitable.

Dans le cas de $\vec{\mathbf{0}}$, le coefficient de variation n'est pas défini. Nous pouvons l'étendre et considérer qu'il est nul. Nous posons donc,

$$Canti_P(\vec{\mathbf{0}}, C) = \vec{\mathbf{0}} \quad \forall C \tag{3.11}$$

Expérimentalement, la formule (3.11) est satisfaisante dans de nombreux cas. Toutefois, il arrive que certains résultats soient décevants. Ainsi, V(‹*froid*›) dans le contexte V(‹*température*›) ne nous rend pas un vecteur proche de V(‹*chaud*›). Le problème vient du fait que le contexte n'allume pas les mêmes idées que V(‹*froid*›). La faiblesse du concept de FROID dans V(‹*température*›) entraîne l'écrasement de la composante ‹*chaleur*› dans le vecteur antonyme par rapport aux autres

composantes. D'où la nécessité de conserver les idées importantes de $V(item)$ quel que soit le contexte (le contexte ne servant plus alors qu'à insister sur ses idées les plus fortes).

La formule actuelle de construction du vecteur antonyme se base sur les notions saillantes à la fois dans $V(item)$ et $V(contexte)$ tout en ne négligeant pas celles qui ne sont importantes que dans $V(item)$. Si elles peuvent être opposées alors l'antonyme doit posséder l'idée inverse en proportion identique. D'où la redéfinition du poids :

$$P_i = V(item)_i^{1+CV(V(item))} \times max(V(item)_i, V(contexte)_i \tag{3.12}$$

En fait, nous ajoutons proportionnellement à son importance dans $V(item)$ et $V(contexte)$ chaque concept de base. Nous construisons ainsi un vecteur qui possède l'inverse des propriétés de $V(item)$.

Pour résumer, la figure 3.10 explicite le calcul du vecteur antonyme $Canti_P$ de $V(item)$ dans le contexte $V(contexte)$.

$$\begin{aligned} \vartheta^2 \rightarrow \vartheta \quad &: \quad X, C \rightarrow Z = Canti_P(X, C) \\ Canti_P(V(item), V(contexte)) \quad &= \quad \bigoplus_{i=1}^{N} P_i \times CantiC(C_i, V(contexte)) \\ \text{où } P_i \quad &= \quad V(item)_i^{1+CV(V(item))} \times max(V(item)_i, V(contexte)_i) \end{aligned}$$

FIG. 3.10 – Construction du vecteur antonyme : fonction $Canti_P$

Nous présenterons en 3.3.5.1 et 3.5.2.2 une évaluation des résultats de cette méthode de construction d'antonymes basée sur le voisinage. Ces résultats sont réalisés, pour la première, avant l'introduction dans l'apprentissage des fonctions de synonymie et d'antonymie et pour la seconde après leur introduction.

3.3.3.3 Mesures de potentiel d'antonymie et fonction de construction d'un antonyme global

Mesure de potentiel d'antonymie

Définitions Nous avons déjà remarqué que tous les items n'ont pas d'antonyme avéré. La mesure du potentiel d'antonymie évalue si un vecteur peut raisonnablement avoir un antonyme. La mesure est basée sur l'idée qu'un vecteur possède un antonyme si ses concepts importants peuvent être inversés et n'en possède pas dans le cas contraire. Sa formule est présentée dans la figure 3.11 où V_i, $CV(V)$ et $\mu(V)$ représentent respectivement le *i-ème* élément, le coefficient de variation et la moyenne arithmétique du vecteur V.

Le principe de cette mesure est de considérer l'importance des idées suivant l'écart à la moyenne des composantes. Si un concept qui possède un antonyme est une idée importante du vecteur il doit renforcer l'indice, sinon il doit l'affaiblir. L'inverse est appliqué si le concept ne possède pas d'antonyme. L'écart à la moyenne est renforcé par une mise à la puissance par le coefficient de variation du vecteur. Les écarts à la moyenne doivent avoir d'autant plus d'influence les écarts entre les composantes sont importants. En effet, la composante principale d'un vecteur génératif doit avoir plus de poids que la composante principale d'un terme quelconque.

$$\vartheta \rightarrow \mathbb{R} : V \rightarrow Z = \text{Anti}_{\text{Pot}}(V) = \log \frac{A}{B}$$

$$\text{où } A = \sum_{i=1}^{dimC} max((V_i - \mu(V))^{CV(V)} \times \theta(i), 0)$$

$$B = \sum_{i=1}^{dimC} max(-(V_i - \mu(V))^{CV(V)} \times \theta(i), 0)$$

$$\text{et } \theta(c) = \begin{cases} 1 & \text{si } opposé(c) \neq c \\ -1 & \text{si } opposé(c) = c \end{cases}$$

FIG. 3.11 – Calcul de la mesure de potentiel d'antonymie

Résultats et interprétation Nous avons, par exemple, les résultats suivants :

$Anti_{Pot_c}(V(\text{‹}existence\text{›}))$	=	$+0,82$	$Anti_{Pot_c}(V(\text{‹}femme\text{›}))$	=	$+0,54$
$Anti_{Pot_s}(V(\text{‹}existence\text{›}))$	=	$-1,47$	$Anti_{Pot_c}(V(FEMME))$	=	$+3,79$
$Anti_{Pot_d}(V(\text{‹}existence\text{›}))$	=	$-1,33$	$Anti_{Pot_c}(V(\text{‹}automobile\text{›}))$	=	$-2,15$
$Anti_{Pot_c}(V(EXISTENCE))$	=	$+2,81$	$Anti_{Pot_c}(V(AUTOMOBILE))$	=	$-8,16$
$Anti_{Pot_s}(V(EXISTENCE))$	=	$-6,02$	$Anti_{Pot_d}(\gamma(\text{‹}père\text{›}, \text{‹}homme\text{›}))$	=	$-0,28$

avec γ qui calcule la contextualisation faible de ‹*père*› dans le contexte ‹*homme*› (cf. 2.1.4.5). Nous constatons que l'item ‹*existence*› est susceptible de posséder un antonyme complémentaire mais ni antonyme scalaire ni antonyme dual. Les items lexicaux ‹*automobile*› et ‹*femme*› n'ont pas d'antonyme en complémentaire. Le concept ‹*existence*› obtient une mesure plus importante que le terme homonyme en antonymie complémentaire ce qui est dû à la polysémie. Nous retrouvons ce phénomène qui agit de manière inverse avec l'antonymie scalaire, le concept obtient dans ce cas une moins bonne note que l'item. La remarque peut être renouvelée avec ‹*femme*› qui est susceptible de posséder un antonyme et ‹*automobile*› qui lui n'en possède pas.

Le cas de ‹*père*› est très intéressant. La mesure $Anti_{Pot_d}$ du vecteur de ‹*père*› dans le contexte ‹*homme*› est négative, pourtant il est clair que ce mot possède au moins ‹*fils*› comme antonyme. De fait l'expérience montre que le vecteur antonyme obtenu grâce à $Canti_P$ est proche de celui de ‹*mère*› (0.3 radian soit 17°).

A priori, $Anti_{Pot}$ devrait être pourtant très facilement interprétable :
- si $Anti_{Pot} \leq 0$, le vecteur n'a pas d'antonyme ;
- si $Anti_{Pot} > 0$, le vecteur possède un antonyme ;
- si $Anti_{Pot} = 0$, ce cas ne peut survenir que si aucune composante correspondant à un concept qui possède un antonyme n'est supérieure à la moyenne. On peut donc en déduire que le vecteur n'a pas d'antonyme.

L'expérimentation, ‹*père*› par exemple, nous conduit à nuancer cette vision des choses. On peut considérer trois zones définies empiriquement :
- si $Anti_{Pot} < 0,4$ le vecteur n'a pas d'antonyme.
- si $Anti_{Pot} > 0,4$ le vecteur possède un antonyme.
- si $Anti_{Pot} \in [-0,4;0[\cup]0;0,4]$ il s'agit d'une zone de flou où l'on trouve des mots sans antonyme dont la note est positive et des mots avec antonyme dont la note est négative. Comme pour père, cette note négative n'empêche pas la fonction $Canti_P$ de renvoyer un vecteur acceptable dans la plupart des cas (après apprentissage cf. 3.5.2.2).
- le cas $Anti_{Pot} = 0$ est particulier. Il ne peut survenir que si aucune composante correspondant à un concept qui possède un antonyme n'est supérieure à la moyenne. On peut donc en déduire que, dans ce cas, le vecteur n'a pas d'antonyme.

Nous allons nous baser sur cette mesure et les fonctions d'antonymie complémentaire, scalaire et duale pour définir la fonction d'antonymie globale.

Fonction de construction d'un antonyme global Il s'agit d'essayer de simuler la fonction d'antonymie classique dont les gens ont l'intuition. En effet, seules les personnes qui ont réellement étudié le sujet savent qu'il existe plusieurs types d'antonymie. Pour toutes les autres, pour les dictionnaires, et a fortiori pour ce qui nous intéresse particulièrement, l'analyse sémantique de texte, elles se détachent peu l'une de l'autre voire sont confondues.

Nous devons donc fabriquer une fonction d'antonymie globale qui pour un vecteur V renvoie son vecteur antonyme. Une solution première consisterait à faire la somme des vecteurs résultant des trois fonctions d'antonymie mais elle n'est pas satisfaisante. En effet, ‹*existence*› a pour antonyme complémentaire ‹*inexistence*› mais est son propre antonyme en antonymie duale et en antonymie scalaire. Faire la somme des trois vecteurs donnerait un vecteur dont la direction ne semble pas clairement correspondre aux antonymes. Une solution qui semble plus appropriée considère les trois vecteurs antonymes en fonction de la mesure du potentiel d'antonymie. Ainsi, plus la mesure est importante, plus le vecteur de ce type d'antonymie sera prépondérant dans la construction du vecteur antonyme.

$$\vartheta^2 \rightarrow \vartheta : X, C \rightarrow Z = Canti_{P_g}(X, C)$$

$$Canti_{P_g}(X, C) = \bigoplus_{i \in \alpha} e^{Anti_{Pot_i}} \times CantiC_i(X, C)$$

$$avec \quad \alpha = \{complémentaire, scalaire, dual\}$$

FIG. 3.12 – Fonction de construction d'un antonyme global : $Canti_{P_g}$

Nous utilisons cette fonction dans l'analyse des textes et donc pour l'apprentissage des vecteurs conceptuels à partir des dictionnaires d'antonymes ou des définitions des dictionnaires classiques. Nous verrons ces effets dans les sections suivantes. Cette fonction est aussi utilisée pour la fonction lexicale d'évaluation de l'antonymie $Anti_R$.

3.3.4 Fonctions lexicales d'évaluation de l'antonymie

Les fonctions lexicales d'évaluation de l'antonymie permettent de savoir dans quelle mesure deux items lexicaux peuvent être antonymes l'un de l'autre. Comme pour la synonymie, on distingue la fonction d'évaluation relative de l'antonymie $Anti_R$ et la fonction d'évaluation partielle de l'antonymie $Anti_P$.

3.3.4.1 Fonction d'évaluation de l'antonymie relative : fonction $Anti_R$

Définitions Soient les vecteurs X et Y. La question est de savoir si on peut dire s'ils sont antonymes dans le contexte C. La distance d'antonymie $Anti_R$ est la mesure de l'angle formé par la somme des vecteurs X et Y et la somme de leur opposés $Anti_R(X,C)$ et $Anti_R(X,C)$ (figure 3.13).

$$\vartheta^3 \rightarrow [0, \tfrac{\pi}{2}] : X, Y, C \rightarrow Z = Anti_R(X, Y, C)$$

$$Anti_R(X, Y, C) = D_A(X \oplus Y,\ Canti_P(X, C) \oplus Canti_P(Y, C))$$

FIG. 3.13 – Calcul de la fonction d'évaluation de l'antonymie relative $Anti_R$

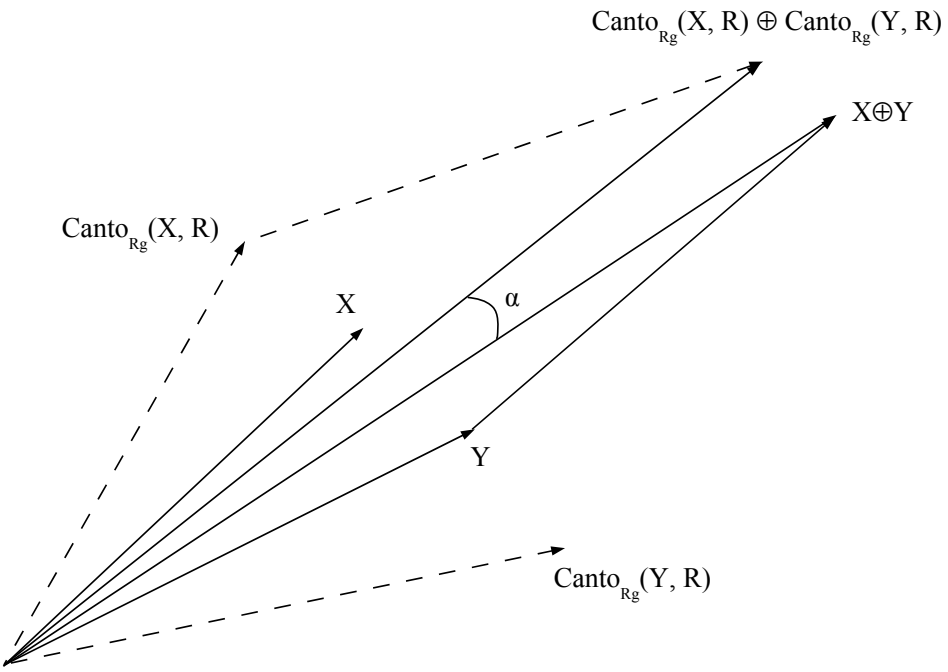


FIG. 3.14 – Représentation géométrique en 2D de la fonction lexicale d'évaluation de l'antonymie (angle α)

Propriétés La mesure d'évaluation de l'antonymie relative n'est pas une distance. Ce n'est qu'une pseudo-distance. Elle vérifie les propriétés de réflexivité, symétrie et inégalité triangulaire uniquement dans le sous-ensemble des items qui n'ont pas d'antonyme. Dans le cas général, elle ne vérifie pas la réflexivité. Les composantes des vecteurs sont positives et nous avons la propriété $Anti_R \in [0, \frac{\pi}{2}]$. Plus la mesure est petite, plus les deux termes sont antonymes dans le contexte. En revanche, ce serait une erreur de considérer que deux antonymes seraient à une distance avoisinant $\frac{\pi}{2}$ puisque, à ce moment là, ils n'auraient aucune idée en commun[66]. Il faut plutôt voir ici l'illustration que deux antonymes ont certaines idées en commun, celles qui ne sont pas opposables ou celles qui le sont mais dont l'activation est proche. Ils ne s'opposent que par certaines activations de concepts. Une distance proche de $\frac{\pi}{2}$ entre deux items lexicaux devrait être plutôt interprétée comme le fait que ces deux termes n'ont que peu d'idées en commun, une sorte d'anti-synonymie. Ce résultat confirme le fait que l'antonymie n'est pas exactement l'inverse de la synonymie mais lui est très liée. L'antonyme d'un item X n'est pas un terme qui ne partage aucune idée avec X mais un item qui s'oppose à X sur certaines idées !

[66] ce cas de figure est purement théorique, il n'existe dans aucune langue deux items lexicaux qui ne partagent aucune idée.

3.3.4.2 Fonction d'évaluation partielle de l'antonymie : fonction $Anti_P$

Définitions Soient les vecteurs X et Y. La question est de savoir dans quelle mesure on peut dire qu'ils sont antonymes. La fonction d'évaluation partielle de l'antonymie $Anti_R$, aussi appelée mesure d'évaluation partielle de l'antonymie, est la mesure de l'angle formé par la somme des vecteurs X et Y et la somme de leurs opposés dans un contexte caractérisé par un renforcement des idées communes aux 2 vecteurs à la manière de ce que nous avons présenté pour la fonction de synonymie partielle (cf. 4.1.1.2). Soit :

$$\vartheta^3 \rightarrow [0, \tfrac{\pi}{2}] : X, Y, C \rightarrow Z = Anti_P(X,Y)$$

$$Anti_P(X,Y) = Anti_R(X,Y,\gamma(X, X \oplus Y) \oplus \gamma(Y, X \oplus Y))$$

FIG. 3.15 – Calcul de la fonction partielle d'évaluation de l'antonymie $Anti_P$

Propriété Par héritage de $Anti_R$, cette fonction n'est pas non plus une distance.

3.3.5 Premiers résultats des fonctions d'antonymie sans apprentissage

Nous avons donc deux méthodes pour valider les fonctions d'antonymie. La première qui utilise uniquement la fonction lexicale de construction consiste à observer les termes les plus proches du vecteur qu'elle renvoie. La deuxième utilise la fonction lexicale d'évaluation qui repose en partie sur la fonction de construction. Dans ce cas, il s'agira d'observer si les rapports d'antonymie entre les termes calculés par la fonction semblent conformes à la perception que l'on peut en avoir naturellement.

Dans cette section, nous observons les résultats obtenus sur une base fabriquée sans l'aide de la fonction d'antonymie. Il ne s'agit donc que d'un premier aperçu, partiel car ne portant que sur peu d'items, de cette fonction. Les résultats ne peuvent être, pour l'instant, observés que sur des items dont les définitions ne comportent pas (ou du moins peu) de négations. Nous verrons dans la section 3.5 les premiers effets sur la base de l'utilisation de la fonction lexicale de construction d'antonymes dans l'apprentissage et en 4.1.2 une amélioration de la fonction de constructions d'antonymes et ses effets sur les vecteurs de la base.

3.3.5.1 Voisinage anti-thématique sans apprentissage : évaluation de la fonction de construction d'antonymes

Une première méthode de validation des résultats consiste à comparer les vecteurs conceptuels calculés par $Canti_{P_g}$ aux vecteurs conceptuels de la base grâce à la méthode de voisinage. Les plus proches, au sens de la distance angulaire, c'est-à-dire ceux qui sont les plus thématiquement proches de $Canti_{P_g}(V(item), V(contexte))$, devraient aussi être en antonymie thématique de $V(item)$. Les méthodes de proximité anti-thématiques *AntiLex* présentées en 3.3.3.2 nous permettent de réaliser cette opération. Remarquons que ce sont ces méthodes qui nous ont permis de concevoir empiriquement les fonctions $Anti_P$. Bien entendu, les fonctions d'antonymie peuvent être testées sur l'intégralité de la base mais les résultats ne sont pas, à ce stade de notre exposé, satisfaisant pour l'ensemble de ces items.

Observons tout d'abord l'action de la fonction sur VIE et MORT, des concepts qui sont en situation d'antonymie complémentaire et scalaire mais qui ne sont pas en situation d'antonymie duale :

$\mathcal{V}(\mathit{CantiC}_c(\textsc{vie}))$ = (*MORT* ; 0,0344) (‹*être guéri de tous les maux*› ; 0,0344) (‹*mort*› ; 0,4272) (‹*haschischin*› ; 0,568) (‹*assassin*› ; 0,5683) (‹*assassineur*› ; 0,5885) (‹*mortel*› ; 0,6631) ...
$\mathcal{V}(\mathit{CantiC}_c(\textsc{mort}))$ = (*VIE* ; 0,023) (‹*vie quotidienne*› ; 0,023) (‹*vie*› ; 0,382) (‹*vivant*› ; 0,534) (‹*naissance*› ; 0,621) ...
$\mathcal{V}(\mathit{CantiC}_d(\textsc{vie}))$ = (*VIE* ; 0,01) (‹*vie quotidienne*› ; 0,01) (‹*vie*› ; 0,33) (‹*vivant*› ; 0,516) (‹*naissance*› ; 0,598) ...
$\mathcal{V}(\mathit{CantiC}_d(\textsc{mort}))$ = (*MORT* ; 0,025) (‹*être guéri de tous les maux*› ; 0,025) (‹*mort*› ; 0,38) (‹*haschischin*› ; 0,529) (‹*assassin*› ; 0,532) (‹*assassineur*› ; 0,564) (‹*mortel*› ; 0,617) ...

Remarquons que dans le cas d'opposition de concepts, il est souvent inutile de prendre un référent puisque ce dernier a un poids très faible du fait de l'importance du coefficient de variation du vecteur correspondant au concept (cf. formule de Canti_P : figure 3.10). En ce qui concerne le voisinage de $\mathcal{V}(\mathit{CantiC}_c(\textsc{vie}))$, l'item lexical le plus proche est le concept *MORT*, ce qui est plus qu'acceptable en antonymie complémentaire (cf 3.3.1). Les résultats de $\mathcal{V}(\mathit{CantiC}_c(\textsc{mort}))$ sont en tout point comparables, avec cette fois, le vecteur du concept *VIE* comme plus proche thématique.

Dans ces résultats, on peut s'apercevoir que la distance entre le vecteur antonyme calculé et le vecteur antonyme réel n'est pas nulle. Ce phénomène est explicable par deux facteurs :

- *la construction des vecteurs* : les vecteurs génératifs, les vecteurs qui correspondent aux concepts de base, ne sont pas indépendants les uns des autres à deux niveaux. Au premier niveau, leur construction est basée sur leur distance ultramétrique à l'intérieur d'une hiérarchie de concepts issue, dans notre cas, du thésaurus Larousse [Larousse, 1992]. Au deuxième, la prise en compte des idées transversales à cette hiérarchie fait qu'ils se renforcent entre eux (voir la construction des vecteurs génératifs 2.3.2). Avec des vecteurs booléens, où seule la composante correspondant au concept ne serait pas nulle, les fonctions Canti_P rendraient exactement le vecteur du concept antonyme.
- *le bruit dû à la méthode de construction* : la méthode de construction d'un vecteur antonyme n'est pas d'une précision absolue. Son seul but est de construire un vecteur approchant le plus possible du vecteur antonyme tel qu'il devrait être.

Observons maintenant les résultats de la fonction *AntiLex* sur les termes ‹*mort*› et ‹*vie*›. Nous avons en antonymie complémentaire et en antonymie duale :

$\mathit{AntiLex}_{P_c}$(‹*mort*›, ‹*décédé*›) = (‹*vie*› ; 0,376) (*VIE* ; 0,421) (‹*vie quotidienne*› ; 0,421) (‹*s'animer*› ; 0,644) (‹*demi-vie*› ; 0,6443) (‹*avoir la vie devant soi*› ; 0,6752) (‹*viabilité*› ; 0,6914) (‹*naître*› ; 0,7083) (‹*vivant*› ; 0,7592) ...
$\mathit{AntiLex}_{P_c}$(‹*vie*›, ‹*existence*›) = (‹*mort*› ; 0,3367) (‹*être guéri de tous les maux*› ; 0,3573) (*MORT* ; 0,3573) (‹*haschischin*› ; 0,3672) (‹*assassin*› ; 0,3675) (‹*assassineur*› ; 0,3774) (‹*mort*› ; 0,407) ...
$\mathit{AntiLex}_{P_d}$(‹*mort*›, ‹*décédé*›) = (‹*mort*› ; 0,283) (‹*être guéri de tous les maux*› ; 0,33) (*MORT* ; 0,33) (‹*haschischin*› ; 0,3543) (‹*assassin*› ; 0,3647) (‹*assassineur*› ; 0,37) ...
$\mathit{AntiLex}_{P_d}$(‹*vie*›, ‹*existence*›) = (‹*vie*› ; 0,376) (‹*vie quotidienne*› 0,421) (*VIE* 0,421) (‹*s'animer*› 0,644) (‹*demi-vie*› 0,6443) (‹*avoir la vie devant soi*› 0,6752) (‹*viabilité*› 0,6914) (‹*naître*› 0,7083) (‹*vivant*› 0,7592) ...

Ces premiers résultats semblent relativement satisfaisants. En antonymie complémentaire, la thématique associée aux vecteurs opposés à ‹*vie*› dans un contexte d'‹*existence*› calculée par la fonction de construction est proche de l'idée de ‹*mort*›. Inversement, la thématique associée au vecteur opposé à ‹*mort*› dans un contexte de ‹*décédé*› est proche d'une thématique de ‹*vie*›. Il en est de même en antonymie scalaire tandis qu'en antonymie duale, on constate bien que la propriété des points fixes est vérifiée. On peut tout de même noter que la distance entre le vecteur

calculé et l'item le plus proche est relativement élevée si on la compare aux distances thématiques observées. Cette importance n'est pas le seul fait de la polysémie des termes (rappelons que le voisinage tient compte du vecteur conceptuel général du terme et n'est absolument pas contextualisé). L'introduction des méthodes d'antonymie dans l'apprentissage ainsi que l'ajout de dictionnaires d'antonymes améliore grandement ces résultats comme nous le verrons en 3.5.2.2.

Observons maintenant le cas d'un terme, ‹*action*›, qui possède à la fois un antonyme complémentaire ‹*inaction*› et un antonyme dual ‹*réaction*›. Nous avons :

> $\mathit{AntiLex}_{P_c}$(‹*action*›, ‹*mouvement*›) = (‹*inaction*› ; 0,513) (‹*repos*› ; 0,491) (‹*oisiveté*› 0,644) (‹*flemmarder*› 0,664) (‹*éteint*› 0,852) (‹*inactiver*› 0,852) ...
> $\mathit{AntiLex}_{P_d}$(‹*action*›, ‹*mouvement*›) = (‹*réaction*› ; 0,42) (‹*contrecoup*› 0,5756) (‹*réagir contre*› 0,6674) (‹*réverbération*› 0,6715) (‹*répercussion*› 0,703) (‹*mutuellement*› 0,742) (‹*rebondissement*› 0,7711) (‹*choc en retour*› 0,7969) ...

Là encore, nous obtenons des items qui semblent très pertinents. On peut en particulier noter que les fonctions retrouvent bien que ‹*action*› $\perp\!\!\!\perp_c$ ‹*inaction*› et ‹*action*› $\perp\!\!\!\perp_d$ ‹*réaction*› mais toujours avec des distances relativement élevées.

Voyons maintenant les résultats obtenus avec la fonction globale de construction d'antonymes (cf. 3.3.3.3). Avec les termes précédemment utilisés, nous obtenons les résultats suivants :

> $\mathcal{V}(\mathit{CantiC}(\textsc{vie}))$ = (*MORT* ; 0,0363) (‹*être guéri de tous les maux*› ; 0,0363) (‹*mort*› ; 0,44) (‹*assassin*› ; 0,585) (‹*haschischin*› ; 0,592) (‹*assassineur*› ; 0,61) (‹*mortel*› ; 0,674) ...
> $\mathit{AntiLex}_P$(‹*vie*›, ‹*existence*›) = (‹*mort*› ; 0,41) (‹*être guéri de tous les maux*› ; 0,41) (*MORT* ; 0,467) (‹*haschischin*› ; 0,47) (‹*assassin*› ; 0,4948) (‹*assassineur*› ; 0,5032) ; ...
> $\mathit{AntiLex}$(‹*action*›, ‹*mouvement*›) = (‹*inaction*› ; 0,41) (‹*réaction*› ; 0,42) (‹*repos*› ; 0,49) (‹*effet*› ; 0,64) ...
> $\mathit{AntiLex}$(‹*lenteur*›, ‹*lenteur*› et ‹*rapidité*›) = (*RAPIDITÉ* ; 0,41) (‹*rapide*› ; 0,42) (‹*rapidité*› ; 0,58) (‹*vitement*› ; 0,65) (‹*vélocité*› ; 0,67) (‹*dare-dare*› ; 0,71)...
> $\mathit{AntiLex}$(‹*lent*›, ‹*lent*› et ‹*rapide*›) = (‹*promptitude*› ; 0,35) (‹*va-vite*› ; 0,39) (‹*laisser-aller*› ; 0,42) (‹*speed*› 0,43) ... (‹*rapidité*› 0,57) (‹*rapide*› ; 0,59)...

Ces résultats sont réalisés sur les vecteurs construits uniquement avec les définitions. L'apprentissage n'utilise pas encore, à ce stade, la fonction d'antonymie. Les résultats semblent toutefois très cohérents.

Le vecteur antonyme global du concept *VIE* est proche du vecteur du concept *MORT*. Si on compare à l'antonyme global, la variation est très légère puisque les vecteurs opposés en antonymie scalaire et duale n'interviennent que de façon très limitée contrairement au vecteur antonyme complémentaire. Nous avons, en effet, $\mathit{Anti}_{\mathit{Pot}_c}(\textsc{vie}) = +4,21$, $\mathit{Anti}_{\mathit{Pot}_s}(\textsc{vie}) = -2,54$ et $\mathit{Anti}_{\mathit{Pot}_d}(\textsc{vie}) = -5,34$. Il en est de même pour $\mathit{AntiLex}_P$(‹*vie*›, ‹*existence*›) de façon nettement moins contrastée toujours du fait de la polysémie. Dans ce cas, nous avons $\mathit{Anti}_{\mathit{Pot}_c}(\textsc{vie}) = +1,2$, $\mathit{Anti}_{\mathit{Pot}_s}(\textsc{vie}) = -1,35$ et $\mathit{Anti}_{\mathit{Pot}_d}(\textsc{vie}) = -1,86$.

L'exemple suivant concerne le terme ‹*action*›. L'antonyme global calculé semble là encore satisfaisant puisqu'il est à peu près médian au vecteur de son antonyme complémentaire ‹*inaction*› et de son antonyme dual ‹*réaction*›.

Pour finir, nous présentons un exemple qui ne renvoie pas le terme souhaité. La thématique associée à l'opposé de ‹*lent*› est certes relativement pertinente mais pas aussi bonne que ce qu'on aurait pu attendre. Nous essayerons de voir quelle amélioration nous aurons après apprentissage.

3.3.5.2 Résultats de la fonction d'évaluation de l'antonymie relative sans apprentissage

Une deuxième méthode d'évaluation des méthodes présentées est d'observer les résultats obtenus par la fonction lexicale d'évaluation de l'antonymie que celle-ci soit relative ou partielle.

En antonymie relative, nous obtenons :

$$
\begin{array}{lcl}
\mathit{Anti}_{R_c}(\textsc{existence},\textsc{inexistence},\ \textsc{vie}) & = & 0,05 \\
\mathit{Anti}_{R_c}(\text{‹}\mathit{existence}\text{›},\text{‹}\mathit{inexistence}\text{›},\text{‹}\mathit{vie}\text{›}) & = & 0,52 \\
\mathit{Anti}_{R_c}(\textsc{existence},\ \textsc{automobile},\ \textsc{vie}) & = & 1,48 \\
\mathit{Anti}_{R_c}(\text{‹}\mathit{existence}\text{›},\ \text{‹}\mathit{automobile}\text{›},\ \text{‹}\mathit{vie}\text{›}) & = & 1,03 \\
\mathit{Anti}_{R_c}(\textsc{automobile},\ \textsc{automobile},\ \textsc{automobile}) & = & 0,006 \\
\mathit{Anti}_{R_c}(\text{‹}\mathit{automobile}\text{›},\ \text{‹}\mathit{automobile}\text{›},\ \text{‹}\mathit{automobile}\text{›}) & = & 0,255 \\
\mathit{Anti}_{R_c}(\text{‹}\mathit{action}\text{›},\ \text{‹}\mathit{inaction}\text{›},\ \text{‹}\mathit{mouvement}\text{›}) & = & 0,376 \\
\mathit{Anti}_{R_d}(\text{‹}\mathit{action}\text{›},\ \text{‹}\mathit{inaction}\text{›},\ \text{‹}\mathit{mouvement}\text{›}) & = & 0,875 \\
\mathit{Anti}_{R_c}(\text{‹}\mathit{action}\text{›},\ \text{‹}\mathit{réaction}\text{›},\ \text{‹}\mathit{mouvement}\text{›}) & = & 0,907 \\
\mathit{Anti}_{R_d}(\text{‹}\mathit{action}\text{›},\ \text{‹}\mathit{réaction}\text{›},\ \text{‹}\mathit{mouvement}\text{›}) & = & 0,346 \\
 & & \\
\mathit{Anti}_{R}(\text{‹}\mathit{action}\text{›},\ \text{‹}\mathit{inaction}\text{›},\ \text{‹}\mathit{mouvement}\text{›}) & = & 0,408 \\
\mathit{Anti}_{R}(\text{‹}\mathit{action}\text{›},\ \text{‹}\mathit{réaction}\text{›},\ \text{‹}\mathit{mouvement}\text{›}) & = & 0,395
\end{array}
$$

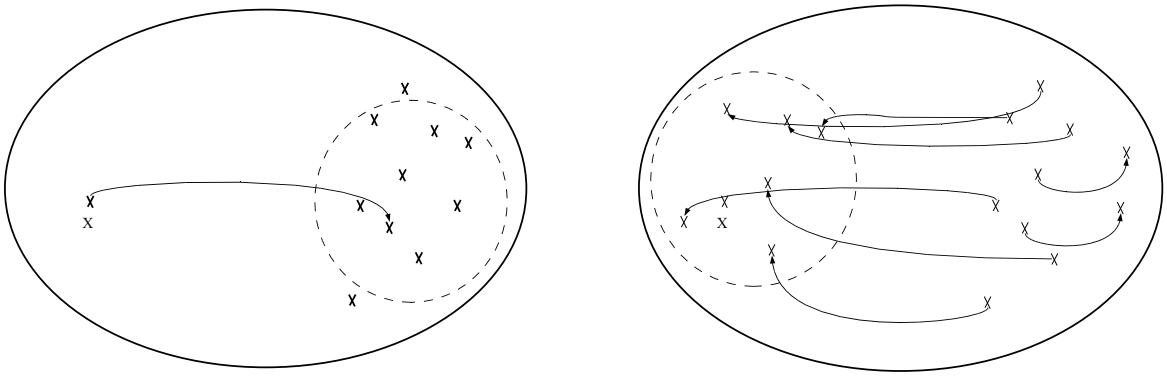
Les exemples ci-dessus illustrent bien nos propos précédents. Les concepts *EXISTENCE* et *INEXISTENCE* sont très fortement antonymes en antonymie complémentaire. L'effet de la polysémie explique que les items ‹*existence*› et ‹*inexistence*› soient moins antonymes que les concepts. On peut d'ailleurs le relever, pour chacun des exemples, les concepts non polysémiques par essence accentuent toujours les effets que l'on peut constater avec les termes polysémiques. En antonymie complémentaire, ‹*automobile*› est son propre antonyme mais de façon nettement moins prononcée que *AUTOMOBILE*.

La mesure de l'antonymie entre *AUTOMOBILE* et *EXISTENCE* est un exemple de notre remarque précédente sur les vecteurs qui ne partagent que peu d'idées. Aux alentours de $\frac{\pi}{2}$, cette mesure se comporte comme la distance angulaire. D'ailleurs, nous avons $D_A(\textsc{existence},\ \textsc{automobile}) = 1,534$.

Les résultats concernant ‹*action*› ainsi que ses antonymes semblent particulièrement intéressants. On constate ainsi, que la fonction donne une évaluation plutôt positive dans un contexte de ‹*mouvement*› pour l'opposition ‹*action*› ||| ‹*inaction*› en antonymie complémentaire et une évaluation plutôt négative en antonymie duale. L'inverse est vérifié pour l'opposition ‹*action*› ||| ‹*réaction*›. Dernier point important, la fonction d'évaluation générale de l'antonymie renvoie un résultat positif pour ces deux oppositions puisque nous avons $\mathit{Anti}_R(\text{‹}\mathit{action}\text{›},\ \text{‹}\mathit{inaction}\text{›},\ \text{‹}\mathit{mouvement}\text{›}) = 0,408$ et $\mathit{Anti}_R(\text{‹}\mathit{action}\text{›},\ \text{‹}\mathit{réaction}\text{›},\ \text{‹}\mathit{mouvement}\text{›}) = 0,395$ qui sont des valeurs très acceptables.

On pourrait envisager d'utiliser la fonction d'évaluation de l'antonymie globale pour chercher, dans le lexique conceptuel, les meilleurs antonymes d'un certain vecteur X. Cette méthode appliquerait cette fonction à chacun des vecteurs de la base et renverrait les plus proches de X. Ainsi, nous pourrions avoir une anti-thématique dont le calcul est inversé par rapport à celui de la fonction *AntiLex* qui, elle, oppose le vecteur X et renvoie le voisinage de cet opposé (voir figure 3.16). Deux arguments s'opposent à cette idée :

– La première objection est d'ordre technique. Le coût en temps, environ dix secondes pour une mesure sur un Sun à 8 processeurs 800 Mhz, reste, en effet, prohibitif. Sur une base d'environ 150 000 termes, il s'agirait d'une opération d'environ 17 jours, 8 heures et 40 minutes. Même si on peut noter que l'amélioration est importante depuis mon DEA puisque à l'époque la fonction d'évaluation de l'antonymie mettait de l'ordre d'une minute sur un PC 1.4Ghz, une telle opération reste totalement inenvisageable à l'heure actuelle. Ce changement important est due à une meilleure utilisation des ressources matérielles (parallélisation de l'exécution des trois fonctions d'antonymie) et des structures de données utilisées et non à une amélioration de la complexité des opérations (chacune des fonctions d'antonymie étant en temps constant de $O(n^2)$ avec n = 873 pour notre expérimentation).



(a) Fonction *AntiLex* : La fonction calcule le vecteur opposé à X puis renvoie ses voisins.

(b) La fonction inverse calcule les opposés de chacun des vecteurs de la base et renvoie les plus proches de X.

FIG. 3.16 – Comparaison entre l'action de la fonction *AntiLex* et celle de la fonction inverse.

– La deuxième objection est son manque d'intérêt. Cette fonction n'apporterait que peu de renseignements en comparaison de la fonction *AntiLex*. Il est sans doute beaucoup plus difficile d'évaluer des résultats avec cette seconde méthode qu'avec la première. En effet, avec la fonction *AntiLex*, il n'y a que le comportement d'un seul vecteur à envisager c'est-à-dire observer l'endroit de l'espace où il se situe par les voisins qui s'y trouvent, tandis que dans le cas de cette nouvelle fonction, il faudrait surtout observer les opposés renvoyés les uns par rapport aux autres ce qui rend extrêmement difficile voire impossible toute interprétation.

En antonymie partielle, Nous obtenons les résultats suivants :

$$
\begin{aligned}
Anti_{P_c}(\textsc{existence},\textsc{inexistence}) &= 0,03\\
Anti_{P_c}(\text{‹}existence\text{›},\text{‹}inexistence\text{›}) &= 0,44\\
Anti_{P_c}(\textsc{existence},\ \textsc{automobile}) &= 1,37\\
Anti_{P_c}(\text{‹}existence\text{›},\ \text{‹}automobile\text{›}) &= 0,907\\
Anti_{P_c}(\textsc{automobile},\ \textsc{automobile}) &= 0,006\\
Anti_{P_c}(\text{‹}automobile\text{›},\ \text{‹}automobile\text{›}) &= 0,255\\
Anti_{P_c}(\text{‹}action\text{›},\ \text{‹}inaction\text{›}) &= 0,347\\
Anti_{P_d}(\text{‹}action\text{›},\ \text{‹}inaction\text{›}) &= 0,798\\
Anti_{P_c}(\text{‹}action\text{›},\ \text{‹}réaction\text{›}) &= 0,980\\
Anti_{P_d}(\text{‹}action\text{›},\ \text{‹}réaction\text{›}) &= 0,312\\
\\
Anti_{P}(\text{‹}action\text{›},\ \text{‹}inaction\text{›}) &= 0,387\\
Anti_{P}(\text{‹}action\text{›},\ \text{‹}réaction\text{›}) &= 0,365
\end{aligned}
$$

En ce qui concerne les relations entre les termes, les fonctions d'antonymie partielles permettent de tirer les mêmes conclusions que les fonctions d'antonymie relatives. On retrouve bien, entre autres, que ‹*existence*› $\Vdash_c$ ‹*inexistence*› et ‹*action*› $\Vdash_c$ ‹*inaction*› et ‹*action*› $\Vdash_d$ ‹*réaction*› en antonymie duale.

Comparons maintenant les résultats concernant les deux antonymies. Comme on pouvait s'y attendre, chacune des valeurs en antonymie partielle est inférieure à celles observées en antonymie relative. En effet, l'antonymie absolue est, rappelons le, l'antonymie relative dont le référent est constitué par la somme contextualisée des deux vecteurs. On assiste en fait au même phénomène que celui qui nous avons pu observer pour la synonymie partielle (cf 3.2.3.2). Ici

aussi, l'opération se fait relativement à un contexte "artificiel" qui permet d'insister d'avantage sur les idées qui sont en commun aux deux vecteurs, ce qui les rapproche d'autant plus.

Les exemples qui concernent le concept *AUTOMOBILE* et le terme ‹*automobile*› sont eux des cas particuliers. Nous avons, en effet, $Anti_{R_c}$(‹*automobile*›, ‹*automobile*›, ‹*automobile*›) = $Anti_P$(‹*automobile*›, ‹*automobile*›). Cette propriété découle directement des propriétés d'idempotence de la somme vectorielle (cf. 2.1.4.1) et de la contextualisation faible (cf. 2.1.4.5).

3.3.5.3 Fonctions de voisinage antonymique

Nous exposons ici la fonction de voisinage antonymique dont nous ne donnons que les définitions. En effet, elles ont été inventées postérieurement aux recherches présentées dans ce chapitre, nous n'avons donc pas pu observer les résultats qu'elles auraient effectivement donnés mais tout laisse à penser que nous aurions eu des résultats aussi affinés que ceux observés si on compare ceux donnés par la fonction de voisinage synonymique et ceux de la fonction de proximité thématique. Nous introduisons dans cette section, deux outils de proximité basés sur les fonctions de synonymie.

Principes et définitions La *fonction de voisinage antonymique* permet de connaître les items lexicaux dont les vecteurs sont les plus antonymes d'un vecteur X donné selon les fonctions d'antonymie et de synonymie relatives. Le principe de cette fonction est de calculer $\overline{X}$ le vecteur antonyme de X puis de calculer le voisinage synonymique de $\overline{X}$. Soit :

$$\begin{gathered}\left|\mathcal{V}_{Anti_{R_k}}(X,C,R)\right| = k,\ \forall Y \in \mathcal{V}_{Anti_{R_k}}(X,C,R),\ \forall X \notin \mathcal{V}_{Anti_{R_k}}(X,C,R),\\ Syn_R(Canti_R(X,C,R),Y,C) \leq Syn_R(Canti_R(X,C,R),Z,C)\end{gathered} \tag{3.13}$$

De même, la *fonction de voisinage antonymique partielle* permet de connaître les items lexicaux dont les vecteurs sont les plus antonymes d'un vecteur X donné selon les fonctions d'antonymie partielle et de synonymie relative. Soit :

$$\begin{gathered}\left|\mathcal{V}_{Anti_{P_k}}(X,C)\right| = k,\ \forall Y \in \mathcal{V}_{Anti_{P_k}}(X,C),\ \forall X \notin \mathcal{V}_{Anti_{P_k}}(X,C),\\ Syn_R(Canti_P(X,C),Y,C) \leq Syn_R(Canti_P(X,C),Z,C)\end{gathered} \tag{3.14}$$

Enfin, la *fonction de voisinage antonymique absolue* permet de connaître les items lexicaux dont les vecteurs sont les plus antonymes d'un vecteur X donné selon la fonction d'antonymie absolue et la fonction de synonymie relative. Soit :

$$\begin{gathered}\left|\mathcal{V}_{Anti_{A_k}}(X)\right| = k,\ \forall Y \in \mathcal{V}_{Anti_{A_k}}(X),\ \forall X \notin \mathcal{V}_{Anti_{A_k}}(X),\\ Syn_R(Canti_A(X),Y,C) \leq Syn_R(Canti_A(X),Z,C)\end{gathered} \tag{3.15}$$

En ce qui concerne l'utilisation pratique de ces fonctions, elles souffrent de la même difficulté d'utilisation que les méthodes de voisinage synonymique auxquelles s'ajoute le temps de calcul du vecteur antonyme c'est-à-dire un temps aux alentours des cinquante secondes sur un Sun à 8 processeurs 800 Mhz (une dizaine pour le calcul du vecteur antonyme et un peu plus de trente pour le voisinage synonymique).

3.4 Utilisation des fonctions lexicales de construction des relations symétriques dans l'apprentissage

Dans les sections qui ont précédé, nous avons présenté des fonctions de construction pour chaque relation symétrique : pour la synonymie, la fonction de construction relative $Csyn_R$ et la

fonction de construction absolue $Csyn_P$ et pour l'antonymie la fonction $Canti_P$. Nous allons maintenant étudier les méthodes qui utilisent ces fonctions en vue de l'amélioration des vecteurs. Dans le cas de la synonymie, l'utilisation est basée sur l'ajout de dictionnaires de synonymes tandis que les fonctions lexicales de construction d'antonymes interviennent dans l'analyse sémantique pour traiter les phrases à tournures négatives.

3.4.1 Utilisation de la synonymie dans l'apprentissage

Il s'agit d'utiliser les informations contenues dans les dictionnaires de synonymes pour fabriquer des objets LEXIES. Les dictionnaires de synonymes disponibles en ligne en tout cas au moment de l'expérience et toujours au moment où j'écris ces lignes ne différencient pas les synonymes en fonction de leur sens. Ainsi, pour chaque terme, on a une liste de ses synonymes telle celle de ‹*souris*› extraite du dictionnaire des synonymes du CRISCO (cf. 4.2.1.2) : ‹*amante*›, ‹*fille*›, ‹*nana*›, ‹*prostituée*›, ‹*rat*›, ‹*rire*›, ‹*rongeur*›, ‹*sourire*›. Dans une première approche, présentée ici, une seule lexie est donc considérée pour ce dictionnaire. Le vecteur conceptuel de cette lexie sera calculé par la fonction de synonymie partielle généralisée présentée à la section 3.2.2.4.

Bien que, comme nous le notions alors, cette généralisation doit être utilisée si le dictionnaire sépare les différents sens ou après classification des termes en fonction de leur sens, nous l'employons donc ici sur une liste fusionnant tous les sens. Il ne s'agit bien ici que d'une première approche pour utiliser les dictionnaires de synonymes. Nous verrons dans la section 4.2.1 que nous avons, par la suite, cherché à améliorer cette utilisation en essayant de regrouper les synonymes en fonction de leur sens et de créer alors une LEXIE pour chacun de ces groupes.

3.4.2 Utilisation de l'antonymie dans l'apprentissage

Dans les textes, en particulier ceux des définitions, il est relativement fréquent de trouver des tournures de phrases négatives. Il est alors intéressant pour une analyse sémantique de tenir compte de ces phénomènes. Prenons l'exemple de la définition de l'item ‹*existence*› issue de [Larousse, 2004] : « *Qui n'existe pas.* ». Son analyse morpho-syntaxique effectuée par SYGFRAN renvoie la structure présentée à la figure 3.17.

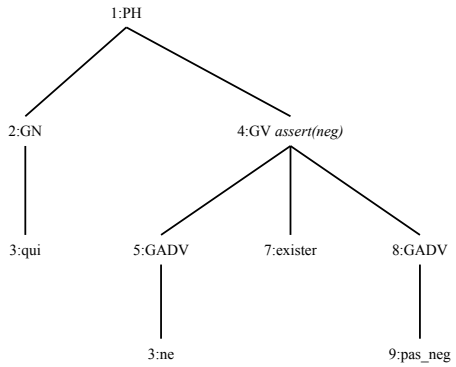


FIG. 3.17 – Analyse morpho-syntaxique de la définition de l'item ‹*existence*› issue de [Larousse, 2004] : « *Qui n'existe pas.* »

Une analyse sémantique telle que celle présentée en 2.1.6 affecterait à chaque feuille le vecteur conceptuel correspondant à l'item (ici uniquement à ‹*exister*› qui est le seul terme porteur de sens). Ensuite la remontée-descente s'effectue par des opérations de contextualisation. Nous aurons donc, dans ce cas, comme vecteur conceptuel final à associer à cette définition d'‹*inexister*› exactement le vecteur d'‹*exister*›, un parfait contre-sens !

SYGFRAN nous fournit une information importante qu'il déduit de différents schémas possibles de négation comme « *ne... pas* » « *ne... jamais* » « *ne... personne* » « *ne... rien* ». Il rajoute au noeud racine du sous-arbre l'information symbolique *assert(neg)*. Notre méthode consiste simplement à considérer ce symbole comme l'indicateur d'une situation favorable pour une inversion de la sémantique du sous-arbre considéré. Nous utilisons alors au niveau de ce nœud la méthode de construction d'antonyme $Canti_P$ au lieu de la méthode de contextualisation faible. Le référent utilisé est alors le contexte descendant du nœud père dans l'arbre. Cette méthode est utilisable tant pour les vecteurs conceptuels que pour les vecteurs sémantiques.

Dans notre exemple, le schéma « *ne... pas* » permet à SYGFRAN d'ajouter aux informations du nœud 4 le symbole *assert(neg)*. L'analyse sémantique va commencer par affecter à chaque feuille de l'arbre le vecteur conceptuel correspondant aux différents items : un vecteur vide pour les feuilles 3, 6, 9 qui correspondent aux items vides de sens ‹*qui*›, ‹*ne*› et ‹*pas*› et le vecteur conceptuel relatif à l'item ‹*exister*› pour la feuille 7. L'analyse fait remonter les différents vecteurs au niveau supérieur en faisant leur somme pondérée par rapport aux fonctions syntaxiques. Le nœud 4 aura donc un vecteur correspondant exactement au vecteur d'‹*exister*›. Puisque ce nœud contient l'information *assert(neg)*, on applique la fonction de construction globale d'antonymes $Canti_P$ avec comme contexte le vecteur du nœud père, c'est-à-dire, dans notre exemple, un vecteur nul, à la première remontée et ensuite, le vecteur global de la définition. Ainsi, cette méthode permet bien d'obtenir le vecteur antonyme à ‹*exister*› pour le calcul du vecteur ‹*inexister*›.

Il aurait été logique, à l'image de ce que nous faisons pour la synonymie, d'utiliser des dictionnaires d'antonymes dans notre apprentissage mais à l'époque des recherches présentées dans cette section, il n'en n'existait pas de gratuit pour le français. Nous n'avons eu accès à une telle source que tard dans nos travaux, en Octobre 2004 (cf. 4.2.2.1).

3.5 Premiers effets sur l'apprentissage

Les fonctions lexicales de construction des relations symétriques ont donc été introduites dans l'apprentissage suivant les méthodes précédemment présentées. Dans cette section, nous cherchons à en évaluer les effets. Ici, nous n'étudions pas l'apport de l'une ou de l'autre des deux relations. Il s'agit ici de la manifestation de la volonté holistique qui nous anime dans nos recherches et sur laquelle nous reviendrons particulièrement au chapitre 5. Nous considérons ainsi que les apports bénéfiques de chacune des deux relations sont supérieurs à ceux que nous obtiendrions si les relations étaient prises indépendamment. En d'autres termes, le tout est supérieur à la somme des parties. C'est pour cette raison que les relations de synonymie et d'antonymie ont été ajoutées quasi-simultanément à l'apprentissage et que nous en étudions parallèlement les effets.

Pour évaluer la base, nous avons un certain nombre d'outils à notre disposition : le voisinage thématique présenté à la section 2.1.3.2 ainsi que quatre outils introduits dans ce chapitre : le *voisinage synonymique*, le *voisinage anti-thématique*, la *fonction d'évaluation de la synonymie* et la *fonction d'évaluation de l'antonymie*. L'idée générale d'évaluation de la base consiste à essayer de mettre en évidence les relations sémantiques entre les items et de les comparer aux résultats précédemment trouvés. Nous allons constater dans les sections suivantes une amélioration de la structuration du lexique de notre base à travers les relations de synonymie et d'antonymie par les proximités thématique et anti-thématique et les fonctions lexicales d'évaluation.

Bien entendu, il est possible de tester les résultats et les améliorations sur toutes les entrées du lexique puisque celles-ci ont été modifiées de façon plus ou moins directe par l'ajout des fonctions symétriques dans l'apprentissage.

3.5.1 Fonctions lexicales d'évaluation après apprentissage

3.5.1.1 Fonction d'évaluation de la synonymie

La figure 3.18 présente l'évolution de l'évaluation de la synonymie entre quelques items après l'apprentissage. Le tableau du haut présente les valeurs de la synonymie partielle au moment de l'expérience tandis que le tableau du bas présente l'évolution en pourcentage des valeurs par rapport à celles présentées avant l'apprentissage (cf. 3.8).

avant / après	destinée	destin	vie	existence	mort	automobile	train	action	inaction	réaction
destinée	0	0,25	0,57	0,42	0,74	1,01	1,15	1,06	0,91	0,96
destin	**0,16**	0	0,58	0,45	0,74	1,02	1,14	1,01	0,86	0,9
vie	**0,47**	**0,46**	0	0,43	0,68	1,06	1,12	1,05	0,88	0,99
existence	**0,4**	**0,39**	**0,32**	0	0,76	1,14	1,23	1,15	1,04	0,98
mort	**0,78**	**0,79**	**0,83**	**0,85**	0	1,1	1,17	1,06	0,88	1,0
automobile	**1,23**	**1,21**	**1,25**	**1,25**	**1,26**	0	0,62	1,16	0,97	1,04
train	**1,24**	**1,26**	**1,21**	**1,29**	**1,22**	**0,59**	0	1,2	1,08	1,15
action	**1,18**	**1,21**	**1,22**	**1,19**	**1,23**	**1,25**	**1,24**	0	0,75	0,47
inaction	**1,13**	**1,2**	**1,18**	**1,23**	**1,17**	**1,3**	**1,25**	**0,85**	0	0,68
réaction	**1,14**	**1,18**	**1,2**	**1,21**	**1,25**	**1,22**	**1,28**	**0,44**	**0,82**	0

Rapprochement (%)	destinée	destin	vie	existence	mort	automobile	train	action	inaction	réaction
destinée	**0**									
destin	**36,0**	**0**								
vie	**17,5**	**20,7**	**0**							
existence	**4,7**	**13,3**	**25,6**	**0**						
mort	**5,1**	**-6,8**	**-22,1**	**-11,8**	**0**					
automobile	**-21,8**	**-18,6**	**-17,9**	**-9,6**	**-2,4**	**0**				
train	**-7,82**	**-10,5**	**-8,0**	**-10,9**	**-4,3**	**4,8**	**0**			
action	**-11,3**	**-19,8**	**-16,2**	**-3,5**	**-16,0**	**-7,8**	**3,3**	**0**		
inaction	**-24,2**	**-39,5**	**-34,1**	**-18,3**	**-33,0**	**-34,0**	**-15,7**	**-13,3**	**0**	
réaction	**-18,8**	**-31,1**	**-21,1**	**-23,5**	**-25**	**-17,3**	**-11,3**	**6,4**	**-20,6**	**0**

FIG. 3.18 – Évolution de l'évaluation de la synonymie partielle Syn_P entre quelques items après l'apprentissage.

On peut remarquer :
- une forte augmentation de la valeur de la synonymie partielle entre les termes ‹*destin*› et ‹*destinée*› (+36%) ou ‹*existence*› et ‹*vie*› (+25,6%) qui sont des items fortement synonymes dans l'usage ;
- une augmentation moindre pour ‹*existence*› et ‹*destinée*› (+4,7%) ainsi que pour ‹*action*› et ‹*réaction*› (+6,4%) qui sont des termes plus polysémiques et dont les champs sémantiques ont ainsi moins de points en commun ;
- un éloignement en particulier en ce qui concerne des termes qui n'ont que peu de rapport en langue comme ‹*automobile*› et ‹*inaction*› (−34%) ou ‹*réaction*› et ‹*destin*› (−31,1%).

3.5.1.2 Fonction d'évaluation de l'antonymie

Regardons maintenant les incidences sur les résultats de la fonction lexicale d'évaluation relative de l'antonymie présentés par la figure 3.19. Nous notons entre parenthèses les résultats avant apprentissage. Plusieurs points peuvent être soulignés :
- *l'incidence sur les concepts* est nulle puisque les vecteurs correspondant ne sont en aucun cas modifiés par l'apprentissage ;

$Anti_{R_c}$(*EXISTENCE*,*INEXISTENCE*, *VIE*)	=	0,05	(0,05)
$Anti_{R_c}$(‹*existence*›,‹*inexistence*›,‹*vie*›)	=	0,31	(0,52)
$Anti_{R_c}$(*EXISTENCE*, *AUTOMOBILE*, *VIE*)	=	1,48	(1,48)
$Anti_{R_c}$(‹*existence*›, ‹*automobile*›, ‹*vie*›)	=	1,19	(1,03)
$Anti_{R_c}$(*AUTOMOBILE*, *AUTOMOBILE*, *AUTOMOBILE*)	=	0,006	(0,006)
$Anti_{R_c}$(‹*automobile*›, ‹*automobile*›, ‹*automobile*›)	=	0,17	(0,255)
$Anti_{R_c}$(‹*action*›, ‹*inaction*›, ‹*mouvement*›)	=	0,264	(0,376)
$Anti_{R_d}$(‹*action*›, ‹*inaction*›, ‹*mouvement*›)	=	0,892	(0,875)
$Anti_{R_c}$(‹*action*›, ‹*réaction*›, ‹*mouvement*›)	=	0,92	(0,907)
$Anti_{R_d}$(‹*action*›, ‹*réaction*›, ‹*mouvement*›)	=	0,249	(0,346)
$Anti_R$(‹*action*›, ‹*inaction*›, ‹*mouvement*›)	=	0,21	(0,408)
$Anti_R$(‹*action*›, ‹*réaction*›, ‹*mouvement*›)	=	0,24	(0,395)

FIG. 3.19 – Résultats de la fonction lexicale d'évaluation relative de l'antonymie après apprentissage

- *l'incidence sur les items* semble être bénéfique :
  - les vecteurs ‹*existence*› et ‹*inexistence*› ont été largement modifiés. Maintenant, les deux items sont considérés comme "plus antonymes complémentaires" que précédemment.
  - L'item ‹*automobile*› apparaît d'autant plus comme un point fixe (0,17 radian contre 0,255 radian précédemment).
  - Les résultats concernant ‹*action*›, ‹*réaction*› et ‹*inaction*› sont plus contrastés : les paires déjà vues comme antonymes (‹*action*› $|||_c$ ‹*inaction*›, ‹*action*› $|||_d$ ‹*réaction*›) le sont davantage, celles qui l'étaient déjà moins le sont en proportion encore moindre (‹*action*› $\not|||_d$ ‹*inaction*›, ‹*action*› $\not|||_c$ ‹*réaction*›). En effet, même si les termes ‹*action*› et ‹*inaction*› sont antonymes ils partagent tout de même un grand nombre d'idées ce qui a pour effet de les maintenir à une certaine distance limite.

3.5.2 Voisinages

3.5.2.1 Voisinages thématique et synonymique

Voici quelques résultats obtenus avec la fonction de voisinage après apprentissage. Les valeurs entre parenthèses correspondent aux distances si les valeurs apparaissaient dans le voisinage avant l'apprentissage :

$\mathcal{V}$(‹*mort*›) = (‹*être guéri de tous les maux*› ; 0,35 (*0.46*)) (*MORT* ; 0,35 (*0,46*)) (‹*décédé*› ; 0,56) (‹*décès*› ; 0,58) (‹*agonie*› ; 0,62) (‹*disparition*› ; 0,65) (‹*extinction*› 0,66) ...
$\mathcal{V}$(‹*vie*›) = (‹*vie quotidienne*› ; 0,29) (*VIE* ; 0,29) (‹*existence*› ; 0,43) (‹*destin*› ; 0,47 (0,58)) (‹*longévité*› ; 0,61) (‹*sort*› ; 0,7) ...

Les résultats obtenus avec ces exemples montrent essentiellement l'apparition de nouveaux termes par rapport aux voisinages avant apprentissage. Une étude plus approfondie des items apparus nous a permis de constater qu'il s'agissait essentiellement des termes contenus dans la liste de synonymes de l'item dont on cherche le voisinage. Cette hypothèse semble confirmée par le voisinage synonymique partiel qui, bien entendu, rapproche les valeurs mais conserve globalement les mêmes voisins.

$\mathcal{V}_{Syn_P}$(‹*mort*›) = (‹*être guéri de tous les maux*› ; 0,3) (*MORT* ; 0,3) (‹*décès*› ; 0,44) (‹*décédé*› ; 0,45) (‹*disparition*› ; 0,49) (‹*agonie*› ; 0,59) (‹*extinction*› ; 0,64) ...

$\mathcal{V}_{Syn_P}$(‹*vie*›) = (‹*vie quotidienne*› ; 0,23) (*VIE* ; 0,23) (‹*existence*› ; 0,32) (‹*destin*› ; 0,37) (‹*longévité*› ; 0,58) (‹*sort*› ; 0,62) ...

On assiste par l'introduction dans l'apprentissage des fonctions de synonymie à un certain rapprochement du voisinage thématique et du voisinage synonymique.

Pour finir, afin d'être relativement complets, bien que ces résultats n'apportent pas vraiment d'informations supplémentaires, nous présentons les résultats de ‹*destin*› et ‹*vie*› après apprentissage.

$\mathcal{V}$(‹*destin*›) = (‹*destin*› ; 0,0) (‹*destinée*› ; 0,33 (0,51)) (‹*sort*› ; 0,36 (0,56)) (‹*fatalité*› ; 0,4 (0,62)) (‹*vie*› ; 0,42) (‹*providence*› ; 0,5) (‹*fatidique*› ; 0,51) (‹*détermination*› ; 0,63 (0,57))

$\mathcal{V}_{Syn_R}$(‹*destin*› ; ‹*vie*›) = (‹*destin*› ; 0,0) (‹*destinée*› ; 0,23(0,25)) (‹*sort*› ; 0,32 (0,38)) (‹*fatalité*› ; 0,35 (0,46)) (‹*providence*› ; 0,38) (‹*fatidique*› ; 0,4) (‹*déterminisme*› ; 0,41 (0,44)) (‹*vie*› ; 0,53) (‹*détermination*› ; 0,54 (0,41)) (‹*déterminer*› ; 0,6 (0,43))

3.5.2.2 Voisinage anti-thématique

Voici quelques résultats obtenus avec la fonction globale de construction d'antonymes après apprentissage. Les valeurs entre parenthèses correspondent aux distances si les valeurs apparaissaient dans le voisinage anti-thématique avant l'apprentissage :

AntiLex$_P$(‹*vie*›, ‹*existence*›) = (‹*mort*› ; 0,25 (*0,41*)) (‹*être guéri de tous les maux*› ; 0,25 (*0,41*)) (*MORT* ; 0,35 (*0,46*)) (‹*letal*› ; 0,56) (‹*décès*› ; 0,61) (‹*décédé*› ; 0,63) ...

AntiLex(‹*action*›, ‹*mouvement*›) = (‹*réaction*› ; 0,27 (*0,42*)) (‹*inaction*› ; 0,28 (*0,41*)) (‹*inactif*› ; 0,54) (‹*oisif*› ; 0,69) (‹*effet*› ; 0,72 (*0,64*)) ...

AntiLex(‹*lenteur*›, ‹*lenteur*› et ‹*rapidité*›) = (*RAPIDITÉ* ; 0,3 (*0,41*)) (‹*rapidité*› ; 0,34 (*0,58*)) (‹*rapide*› ; 0,37 (*0,42*)) (‹*instantané*› ; 0,63) (‹*expéditif*› ; 0,64)...

AntiLex(‹*lent*›, ‹*lent*› et ‹*rapide*›) = (‹*hâte*› ; 0,28) (‹*speed*› ; 0,35 (0,43)) ... (‹*rapide*› 0,41 (0,59))... (‹*rapidité*› ; 0,43 (0,57))...

AntiLex(‹*incombustibilité*›, ‹*combustibilité*› et ‹*incombustibilité*›) = (‹*incombustibilité*› ; 0,21) (‹*monergol*› ; 0,32) (‹*métaldéhyde*› ; 0,32) (‹*postcombustion*› ; 0,34) (‹*combustibilité*› ; 0,34) (‹*phlogistique*› ; 0,36) (‹*précombustion*› ; 0,37)...

AntiLex(‹*combustibilité*›, ‹*combustibilité*› et ‹*incombustibilité*›) = (‹*combustibilité*› ; 0,14) (‹*précombustion*› ; 0,23) (‹*incombustibilité*› ; 0,24) (‹*monergol*› ; 0,29) (‹*phlogistique*› ; 0,33) (‹*métaldéhyde*› ; 0,38) (‹*postcombustion*› ; 0,39)...

Ces résultats permettent de voir une amélioration globale du voisinage anti-thématique. Les voisins de l'opposé de ‹*vie*› par rapport à ‹*existence*› sont toujours constitués par le lexique proche de ‹*mort*› mais de façon nettement plus proche qu'avant l'apprentissage. On a le vecteur de ‹*mort*› à une distance de $0,25$ radian ($14,3^\circ$) contre une distance de $0,41$ radian ($23,5^\circ$) précédemment. De telles constatations peuvent être formulées avec ‹*action*› par rapport à un contexte de ‹*mouvement*› ainsi qu'avec ‹*lenteur*› par rapport à un contexte constitué de ‹*lenteur*› et de ‹*rapidité*›.

En revanche, le résultat qui concerne le substantif ‹*lent*› est surprenant surtout en comparaison de celui de l'adjectif ‹*lenteur*›. Pour comprendre, il faut examiner les définitions des termes et remarquer que dans aucune des définitions de ‹*lent*› dont nous disposons ne figure de tournure négative utilisant ‹*rapide*›. Ainsi, la fonction d'antonymie ne rentre pas dans la construction de ces vecteurs, et, de leur côté, les informations disponibles (définitions, synonymes) ne semblent pas avoir permis de faire émerger cette opposition d'une autre façon. Nous ne pourrons combler ce type de déficits que lorsque des dictionnaires d'antonymes seront à notre disposition (cf. 4.2.2).

Les deux derniers résultats semblent montrer que les termes ‹*combustibilité*› et ‹*incombustibilité*› sont des points fixes, on peut même constater que les deux voisinages sont similaires ce qui est caractéristique de termes thématiquement proches. Cet exemple paraît donc complètement en décalage avec la réalité. Lorsque nous sommes confrontés à une telle situation, lorsque visiblement l'indexation automatique n'est pas suffisante pour donner un vecteur pertinent à un terme, nous l'indexons manuellement de la même manière que celle que nous avons présentée pour le noyau à la section 2.3.5.2. Ici pourtant, cette opération est rendue difficile. En effet, lorsqu'on observe mieux les concepts associés dans le thésaurus Larousse à ‹*incombustibilité*› et ‹*combustibilité*›, on comprend mieux que le système vectoriel ne soit pas très efficace : ils sont tous les deux définis à l'aide du même concept unique : COMBUSTIBILITÉ. Nous avons donc affaire ici à un problème dû à la hiérarchie utilisée et non pas aux méthodes d'apprentissage.

Nous examinerons plus particulièrement ce dernier résultat dans la conclusion de cette partie mais globalement, nous pouvons constater que lors de l'apprentissage, l'analyse des définitions s'est affinée et la base de vecteurs est maintenant d'autant plus cohérente.

3.6 Conclusions du chapitre

Dans ce chapitre nous avons présenté comment les relations paradigmatiques ou syntagmatiques qu'il existe entre les items lexicaux peuvent être utiles dans le cadre de l'analyse des textes en général et des définitions en particulier. Nous avons proposé de modéliser ces relations sous la forme de fonctions lexicales de deux types : (1) les *fonctions lexicales de construction* qui permettent de construire un vecteur conceptuel en fonction des informations lexicales disponibles. Elles peuvent, au cours d'une analyse sémantique, rendre possible la gestion de certains phénomènes de négation ou alors donner la possibilité de construire des vecteurs à partir de listes de synonymes ou d'antonymes ; (2) les *fonctions lexicales d'évaluation* qui permettent d'estimer la pertinence d'une relation entre deux items. L'une des utilisations de ces dernières concerne l'évaluation globale d'une base.

Nous avons ensuite présenté les deux premières modélisations de fonctions lexicales qui ont été développées : les fonctions lexicales symétriques. Nous avons repris des travaux précédents concernant la fonction de synonymie relative [Lafourcade & Prince, 2001a] qui admet des propriétés mathématiques d'équivalence (ou de quasi-équivalence). Nous avons proposé une méthode partielle d'évaluation de la synonymie qui bien que ne respectant pas les critères d'équivalence peut s'avérer utile dans les cas où l'on veut évaluer la synonymie entre deux termes sans autre précision de contexte. Elles sont employées dans un autre outil introduit ici : le voisinage synonymique.

Les pendants de ces fonctions lexicales d'évaluation pour la construction de vecteurs sont la fonction relative et la fonction partielle de construction de synonymes introduite également dans ce chapitre. Nous les utilisons dans le cadre de l'apprentissage sur des vecteurs conceptuels à l'aide de dictionnaires de synonymes.

Notre exposé s'est poursuivi par les travaux sur l'antonymie entrepris en DEA et continués au cours de ma thèse. Nous avons proposé là encore des fonctions de construction et des fonctions d'évaluation. Nous présentons une méthode pour utiliser ces fonctions en analyse sémantique et une autre dans le cadre de l'apprentissage à partir de dictionnaires d'antonymes.

Ce chapitre se termine par un bilan sur l'apport de ces fonctions lexicales sur la cohérence globale des vecteurs. Nous avons montré que dans de nombreux cas, le gain qualitatif était relativement conséquent. Toutefois, certains exemples posent encore problème et nous allons en discuter dans la conclusion de cette partie.

Conclusions de la première partie

Dans cette première partie, nous avons cherché à montrer en quoi la question du sens pouvait être importante dans de multiples applications du TALN comme la traduction automatique, la catégorisation automatique, le résumé automatique ou la recherche d'informations. De nombreuses modélisations ont été réalisées et nous en avons présenté quelques exemples comme les réseaux sémantiques, les bases d'acceptions et les vecteurs saltoniens.

Dans cette thèse, nous nous intéressons plus particulièrement aux vecteurs conceptuels qui, avec les vecteurs sémantiques, constituent les vecteurs d'idées. Dans ce modèle, tout segment textuel peut-être caractérisé à partir de concepts censés pouvoir générer l'ensemble des idées exprimables en langue. Il s'agit de la projection de la notion linguistique de champ sémantique dans le modèle mathématique d'espace vectoriel. Il est possible de réaliser dans l'espace des vecteurs d'idées ϑ toute opération réalisable dans un espace vectoriel, avec la différence que, dans ϑ, ces opérations peuvent avoir des interprétations linguistiques voire psycholinguistiques. Nous avons ainsi défini un certain nombre de fonctions et de méthodes pour calculer ou interpréter les vecteurs d'idées (distance thématique, voisinage thématique, analyse sémantique, contextualisation faible, fonctions lexicales de production et d'évaluation de la synonymie et de l'antonymie, voisinage synonymique, voisinage antonymique, ...).

Les vecteurs conceptuels sont construits à partir de différentes sources (dictionnaires classiques, dictionnaires de synonymes, d'antonymes, sites Web, ...).L'idée est de permettre l'introduction d'autant de sources variées que possible afin que leurs informations se recoupent, se complètent les unes les autres et apportent ainsi une meilleure cohérence à la base de données vectorielle. L'apprentissage initial utilise des dictionnaires à usage humain classiques présentés sous forme électronique et recourt à une méthode d'analyse sémantique pour calculer le vecteur conceptuel de la définition. Afin d'améliorer l'analyse thématique en gérant de meilleure façon certaines tournures négatives grâce à l'antonymie et de rendre possible l'utilisation de dictionnaires de synonymes, nous avons étudié une modélisation de ces deux fonctions symétriques. Une amélioration globale de la pertinence des vecteurs a été obtenue grâce à un apprentissage complémentaire basé sur ces méthodes.

Un des outils introduits dans cette thèse est la méthode de voisinage anti-thématique *AntiLex*. Cette méthode utilise la fonction de construction d'un vecteur antonyme $Canti_R$ pour fabriquer le vecteur $\overline{X}$ antonyme d'un vecteur X dans un contexte C par rapport à un référent R, puis renvoie son voisinage thématique. Cette méthode nous a permis de mettre au point les fonctions de construction d'un vecteur antonyme. Elle est le seul outil à notre disposition pour nous permettre leurs évaluations directes[67].

Les expériences menées avec cette fonction nous ont permis de constater son efficacité globale. Ainsi, ses résultats semblent pertinents dans de nombreux cas comme ‹*action*› ou ‹*lenteur*› mais dans d'autres comme ‹*lent*›, ‹*incombustibilité*› et ‹*combustibilité*› ils ne sont pas cohérents. Si le premier exemple est facilement améliorable grâce à l'apport d'une nouvelle source, les deux autres souffrent d'une certaine incohérence dans la hiérarchie utilisée. En effet, il n'existe qu'un

[67] Rappelons que la notion de voisinage antonymique est postérieure à ces recherches.

seul concept qui définit à la fois l'idée de *COMBUSTIBILITÉ* et celle d'*INCOMBUSTIBILITÉ*. Ainsi lors de la création des VAC, les listes de vecteurs antonymes aux concepts, il n'a pas été possible de les opposer.

Mais le problème de la hiérarchie n'est pas le problème le plus difficile à résoudre puisqu'on pourrait, par exemple, en considérer une qui ne présenterait pas ce type de difficultés. Pour que la fonction *AntiLex* soit vraiment fiable, il faudrait que ces VAC soient satisfaisantes en toutes circonstances. Or, il semble évident que la construction de ces listes est parfois très subjective et donc soumise à caution. Aucune solution d'apprentissage des listes à partir d'antonymes n'est non plus envisageable. En effet, pour pouvoir trouver de façon automatique quels concepts se trouvent en opposition pour un couple, il faudrait disposer de vecteurs "bien indexés". Or nous cherchons justement à fabriquer ces fonctions pour avoir des vecteurs "bien indexés".

La modélisation des fonctions lexicales d'antonymie n'est donc pas parfaite et leur amélioration doit donc être envisagée. Il va falloir essayer de s'affranchir de la hiérarchie autant que faire se peut.

Suivant plusieurs théories [Damasio & Damasio, 1992], la cognition représenterait le sens en fonction, non seulement de la thématique, mais aussi en fonction des rapports lexicaux que les termes entretiennent entre eux. Par exemple, pour amplifier l'item ‹*fièvre*›, on emploiera plutôt l'adjectif ‹*forte*› au lieu de ‹*haute*› contrairement à l'anglais où l'on emploiera plutôt ‹*high*› alors que rien ne semble les prédestiner à un emploi plutôt qu'à un autre.

À l'inverse, la théorie de la sémantique componentielle permet de penser qu'un nombre suffisant de composantes, on parle parfois de 40000, pourrait permettre une représentation "complète" du sens. Dans ce cas, les capacités actuelles des machines ne nous permettrait pas d'atteindre ce niveau de précision. Nos vecteurs ont actuellement 873 dimensions, et certaines opérations, bien que de complexité linéaire, sont coûteuses en temps. Des analyses sémantiques sur de longues définitions peuvent prendre de l'ordre d'une minute, la moyenne devant se situer aux alentours de 2 secondes. Avec des vecteurs de dimension jusqu'à cinquante fois supérieure, nous atteindrions des temps de calculs peu faciles à envisager dans des conditions pratiques.

Les expériences menées jusqu'à présent sur les vecteurs d'idée tendent à exhiber leurs limites pour représenter le sens. Ils représentent plutôt le thème comme en témoigne, par exemple la distance angulaire qui semble devoir être interprétée comme une distance thématique.

De plus, comme nous l'évoquions la section 2.2.6, les expériences menées avec les vecteurs sémantiques sur la catégorisation ont montré la nécessité de ne pas se limiter à ce modèle pur. La solution apportée alors a été d'adjoindre à cette représentation d'autres vecteurs, ceux-là de type saltoniens. Dans le domaine particulier de ces recherche, « *la représentation statistique permet de mettre en évidence le vocabulaire discriminant tandis que la représentation conceptuelle permet, quant à elle, d'obtenir une vision globale du texte en projetant ce dernier sur un ensemble de concepts.* ». Or, les vecteurs statistiques sont une façon de représenter certains des rapports de surface entre les mots. Comme nous le remarquions en 1.2.2 « *L'analyse distributionnelle cherche à décrire les objets linguistiques en fonction du pouvoir d'associativité qu'ils possèdent ou ne possèdent pas entre eux* ». Les auteurs de cette méthode ont donc adopté, par la méthode mixte, une approche conforme aux théories cognitivistes actuelles.

Toutes ces expériences semblent montrer qu'il est encore illusoire de pouvoir se satisfaire d'informations uniquement conceptuelles pour fabriquer une base lexicale sémantique c'est pourquoi dans la suite de notre thèse nous allons étudier les moyens à notre disposition pour améliorer nos résultats. Dans le chapitre suivant, nous nous attacherons en particulier à ajouter aux fonctions lexicales symétriques des informations purement lexicales.

Deuxième partie

Vers la construction d'une Base Lexicale Sémantique

4

Apport d'informations purement lexicales pour les fonctions symétriques

La conclusion de la première partie a mis l'accent sur la difficulté à se limiter aux vecteurs d'idées pour représenter le sens des termes et les fonctions lexicales. Dans ce chapitre, nous cherchons à introduire des informations purement lexicales dans les fonctions symétriques. Cette introduction se fait par l'utilisation de la méthode de contextualisation forte pour les deux relations à laquelle s'ajoute, dans le cas de l'antonymie, celle des oppositions déjà rencontrées. La deuxième partie de ce chapitre concerne la représentation des informations que nous donnent les fonctions lexicales dans la base lexicale sémantique. Nous explorons des méthodes pour regrouper les synonymes et les antonymes en fonction de leur sens afin de fabriquer des LEXIES. Le manque de source pour les antonymes pendant une grande partie de ces travaux a été un grand problème. Nous l'avons, dans un premier temps, partiellement réglé en construisant, de façon semi-supervisée, une liste de couples d'antonymes en nous basant sur les oppositions de morphologie qu'il peut exister entre eux. Nous concluons par l'apport de ces informations dans la construction de la base lexicale sémantique.

Sommaire

DAns le chapitre précédent, nous avons présenté deux catégories de fonctions lexicales : les fonctions lexicales de production et les fonctions lexicales d'évaluation. Créées plus particulièrement dans le cadre des vecteurs conceptuels, les premières modélisations de ces fonctions ne considèrent pourtant que des informations de type vectoriel. En effet, à l'époque de leur conception, les limites du modèle vectoriel pur que nous évoquions dans la conclusion de la partie précédente n'étaient pas encore véritablement ressenties. De plus, les limites techniques des machines d'alors pouvaient empêcher une véritable utilisation de la contextualisation forte. Ainsi, les fonctions lexicales présentées dans la partie précédente reposent sur la méthode de contextualisation faible et peuvent être employées pour tout type de vecteurs d'idées que ce soient des vecteurs sémantiques ou des vecteurs conceptuels.

Dans le cadre plus particulier des vecteurs conceptuels, un autre type d'objet lexical s'ajoute à l'unique type des vecteurs sémantiques : le type LEXIE (cf. 2.3.4.2). Il constitue le socle d'une base lexicale sémantique et est composé à la fois d'informations vectorielles (vecteur conceptuel) et d'informations lexicales (morphologie, fréquence). Chacun d'eux est construit à partir de différentes sources d'informations (dictionnaires classiques, fonctions lexicales, Web, ...).

La méthode de contextualisation forte est basée sur ces objets LEXIE. Elle permet de considérer les vecteurs conceptuels de chacune des LEXIES d'un terme en fonction des informations contextuelles disponibles, informations qui peuvent être d'ordre vectoriel ou d'ordre morphologique.

Dans ce chapitre, nous redéfinissons les fonctions lexicales symétriques à l'aide de cette contextualisation forte. Nous tenons ensuite compte des observations que nous faisions dans la conclusion de la première partie de cette thèse pour améliorer la fonction d'antonymie. À la fonction lexicale de construction d'antonymes présentée dans le chapitre précédent, et que nous qualifierons maintenant de fonction naïve, nos ajoutons un module qui permet de tenir compte des oppositions déjà rencontrées. Ainsi, si la fonction découvre qu'‹*existence*› $|||_c$ ‹*inexistence*› elle utilisera ensuite ce résultat lorsqu'elle rencontrera à nouveau cette opposition. Une amélioration synchronique de la fonction lexicale et de la base vectorielle est dès lors obtenue.

La deuxième grande partie de ce chapitre concerne la représentation des informations que nous donnent les fonctions lexicales dans la base lexicale sémantique. Partant de la constatation qu'une LEXIE devrait correspondre à un sens particulier d'un terme selon une source, nous essayons d'améliorer la construction des LEXIES issues de fonctions lexicales en regroupant les synonymes et les antonymes en fonction de leur sens.

Notre tâche s'est révélée plus particulièrement ardue avec les antonymes à cause d'un manque de sources gratuitement disponibles jusqu'à une période très récente. Nous présentons la méthode que nous avons utilisée pour construire nous-même une telle source. Cette méthode, basée sur les oppositions morphologiques qu'il existe entre les termes, nous a permis de constituer un dictionnaire de près de 2000 couples d'antonymes. Nous comparons l'apprentissage réalisé avec cette source et celui réalisé avec le dictionnaire des antonymes du CRISCO qui est apparu en

octobre 2004.

Pour conclure, nous présentons les résultats de ces améliorations sur l'apprentissage et la représentation des fonctions lexicales symétriques.

4.1 Introduction d'informations lexicales dans les fonctions symétriques

Comme nous le constatons dans la conclusion de la partie précédente, il ne semble pas pertinent à la fois pour des raisons d'ordre cognitif et des raisons pratiques de se limiter aux seuls vecteurs conceptuels pour représenter le sens. Nous montrons, ici, comment nous avons cherché à introduire des informations de type lexical dans les fonctions lexicales. Cette introduction se fait de deux manières. Pour la synonymie grâce à la contextualisation forte et pour l'antonymie, non seulement grâce à la contextualisation forte mais aussi par l'utilisation d'oppositions lexicales rencontrées dans des textes ou des listes d'antonymes.

4.1.1 Introduction d'informations lexicales dans les fonctions de synonymie : utilisation de la contextualisation forte

Dans cette section, nous reprenons l'ensemble des fonctions lexicales de construction et d'évaluation de la synonymie et nous les adaptons aux contraintes de la contextualisation forte.

4.1.1.1 Fonctions lexicales de construction d'un vecteur synonyme

Fonction lexicale de construction de synonymie relative Nous définissons la fonction lexicale de construction d'un vecteur synonyme $Csyn_R$ qui donne un vecteur synonyme aux ITEMS LEXICAUX x et y dans un contexte vectoriel V_c et un contexte morphologique M_c (figure 4.1).

$$\begin{array}{rcl} \omega^2 \times m \times \vartheta \rightarrow \vartheta & : & x,y,M_c,V_c \rightarrow Z = Csyn_R(x,y,M_c,V_c) \\ Csyn_R(x,y,M_c,V_c) & = & \Gamma(x,M_c,V_c) \oplus \Gamma(y,M_c,V_c) \end{array}$$

FIG. 4.1 – Construction d'un vecteur synonyme : fonction $Csyn_R$

Généralisation de la fonction lexicale de construction de synonymie relative Nous redéfinissons ici la fonction lexicale généralisée de construction d'un vecteur synonyme $Csyn_R$ comme la fonction qui donne le vecteur synonyme à un ensemble d'ITEMS LEXICAUX $x_1, x_2, \ldots, x_n$ dans un contexte vectoriel V_c et un contexte morphologique M_c (figure 4.2).

$$\begin{array}{rcl} \omega^n \times m \times \vartheta \rightarrow \vartheta & : & x_1,x_2,\ldots,x_n \rightarrow Z = Csyn_R(x_1,x_2,\ldots,x_n,M_c,V_c) \\ Csyn_R(x_1,x_2,\ldots,x_n,M_c,V_c) & = & \bigoplus_{i=1}^{n} \Gamma(x_i,M_c,C_c) \end{array}$$

FIG. 4.2 – Fonction $Csyn_R$ généralisée à n termes

Fonction lexicale de construction de synonymie partielle Rappelons que cette fonction a été créée pour permettre d'analyser les dictionnaires de synonymes dans lesquels figurent rarement des indications sur le ou les contextes où cette équivalence peut être rencontrée. On peut considérer qu'un tel contexte serait constitué des idées communes aux deux items. Pour trouver le ou les sens partagés par les deux termes, il faut donc construire un vecteur correspondant au "dénominateur commun", aux idées communes aux deux termes synonymes. De même, deux synonymes sont de nature grammaticale identique, on peut donc considérer le contexte morphologique comme l'intersection des deux ensembles morphologiques $M_c = m(x) \cap m(y)$. Mathématiquement, le vecteur contexte peut être ainsi le vecteur médian aux vecteurs fortement contextualisés par la morphologie commune et le vecteur correspondant à la somme vectorielle des vecteurs des deux ITEMS LEXICAUX. Une bonne heuristique dans un cas d'analyse est donc de considérer que le contexte V_c vaut $\Gamma(x, M_c, V(x) \oplus V(y)) \oplus \Gamma(y, M_c, V(x) \oplus V(y))$.

Nous pouvons ainsi proposer la fonction lexicale de construction partielle d'un synonyme $Csyn_P$ qui calcule le vecteur synonyme aux ITEMS LEXICAUX x et y (figure 4.3).

$$\begin{aligned} \omega^2 \rightarrow \vartheta \quad &: \quad x, y \rightarrow Z = Csyn_P(x,y) \\ Csyn_P(x,y) \quad &= \quad Csyn_R(x, y, M_c, V_c) \\ \text{où } M_c \quad &= \quad m(x) \cap m(y) \\ \text{et } V_c \quad &= \quad \Gamma(x, M_c, V(x) \oplus V(y)) \oplus \Gamma(y, M_c, V(x) \oplus V(y)) \end{aligned}$$

FIG. 4.3 – Construction d'un vecteur synonyme : fonction $Csyn_P$

Généralisation de la fonction lexicale de construction de synonymie partielle Dans le cas de la généralisation de la fonction lexicale de construction de synonymie partielle, il s'agit de trouver le ou les sens partagés par l'ensemble des termes d'une liste de synonymes. Le contexte morphologique est donc $M_c = m(x_1) \cap m(x_2) \cap \ldots \cap m(x_n)$ et le contexte vectoriel $V_c = \bigoplus_{i=1}^{n} \Gamma(x_i, M_c, V(x_1) \oplus V(x_2) \oplus \ldots \oplus V(x_n))$ (figure 4.4).

$$\begin{aligned} \omega^n \rightarrow \vartheta \quad &: \quad x_1, x_2, \ldots, x_n \rightarrow Z = Csyn_P(x_1, x_2, \ldots, x_n) \\ Csyn_P(x_1, x_2, \ldots, x_n) \quad &= \quad Csyn_R(x_1, x_2, \ldots, x_n, M_c, V_c) \\ \text{où } M_c \quad &= \quad m(x_1) \cap m(x_2) \cap \ldots \cap m(x_n) \\ \text{et } V_c \quad &= \quad \bigoplus_{i=1}^{n} \Gamma(x_i, M_c, V(x_1) \oplus V(x_2) \oplus \ldots \oplus V(x_n)) \end{aligned}$$

FIG. 4.4 – Fonction $Csyn_P$ généralisée à n termes

Nous verrons en 4.2.1 l'utilisation de cette fonction dans l'apprentissage en ne considérant plus une LEXIE par liste de synonymes mais une granularité plus proche du sens.

4.1.1.2 Fonctions lexicales d'évaluation de la synonymie

Fonction de synonymie relative Syn_R La fonction de synonymie relative Syn_R est la fonction qui évalue la synonymie entre deux ITEMS LEXICAUX x et y par rapport à un contexte morphologique M_c et un contexte vectoriel V_c (figure 4.5).

$$\omega^2 \times m \times \vartheta \rightarrow [0, \tfrac{\pi}{2}] : \qquad x,y,M_c,V_c \rightarrow D = Syn_R(x,y,M_c,V_c)$$

$$Syn_R(x,y,M_c,V_c) = D_A(\Gamma(x,M_c,V_c),\ \Gamma(y,M_c,V_c))$$

FIG. 4.5 – Calcul de la fonction de synonymie relative Syn_R

Propriétés L'introduction d'informations lexicales ne modifie pas la propriété de distance de la fonction de synonymie relative. Elle respecte :

1. la *réflexivité* : $Syn_R(x,x,M_c,V_c) = 0$ La réflexivité est héritée de celle de la distance angulaire (équation 2.6).
2. la *symétrie* : $Syn_R(x,y,M_c,V_c) = Syn_R(y,x,M_c,V_c)$ La symétrie pour les deux premiers arguments provient de celle de la distance angulaire (équation 2.7).
3. la *pseudo-transitivité* : Nous avons par héritage de l'inégalité triangulaire de D_A (équation 2.8) : $Syn_R(x,y,M_c,V_c) + Syn_R(y,z,M_c,V_c) \geq Syn_R(x,y,M_c,V_c)$.

Fonction de synonymie partielle Syn_P

Principes et définitions La fonction de synonymie partielle Syn_P est la fonction qui évalue la synonymie entre deux ITEMS LEXICAUX x et y (figure 4.6)

$$\omega^2 \rightarrow [0, \tfrac{\pi}{2}] : \qquad x,y \rightarrow D = Syn_P(x,y)$$

$$Syn_P(x,y) = Syn_R(x,y,\Gamma(x,M_c,V_c) \oplus \Gamma(y,M_c,V_c))$$

$$\text{où } V_c = V(x) \oplus V(y) \text{ et } M_c = m(x) \cap m(y)$$

FIG. 4.6 – Calcul de la fonction de synonymie partielle Syn_P

Il s'agit de la fonction de synonymie relative pour laquelle le contexte morphologique est donné par l'intersection des morphologies des deux items et le contexte vectoriel par la somme vectorielle des vecteurs correspondant aux contextualisations fortes des deux items par ce contexte morphologique et par la somme vectorielle des deux items lexicaux.

Propriétés La fonction de Synonymie partielle hérite des propriétés de réflexivité et de symétrie de la fonction de synonymie relative.

1. *réflexivité* : $Syn_P(x,x) = Syn_R(x,x,m(x),\Gamma(x,m(x),V(x))) = 0$ La réflexivité est héritée de celle de la fonction de synonymie relative.
2. *symétrie* : $Syn_P(x,y) = Syn_P(y,x)$ La symétrie est héritée de celle de la synonymie relative. On a en effet $Syn_R(x,y,M_c,V_c) = Syn_R(y,x,M_c,V_c)\ \forall\ M_c,V_c$

En revanche, la relation de pseudo-transitivité n'est pas respectée. La fonction de synonymie partielle n'est donc pas une distance.

Résultats Les tableaux de la figure 3.7 présentent quelques résultats obtenus avec la fonction de synonymie partielle. Dans le premier tableau, la partie supérieure droite rappelle les résultats obtenus avec la distance thématique et la partie inférieure gauche présente les résultats de la

fonction de synonymie partielle. Le deuxième tableau indique le rapprochement en pourcentage de la fonction de synonymie partielle par rapport à la distance thématique.

Syn_P \ D_A	destinée	destin	vie	existence	mort	automobile	train	action	inaction	réaction
destinée	0	0,51	0,82	0,7	0,99	1,29	1,38	1,31	1,14	1,2
destin	**0,25**	0	0,83	0,75	0,99	1,3	1,38	1,25	1,07	1,16
vie	**0,57**	**0,58**	0	0,61	0,89	1,28	1,35	1,3	1,1	1,2
existence	**0,42**	**0,45**	**0,43**	0	0,98	1,37	1,43	1,37	1,25	1,3
mort	**0,74**	**0,74**	**0,68**	**0,76**	0	1,33	1,4	1,32	1,15	1,26
automobile	**1,01**	**1,02**	**1,06**	**1,14**	**1,1**	0	0,88	1,4	1,22	1,29
train	**1,15**	**1,14**	**1,12**	**1,23**	**1,17**	**0,62**	0	1,43	1,3	1,39
action	**1,06**	**1,01**	**1,05**	**1,15**	**1,06**	**1,16**	**1,2**	0	1,01	0,67
inaction	**0,91**	**0,86**	**0,88**	**1,04**	**0,88**	**0,97**	**1,08**	**0,75**	0	0,9
réaction	**0,96**	**0,9**	**0,99**	**0,98**	**1,0**	**1,04**	**1,15**	**0,47**	**0,68**	0

Rapprochement (%)	destinée	destin	vie	existence	mort	automobile	train	action	inaction	réaction
destinée	**0**									
destin	**51,0**	**0**								
vie	**30,5**	**30,1**	**0**							
existence	**40,0**	**40,0**	**29,5**	**0**						
mort	**25,3**	**25,3**	**23,6**	**22,4**	**0**					
automobile	**14,8**	**21,5**	**17,2**	**16,8**	**17,3**	**0**				
train	**16,7**	**17,4**	**17,0**	**14,0**	**16,4**	**29,5**	**0**			
action	**19,1**	**19,2**	**19,2**	**16,1**	**19,7**	**17,1**	**16,1**	**0**		
inaction	**20,2**	**19,6**	**20,0**	**16,8**	**23,5**	**20,5**	**16,9**	**25,8**	**0**	
réaction	**20,0**	**22,4**	**17,5**	**24,6**	**20,6**	**19,4**	**17,3**	**29,9**	**24,4**	**0**

FIG. 4.7 – Exemples de résultats de la fonction de synonymie partielle : $Syn_P(X,Y)$ comparé avec la distance thématique D_A(X, Y).

Les résultats que nous obtenons sont en tout point comparables à ceux de la fonction de synonymie relative. En effet, la comparaison de ces chiffres entre eux fait clairement apparaître les mêmes relations que nous avions constatées avec la synonymie relative.

Comparons maintenant les résultats obtenus en synonymie relative et ceux obtenus en synonymie partielle. On peut constater que les rapprochements sont tous supérieurs. Le rapprochement est d'autant plus important que les termes sont en situation de synonymie. Il atteint 51% pour ‹*destin*› et ‹*destinée*› soit une augmentation de près de 20% sur la fonction de synonymie relative avec comme contexte ‹*vie*›. En effet, dans le cas de la synonymie partielle, les idées du contexte utilisé sont celles communes aux deux items. La synonymie relative utilise, par contre, comme référent, le vecteur d'un item qui n'a, en pratique, aucune chance de posséder plus d'idées en commun avec les deux items que le contexte "artificiel" fabriqué par la fonction de synonymie partielle. Effectivement, pour posséder plus d'idées en commun avec les deux vecteurs, il faudrait que ce vecteur ait une norme plus importante ce qui n'est pas conforme à notre modèle.

4.1.2 Introduction d'informations lexicales dans les fonctions d'antonymie

Comme nous l'avons vu, la fonction de construction d'antonymes que nous avons présenté dans le chapitre précédent et que nous appellerons dorénavant fonction naïve présente plusieurs faiblesses :

– *Elle est fixe* : la seule manière de l'améliorer est de modifier les listes des vecteurs antonymes aux concepts (cf. 3.3.3.2). Pour que la fonction soit vraiment fiable, ces listes doivent être satisfaisantes en toutes circonstances or, il semble évident que la construction de ces listes est très subjective et donc soumise à caution. Cela suppose, de plus, que la langue soit figée, ce qui est loin d'être le cas. Il y a une évolution dans le sens des mots et dans la

perception de l'antonymie. La fonction d'antonymie ne devrait donc pas être figée pour tenir compte à la fois des éventuelles "erreurs" des listes et des variations de la langue.
- *les concepts opposés peuvent ne pas exister ni être construits à partir d'autres* : c'est le cas, par exemple, dans le thésaurus Larousse où le concept *COMBUSTIBILITÉ* existe mais le concept qui semble opposé *INCOMBUSTIBILITÉ* n'existe pas. On trouve d'ailleurs dans la partie index du thésaurus comme définition de ‹*incombustibilité*› « *COMBUSTIBILITÉ* ». C'est aussi le cas d'*AMOUR* qui existe alors que *HAINE* n'existe pas.

Nous avons donc cherché à développer une méthode qui permettrait à la fonction de se modifier grâce à des exemples de couples antonymes. L'idée est d'utiliser les couples d'antonymes attestés afin que notre fonction s'améliore en fonction de ces informations.

4.1.2.1 Auto-modification des VAC

Nous l'avons vu, la construction du vecteur antonyme est basée sur les listes de vecteurs antonymes aux concepts (VAC cf. 3.3.3.2). L'idée d'apprentissage la plus directe serait donc une modification automatique de ces listes. Cette idée semble toutefois être difficile à mettre en œuvre pour deux raisons :
- Les concepts sont à un niveau trop bas pour être modifiés correctement par des termes. L'apprentissage se fait à partir de couples d'items antonymes rencontrés. Comment savoir, de façon automatique, sur quels concepts se fait l'opposition ? Dans les listes, l'opposition se fait sur des mots (des vecteurs) et non sur les idées (les composantes des vecteurs). De plus, pour découvrir sur quelles composantes se font les oppositions, il faudrait avoir des vecteurs "parfaits" c'est-à-dire des vecteurs "bien indexés" or c'est justement l'objectif que nous cherchons à atteindre.
- Même si nous pouvions dégager des oppositions de concepts grâce aux antonymes, il semble difficile de savoir automatiquement sur quel type d'antonymie se fait cette distinction. Trouver ce type demande, en effet, des connaissances sur le monde qu'il semble difficile d'acquérir automatiquement à l'heure actuelle.

Nous nous sommes donc orientés vers une solution qui utilise les résultats de la fonction naïve d'antonymie tout en les ajustant par des listes de vecteurs antonymes créés et modifiés par la fonction.

4.1.2.2 Principe général : apprentissage de et par les fonctions lexicales d'antonymie

Contrairement aux fonctions lexicales de construction d'un vecteur antonyme que nous avons mises au point dans le chapitre précédent (section 3.3.3) et qui pouvaient être utilisées sur une liste d'antonymes ou comme nous le faisions alors dans le cadre d'une analyse sémantique pour gérer certaines tournures négatives, celles que nous définissons ici ne peuvent être utilisées que dans le premier cas et donc uniquement dans le cadre de l'apprentissage.

L'idée est d'ajouter à la fonction d'antonymie un module qui aura un poids équivalent à celui de la fonction naïve. Ce module gère des listes d'items et de vecteurs calculés par la fonction suivant les couples d'antonymes rencontrés. L'algorithme utilise trois listes d'items antonymes (LIA), une par type d'antonymie. Ces listes contiennent des triplets de la forme ⟨item, item antonyme, $V_{opposé}$⟩. Soit la fonction $antiV_\alpha$ qui renvoie en fonction du couple d'antonymes x et $\overline{x}$ ainsi que dans un contexte vectoriel V_c et un contexte morphologique M_c le vecteur conceptuel antonyme à $\overline{x}$ tel qu'il est dans la LIA_α ou tel qu'il devrait y être (figure 4.8).

En quelque sorte, on peut dire que *AntiV* est la mémoire, la base de connaissances de la fonction lexicale de construction d'antonymes, ce qu'elle a appris précédemment. Dans le cas où la ligne n'existe pas, elle renvoie le vecteur conceptuel correspondant à la contextualisation forte

$$\omega^2 \times m \times \vartheta \rightarrow \vartheta \quad : \quad x, \bar{x}, M_c, V_c \rightarrow Z = AntiV(x, \bar{x}, M_c, V_c)$$

$$AntiV(x, \bar{x}, M_c, V_c) = \begin{cases} V_{opposé} & \text{si la ligne} \langle x, \bar{x}, V_{opposé} \rangle \text{ existe} \\ \Gamma(x, M_c, V_c) & \text{sinon} \end{cases}$$

FIG. 4.8 – Fonction *AntiV*

de l'ITEM LEXICAL x dans le contexte morphologique M_c et le contexte vectoriel V_c, c'est-à-dire le vecteur de l'item que l'on sait antonyme de $\bar{x}$.

Lors de la révision d'un item, nous rajoutons une opération qui consiste à envoyer à la fonction d'antonymie le ou les couples d'antonymes dont fait partie l'item. Rappelons ici que nous considérons que chaque item possède au moins un antonyme, lui-même le cas échéant (cf. propriété des points fixes section 3.3.1.2). Grâce à cette information, non seulement la fonction va s'améliorer en complétant ses LIA mais son résultat, exploité par la méthode d'apprentissage, va servir à la construction d'une nouvelle LEXIE correspondant à ce mot qui, elle-même, sera utilisée par la suite pour affiner d'autres vecteurs ainsi qu'en retour la fonction lexicale. Le système global s'enrichit de l'apport de la fonction qui elle-même s'enrichit de l'apport de l'ensemble du système. Ce principe est connu sous le nom de *double boucle* [Lecerf, 1997].

4.1.2.3 Fonctions lexicales de construction d'antonymes : définitions et algorithme

La fonction lexicale de construction d'antonymie relative $Canti_{R_\alpha}$

Définition La fonction $Canti_{R_\alpha}$ renvoie le vecteur que l'ITEM LEXICAL x devrait posséder sachant qu'il est antonyme de $\bar{x}$ en antonymie $\alpha \in \{comp, scal, dual\}$ dans un contexte morphologique M_c et un contexte vectoriel V_c par rapport à un axe de symétrie V_r (figure 4.9).

$$\omega^2 \times m \times \vartheta^2 \rightarrow \vartheta \quad : \quad x, \bar{x}, M_c, V_c, V_r \rightarrow Z = Canti_{R_\alpha}(x, \bar{x}, M_c, V_c, V_r)$$

FIG. 4.9 – Fonction $Canti_{R_\alpha}$

La fonction $Canti_{R_\alpha}$ est calculée par l'algorithme 6.

Algorithme L'algorithme 6 montre la manière dont la fonction se modifie lorsqu'on lui fournit un couple d'items antonymes $(x, \bar{x})$[68]. À chaque cycle d'apprentissage, après l'analyse des définitions, la méthode d'apprentissage fait appel à la fonction d'antonymie. La première fois qu'elle rencontre un couple $(x, \bar{x})$, la fonction $AntiV_\alpha$ renvoie le vecteur de x fortement contextualisé. À ce vecteur va s'ajouter celui de l'antonyme de $\bar{x}$ calculé par la fonction naïve pour être inséré dans la LIA correspondante.

Aux cycles suivants, l'opération se répète mais cette fois, ce n'est plus $\Gamma(x, M_c, V_c)$ mais ce vecteur que renverra V_{mem}. Le vecteur antonyme calculé est alors utilisé dans l'apprentissage des nouvelles définitions et il participe ainsi à la création de LEXIES. Cette méthode a l'avantage de permettre une amélioration synchronique des vecteurs et de la fonction d'antonymie. Elle utilise de plus les propriétés de points fixes de certains mots. Peu à peu, les vecteurs des mots dérivent vers une position cohérente à la fois avec la fonction d'antonymie mais aussi avec les

[68] on peut avoir $x = \bar{x}$.

Algorithme 6: Calcul de $Canti_R$ et auto-modification de la fonction

Entrée : $(x,\bar{x})$ un couple d'items antonymes, V_c le contexte vectoriel, M_c le contexte morphologique, V_r le vecteur référent et α le type d'antonymie.

Sortie : $V_{anto} = Canti_{R_\alpha}(x,\bar{x},M_c,V_c,V_r)$

% on calcule le vecteur contextualisé des deux vecteurs
$V_x \leftarrow \Gamma(x,M_c,V_c)$, $V_{\bar{x}} \leftarrow \Gamma(\bar{x},M_c,V_c)$

% le vecteur antonyme selon la fonction naïve.

$V_{naïf} = Canti_P(V_{\bar{x}},V_r)$

% le vecteur antonyme selon la base de connaissance (LIA_α)
$V_{mem} = AntiV_\alpha(x,\bar{x},M_c,V_c)$

% on tient compte autant de la fonction naïve que de la mémoire issue de la LIA
$V_{anto} = V_{naïf} \oplus V_{mem}$

% avant de renvoyer ce vecteur, on modifie la LIA_α
remplacer $\langle x,\bar{x},V_{mem}\rangle$ par $\langle x,\bar{x},V_{anto}\rangle$ dans LIA_α

retourner V_{anto}

autres méthodes d'apprentissage. Les vecteurs des termes antonymes du lexique se retrouvent en opposition, les items sans contraire avérés dérivent vers des points fixes.

Pour une simple question d'espace mémoire et surtout à cause de l'intérêt très limité des informations concernant les points fixes, en pratique, les LIA ne les conservent pas.

La fonction lexicale de construction d'antonymie partielle $Canti_{P_\alpha}$ Que ce soient des listes d'antonymes établies par des lexicographes ou des listes que nous avons établies nous-même, nous n'en avons pas trouvées qui précisent par rapport à quel axe[69] ni même dans quel contexte les termes sont opposés. En pratique, il faut donc rechercher des heuristiques qui nous permettrons lors de l'apprentissage d'analyser les listes d'antonymes. Ainsi,

- *le contexte vectoriel peut être considéré comme la somme des vecteurs contextualisés des deux* ITEMS LEXICAUX. Dans ce cas, le référent/contexte vectoriel choisi en pratique est un vecteur constitué des idées communes à l'item et à son antonyme. En effet, comme nous le remarquions déjà en 3.3.3.2, il est souvent difficile, parfois sans doute impossible de trouver un contexte lexicalisé permettant d'opposer deux items lexicaux mais, si celui-ci existait, il posséderait au moins les mêmes idées. Par exemple, un axe de symétrie par lesquels ‹*chaleur*› et ‹*froid*› s'opposent pourrait être constitué de la somme vectorielle des vecteurs contextualisés de ces deux termes qui devrait posséder les mêmes idées que ‹*température*› ;
- *le contexte et référent axial peuvent être considérés comme identiques* puisque la somme de deux vecteurs est bien l'axe de symétrie de ces deux vecteurs ;
- *le contexte morphologique peut être considéré comme l'intersection des morphologies des deux items* puisque deux antonymes sont de même nature grammaticale.

Ainsi, nous définissons la fonction $Canti_{P_\alpha}$ qui renvoie le vecteur que l'ITEM LEXICAL x devrait posséder sachant qu'il est antonyme de $\bar{x}$ en antonymie $\alpha \in \{comp, scal, dual\}$ (figure 4.10).

[69] Pour les listes établies par des lexicographes, le contraire eut été surprenant.

$$
\begin{array}{rcl}
\omega^2 \to \vartheta & : & x,\overline{x} \to Z = Canti_{P_\alpha}(x,\overline{x}) \\
Canti_{P_\alpha}(x,\overline{x}) & = & Canti_{R_\alpha}(x,\overline{x},M_c,V_c,V_c) \\
\text{avec } M_c & = & m(x) \cap m(\overline{x}) \\
\text{et } V_c & = & \Gamma(x,M_c,V(x) \oplus V(y)) \oplus \Gamma(y,M_c,V(x) \oplus V(y))
\end{array}
$$

FIG. 4.10 – Fonction $Canti_{P_\alpha}$

4.1.3 Généralisation des fonctions lexicales de construction d'antonymes

Sur le modèle des fonctions lexicales généralisées de construction de synonymes, nous introduisons dans cette partie les fonctions lexicales généralisées de construction d'antonymes. Ces fonctions vont nous servir à fabriquer des LEXIES à partir de listes d'antonymes comme nous le verrons dans la section suivante.

La fonction $Canti_{R_\alpha}$ généralisée renvoie le vecteur que l'ITEM LEXICAL x devrait posséder sachant qu'il est antonyme de $\overline{x_1},\ldots,\overline{x_n}$ dans un contexte morphologique M_c et un contexte vectoriel V_c par rapport à un axe de symétrie V_r (figure 4.11).

$$
\begin{array}{rcl}
\omega^{n+1} \times m \times \vartheta^2 \to \vartheta & : & x,\overline{x_1},\ldots,\overline{x_n},M_c,V_c,V_r \to Z = Canti_{R_\alpha}(x,\overline{x_1},\ldots,\overline{x_n},M_c,V_c,V_r) \\
Canti_{R_\alpha}(x,\overline{x_1},\ldots,\overline{x_n},M_c,V_c,V_r) & = & \bigoplus_{i=1}^{n} Canti_{R_\alpha}(x,\overline{x_i},M_c,V_c,V_r)
\end{array}
$$

FIG. 4.11 – Fonction $Canti_{R_\alpha}$ généralisée à n termes

Il s'agit de la somme vectorielle normée de la fonction $Canti_{R_\alpha}$ définie en 4.1.2.3.

La fonction $Canti_{P_\alpha}$ généralisée à n termes renvoie le vecteur que l'ITEM LEXICAL x devrait posséder sachant qu'il est antonyme de $\overline{x_i},\ldots,\overline{x_n}$ en antonymie $\alpha \in \{comp, scal, dual\}$ (figure 4.10).

$$
\begin{array}{rcl}
\omega^{n+1} \to \vartheta & : & x,\overline{x_1},\ldots,\overline{x_n} \to Z = Canti_{P_\alpha}(x,\overline{x_1},\ldots,\overline{x_n}) \\
Canti_{P_\alpha}(x,\overline{x_1},\ldots,\overline{x_n}) & = & Canti_{R_\alpha}(x,\overline{x_1},\ldots,\overline{x_n},M_c,V_c,V_c) \\
\text{où } M_c & = & m(x) \cap m(\overline{x_1}) \cap \ldots \cap m(\overline{x_n}) \\
V_c & = & \Gamma(x,M_c,v_c) \oplus \bigoplus_{i=1}^{n} \Gamma(\overline{x_i},M_c,v_c) \\
\text{et } v_c & = & V(x) \oplus \bigoplus_{i=1}^{n} V(\overline{x_i})
\end{array}
$$

FIG. 4.12 – Fonction $Canti_{P_\alpha}$ généralisée à n termes

Tout comme nous avons fait avec les fonctions lexicales de construction de synonymes, cette méthode considère $\overline{x_1},\ldots,\overline{x_n}$ comme synonymes entre eux et devrait être ainsi utilisée lors de l'apprentissage.

Le référent/contexte choisi ici est donc un vecteur constitué des idées communes à l'item dont on veut un vecteur et à l'ensemble de ses antonymes.

4.2 Amélioration de l'utilisation des fonctions symétriques dans la base de vecteurs

Nous avons présenté dans le chapitre précédent (section 3.4.1) une première introduction des fonctions lexicales de construction de synonymes dans l'apprentissage. Notre principale source pour ce type de données est le *dictionnaire des synonymes du CRISCO*. L'une des ses caractéristiques est de ne donner pour un terme qu'une liste d'items synonymes et de ne pas les séparer suivant leur sens. L'utilisation que nous faisons alors de la fonction de synonymie partielle généralisée $Csyn_P$ n'est donc pas optimale. En effet, comme nous le notions lors de son introduction en 3.2.2.3, « *Cette généralisation de la fonction lexicale de synonymie partielle devrait être utilisée avec des termes qui sont synonymes dans le même contexte, c'est-à-dire dans le cas où le dictionnaire sépare les sens mais ne donne pas d'indication sur ce sens ou après classification des termes en fonction de leur sens.* ».

De plus, l'architecture de notre base lexicale sémantique s'appuie sur les objets lexicaux LEXIES qui regroupent chacune les informations vectorielles et morphologiques issue d'une définition d'un dictionnaire. Ainsi, on peut considérer qu'une lexie correspond à un des sens d'un terme dans un dictionnaire. La méthode de contextualisation forte est d'ailleurs largement basée sur cette hypothèse (cf. 2.3.6). L'ajout d'une nouvelle source ne va donc pas sans poser de problèmes. La manière de l'exploiter, en particulier le découpage des sens, entraînera sur la base des conséquences au moins aussi importantes que sa qualité (ie. sa facilité d'analyse).

L'ajout d'une source doit donc être étudiée à deux niveaux :

- au niveau LEXIE : les informations de chaque LEXIE doivent décrire un sens particulier du terme.
- au niveau ITEM LEXICAL : il ne faut pas avoir une sur-représentation ou une sous-représentation disproportionnées de la source au regard des autres. Les LEXIES de chaque source pour un ITEM LEXICAL doivent donc être en nombre comparable.

Cette approche suppose ainsi que l'introduction des informations de type antonymie se fasse de la même manière. Dans cette section, après avoir étudié une meilleure façon d'incorporer les données issues des dictionnaires de synonymes, nous nous posons le même problème avec des données de nature antonymique. Nous verrons que la solution adoptée pour la synonymie est aussi adéquate pour l'antonymie bien que ce problème ait été accentué par l'existence des trois types d'antonymie ainsi que, dans un premier temps de nos recherches, le manque de ressources gratuitement disponibles.

4.2.1 Utilisation de la synonymie dans l'apprentissage : dictionnaires de synonymes

Comme nous venons de le faire remarquer, le dictionnaire des synonymes du CISCO, du fait de sa structuration qui ne distingue pas les sens, n'est pas aussi trivialement utilisable que ce que notre première approche semblait le laisser penser. Il nous amène ainsi à réfléchir à une indexation des termes plus fine que celle réalisée jusqu'à présent avec cette source. Nous allons donc étudier ici l'ajout de dictionnaires de synonymes et les problèmes que posent tel ou tel choix. Il s'agit de regrouper les synonymes dans le but de créer des LEXIES, tout le problème étant de choisir la granularité de ces regroupements.

4.2.1.1 Granularité de l'affectation de vecteurs

L'ajout d'informations lexicales nouvelles, en particulier des relations sémantiques, doit permettre l'amélioration globale de la base de données vectorielles. En ce qui concerne l'ajout de dictionnaires de synonymes, la difficulté vient essentiellement de l'affectation de vecteurs, de la

création des LEXIES. Trois solutions sont possibles : une LEXIE pour l'ensemble des synonymes, une pour chacun d'entre eux, une pour un regroupement par sens.

Une lexie pour la liste des synonymes Il s'agit de créer une seule LEXIE pour l'ensemble de la liste de synonymes. Les deux avantages principaux de cette méthode sont qu'il est extrêmement simple de la mettre en place et qu'elle utilise un minimum d'espace mémoire. C'est d'ailleurs pour ces raisons qu'elle avait été dans un premier temps adoptée (cf. 3.4.1). Elle pose en revanche deux inconvénients majeurs pour :

- *la représentation au niveau* LEXIE : Cette méthode regroupe l'ensemble des idées contenues dans les synonymes ce qui n'est pas en conformité avec le modèle des vecteurs conceptuels et ses LEXIES qui devraient correspondre à un sens d'un terme suivant un dictionnaire. De plus, l'emploi de la fonction lexicale de construction d'un vecteur synonyme $Csyn_P$ utilisée avec une telle liste donne un résultat d'autant plus aléatoire que le terme a de sens puisque, dans ce cas, le vecteur contexte utilisé dans l'opération mélange trop d'idées différentes.
- *la représentation au niveau* ITEM LEXICAL : Au niveau ITEM LEXICAL, cette méthode atténue trop l'effet de l'ajout de synonymes par rapport aux autres définitions. Prenons l'exemple d'un terme qui possède trois sens distincts et qui est défini par une source dictionnairique ainsi que par une source synonymes. Pour un certain sens, nous avons alors le vecteur issu des synonymes qui est en moyenne occupé pour un tiers par le sens, et un autre vecteur, issu lui du dictionnaire, occupé lui uniquement par ce sens. C'est-à-dire que ce dernier a alors un poids environ trois fois plus important dans le vecteur de l'ITEM LEXICAL.

Une lexie pour chacun des synonymes Il s'agit ici de créer une LEXIE pour chacun des synonymes. Cette solution offre, comme la précédente, l'avantage d'être extrêmement facile à mettre en œuvre. En revanche, elle entraîne une augmentation importante de la taille de la base. Elle semble tout de même plus intéressante que la précédente en ce qui concerne les représentations tout au moins au niveau LEXIE. En effet, pour :

- *la représentation au niveau* LEXIE : Chacune des LEXIES est fabriquée à partir d'un synonyme, on peut donc considérer qu'une LEXIE correspond bien à un sens du terme. Dès lors, l'utilisation de la fonction $Csyn_P$ généralisée est parfaitement conforme à sa spécification.
- *la représentation au niveau* ITEM LEXICAL : Dans ce cas, nous nous trouvons dans l'excès inverse de l'affectation d'une LEXIE pour la liste. Plus un terme a de synonymes moins ses définitions issues de dictionnaires classiques auront un poids dans le vecteur final. On pourrait objecter que ce fait est un indice indirect de la fréquence d'utilisation du sens mais le vecteur de l'ITEM LEXICAL est censé fusionner les sens en les considérant de façon identique pour ne pas en favoriser un sur les autres en particulier lors d'un apprentissage (cf. analyse sémantique avec les vecteurs sémantiques 2.2.3 ou fonctions lexicales de synonymie et d'antonymie chapitre 3). Rappelons qu'à ce titre, nous ne tenons pas compte de la fréquence dans la contextualisation forte pour le calcul du vecteur de l'ITEM LEXICAL.

Une lexie pour un regroupement par sens Cette solution consiste à regrouper les synonymes par sens et à affecter un vecteur à chaque ensemble. Elle offre une situation intermédiaire aux deux précédentes pour ce qui est relatif à l'espace occupé. En ce qui concerne les représentations, nous avons alors pour :

- *la représentation au niveau* LEXIE : Chacune des LEXIES est fabriquée à partir d'un ensemble de synonymes qui partagent les mêmes idées ce qui est parfaitement en adéquation

à la fois avec la description d'une LEXIE et avec l'utilisation optimale de la fonction $Csyn_P$ généralisée.
- *la représentation au niveau* ITEM LEXICAL : Le niveau de granularité est sensiblement identique à celui que l'on peut trouver dans des dictionnaires ce qui garantit une considération des sens qui en sont issus similaire à celle faite aux synonymes. Ainsi, les synonymes deviennent une source à part entière ni privilégiée ni sous-estimée au regard des autres.

Cette dernière solution semble donc être la plus séduisante en ce qui concerne la représentation mais c'est aussi celle qui est la plus difficile à mettre en oeuvre. En effet, elle nécessite de mettre au point des méthodes qui permettent de séparer les items suivant leurs différents sens.

4.2.1.2 Regroupement de synonymes en fonction de leur sens : Une approche purement lexicale

Dans cette section, nous allons donc chercher à regrouper les synonymes en fonction de leur sens. Le principal outil nous fournissant des synonymes est *Le Dictionnaire Électronique des synonymes du CRISCO*[70] (Centre de Recherches Interlangues sur la Signification en Contexte) [Manguin *et al.*, 2004].

Le dictionnaire des synonymes du CRISCO Le dictionnaire des synonymes du CRISCO est actuellement l'un des deux seuls dictionnaires des synonymes pour le Français en accès libre sur Internet. Le deuxième, que l'on peut trouver sur le site de l'Institut des Sciences Cognitives à Lyon[71], est constitué d'une ancienne version de celui du CRISCO. Couplé à un équivalent anglais il forme un dictionnaire bilingue. Dans l'objectif de cette section, pour l'exploitation des synonymes comme données lexicales, nous préférons utiliser le dictionnaire du CRISCO dont les données sont régulièrement mises à jour.

Objectifs de la constitution du dictionnaire Tout d'abord, remarquons que les auteurs utilisent une définition de la synonymie proche de la nôtre (cf. 3.2) : « *Deux unités lexicales sont en relation de synonymie si toute occurrence de l'une peut-être remplacée par une occurrence de l'autre dans un certain nombre d'environnements sans modifier notablement le sens de l'énoncé dans lequel elle se trouve* » [Ploux & Victorri, 1998]. L'objectif premier de la constitution du dictionnaire des synonymes du CRISCO est la création d'un graphe de synonymie. Dans le cadre d'une approche continuiste de la polysémie, chaque item lexical est associé à un espace sémantique où se déploient les divers emplois des termes matérialisés donc ici par les synonymes possibles. Chaque sommet du graphe est constitué d'un item lexical et les arêtes correspondent à une relation de synonymie. L'exploration des différents sens des items se fait par la découverte de sous-graphes ayant des propriétés particulières : des composantes connexes et des cliques.

Constitution du dictionnaire Ce dictionnaire électronique rassemble, selon les responsables actuels du projet [Manguin *et al.*, 2004], « *sept dictionnaires classiques : deux dictionnaires analogiques (le Grand Larousse et le Grand Robert), deux dictionnaires des synonymes du 19e siècle (Lafaye et Guizot), et trois dictionnaires des synonymes du milieu et de la fin du 20e siècle (Bailly, Bénac et Du Chazaud)* » [Larousse, 1971] [Robert, 1995] [Lafaye, 1841] [Guisot, 1864] [Bailly, 1947] [Bénac, 1956] [Chazaud, 1979]. La mise au point du dictionnaire a nécessité trois opérations principales :
- *l'unification des formats des dictionnaires* : cette opération a fait disparaître les éventuels commentaires ainsi que la structure des articles d'origine.

[70] http://elsap1.unicaen.fr/cgi-bin/cherches.cgi
[71] http://dico.isc.cnrs.fr/index.html

– *la fusion des dictionnaires* : Cette opération a fusionné pour chaque entrée les différentes listes de termes synonymes. L'origine de la relation a été conservée mais son utilisation n'est possible qu'à l'aide d'un programme interne au CRISCO et en aucun cas possible via le formulaire Web.
– *la symétrisation des relations* : Dans certain cas, un dictionnaire peut présenter une relation uniquement dans un sens. En général, il s'agit alors plus d'une relation d'hyperonymie que d'une relation de synonymie. Pour les concepteurs de ce dictionnaire, deux réponses étaient alors possibles : soit supprimer les relations sans symétrique soit les symétriser. Dans le premier cas, de nombreux liens seraient alors éliminés tandis que dans le deuxième un maximum serait conservés. Afin d'avoir plus de relations et ainsi plus de matériaux sur lesquels travailler, les relations ont été symétrisées [Manguin, 2004]. L'auteur présente tout de même une justification théorique en citant les travaux de [Kahlmann, 1975] qui place dans son modèle de dictionnaires de synonymes des relations orientées puis étudie le graphe inversé. Il décèle des anomalies dues à l'orientation du graphe et « *constate que la symétrisation améliore sensiblement la qualité du dictionnaire sans pour autant engendrer de relations aberrantes* ».

Découpage de sens Nous l'avons dit, l'exploration des sens des items lexicaux du graphe ainsi construit se fait par la recherche des composantes connexes et des cliques du graphe. Nous illustrons nos propos par un exemple extrait de [Manguin *et al.*, 2004]. Nous considérons l'item ‹*baie*› dont la figure 4.13 présente un extrait simplifié du graphe de synonymie.

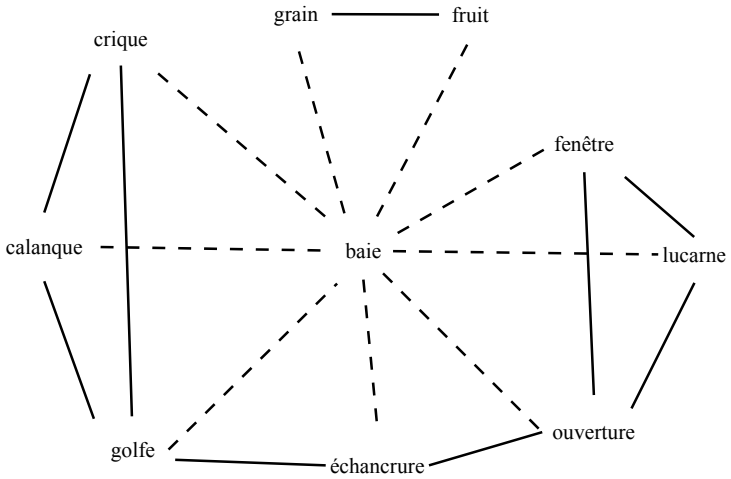


FIG. 4.13 – Exemple de graphe de synonymie simplifié centré sur l'item lexical ‹*baie*›

– *Composantes connexes :* On appelle *composante connexe* tout groupe de sommets d'un graphe pour lesquels il existe un chemin de longueur quelconque entre toute paire de sommets [Berge, 1967]. Pour la recherche des différentes composantes connexes auxquelles est associé un item, seules les relations n'impliquant pas ce mot sont conservées. On peut voir apparaître dans l'exemple deux composantes connexes, chacune correspondant à un des sens homonymes de l'item considéré. La première se rapporte à la **baie sauvage** (‹*grain*›, ‹*fruit*›) tandis que la seconde se rapporte plutôt à une **ouverture** (‹*crique*›, ‹*calanque*›, ‹*golfe*›, ‹*échancrure*›, ‹*ouverture*›, ‹*fenêtre*›, ‹*lucarne*›). Pour cette dernière, il est clair qu'il n'y a aucune relation entre les sens liés à *ouverture*/**mur** et ceux liés à *ouverture*/**mer**, seul ‹*échancrure*› joue un rôle de pivot.

– *Cliques :* Il s'agit des sous-graphes dont chaque sommet est relié à tous les autres. Ce sont donc des groupes de forte cohésion qui correspondent, selon les concepteurs de la méthode, à une meilleure dissociation des sens. Dans l'exemple de la figure 4.13, ‹*baie*› a cinq cliques : (‹*grain*›, ‹*fruit*›), (‹*golfe*›, ‹*échancrure*›), (‹*échancrure*›, ‹*ouverture*›), (‹*crique*›, ‹*calanque*›, ‹*golfe*›) et (‹*ouverture*›, ‹*fenêtre*›, ‹*lucarne*›).

En Avril 2005, le dictionnaire contient approximativement 49 000 entrées et 396 000 relations synonymiques.

Affectation de vecteurs : une approche pragmatique Notre but est de chercher la meilleure granularité possible pour affecter un vecteur conceptuel. Suivant ce que nous venons de dire, les cliques représenteraient la meilleure dissociation possible des sens. En effet, tous les mots d'une clique devraient pouvoir être échangés dans n'importe quel contexte selon la définition de la synonymie. Cela n'est pas aussi simple à cause de la symétrisation réalisée. En effet, les dictionnaires de synonymes sont avant tout utilisés pour aider à la rédaction des textes et non dans un cadre purement lexicographique. Ainsi, bien souvent, on trouve une liste de termes pouvant remplacer le mot cible plutôt qu'une liste de termes pouvant se substituer l'un l'autre. Nous avons alors non seulement des synonymes mais aussi des hyperonymes. C'est pour cette raison que l'on peut trouver dans certains dictionnaires des relations de synonymie aussi surprenantes que ‹*chat*›≅‹*félin*› ou ‹*chaussure*›≅‹*snow-boots*›. Si on peut comprendre que dans le cadre des travaux que mènent les auteurs une méthode de symétrisation alliée à une étude des cliques puisse faire émerger un certain nombre d'enseignements sur les différents sens d'un terme il nous semble évident que la symétrisation entraîne une grande quantité de bruit et augmente ainsi considérablement le nombre de cliques.

item lexical	*synonymes*	*composantes connexes*	*cliques*	*moyenne sens Robert, Larousse*	*rapport cliques/sens*	*rapport comp.conn./sens*
‹*chaussure*›	44	6	35	1,5	23,3	4
‹*chat*›	22	14	19	4,5	4,22	3,1
‹*avoir*›	72	5	82	10	8,2	0,5
‹*être*›	53	5	52	19	2,7	0,2
‹*baie*›	24	6	18	4,5	4	1,3
‹*fin*›	149	2	197	11,5	17,3	0,2
‹*maison*›	95	9	127	13	9,8	0,7
‹*propre*›	102	3	112	15,5	7,2	0,2
‹*monter*›	104	3	140	24,5	5,7	0,1
‹*entendre*›	41	5	54	12	4,5	0,4

FIG. 4.14 – Nombre de cliques et de composantes connexes dans le graphe de synonymie du CRISCO par rapport au nombre de sens de [Larousse, 2004] et [Robert, 2000].

Une lexie pour une clique En ce qui concerne la représentation au niveau LEXIE, une clique semble effectivement représenter un certain sens du terme ce qui est notre exigence première.

En revanche, la représentation au niveau ITEM LEXICAL ne semble pas remplir les conditions souhaitées. En effet, si on observe le tableau de la figure 4.14, on peut constater que le nombre de cliques pour un item est nettement plus élevé que le nombre de sens identifiés par les dictionnaires classiques. Le rapport le moins important est déjà de 2,7 tandis que le rapport le plus important atteint le chiffre de 23,3. Les différences sont ici si importantes que l'affectation d'un vecteur conceptuel par clique entraînerait une sur-représentation des synonymes par rapport aux autres dictionnaires.

On pourrait argumenter que cette source est peut-être plus fiable que les sources dictionnairiques classiques. En effet, il est possible qu'elle identifie des sens que les lexicographes qui ont créé les dictionnaires n'ont pas su reconnaître. Deux objections s'opposent à cette théorie. La première concerne tout simplement le mode de construction du dictionnaire des synonymes, en particulier la symétrisation qui, comme nous l'avons vu, entraîne la création d'un grand nombre de cliques. La seconde, concerne le travail des lexicographes rédacteurs des dictionnaires [Larousse, 2004] et [Robert, 2000]. Au cours de leurs travaux, il est possible qu'ils n'aient pas identifié quelques-uns des sens du terme mais de là à penser qu'ils sont passés à côté de 95,8% des sens de chaussure, il y a une marge difficile à franchir.

Pour en finir avec cette solution, on peut remarquer que, pour des termes comme ‹*avoir*›, ‹*fin*›, ‹*maison*› ou ‹*monter*›, le nombre de cliques est même plus important que le nombre de synonymes ce qui rend, pour ces termes, cette méthode encore moins économique en espace que celle qui donnerait une LEXIE par synonyme.

Une lexie pour une composante connexe Comme nous l'avons vu, une composante connexe correspond à un sens homonymique. En effet, il arrive qu'au cours de l'évolution de la langue, deux items lexicaux différents prennent la même forme (cf. 5.1.2.1). C'est ainsi le cas de ‹*baie*›, dont le sens d'**ouverture** proviendrait de l'ancien français ‹*baer*› signifiant « *être ouvert* » et le sens de **fruit** du latin ‹*baca*›. Même si les formes ont évolué, les relations de synonymie sont conservées ce qui explique que les composantes connexes correspondent à des sens homonymiques.

Ces sens peuvent ensuite dériver et devenir ainsi polysémiques. En observant une forme particulière, on peut donc voir une sorte de hiérarchie des sens, d'arbre des sens. À un premier niveau le découpage homonymique et pour les niveaux inférieurs les découpages polysémiques.

Pour ce qui est de la représentation au niveau LEXIE, nous avons donc avec cette solution un découpage des sens, certes moins précis qu'avec un dictionnaire classique, mais tout ce qu'il y a d'acceptable.

En ce qui concerne la représentation au niveau ITEM LEXICAL, on peut constater sur le tableau de la figure 4.14 que le nombre de composantes connexes est nettement moins important que le nombre de synonymes. En effet, une des propriétés des graphes est de posséder un nombre de composantes connexes inférieur ou égal au nombre de sommets (égal s'il n'y a aucune arête). Dans le cas d'un sous-graphe relativement dense comme celui qui nous intéresse ici, le nombre de composantes connexes est ainsi fortement inférieur au nombre de sommets. En ce qui concerne les rapports sens/composantes connexes, on passe de 0,1 pour ‹*monter*› à 4 pour ‹*chaussure*› ce qui paraît raisonnable pour éviter toute sur-représentation ou sous-représentation disproportionnée des synonymes par rapport aux autres sources. En pratique, les composantes connexes correspondent, comme nous l'avons vu, à des sens homonymes plutôt qu'à de la réelle polysémie or le phénomène d'homonymie est moins probable que le phénomène de polysémie. Ainsi, on peut constater en règle générale une légère sous-représentation des synonymes au regard des sources dictionnairiques dans notre base.

Les composantes connexes semblent être un bon compromis puisqu'elles allient les deux critères importants pour fabriquer une LEXIE : une bonne délimitation des sens et une représentativité raisonnable de la source au regard des autres. C'est pour ces deux raisons, purement pragmatiques, que nous avons choisi d'affecter les vecteurs aux composantes connexes.

4.2.2 Utilisation de l'antonymie dans l'apprentissage

Les informations de type antonymique sont importantes pour les fonctions d'antonymie car elles sont utilisées à deux titres. Non seulement elles peuvent être exploitées pour créer de nou-

velles LEXIES mais surtout elles enrichissent la fonction elle-même grâce à une base de connaissance contenant les informations sur des oppositions déjà rencontrées (cf. 4.1.2.2).

Comme pour toutes les autres sources, l'introduction de données de nature antonymique ne peut se faire sans étude préalable des sources disponibles. Ces études doivent être menées à l'aune des caractéristiques spécifiques de la source, dans le cas particulier de l'antonymie l'existence dans notre modèle de trois types.

Une de nos grandes difficultés a été que, pendant plus de trois ans, nous n'avions pas accès à de telles sources spécialisées. Nous avons donc dû fabriquer nous-même des listes d'antonymes.

Dans cette section, après avoir exposé le problème du manque de source sur l'antonymie, nous présentons la méthode semi-automatique que nous avons adoptée pour en construire une. Cette méthode est basée sur les oppositions morphologiques dans les couples d'items lexicaux antonymes. Ainsi, comme le couple ‹*monothéiste*› ||| ‹*polythéiste*› s'opposent par l'opposition des préfixes *mono-* (qui signifie « *un seul* ») et *poly-* (qui signifie « *plusieurs* ») on peut penser que deux termes qui s'opposent aussi sur le même préfixe doivent être antonymes.

Puisqu'elles sont de notre création, ces sources offrent l'avantage de préciser le type d'antonymie qui caractérise les couples. Cette étude en revanche n'a pas découpé les sens et le même problème que pour la synonymie est posé. Nous allons voir que la réponse apportée est la même. En revanche cette méthode bien que donnant de nombreux couples antonymes n'est pas complète et ne permet donc pas d'utiliser les propriétés de points fixes. Ce problème sera résolu lors de l'apparition en octobre 2004 d'une nouvelle source qui nous offre enfin des antonymes couvrant suffisamment le lexique pour être considérée comme fiable.

4.2.2.1 Le problème des sources

Une première version de ce qui allait devenir plus tard la fonction lexicale de construction d'antonymes a été mise au point au cours de mon DEA entre mars et juin 2001. Si son utilisation a été immédiatement possible dans l'analyse sémantique des textes, elle a été en revanche nettement moins aisée dans le cadre d'une analyse de dictionnaire d'antonymes à l'image de ce que nous réalisions déjà avec les synonymes. Les seules données disponibles étaient les rares antonymes spécifiés dans les définitions de certains dictionnaires, c'est-à-dire quelques centaines de couples qui ne comptent même pas forcément parmi eux ceux qui paraissent les plus représentatifs comme ‹*existence*› ||| ‹*inexistence*› ou ‹*vie*› ||| ‹*mort*›. De tels manques empêchent toute utilisation de la propriété des points fixes dans l'apprentissage et donc limitent fortement l'usage de l'antonymie pour l'apprentissage spécifiquement sur de telles sources. C'est d'ailleurs à cause de ce déficit de couverture lexicale que nous n'utilisons pas une telle méthode dans le chapitre précédente.

Il faut attendre Octobre 2004 pour que de telles données, en version électronique et gratuite, apparaissent pour le français. Depuis cette date, le dictionnaire électronique du CRISCO fournit, en sus des synonymes, des listes d'antonymes. Dans l'intervalle, il nous a fallu essayer de trouver des méthodes pour fabriquer le plus automatiquement possible de telles sources.

4.2.2.2 Extraction semi-supervisée de couples d'antonymes grâce à leur morphologie

Notre méthode est basée sur les oppositions de nature morphologique qui peuvent exister entre les items lexicaux. Par exemple, les préfixes *ante-* et *post-* s'opposent sur une idée de durée tandis que *poly-* et *mono-* s'opposent, eux, sur une idée de nombre. Cette méthode, à partir d'un premier ensemble de couples connus comme antonymes, extrait les préfixes susceptibles de l'opposer puis cherche, dans un corpus constitué des entrées de notre base lexicale, d'autres termes susceptibles d'être antonymes. Un expert valide les couples ainsi extraits et par là-même

les couples de préfixes qui leur correspondent. Une recherche automatique de nouveaux préfixes opposés est effectuée dans le corpus à partir de ces nouveaux couples. La méthode est itérée jusqu'au moment où il n'y a plus de préfixes candidats.

Nous étudions les résultats obtenus par cette méthode en particulier, nous présentons les oppositions de préfixes ainsi découvertes et leur validité sur le corpus puis nous discutons de la répartition des types d'antonymie en fonction des couples opposés de préfixes.

Morphologie et antonymie

Constructions savantes et constructions populaires des termes Au milieu du Haut Moyen-Âge, Charlemagne (*747 - 814*) constate que le Latin a perdu son lustre d'antan. Peu à peu, au fil des siècles, les anciennes provinces de Rome se sont mises à ne plus parler le romain mais des langues devenues romanes : le peuple ne le comprend plus, la plupart des religieux ignore la signification des sermons qu'ils donnent pendant la messe. Seuls certains moines reclus dans leur monastère gardent précieusement le souvenir de la culture latine. Dans la volonté d'organiser son royaume en essayant de dépasser les cultures païennes et profanes grâce au christianisme, Charlemagne va chercher à souder le lien entre le peuple, sa langue vulgaire[72] et la traduction latine de la Bible écrite plus de 400 ans auparavant. L'empereur charge ainsi quelques religieux érudits venus des quatre coins d'Europe de l'aider dans cette tâche. Suivant les conseils du théologien anglo-saxon Alcuin (*747 - 804*), Charlemagne crée dans sa capitale d'Aix-La-Chapelle l'école palatine dont le but est d'éduquer l'élite de l'empire qui sera ensuite chargée de l'enseignement du peuple.

Même si cette *renaissance carolingienne*, incomprise des provinces car provenant d'un pouvoir central lointain au regard des habitudes prises durant les siècles précédents, a, en grande partie, échoué, c'est à partir de cette époque que l'habitude a été prise de créer des termes à partir de racines latines mais aussi grecques [Walter, 1988]. C'est pourquoi en français il n'est pas rare de trouver des termes dits "*populaires*" c'est-à-dire ayant subit des déformations normales dans une langue (‹*mère*›, ‹*ciel*›) à côté de termes dits "*savants*" qui eux sont directement construits à partir de morphèmes issus du latin ou du grec (‹*maternel*›, ‹*céleste*›).

Morphèmes antonymes Les morphèmes, en particulier ceux directement issus des langues antiques, sont porteurs de sens. Par exemple, le préfixe latin *bi-* correspond à *deux* (‹*binaire*›, ‹*bisexuel*›), le préfixe latin *semi-* ainsi que le préfixe grec *hémi-*, à une idée de moitié (‹*semi-conducteur*›, ‹*semi-rigide*›, ‹*hémisphère*›) et le préfixe grec *péri-* à ‹*tour*› (‹*périmètre*›, ‹*péricarde*›).

De nombreux mots "*savants*" ont donc été (et sont encore) créés "de toutes pièces" en utilisant des préfixes ou des suffixes marquant une idée négative par rapport à la racine ou alors provenant de mots opposés en latin, en grec et aujourd'hui en français. Ce sont ces deux types d'affixes que nous allons utiliser pour construire automatiquement des listes d'antonymes. Ainsi, les préfixes *poly-* (‹*plusieurs*›) et *mono-* (‹*un*›) s'opposent sur le *nombre*, *hyper-* (superlatif) et *hypo-* (au-dessous) s'opposent par rapport à une valeur de *référence* tandis que *non-* ou *més-* marquent la négation. Pour les suffixes, on remarquera l'opposition *-phobe* ||| *-phile* (‹*homophobe*› ||| ‹*homophile*›) ou l'opposition *-dynamique* ||| *-statique* (‹*hydrostatique*› ||| ‹*hydrodynamique*›). Nous n'étudions ici que l'extraction des couples d'antonymes à partir de préfixes mais la méthode pour extraire les suffixes est la même.

Il existe des études comme [Béchade, 1992] sur les préfixes du français et leur signification. Elles permettent de faire une première étude du comportement des préfixes mais ne permettent pas de façon rigoureuse d'opposer deux préfixes et encore moins de chercher à savoir de quel type d'antonymie ils sont marqueurs. Ainsi, une vérification en corpus s'avère impérative. C'est

[72] Dans le sens étymologique du terme : du latin ‹*vulgare*›, ‹*peuple*›.

pour cette raison que nous avons mis au point un processus semi-automatique de constructions de listes d'antonymes qui permet aussi d'extraire les préfixes opposés.

Processus

Principe Notre méthode est proche de celle utilisée par [Morin, 1999] pour l'acquisition de schémas lexico-syntaxiques mais elle s'en différencie sur deux points principaux :

- Notre problématique première n'est pas de récupérer des morphèmes opposés mais bien de construire, le plus automatiquement possible, des listes d'antonymes. Nous avons tout de même conservé les informations concernant ces préfixes afin de savoir s'ils caractérisent plus particulièrement telle ou telle antonymie. Nous présentons ces résultats dans la section 4.2.2.2.
- Dans [Morin, 1999], les experts valident directement les schémas. Ici, les schémas sont validés indirectement si des couples d'items antonymes, caractérisés par ces schémas, le sont. Cette méthode semble augmenter le nombre de cas à examiner mais il nous paraît difficile d'éliminer des couples de préfixes sans observer leur comportement. Un filtrage automatique basé sur le nombre de couples validés permet de limiter ces cas.

La figure 4.15 présente le processus de construction de ces listes qui est composé de sept étapes. La méthode utilisant les suffixes est identique.

1. *Fournir une liste de couples d'items lexicaux antonymes* : cette liste peut être fournie par des dictionnaires ou bien spécifiées manuellement. Elle va permettre d'avoir un noyau de référence pour amorcer le processus.
2. *Extraire les préfixes par lesquels s'opposent ces items* : la méthode consiste à enlever aux mots le plus long suffixe commun aux deux. Ainsi, avec ‹*monosémique*› et ‹*polysémique*›, on enlève le suffixe *sémique*. On obtient ainsi une liste de couples de préfixes candidats.

   Ces deux premières étapes sont facultatives. On peut directement fournir une liste de préfixes censés s'opposer.
3. *Extraire les couples qui s'opposent par ces préfixes* : on extrait du corpus ces couples d'items. On obtient ainsi une liste de couples d'items candidats.
4. *Valider les couples candidats* : on vérifie manuellement les termes candidats et on ne conserve que ceux qui sont effectivement des antonymes. Cette phase est réalisée par un "expert". Si au moins un des couples validés est caractérisé par un des couples de préfixes candidats, ce dernier est validé.
5. *Extraire des préfixes candidats* : parmi la liste des couples retenus, on extrait les suffixes pour trouver dans la base de nouveaux préfixes marquant l'opposition. Le principe consiste à sortir du corpus l'ensemble des termes ayant ce suffixe et d'extraire les préfixes de chacun.
6. *Filtrer les préfixes candidats* : il s'agit de ne retenir que les couples de préfixes qui caractérisent au moins n couples d'items lexicaux dans le corpus.

Le choix de n est important puisque si n est grand, on élimine un grand nombre de préfixes à vérifier manuellement ce qui facilite la tâche de l'expert mais ne garantit pas l'extraction de l'ensemble des oppositions de préfixes. En revanche, le choix d'un n trop petit multiplie le nombre de couples à vérifier et s'avère difficile à concevoir (un choix de $n = 2$ sur la liste des "a privatifs" entraîne la vérification de 3251 couples de préfixes !). Notre choix s'est porté sur un compromis de $n = 5$ qui élimine un nombre suffisant de candidats (nous n'avons plus alors que 110 vérifications à effectuer sur la liste des "a privatifs").

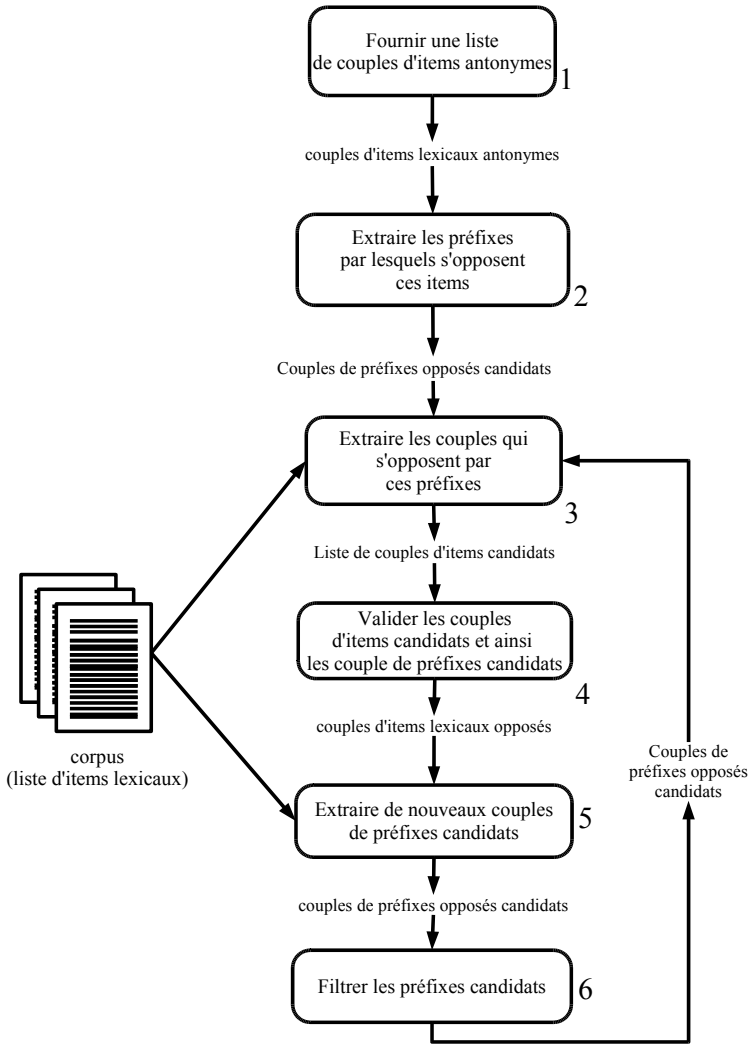


FIG. 4.15 – Processus d'acquisition de préfixes et de termes antonymes.

Déroulement Considérons le corpus très réduit suivant : ‹*acyclique*›, ‹*anticyclique*›, ‹*anticyclone*›, ‹*antimoine*›, ‹*bisémique*›, ‹*biphonie*›, ‹*cyclique*›, ‹*cyclone*›, ‹*moine*›, ‹*monophonie*›, ‹*monosémique*›, ‹*polyphonie*›, ‹*polysémique*›, ‹*souris*›.

1. *étape 1* : On prend une liste de couples que l'on considère comme antonymes :
   ‹*monosémique*› ||| ‹*polysémique*›, ‹*acyclique*› ||| ‹*cyclique*›
2. *étape 2* : On extrait les préfixes par lesquels ces couples s'opposent :
   mono- ||| *poly-*, *a-* ||| ε-
3. *étape 3* : On extrait les couples qui s'opposent par ces préfixes :
   pour *mono-* ||| *poly-*, ‹*monosémique*› ||| ‹*polysémique*›, ‹*monophonie*› ||| ‹*polyphonie*›
   pour *a-* ||| ε-, ‹*acyclique*› ||| ‹*cyclique*›
4. *étape 4* : L'expert valide les paires extraites :
   ‹*monosémique*› ||| ‹*polysémique*› et ‹*acyclique*› ||| ‹*cyclique*› sont validées, les deux couples de préfixes sont validés.
   ‹*monophonie*› ||| ‹*polyphonie*› est rejetée, cela n'entraîne aucune conséquence sur la validité des préfixes.
5. *étape 5* : On extrait de nouveaux préfixes candidats grâce aux paires validées et au corpus.

‹*monosémique*› ||| ‹*polysémique*› possèdent en commun le suffixe *sémique*, on recherche dans le corpus les termes qui ont cette même caractéristique. On extrait ainsi ‹*polysémique*›, ‹*monosémique*› et ‹*bisémique*›. En excluant l'opposition déjà considérée *mono-* ||| *poly-*, ces couples de termes nous permettent comme nouveaux couples de préfixes candidats *mono-* ||| *bi-* et *poly-* ||| *bi-*.
De même, ‹*acyclique*› ||| ‹*cyclique*› permet d'obtenir *anti-* ||| ε- et *a-* ||| *anti-*.

6. *étape 6* : On filtre les couples de préfixes :
   - *mono-* ||| *bi-* apparaît dans deux couples (‹*monosémique*› ||| ‹*bisémique*›, ‹*monophonie*› ||| ‹*biphonie*›), il est donc conservé.
   - *poly* ||| *bi-*, en revanche, n'apparaît qu'une fois dans le corpus, il est donc supprimé. Dans notre expérience, ces préfixes candidats n'ont pas été rejetés par le filtrage automatique mais par les experts qui ont invalidé les dix couples d'items extraits pour ce couple de préfixes. Il ne semble donc pas y avoir d'exception pour les préfixes *bi-* (marquant une idée de ‹*deux*›) et *poly* (marquant une idée de ‹*plusieurs*›) qui sont incluses l'une dans l'autre.
   - *anti-* ||| ε- apparaît lui dans trois couples ‹*cyclone*› ||| ‹*anticyclone*›, ‹*cyclique*› ||| ‹*anticyclique*› et ‹*moine*› ||| ‹*antimoine*›. En pratique la validation ne cherche bien sûr pas tous les couples et s'arrête dès qu'elle en a trouvé n (ici 2).
   - *a-* ||| *anti-* n'apparaît aussi qu'une seule fois et est donc supprimé.

   On réitère l'étape 3

7. *étape 3* : On extrait les couples qui s'opposent par ces préfixes :
   pour *mono-* ||| *bi-*, ‹*monosémique*› ||| ‹*bisémique*›, ‹*monophonie*› ||| ‹*biphonie*›
   pour *anti-* ||| ε-, ‹*cyclone*› ||| ‹*anticyclone*›, ‹*cyclique*› ||| ‹*anticyclique*› et ‹*moine*› ||| ‹*antimoine*›.

8. *étape 4* : L'expert valide les paires extraites :
   ‹*monosémique*› ||| ‹*bisémique*›, ‹*monophonie*› ||| ‹*biphonie*› ‹*cyclone*› ||| ‹*anticyclone*›, et ‹*cyclique*› ||| ‹*anticyclique*› sont validées, les couples de préfixes *mono-* ||| *bi-* et *anti-* ||| ε- sont validés. ‹*moine*› ||| ‹*antimoine*› est rejetée, cela n'entraîne aucune conséquence sur la validité des préfixes.

9. *étape 5* : On cherche à extraire de nouveaux préfixes candidats grâce aux paires validées et au corpus.
   Il n'y en a plus, le processus s'arrête.

Résultats Le corpus que nous avons utilisé est constitué de 79 220 items lexicaux issus de notre base lexicale sémantique [Schwab *et al.*, 2004]. Ces termes correspondent globalement aux entrées hors noms propres de dictionnaires sous forme électronique : dictionnaires classiques [Larousse, 2004] [Robert, 2000], dictionnaires de synonymes (CRISCO[73] cf. 4.2.1.2), thésaurus [Larousse, 1992].

Lors de cette constitution j'ai joué le rôle d'expert-validateur. On peut estimer le temps de réalisation de l'expérience à une cinquantaine d'heures utilisées à plus de 99% par la phase de validation (étape 4).

Cette méthode nous a permis d'extraire 59 couples de préfixes opposés. Le tableau de la figure 4.16 en présente quelques-uns.

[73] http://elsap1.unicaen.fr/cgi-bin/cherches.cgi

préfixe 1	préfixe 2	exemple	contre-exemple
a-	ε	‹*chromatique*› ||| ‹*achromatique*›	‹*afin*›|/|‹*fin*›
an-	ε	‹*aérobie*› ||| ‹*anaérobie*›	‹*anatomiste*›|/|‹*atomiste*›
anti-	ε	‹*communiste*› ||| ‹*anticommuniste*›	‹*moine*›|/|‹*antimoine*›
dés-	ε	‹*accord*› ||| ‹*désaccord*›	‹*avouer*›|/|‹*désavouer*›
in-	ε	‹*imaginable*› ||| ‹*inimaginable*›	‹*incas*›|/|‹*cas*›
il-	ε	‹*licite*› ||| ‹*illicite*›	‹*illustre*›|/|‹*lustre*›
pré-	post-	‹*préface*› ||| ‹*postface*›	ε
hyper-	hypo-	‹*hyperonymie*› ||| ‹*hyponymie*›	‹*hyperstatique*›|/|‹*hypostatique*›
syno-	anto-	‹*synonymie*› ||| ‹*antonymie*›	ε
méro-	holo-	‹*méronymie*› ||| ‹*holonymie*›	ε
mono-	poly-	‹*polysémique*› ||| ‹*monosémique*›	‹*monophonie*›|/|‹*polyphonie*›
mono-	stéréo	‹*monophonie*› ||| ‹*stéréophonie*›	‹*monotype*›|/|‹*stéréotype*›

FIG. 4.16 – Exemples de préfixes antonymes extraits

préfixe 1	préfixe 2	nombre de paires extraites	nombre de paires validées	nombre de paires invalidées	pourcentage
a-	ε	288	71	217	24,6%
an-	ε	61	15	41	24,6%
anti-	ε	156	147	9	94,2%
dés-	ε	195	179	16	91,7%
in-	ε	616	539	77	87,5%
il-	ε	31	18	13	58%
anté-	ε	15	10	5	80%
post-	ε	37	33	4	89%
anté-	post-	3	2	1	66,6%
pré-	post-	11	11	0	100%
hyper-	hypo-	25	24	1	96%
syno-	anto-	4	4	0	100%
méro-	holo-	2	2	0	100%
mono-	poly-	28	25	3	89,2%
mono-	stéréo-	5	4	1	80%

FIG. 4.17 – Extraits des résultats d'extraction

préfixe 1	préfixe 2	complémentaires	scalaires	duals
a-	ε	71 (100 %)		
an-	ε	14 (92,8%)		1 (7,2%)
anti-	ε			147 (100 %)
dés-	ε		3 (1,6 %)	176 (98,3 %)
in-	ε	539 (100 %)		
il-	ε	18 (100%)		
anté-	ε		9 (90%)	1 (10%)
post-	ε		23 (100 %)	
anté-	post-		2 (100%)	
pré-	post-		11 (100%)	
hyper-	hypo-		24 (100 %	
mono-	poly-	25 (100%)		
mono-	stéréo-	4 (100 %)		

FIG. 4.18 – Répartition des schémas suivant le type

La figure 4.17 présente le pourcentage de paires validées par l'expert, c'est-à-dire le nombre de paires considérées comme valides pour chaque couple de préfixes antonymes candidats. On peut constater que,si le taux de validation est très important pour la plupart des couples de préfixes, il est, en revanche, très faible pour le *a privatif* (*a-* et *an-*).

Finissons par la typologie des paires d'antonymes extraites. La figure 4.18 présente les résultats obtenus. On constate que globalement la morphologie permet de relativement bien connaître le type d'antonymie. Ainsi, on peut donner quelques indications sur les couple de préfixes suivant leur sémantique :

- *temporels* (*anté-* ||| ε, *post-* ||| ε, *anté-* ||| *post-*, *pré-* ||| *post-*) : Tous sont scalaires sauf le couple dual ‹*christ*› $|||_d$ ‹*antéchrist*›. La définition de [Larousse, 2004] nous donne « *Imposteur qui, suivant l'Apocalypse, doit venir quelque temps avant la fin du monde pour essayer d'établir une religion opposée à celle de Jésus-Christ.* ». Dans ce cas, l'opposition sémantique s'explique par l'opposition ‹*dieu*› ||| ‹*démon*› tandis que la construction de l'opposition morphologique exprime l'idée de l'arrivée du démon avant le retour du Christ.
- *médicaux* (*hyper-* ||| *hypo-*) : Ils caractérisent des mesures qui sont donc au-dessus ou au-dessous de la normale (‹*hyperthyroïdie*› ||| ‹*hypothyroïdie*›).
- *nombre* (*bi-* ||| ε, *tri-* ||| ε, *mono-* ||| *poly-*, *mono-* ||| *stéréo-*) : Ils s'opposent par une propriété possédée une fois (*mono*) ou plusieurs (*bi-*, *tri-*, *poly*, *stéréo-*). Ils sont complémentaires.
- *absence de propriété* : *il-* ||| ε (‹*illicite*› ||| ‹*licite*›, ‹*illimité*› ||| ‹*limité*›), *a-* ||| ε (‹*typique*› ||| ‹*atypique*›, ‹*sociabilité*› ||| ‹*asociabilité*›), *an-* ||| ε (‹*anencéphale*› ||| ‹*encéphale*›) ils sont tous complémentaires sauf ‹*anion*› ||| ‹*ion*› qui relève plutôt de l'antonymie duale.
- *opposition culturelle ou produit permettant de lutter contre quelque chose* : *anti-* ||| (‹*anticlérical*› ||| ‹*clérical*›, ‹*antiviral*› ||| ‹*viral*›). Ils sont duals.

4.2.2.3 Introduction de couples d'antonymes dans l'apprentissage

Lors de l'introduction d'une nouvelle source d'informations, il est fondamental d'étudier ses caractéristiques pour que les conséquences sur la base de données lexicales soient bénéfiques en termes de pertinence. En particulier, il ne faut pas que cette source soit trop fortement sur-représentée ou sous-représentée par rapport aux autres.

Caractéristiques générales d'un apprentissage basé sur la relation d'antonymie Le premier point à remarquer est qu'il n'existe pas une mais trois antonymies. Il s'agit donc de regrouper les antonymes par type.

Comme nous l'avons déja vu, une LEXIE correspond à un sens d'un item lexical suivant une source de données. Il existe une analogie relativement évidente entre une liste d'antonymes et une liste de synonymes. Dans chacune des listes, certains termes sont synonymes entre eux. Considérons par exemple, les antonymes de ‹*vie*› selon le dictionnaire du CRISCO : ‹*atonie*›, ‹*langueur*›, ‹*molesse*›, ‹*mort*›, les trois premiers sont synonymes.

La création des LEXIES doit donc se faire pour l'antonymie de la même manière que pour la synonymie en regroupant les termes selon leurs composantes connexes dans le graphe de relation de synonymie.

Il est important de noter que ce double regroupement entraîne une conséquence importante sur les sources de type antonymie. Considérons l'item ‹*action*›, il a pour antonymes d'un de ses sens deux termes : ‹*inaction*› en complémentaire et ‹*réaction*› en dual. Nous allons donc créer deux LEXIES pour le même sens ! La solution consiste simplement à considérer non plus une seule source par antonymie mais trois, une par type. Nous aurons bien alors une LEXIE pour un sens.

Caractéristiques des sources et conséquences sur l'apprentissage

Couples extraits grâce à leur morphologie Nous avons dû construire cette source afin de pallier un manque de ressources sur l'antonymie gratuitement accessible. Deux propriétés la caractérisent et vont influer l'apprentissage que nous voulons mettre en place :

- *le type d'antonymie est indiqué* : nous avons construit nous-même cette source en suivant notre définition de l'antonymie nous avons donc séparé les différentes typologies d'antonymie ce qui ne sera plus à faire automatiquement ;
- *sa couverture lexicale est relativement faible* : nous avons extrait environ 2000 couples ce qui, malgré le manque de repères, nous a semblé relativement peu à l'époque. Certaines oppositions "célèbres" ont certes été ajoutées comme ‹*vie*› ||| ‹*mort*› ou ‹*jour*› ||| ‹*nuit*› faisant monter le nombre de couples à 2300 mais la couverture nous semblait encore très faible. Ce pressentiment s'est révélé exact lors de l'introduction du dictionnaire des antonymes du CRISCO qui comprend 11404 entrées et 26991 couples. Nous pouvons donc estimer que nous ne couvrions, au mieux, que 8,52% des relations d'antonymie de la langue française. Du fait de cette faible couverture, nous ne pouvions pas raisonnablement utiliser les propriétés de points fixes des termes sans antonymes.

Dictionnaire des antonymes du CRISCO Le dictionnaire des antonymes du CRISCO est venu compléter leur dictionnaire des synonymes en Octobre 2004. Il n'existe à l'heure actuelle aucune publication sur la manière dont il a été construit. Nous ne pouvons que nous référer à un échange effectué par courriel avec Jean-Luc Manguin, responsable du projet, à la fin Avril 2005.

Ce dictionnaire comprend 11404 entrées, reliées par 53982 relations. Ces relations sont toujours doubles, c'est-à-dire que parmi les 53982, on va trouver la relation entre A et B, et celle entre B et A (dictionnaire symétrisé). Il a été construit à partir de données de la société Memodata[74], et de programmes analysant le dictionnaire des synonymes (synonymes des antonymes, ...).

Cette source a deux propriétés opposées à celle que nous avons construite grâce aux oppositions morphologiques :

- *Elle ne regroupe pas les antonymes selon leur type* : il va falloir mettre au point une méthode automatique pour classifier les antonymes.
- *Elle a une bonne couverture lexicale* : Elle permet ainsi l'utilisation de la propriété des points fixes.

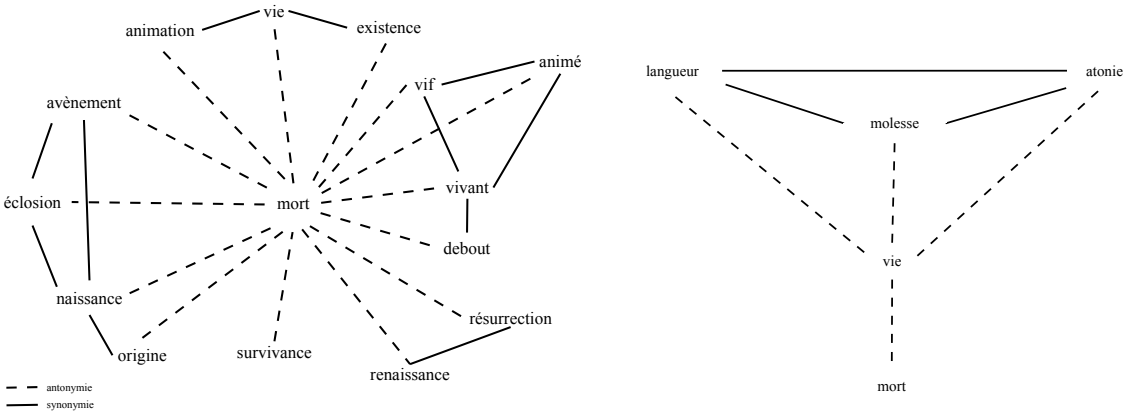
Classement d'antonymes suivant les composantes connexes La méthode utilisée est la même que celle présentée en 4.2.1.2 pour les synonymes. Il s'agit de créer une lexie par composante connexe des antonymes. Prenons les exemples de ‹*vie*› et ‹*mort*›. Les figures 4.19(a) et 4.19(b) présentent pour ces deux items les termes avec lesquels il sont en relation d'antonymie et les relations de synonymie que ces derniers entretiennent entre eux.

Considérons tout d'abord l'exemple de ‹*vie*›. Il y a deux composantes connexes qui correspondent pour la première à l'opposé de *vie*/**existence** avec pour antonyme ‹*mort*› et pour la seconde *vie*/**entrain** avec pour antonymes ‹*langueur*›, ‹*molesse*›, ‹*atonie*›.

Le deuxième exemple, celui de ‹*mort*›, possède beaucoup plus d'antonymes. Le regroupement suivant la synonymie fait apparaître cinq composantes connexes.

1. ‹*animation*›, ‹*vie*›, ‹*existence*›
2. ‹*animé*›, ‹*vif*›, ‹*vivant*›, ‹*debout*›

[74] http://www.memodata.com (cf 1.4.2)



(a) Exemple de graphe d'antonymie centré sur l'item lexical ‹*mort*›

(b) Exemple de graphe d'antonymie centré sur l'item lexical ‹*vie*›

FIG. 4.19 – Exemples de graphe d'antonymie

3. ‹*survivance*›
4. ‹*avènement*›, ‹*éclosion*›, ‹*naissance*›, ‹*origine*›
5. ‹*renaissance*›, ‹*résurrection*›

Cet exemple est très intéressant car il montre bien que le découpage en composantes connexes regroupe les antonymes en fonction de leur sens et donc par effet de bord les regroupe aussi en fonction du type d'antonymie. Ainsi, la première composante correspond aux antonymes complémentaires du nom *mort*/**cadavre**, la deuxième aux antonymes complémentaires de l'adjectif *mort*/**décédé**, le troisième à l'antonyme complémentaire du nom *mort*/**inexistence**. Les deux derniers correspondent aux antonymes duals du même sens de mort, le sens de **décès**. La différence entre les deux s'explique par l'axe de symétrie qui diffère. Dans le quatrième ensemble, nous avons une opposition due d'un côté au passage de l'état d'‹*inexistence*› à l'état d'‹*existence*› pour les évènements ‹*naissance*› ou ‹*origine*› et de l'autre le passage de l'état d'‹*existence*› à l'état d'‹*inexistence*› pour la ‹*mort*› par rapport à la *vie*/**existence**. Dans le cinquième, cette même opposition se fait par rapport à la *mort*/**inexistence** comme le montre la figure 4.20.

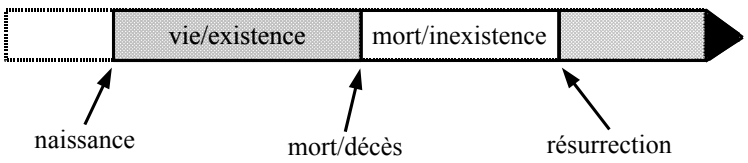


FIG. 4.20 – *mort*/**décès**, ses antonymes duals et leur axe de symétrie.

Une méthode automatique présentée par l'algorithme 7 permet d'identifier le ou les types d'antonymie qui caractérisent ces ensembles[75].

Pour chacune des oppositions,

1. si l'opposition est connue dans la source basée sur la morphologie, source manuelle que nous considérons de ce fait comme très fiable, alors nous prenons le ou les types de cette opposition.

[75] rappelons, par exemple, que ‹*avant*› $|||_s$ ‹*après*› et ‹*avant*› $|||_d$ ‹*après*›

Algorithme 7: Calcul du type d'antonymie caractérisant un ensemble d'items antonymes

Entrée : x un item, $\{\overline{x_1},\ldots,\overline{x_n}\}$ un ensemble d'items antonymes
Sortie : Le type ou les types d'antonymie qui caractérisent cet ensemble
$ens_{type} = \emptyset$
pour *chaque couple* $(x,\overline{x_i})$ **faire**
  si $type_{morpho}((x,\overline{x_i})) \neq \emptyset$ **alors**
    $ens_{type} = ens_{type} \cup type_{morpho}((x,\overline{x_i}))$
    % $type_{morpho}((x,\overline{x_i}))$ renvoie l'ensemble des types du couple $(x,\overline{x_i})$ suivant la source construite grâce à la morpho
  sinon
    pour $\alpha \in \{comp, scal, dual\}$ **faire**
      si $Anti_{Pot_\alpha}(x) > 0 \wedge Anti_{Pot_\alpha}(\overline{x_i}) > 0$ **alors**
        $ens_{type} = ens_{type} \cup \alpha$
  si $ens_{type} = \emptyset$ **alors**
    retourner {*comp, scal, dual*}
  sinon
    retourner ens_{type}

2. pour chacun des types d'antonymie, si le potentiel d'antonymie des deux items est positif pour les deux items du couple alors cette opposition est susceptible d'être de ce type.

4.3 Conclusion et perspectives

Nous avons noté dans la conclusion de la première partie les limites des vecteurs d'idées pour représenter non seulement le sens des termes mais aussi pour modéliser certains aspects des fonctions lexicales. Une solution à la fois facile à mettre en œuvre et qui semble être compatible avec les théories cognitives actuelles est d'allier des informations lexicales aux informations vectorielles déjà utilisées. Nous avons ainsi redéfini l'ensemble des fonctions lexicales de construction et d'évaluation des relations symétriques présentées au chapitre précédent à l'aide de la contextualisation forte. Nous avons aussi amélioré la fonction d'antonymie par l'apport d'un module d'apprentissage basé sur les oppositions rencontrées par la fonction. Grâce à ce module, il y a amélioration synchronique de la base de données vectorielle et de la fonction. Le système global s'enrichit de l'apport de la fonction qui elle-même s'enrichit de l'apport de l'ensemble du système (*double boucle*) [Lecerf, 1997].

5

Construction d'une Base Lexicale Sémantique

DANS ce chapitre, nous tirons le bilan de l'expérience acquise dans les précédents pour revoir en partie le modèle de Base Lexicale Sémantique, c'est-à-dire le modèle d'une base de données permettant la représentation et l'exploitation du sens des items lexicaux. Nous présentons les six hypothèses de départ qu'il nous paraît nécessaire d'adopter dans le but de construire une telle base. Ces hypothèses nous ont conduit à choisir une architecture à trois niveaux d'objets lexicaux (LEXIE, ACCEPTION, ITEM LEXICAL). Nous montrons comment ces hypothèses, les applications hétérogènes visées ainsi que des caractéristiques techniques nous ont amenés à adopter une architecture multi-agent dont nous présentons les caractéristiques conceptuelles et techniques. Le système multi-agent proposé, appelé Blexisma, a pour but d'intégrer tout agent pouvant permettre de créer, d'améliorer et/ou d'exploiter une ou plusieurs Bases Lexicales Sémantiques. Nous exposons enfin les différents agents déjà implémentés ainsi qu'un exemple de leur interaction dans le cadre de l'acquisition d'informations sémantiques et de leur exploitation pour fabriquer des objets lexicaux.

Sommaire

Nous l'avons vu tout au long des chapitres précédents, nos travaux se placent dans le cadre d'un projet plus vaste visant des champs applicatifs aussi variés que la recherche d'informations, la traduction automatique ou encore le résumé automatique. Les applications étant variées, le système que nous développons pour permettre la représentation et l'exploitation du sens nécessite d'être générique et évolutif. La couverture du lexique des langues traitées doit ainsi être la plus large possible, ce qui pose en particulier un problème d'apprentissage des connaissances, non seulement pour l'acquisition des termes de ce lexique mais surtout pour l'identification de leur sens. C'est donc une triple problématique que nous essayons principalement de résoudre dans nos recherches : découvrir un maximum d'items lexicaux constituant le lexique des langues traitées, récupérer les informations concernant plus particulièrement le sens de ces items et utiliser ces informations pour fabriquer des objets représentant ces sens.

Dans les chapitres précédents, nous avons cherché à améliorer une Base Lexicale Sémantique déjà existante à l'aide des fonctions lexicales symétriques. Nous avons ainsi pu constater les limites du modèle purement vectoriel non seulement pour ce qui nous intéressait de prime abord, la modélisation des relations sémantiques, mais aussi, et c'est ce qui nous intéressera d'avantage ici, pour la représentation du sens des items lexicaux. Dans ce chapitre, nous tirons le bilan de cette expérience pour revoir en partie le modèle de BLS. L'acquisition des termes et de leur sens est basée sur quelques hypothèses fortes. Nous considérons ces hypothèses comme les axiomes de nos recherches et elles sont à ce titre fondamentales à nos yeux. Elles justifient ainsi non seulement l'architecture conceptuelle de notre système mais aussi l'architecture implémentationnelle que nous avons choisies. Ces hypothèses nous ont amenés à retenir une approche totalement holistique[76], d'un côté en essayant de tenir compte du maximum d'informations qu'il est possible de trouver sur les mots et leurs usages et d'un autre en utilisant les résultats de tous les traitements effectués afin d'améliorer ces derniers dans l'esprit des travaux déjà effectués pour la fonction d'antonymie (cf. section 4.1.2). L'exploitation et l'apprentissage des vecteurs conceptuels sont ainsi fortement liés. Il est donc nécessaire de pouvoir facilement ajouter des modules apportant tel ou tel service et capables de s'auto-modifier. C'est une des raisons pour lesquelles notre vision de l'architecture nécessaire s'est rapidement rapprochée des systèmes multi-agents (SMA).

Ce chapitre présente, dans un premier temps, les hypothèses retenues pour bâtir une base lexicale sémantique. Nous considérons une architecture basée sur des objets lexicaux LEXIES, ACCEPTIONS et ITEM LEXICAL qui regroupent toutes les informations vectorielles et lexicales qu'il est possible de rassembler sur les sens que peut avoir un terme à partir de diverses sources.

[76] Nous entendons par holistique le fait de ramener les connaissances d'un élément à celles de l'ensemble. C'est-à-dire que nous pensons que tout élément d'un système plus complexe doit être considéré comme plongé dans un environnement sur lequel il intervient et qui, en retour, intervient sur lui. L'élément doit alors être observé dans cette perspective afin de pouvoir décrire le système dans sa globalité. L'holistique s'oppose ainsi au réductionnisme qui, au contraire, réduit les phénomènes complexes à leurs éléments plus simples et considère ces derniers comme plus fondamentaux que les phénomènes complexes observés.

Nous montrons ensuite comment les hypothèses nous ont amenés à adopter une architecture multi-agent dont nous précisons les diverses caractéristiques conceptuelles et techniques. Nous présentons quelques agents avant de conclure par un premier exemple de collaboration pour la construction des objets lexicaux d'un terme.

5.1 Hypothèses de construction d'une base sémantique lexicale

Dans cette section, nous tirons le bilan de l'expérience acquise dans les chapitres précédents pour revoir les hypothèses de départ que nous avons prises pour la construction d'une base lexicale sémantique, c'est-à-dire les moyens utilisés pour fabriquer des objets lexicaux ITEM LEXICAL, ACCEPTION et LEXIE permettant de représenter le sens d'autant de termes de la langue que possible. Ces hypothèses, au nombre de six, sont les suivantes : (I) une *représentation hybride du sens par une approche combinant approche thématique (vectorielle) et approche lexicale (relations symbolique)*, (II) une prise en compte des *relations sémantiques internes* (polysémie), (III) une *génération automatique* des ACCEPTIONS, (IV) la réalisation d'une *analyse multi-source* (à partir de dictionnaires classiques, de listes de synonymes, d'antonymes, de sites web, ...), (V) un *apprentissage permanent* et (VI) l'hypothèse de *double boucle*.

5.1.1 Hypothèse I - Représentation hybride du sens : approche combinant représentation thématique et informations lexicales

Plusieurs raisons nous ont poussés à adopter une approche combinant représentation thématique et informations lexicales (relations entre objets du lexique) : la limitation des vecteurs conceptuels en ce qui concerne les relations lexicales et les phénomènes du langage qui sont plus explicables par l'usage que par les traits sémantiques, l'idée d'apporter au rappel intrinsèque des vecteurs conceptuels la précision des réseaux sémantiques ainsi qu'une adéquation au traitement cognitif du langage par le cerveau humain.

5.1.1.1 Limitations des vecteurs conceptuels dans la modélisation des fonctions lexicales

Nous l'avons vu dans la section 3.6, les vecteurs conceptuels ne permettent pas la représentation fine du sens des segments textuels et ne donnent seulement que des indications sur leur thème. Ce phénomène s'explique par le fait que la distance angulaire entre deux vecteurs conceptuels semble devoir être interprétée plutôt comme une distance thématique entre les deux idées correspondantes plutôt que comme une distance sémantique ou ontologique. Prenons l'exemple de ‹*fourmis*› et ‹*fourmilier*›. Les informations à partir desquelles on calcule le vecteur conceptuel de ce dernier spécifient que les premières constituent la proie quasi-unique de ce dernier, ce qui se retrouve de manière non négligeable dans le vecteur, en tout cas de manière plus importante que les idées associées à ‹*mammifère*›. Le vecteur de ‹*fourmilier*› est ainsi plus proche de celui de ‹*fourmis*› que de celui de ‹*mammifère*›. Les vecteurs conceptuels ne traduisent donc pas directement les relations ontologiques et ne possèdent donc pas intrinsèquement la relation d'hyperonymie ; or cette relation est fondamentale pour la représentation du sens, comme nous l'avons vu dans la partie 1.3.2.

La modélisation de l'hyperonymie en particulier mais aussi des fonctions lexicales en général ne peut se faire sans apport d'informations lexicales comme nous l'avons montré aux chapitres 3 et 4. C'est le cas, par exemple, de l'antonymie dont la modélisation est basée sur la double approche vectorielle et symbolique (cf. section 4.1.2).

Le vecteur exprimant l'idée d'une fièvre de quarante degrés aurait des composantes de *MALADIE* et de *CHALEUR* prépondérantes. La génération d'une phrase correspondante sans l'aide

d'autres informations que celles données par les idées contenues dans le vecteur pourrait donner au moins trois syntagmes différents : *« *une fièvre importante* », *« *une grande fièvre* » ou « *une forte fièvre* » pourtant seul le troisième semble pleinement accepté par l'usage. Seules des informations lexicales, purement symboliques, sur les rapports syntagmatiques que les termes entretiennent entre eux peuvent permettre la génération d'une phrase correcte (cf. cooccurrences section 1.3.1.2).

5.1.1.2 Rappel et précision

Dans la section 2.1.3.2, nous avons montré que les vecteurs conceptuels bénéficiaient d'un fort rappel mais d'une précision bien moindre. À l'inverse, les réseaux sémantiques permettent d'obtenir une bonne précision associée à un rappel faible (cf. section 1.3.2). Une deuxième raison pour combiner représentation thématique et informations lexicales, sans conteste la principale, est d'allier, dans un but d'analyse sémantique, au fort rappel des vecteurs conceptuels la forte précision des réseaux sémantiques. L'idée sous-jacente est de chercher à obtenir des résultats, de pouvoir explorer des solutions plus éloignées tout en restant dans le domaine du possible.

5.1.1.3 Adéquation avec le modèle cognitif

Nos travaux visent deux objectifs duaux : comprendre et produire des textes. Si on se place dans une perspective cognitiviste, on peut considérer que les textes ne sont rien de plus que les résultats d'une activité cognitive exercée par le cerveau humain (cf. section 1.1.2). Nos travaux ont donc pour but d'interpréter les résultats d'une activité cognitive et de produire des résultats aussi proches que possible de ceux que pourrait créer cette même activité. Ainsi, il n'est pas déraisonnable de penser que s'inspirer des recherches en sciences cognitives peut nous aider à résoudre nos problématiques.

Comme nous le notions précédemment ainsi que dans la section sur les collocations (section 1.3.1.2), « *Dans toutes les langues, certaines combinaisons d'items lexicaux prévalent sur d'autres sans qu'il ne semble n'y avoir de motif logique.* » On peut parler de « *forte fièvre* » ou de « *dormir profondément* » mais pas de *« *fièvre importante* » ni de *« *dormir totalement* ». Pourquoi le langage ne permet pas l'expression de certaines idées grâce à des termes qui pourtant sont quasi-synonymes ? Le point de vue cognitif et en particulier les neurosciences peuvent nous aider à mieux appréhender ces phénomènes.

Depuis le milieu du XIX^{e} siècle, l'étude de diverses lésions cérébrales a permis, par recoupements d'informations, de comprendre la manière dont le cerveau traite le langage. Il a ainsi été possible de localiser trois zones neuronales dont les interactions nous permettent de communiquer [Damasio & Damasio, 1992].

La première zone, qui occupe une partie des deux hémisphères cérébraux, permet la représentation non-linguistique des interactions entre le corps et son environnement. Cette aire réalise cette fonction par deux processus. Par le premier, elle bâtit des schémas de tout ce que l'individu voit, entend, touche, accomplit ou conçoit. C'est donc dans cette partie du cerveau que sont bâtis des schémas correspondant aux idées lues et même écrites par l'individu. Par le deuxième processus, elle fabrique les concepts qui, à un niveau supérieur, permettent une classification de toutes ces informations. C'est cette zone, purement conceptuelle (gestion des idées), qui est le siège de nos facultés d'abstraction et de métaphore. Les lésions de cette partie du cerveau semblent être les seules qui entraînent une perte de la perception des couleurs chez des sujets qui n'ont pourtant pas de problème visuel. Ils perdent l'aptitude de conceptualisation des couleurs et conçoivent alors le monde en nuances de gris.

Une deuxième zone, généralement située dans l'hémisphère gauche[77], gère tout ce qui concerne

[77] Cette zone est située dans l'hémisphère gauche de 99% des droitiers et 66% des gauchers

la "surface" du langage humain c'est-à-dire les signes qu'ils soient oraux, textuels ou gestuels. C'est dans cette partie que sont donc représentés les phonèmes, les combinaisons de phonèmes, les règles de combinaisons de mots en phrases, les termes écrits ainsi que ce qui nous intéresse le plus ici, les associations lexicales. Son dysfonctionnement entraîne chez les patients une mauvaise prononciation des mots pourtant conceptuellement retrouvés.

Enfin, la troisième zone située également dans l'hémisphère gauche exerce une activité d'organisation entre les deux premières. Elle permet la production d'idées à partir de termes ou de termes à partir d'idées. Les dysfonctionnements de cette partie du cerveau n'entraînent ni un trouble au niveau des concepts ni une quelconque déformation des mots. Aujourd'hui, l'étude des lésions de certaines parties du cerveau n'est plus le seul moyen de l'étudier. L'imagerie fonctionnelle cérébrale (IFC), par l'observation des variations du débit sanguin et des variations de potentiel électrique, a permis de confirmer l'existence de ces trois zones interconnectées et de les limiter avec une précision anatomique nettement plus grande [Houde *et al.*, 2002].

La structure interne du cerveau explique donc bien pourquoi certaines idées ne peuvent pas être exprimées grâce à des termes qui pourtant sont quasi-synonymes. La gestion des idées et la gestion des formes ne sont pas situées dans la même aire cérébrale, ainsi, une représentation du sens associant des informations complètement conceptuelles à des informations lexicales semble être compatible avec le modèle cognitif du cerveau humain pour le traitement du langage.

5.1.2 Hypothèse II - Utilisation conjointe d'objets lexicaux de type acception et item lexical

Les mots peuvent avoir plusieurs sens. Derrière cette trivialité connue et attestée[78] depuis des millénaires se cache l'un des problèmes les plus importants du TALN, la désambiguïsation sémantique, en d'autres termes, le problème de l'identification des acceptions utilisées dans un texte quelconque (cf. section 1.3.3.1). Rappelons qu'on peut définir une acception comme un sens particulier d'un item lexical admis et reconnu par l'usage. Il s'agit d'une unité sémantique propre à une langue donnée [Sérasset & Mangeot, 2001]. Par exemple, l'item lexical ‹*botte*› a au moins trois acceptions avérées, la **chaussure**, l'**amas de paille** ou le **coup**. Contrairement aux items lexicaux, les acceptions sont donc monosémiques. Il faut toutefois noter que la monosémie n'empêche pas les acceptions de pouvoir être raffinées. Par exemple, l'acception *frégate*/**bateau** peut être raffinée en **bateau ancien** et **bateau moderne**.

Cette section présente dans un premier temps les phénomènes de polysémie et de monosémie puis elle explicite notre deuxième hypothèse en montrant pourquoi nous considérons que nous ne pouvons pas faire l'économie de la création d'un objet par sens d'item et un objet par terme pour représenter le sens des items.

5.1.2.1 Monosémie et polysémie

Il est admis que chacun des mots de la langue couvre une certaine *aire sémantique* qui correspond à l'ensemble des significations dont il est susceptible. On distingue d'une part la *monosémie* qui concerne les termes qui n'ont qu'un seul sens et d'autre part la *polysémie* et l'*homonymie* qui concernent les termes qui ont plusieurs sens.

Monosémie Un item *monosémique* (dit *monosème*) n'a qu'un seul sens. Ainsi, des termes comme ‹*calame*›, ‹*cajou*›, ‹*neuroleptique*›, ‹*polyamide*› semblent n'avoir qu'une seule signification.

[Damasio & Damasio, 1992].

[78] On sait que les Sumériens, réputés pour avoir inventé l'écriture au cours du IVe millénaire avant notre ère, avaient un langage très polysémique [Glassner, 2001].

Dans le vocabulaire général, cette situation est beaucoup moins courante que dans le vocabulaire scientifique ou technique. En effet, il n'est pas rare que des monosèmes dont la fréquence dans l'usage s'élève, tendent à se charger de nouvelles significations et à devenir des polysèmes. Par exemple, ‹*bug*› qui, dans la deuxième partie du vingtième siècle a connu plusieurs dérivations de sens impliquant plusieurs langues. Initialement, ce terme provient de l'anglais signifiant ‹*insecte*›[79]. Le terme ‹*bug*› est rentré dans le vocabulaire français[80] où il n'a eu pendant longtemps que le sens de **problème survenant dans un système informatique**. Signe de la démocratisation de l'informatique mais surtout de ses problèmes, il est aujourd'hui utilisé non plus seulement pour un problème dans un système informatique mais pour un problème en général. Ces dérivations de sens sont aussi très souvent d'ordre métaphorique comme, par exemple, ‹*banane*› qui est devenu le nom d'une coiffure qui ressemble fortement au fruit de même nom.

Polysémie et homonymie On parle habituellement de *polysémie* (l'item est qualifié de *polysémique* ou appelé *polysème*) lorsqu'un item lexical rassemble plusieurs sens entre lesquels existe un lien. On parle d'*homonymie* si un terme rassemble plusieurs sens entre lesquels il n'existe pas de lien. Il est habituel, dans ce cas, de considérer qu'il s'agit de deux items différents et on peut dire que ces deux items ne présentent pas plus d'affinité sémantique ensemble que n'importe quel item du dictionnaire pris au hasard.

Pour différencier homonymie et polysémie, une étude diachronique peut souvent se révéler efficace. Ainsi, il est habituel de considérer qu'il existe deux verbes homonymes ‹*louer*›, l'un signifiant **adresser des louanges** et l'autre **donner/prendre en location** qui dérivent du latin *laudare* pour le premier et du latin *locare* pour le deuxième. Toutefois, le critère étymologique peut parfois ne pas se révéler décisif. Une forte évolution sémantique peut conduire une forme polysémique à une forme homonymique. Par exemple, ‹*grève*› est issu du latin populaire *grava* qui signifie à l'origine **terrain de sable et de gravier au bord de l'eau**. Sur la ‹*place de Grève*› (l'actuelle place de l'Hôtel-de-Ville à Paris) étaient réunis, quand on n'y exécutait pas les condamnés à mort,les ouvriers qui attendaient l'embauche. Au fil du temps, on associa à l'item lexical ‹*grève*› le sens de **arrêt de travail**. En synchronie, les liens entre les deux sens n'existent plus et on peut considérer que la polysémie est devenue homonymie.

Toutefois, une évolution inverse est toujours possible. Des termes sans lien historique sont aujourd'hui ressentis comme un seul item lexical polysémique. C'est, en général, la conséquence d'une *attraction paronymique*, c'est-à-dire la confusion de deux mots due à une ressemblance phonétique (*paronymie*). Il n'est pas rare de voir confondus, l'occasion aidant, ‹*bulletin*› et ‹*butin*›, ‹*extorquer*› et ‹*escroquer*›, ‹*précepteur*› et ‹*percepteur*› ou encore ‹*dénoter*›, ‹*détoner*› et ‹*détonner*›. Dans certains cas, ces attractions paronymiques se lexicalisent. Par exemple, l'item ‹*souffreteux*›, qui signifiait, en ancien français **indigent, miséreux, privé de** et se rattachait au substantif ‹*souffraire*› (**privation, manque, disette, misère**) s'est trouvé rapporté à ‹*souffrir*› et a pris le sens **d'une santé fragile, souvent malade**.

[79] On dit souvent que le sens informatique de cet item a été inventé par Grace Hopper (*1906 - 1992*), la créatrice dans les années 1950 du langage Cobol. Il proviendrait d'un papillon de nuit retrouvé grillé contre un relais électronique de son ordinateur Mark II de l'Université d'Harvard et qui provoqua son arrêt le 9 septembre 1947. Depuis plus de 50 ans, cette histoire nourrit des générations d'informaticiens : des insectes se collent aux circuits des machines et provoquent des pannes. Mais il s'agit en fait une grosse erreur "entomologique". Grace Hopper (On remarquera que son nom se prononce presque comme ‹*grasshopper*› qui signifie, ironie de l'histoire, ‹*sauterelle*›) écrit dans son carnet de note « *Premier cas de "bug" réel à avoir été trouvé* ». Il s'agit d'une remarque qui atteste que le sens de ‹*bug*› est donc préalable à l'anecdote. De fait, le terme est déjà en usage chez les spécialistes du radar pendant la deuxième guerre mondiale. Dès 1896, l'*Hawkin's New Catéchism of Electricity* [Hawkins, 1896] donne : « *le terme bug est utilisé pour désigner tout problème ou erreur dans le fonctionnement d'un appareil électrique* ». La véritable origine du terme remonterait aux débuts du télégraphe électrique. Sur un des appareils d'émission en morse (un clavier Vibroplex), était dessiné un scarabée. Cet appareil était fort connu pour son usage délicat. Les débutants qui l'utilisaient avaient tendance à introduire des perturbations sur la ligne, des bugs.

[80] On trouve aussi parfois ‹*bogue*› au masculin pour équivalent de ‹*bug*›.

Dans la littérature, la différence entre homonymie et polysémie est souvent floue. Pour une situation où certains linguistes considèrent une polysémie d'autres y voient une forme d'homonymie. En général, ceux qui tentent de décrire les faits grammaticaux et sémantiques ont tendance à être homonymistes tandis que ceux qui veulent les expliquer sont plutôt polysémistes.

Si le problème de la différentiation entre l'homonymie et la polysémie est un problème qui reste encore largement ouvert chez les linguistes, il semble toutefois assez peu intéressant au niveau du TALN en général et de la désambiguïsation en particulier. Dans ce dernier cas, il s'agit de trouver le sens le plus adéquat pour un terme et non de savoir comment le terme a acquis ce nouveau sens ni quels sont les rapports qu'entretiennent ces sens entre eux. En d'autres termes, nous réalisons une étude synchronique du lexique plutôt qu'une étude diachronique [81]. Il faut toutefois préciser que les séparations de sens entre homonymes étant souvent beaucoup plus importantes que les séparations entre polysèmes, les cas d'homonymie se révèlent beaucoup plus simples à traiter en pratique. Nous parlerons dans la suite de ce mémoire d'item polysémique sans chercher à distinguer une vraie polysémie d'un cas d'homonymie.

5.1.2.2 Représentation des acceptions

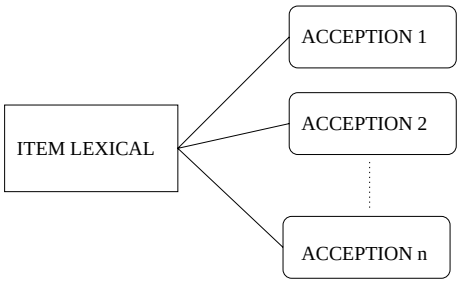
Nous considérons que le fait pour un item lexical d'avoir plusieurs sens doit être pris en compte dans la construction de nos vecteurs conceptuels. Nous pensons qu'il est difficile de considérer qu'une quelconque tâche de désambiguïsation est possible sans la considération d'un objet par sens du terme. En effet, si on envisage pour représenter un item lexical, la présence d'un seul vecteur conceptuel, on peut être confronté à l'émergence de combinaisons de sens non pertinentes au cours du processus d'analyse sémantique. Prenons un exemple. Supposons que les deux concepts les plus importants soient, pour *souris*/**animal** ANIMAL et GRIS et pour *souris*/**ordinateur**, INFORMATIQUE et COMMUNICATION. Si nous regroupons les deux sens dans un même vecteur, les 4 concepts seront prépondérants. L'analyse sémantique d'une phrase comme « *la souris a mangé le fil gris du téléphone* » fait alors ressortir les concepts GRIS et TÉLÉCOMMUNICATION qui ne sont pourtant pas regroupés dans un des sens de souris.

Notre deuxième hypothèse est de prendre en compte les différents sens possibles pour un terme par la construction d'un objet par sens. Cet objet est composé conformément à l'hypothèse I d'un vecteur conceptuel et de fonctions lexicales. Nous appelons cette entité ACCEPTION.

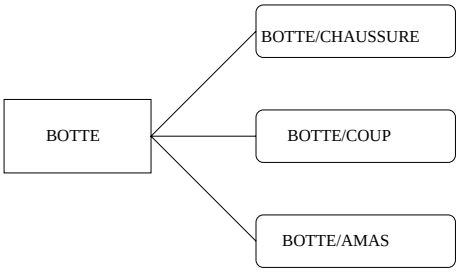
À partir de ces deux premières hypothèses, nous pouvons spécifier la structure interne des objets ACCEPTION qui sont donc composées d'un certain nombre d'*informations linguistiques* :

- un **identifiant**, par exemple le nom de l'item concaténé à un marqueur permettant de différentier ses diverses ACCEPTIONS ;
- la **morphologie** composé des *catégories grammaticales* (*nom, pronom, adjectif, adverbe, etc.*), du *genre* (*masculin, féminin, neutre*) et du nombre (*singulier, pluriel*) ;
- la **fréquence en usage** c'est-à-dire le nombre de fois (ou au moins une évaluation) où l'acception a été rencontrée ;
- un **vecteur conceptuel** ;
- les **fonctions lexicales** associées ;
- des **informations étymologiques** ;

[81] Il est bien entendu que nous entendons synchronisme et diachronisme au sens habituellement utilisé en linguistique [Saussure, 1916]. Une étude en synchronie ne peut se faire véritablement à un instant t, car il est illusoire de vouloir obtenir un cliché du lexique à un moment précis. Il s'agit plutôt d'une étude couvrant une période restreinte dans le temps, ce temps durant généralement plusieurs années. Avoir un apprentissage n'implique donc pas forcément une étude en diachronie.



(a) Organisation générale de la représentation du sens pour un item lexical. Le vecteur conceptuel général d'un item est calculé à partir des vecteurs de chacune des ACCEPTIONS de cet item.



(b) Organisation générale de la représentation du sens pour l'item lexical ‹*botte*› (cf. fig. 5.2 et 5.3)

FIG. 5.1 – Organisation générale de la représentation du sens pour un item lexical.

– des **gloses**, c'est-à-dire des informations que l'on trouve dans certains dictionnaires (principalement les dictionnaires de traduction ou de synonymie) pour préciser le sens d'un terme.

Les informations de chaque ACCEPTION sont fusionnées pour fabriquer un objet *item lexical* structuré de la même façon (cf. fig. 5.11). En pratique, il s'agit de la somme vectorielle normée de chacun des vecteurs, de l'union des morphologies, des fonctions lexicales, ... Les figures 5.2 et 5.3 présentent un exemple d'objets ACCEPTION et ITEM LEXICAL pour ‹*botte*›.

5.1.3 Hypothèse III - Génération automatique

Notre objectif est donc de construire une base de stockage d'objets ACCEPTION et ITEM LEXICAL. La difficulté principale vient de la création de ces objets. Dans des dictionnaires francophones classiques, comme le sont [Larousse, 2004] et [Robert, 2000], sur les environs 80 000 items répertoriés (noms communs, noms propres, expressions, ...), une majorité est polysémique. Dans la première expérience, menée par Mathieu Lafourcade, pour un peu plus de 100 000 entrées (noms communs, noms propres, expressions, ...), le taux de termes polysémiques est d'environ 61%. Le nombre moyen de définitions pour ces derniers étant d'un peu plus de 5, il faudrait indexer à la main au bas mot 400 000 objets lexicaux, ce qui est totalement inenvisageable.

Notre troisième hypothèse forte est qu'il est possible d'automatiser cette tâche grâce à un apprentissage basé sur des informations extraites de diverses sources (dictionnaires, listes de synonymes, indexations manuelles, recherches sur le Web, ...). Pour chaque item lexical, chacune des sources peut nous fournir une ou plusieurs définitions dont on peut extraire un certain nombre d'informations. Les plus riches mais aussi les plus difficiles à analyser sont celles issues des dictionnaires classiques. Ces définitions peuvent permettre d'obtenir les informations de morphologie, de relations lexicales, éventuellement d'étymologie et à partir du texte proprement dit, de calculer un vecteur conceptuel. Nous nommons LEXIE l'objet qui regroupe toutes les informations qu'il est possible d'extraire d'une définition. Sa structure interne est identique à celle d'une ACCEPTION ou d'un ITEM LEXICAL (cf. hypothèse II). L'essentiel de l'apprentissage principal se fait sur des dictionnaires à usage humain (dictionnaires classiques, de synonymes d'antonymes, ...) ou machinaux. Dans la perspective où nous nous plaçons, dans un espace dimensionné en fonction d'une hiérarchie de concepts, un amorçage du système d'apprentissage est nécessaire. Il s'agit d'affecter des vecteurs à un nombre réduit d'entités qui sont choisies en fonction de leur fréquence en langue et/ou de leur polysémie. La taille du noyau est très réduite (un millier de termes environ) et les éléments de ce noyau considérés comme pertinents. À partir de ce noyau, le processus d'apprentissage peut débuter. La méthode d'analyse construit, à partir

– **identifiant** : *botte*/**chaussure**
– **morphologie** : *nom féminin*
– **fréquence en usage** : 10 000
– **vecteur conceptuel** : *C4 :PLUIE*, *C4 :CHAUSSURE* (composantes principales)
– **fonctions lexicales** : hyper = {chaussure}, hypo = {boots, bottine, santiag}, ...
– **informations étymologiques** : *bot*
– **gloses** :

– **identifiant** : *botte*/**amas**
– **morphologie** : *nom féminin*
– **fréquence en usage** : 4 000
– **vecteur conceptuel** : *C4 :HERBE ET FOUGÈRE*, *C4 :AGRICULTURE* (composantes principales)
– **fonctions lexicales** : syn = {balle, bouquet, fagot, gerbe, tas}, hyper = {assemblage}, ...
– **informations étymologiques** : *bote*
– **gloses** :

– **identifiant** : *botte*/**coup**
– **morphologie** : *nom féminin*
– **fréquence en usage** : 700
– **vecteur conceptuel** : *C4 :SPORT* (composante principale)
– **fonctions lexicales** : syn = {secret}, hyper = {coup}, ...
– **informations étymologiques** : italien :*botta*(coup)
– **gloses** :

FIG. 5.2 – Exemple d'objets ACCEPTION pour l'item lexical ‹*botte*›.

– **identifiant** : botte
– **morphologie** : *nom féminin*
– **fréquence en usage** : 14 700
– **vecteur conceptuel** : *C4 :PLUIE*, *C4 :CHAUSSURE*, *C4 :HERBE ET FOUGÈRE*, *C4 :AGRICULTURE*, *C4 :SPORT* (composantes principales)
– **fonctions lexicales** : syn = {balle, bouquet, fagot, gerbe, tas, secret}, hyper={chaussure, assemblage, coup}, ...
– **informations étymologiques** : *bot* ; *bote* ; italien :*botta*(coup)
– **gloses** :

FIG. 5.3 – Exemple d'objet ITEM LEXICAL ‹*botte*› regroupant les informations des ACCEPTIONS de la figure 5.2.

de vecteurs conceptuels déjà existants et des définitions, de nouveaux vecteurs. L'idée est qu'à partir d'un noyau réduit d'items pertinents, un apprentissage sur des définitions permet de créer une cohérence entre les vecteurs et donc de générer une base d'items pertinents.

5.1.4 Hypothèse IV - Analyse multi-source

5.1.4.1 Pourquoi une analyse multi-source ?

L'analyse des définitions pose un certain nombre de problèmes quant à la lecture du sens. C'est le cas de la gestion du *métalangage*, c'est-à-dire le langage utilisé pour structurer le dictionnaire. Le métalangage est aisément utilisable lorsqu'il s'agit de récupérer les catégories grammaticales des items mais certaines constructions de définitions sont difficilement interprétables sans compétence métalinguistique. Il existe un certain nombre de tournures de phrase pour lesquelles la lecture du sens n'est pas aisée comme *partie de*, *se dit de*, *on dit que* ou d'ambiguïtés que même un humain ne peut lever. Comment savoir que *en parlant* est du métalangage dans la définition de l'item ‹*aboyer*› issue de [Robert, 2000] « *Pousser son cri, en parlant du chien* » ou dans celle de ‹*anthropophage*›, « *Qui mange de la chair humaine en parlant de l'Homme* » ? Même si des solutions permettant de gérer en partie ces cas existent (cf. 2.3.5.1), nous utilisons diverses sources lexicales pour pallier de tels manques définitoires. Il s'agit de tempérer statistiquement les diverses incohérences locales. Ainsi, si une définition est mal formée (donc difficilement analysable correctement) une autre définition, mieux formée et provenant d'une autre source pourra corriger l'effet de la première.

Aucune source ne peut couvrir l'intégralité du lexique non seulement parce que celui-ci est en constance évolution mais surtout parce que rechercher, trouver et finalement décrire de façon systématique chaque terme de la langue est une tâche extrêmement difficile. Il faudrait, en effet, connaître l'ensemble des formes mais aussi l'ensemble des sens que peuvent prendre ces formes dans n'importe quel domaine d'activité, dans n'importe quelle structure sociale et quel que soit le niveau de langage. L'utilisation de multi-source permet donc aussi de maximiser la probabilité de récupérer une définition pour des termes qui pourraient être trouvés dans certains dictionnaires mais pas dans d'autres. Par exemple, ‹*liturgiste*› ne se trouve pas dans [Larousse, 2004] mais on le trouve dans [Robert, 2000].

5.1.4.2 Catégorisation des lexies : création d'acceptions

La figure 5.4 présente la structuration générale de la base sémantique lexicale. À partir de diverses sources, on peut obtenir des définitions pour un item lexical. Pour chaque définition, on crée une lexie qui est un objet qui regroupe les diverses informations contenues dans cette définition ainsi qu'un vecteur conceptuel calculé à partir du texte de la définition. Ces lexies doivent alors être catégorisées (regroupées par sens) afin de fabriquer les diverses ACCEPTIONS de l'item. Nous présenterons dans la suite les méthodes utilisées pour fabriquer les LEXIES à partir des définitions, les ACCEPTIONS à partir de ces LEXIES et les vecteurs conceptuels de ces items à partir de ces ACCEPTIONS.

Prenons un exemple, considérons trois sources, le dictionnaire *Hachette de la langue* (noté [HDL]) [Robert, 2000], le *Petit Larousse* (noté [LAR]) [Larousse, 2004] et l'*encyclopédie Club-internet* [82] (notée [CLUI]). Si nous prenons l'exemple de l'item ‹*botte*›, nous trouvons les définitions de la figure 5.5.

Il semble clair que les sens {1,4,7} peuvent se regrouper pour former le sens d'**amas**, les sens {2,5,8} pour former le sens de **coup** et les sens {3,6,9} pour former l'acception **chaussure**.

[82] http://www.club-internet.fr/encyclopedie

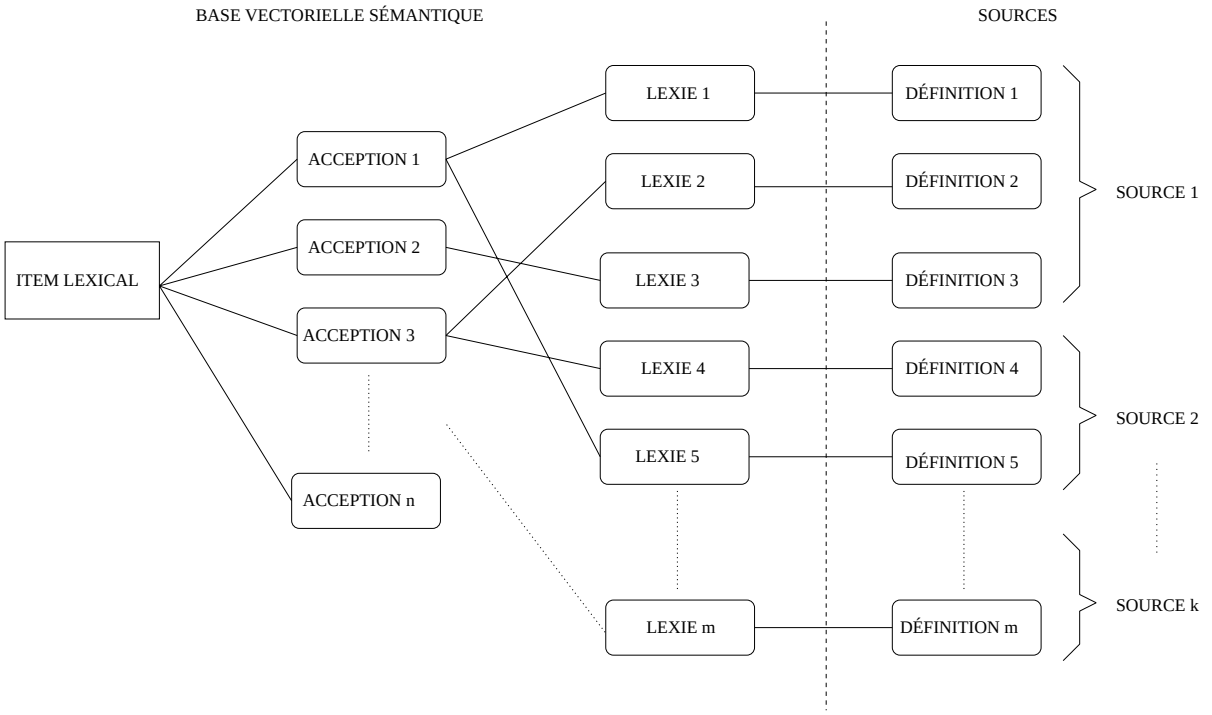


FIG. 5.4 – Organisation générale de la représentation du sens pour un item lexical.

La figure 5.6 présente l'organisation générale de l'item ‹*botte*› telle qu'elle serait si celui-ci était défini par ces neuf définitions.

5.1.4.3 Le particularisme des relations sémantiques

De la même manière, d'autres sources que les dictionnaires classiques peuvent être utilisées comme les thésaurus, lexiques terminologiques et surtout les dictionnaires de synonymes et d'antonymes (cf. chapitre 4). Notre approche se veut ainsi la plus holistique possible en considérant autant d'informations qu'il est permis d'acquérir sur les items et leurs acceptions.

L'introduction de sources concernant les relations sémantiques pose un autre problème. Dans les dictionnaires classiques, les lexicographes s'emploient à décrire aussi bien que possible l'ensemble des sens d'un terme. En revanche, la création de termes synonymes ou antonymes se fait, elle, uniquement par l'usage, de façon totalement ou du moins en grande partie naturelle. Un terme aura ainsi plus de synonymes pour un de ses sens utilisé fréquemment que pour un sens qui l'est moins. Par exemple, [Chazaud, 1979] présente dix synonymes pour le sens **amas** de ‹*botte*›, six pour celui de **chaussure** et seulement deux pour celui de **coup**. Ainsi, cette "description naturelle" se fait-elle en fonction de la fréquence d'utilisation pour les dictionnaires de synonymes ou d'antonymes et non pas de façon systématique comme dans les dictionnaires classiques.

De plus, des particularismes peuvent survenir pour certaines relations lexicales. On pense en particulier à l'antonymie dont certains termes peuvent avoir plusieurs antonymes correspondant au même sens, y compris dans le même type, mais avec un axe de symétrie différent. De même, pour la synonymie, le découpage en fonction des composantes connexes peut se révéler différent de celui des dictionnaires.

Nous avons donc affaire avec les relations sémantiques à deux phénomènes aux conséquences opposées dont il s'agit de tenir compte (cf. 4.2) : d'un côté un manque de description du sens de certaines sources qui n'est pas à considérer comme un déficit dictionnairique et de l'autre

botte.1 : #nf# Réunion de végétaux de même nature liés ensemble. (Une botte de paille, de radis, de fleurs) . *[HDL]*
botte.2 : #nf# En escrime, coup porté à l'adversaire avec un fleuret ou une épée. (Pousser, porter, parer une botte) (Botte secrète.). *[HDL]*
botte.3 : #nf# Chaussure de cuir, de caoutchouc ou de plastique qui enferme le pied et la jambe, parfois la cuisse. (Des bottes de cavalier) – Chaussure d'extérieur basse. (Botte d'hiver, de ski, de marche) . *[HDL]*
botte.4 : #nf# (néerl. bote, touffe de lin) . Assemblage de végétaux de même nature liés ensemble : (Botte de paille. Botte de radis.) . *[LAR]*
botte.5 : #nf# (#ethym-it# botta, coup) . Coup de pointe donné avec le fleuret ou l'épée . *[LAR]*
botte.6 : #nf# (p.-ê. de bot) . Chaussure à tige montante qui enferme le pied et la jambe généralement jusqu'au genou : (Bottes de cuir) . *[LAR]*
botte.7 : #nf# (néerl. (bote) « touffe de lin »). Assemblage de végétaux, de même sorte, tenus par un lien. (Une botte de paille, d'asperges.). *[CLUI]*
botte.8 : #nf# (ital. (botta) « coup »). En escrime, coup de pointe à l'épée ou au fleuret. (Porter une botte.) #fig# Attaque vive . *[CLUI]*
botte.9 : #nf# Chaussure montant jusqu'à mi-jambe ou jusqu'au genou, enfermant parfois une partie de la cuisse. (Des bottes de pêche, d'équitation.). *[CLUI]*

FIG. 5.5 – Exemple de définitions pour l'item lexicale ‹*botte*›.

certains sens qui seront décrits plusieurs fois.

5.1.5 Hypothèse V - Apprentissage permanent

Pour analyser un certain nombre de documents, en particulier les journaux, il est souvent nécessaire de savoir à quoi correspondent des néologismes, qui sont certaines personnalités ou encore quel est le domaine d'activité d'une entreprise. Par exemple, dans un texte, l'usage du nom de l'entreprise ‹*Arcelor*› indique vraisemblablement un contexte axé sur le traitement de l'acier. Les diverses sources et en particulier le Web par les serveurs d'informations (*Le Monde*[83], *Libération*[84], ...) présentent ces nouveautés et peuvent permettre d'acquérir les informations nécessaires à la fabrication des structures appropriées.

Nous l'avons vu, l'apprentissage des vecteurs se fait à partir des définitions et des vecteurs déjà existants. Il est difficile de penser que la base vectorielle deviendrait cohérente dès la première passe. Il est vraisemblable que des mots clés d'une définition n'aient pas encore été appris par le système lors de son analyse. La convergence des vecteurs vers une position quasi-stable ne pourra se faire que dans un nombre de cycles qu'il est sans doute difficile de déterminer à l'avance mais qui est fonction de l'ordre d'apprentissage des items et de leur définition.

Ce sont ces deux raisons, la difficulté à savoir si une base est véritablement stabilisée ainsi que la variabilité lexicale, qui nous ont conduits à considérer cette cinquième hypothèse forte : la base est en apprentissage permanent.

5.1.6 Hypothèse VI - double boucle

5.1.6.1 La double boucle en biologie

La double boucle est, comme le note Christophe Lecerf ([Lecerf, 1997], p. 177), un *élément structurel invariant* qui permet l'action sur son environnement et qui est un produit de cette

[83] http://www.lemonde.fr
[84] http://www.liberation.fr

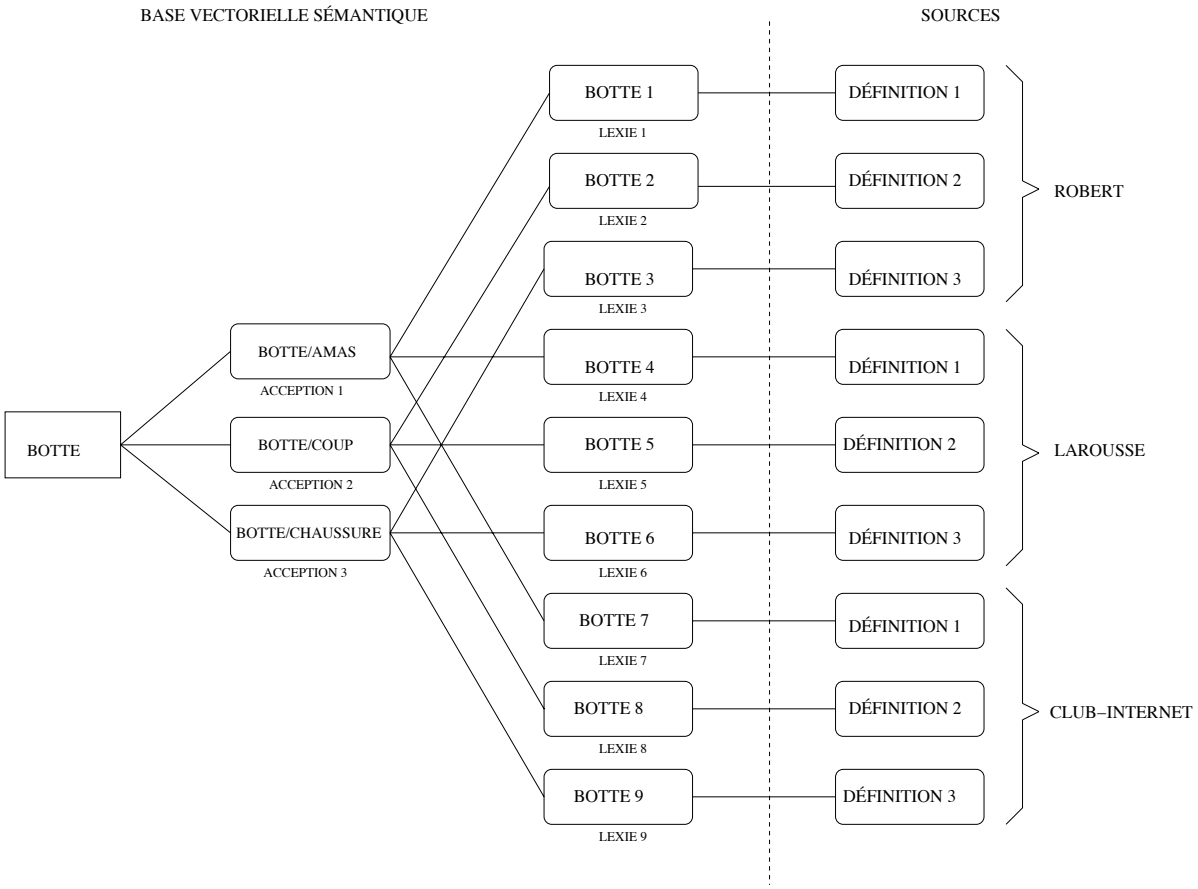


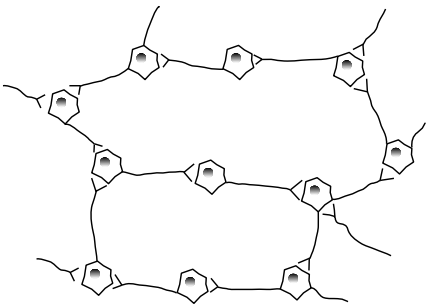
FIG. 5.6 – Exemple d'organisation générale de la représentation du sens pour l'item ‹*botte*›.

action. Chez un individu, on retrouve cette structure du niveau le plus bas, la cellule, au niveau le plus haut, l'organisme.

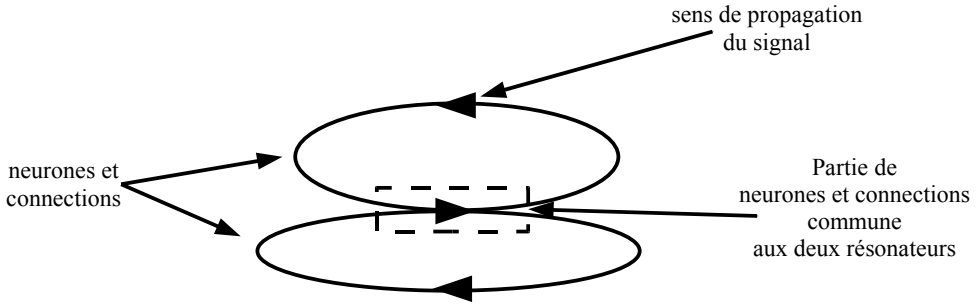
Au niveau des cellules Chez l'être humain, comme chez tous les vertébrés, le système nerveux central (SNC) est formé de l'encéphale (cerveau, cervelet, tronc cérébral) et de la moelle épinière. Les cellules qui le composent, les neurones, sont reliées entre elles grâce à des connexions synaptiques. Ces connexions sont unidirectionnelles et permettent grâce à un signal de nature électrique et chimique, d'exciter le ou les neurones connectés. Ce signal est appelé influx nerveux. Plus ces connexions sont utilisées, plus elles sont renforcées et plus elles ont des chances d'être utilisées.

Parfois, ces neurones et leurs connexions forment des circuits fermés. L'activation, depuis une source extérieure, d'un des neurones de la boucle se transmet au suivant et ainsi de suite, chaque cellule reçoit à son tour l'influx et le reproduit pour les suivantes. Le signal est ainsi auto-entretenu à partir d'une source extérieure par un effet de résonance dû à la boucle et envoyé vers les neurones en aval. On appelle une telle boucle une boucle auto-reproductrice.

Il peut arriver que deux boucles aient une partie de neurones et de connecteurs en commun (cf. figure 5.7). Dans ce cas, de la même manière que deux roues dans un engrenage, les flux circulent dans des sens contraires mais de façon conjointe. Il y a un auto-entretien des flux dans le double circuit, la double boucle. Ce signal sélectionné et stabilisé par l'effet d'une perception répétée est une image intériorisée de l'évènement perçu. On dit que cette double boucle forme un objet mental. L'évènement perçu est ainsi reconnu de plus en plus facilement grâce à cet auto-entretien.



(a) réseau de neurones en double boucle



(b) structure simplifiée du réseau de neurones en double boucle

FIG. 5.7 – La structure en double boucle de l'objet mental ([Lecerf, 1997], p. 41)

Au niveau du système nerveux central L'apprentissage à long terme se déroule en trois phases. Dans une première, un objet mental est construit à la suite de la perception d'un évènement. Sans entretien, le flux circulera entre les neurones durant quelques secondes avant de disparaître. C'est ce qu'il arrive lorsque vous devez retenir une adresse ou un numéro de téléphone le temps de le noter. Il s'agit de ce qu'on appelle la mémoire à court terme (MCT). On entre dans une deuxième phase si l'événement se répète suffisamment de fois pour stabiliser l'objet mental par l'augmentation des liaisons et donc du nombre d'influx à travers l'objet. Enfin, la dernière phase voit la consolidation de l'objet mental non plus grâce à l'apprentissage externe mais grâce à une stabilisation du flux provenant du circuit hypocampique. Nous assistons ainsi au remplacement d'influx dus à des expériences par des influx provoqués par un simple bouclage passant par l'hippocampe. Nous avons donc ici aussi une double boucle : l'une passant par l'environnement du SNC et l'autre, qui la double et qui renforce les objets mentaux qu'elle a créés, qui passe par l'hippocampe (cf. figure 5.8).

Au niveau de l'organisme Le SNC est une partie intégrante de l'organisme qui interagit avec lui grâce à deux structures qui jouent le rôle d'interface. La première, dîte *motrice*, lui permet d'agir sur l'organisme. C'est elle qui régit, entre autres, tous les mouvements qu'ils soient conscients ou non (réflexes). La seconde, le système *perceptif*, permet de percevoir les informations extérieures à l'organisme (ouïe, vue, toucher, ...) mais aussi les informations extérieures au seul SNC, les perceptions sur les postures, les mouvements du corps (perceptions proprioceptives).

Ainsi, comme le présente la figure 5.9, trois sortes de sources sont à l'origine des signaux qui se propagent au sein du SNC. Les premières, dues aux perceptions du monde extérieur, donnent lieu à des échanges avec l'extérieur du SNC et l'extérieur de l'organisme. Les deuxièmes, dues aux perceptions proprioceptives, ne donnent donc lieu qu'à des échanges externes aux SNC mais internes à l'organisme. Enfin la troisième concerne l'activité interne du SNC que nous avons déjà

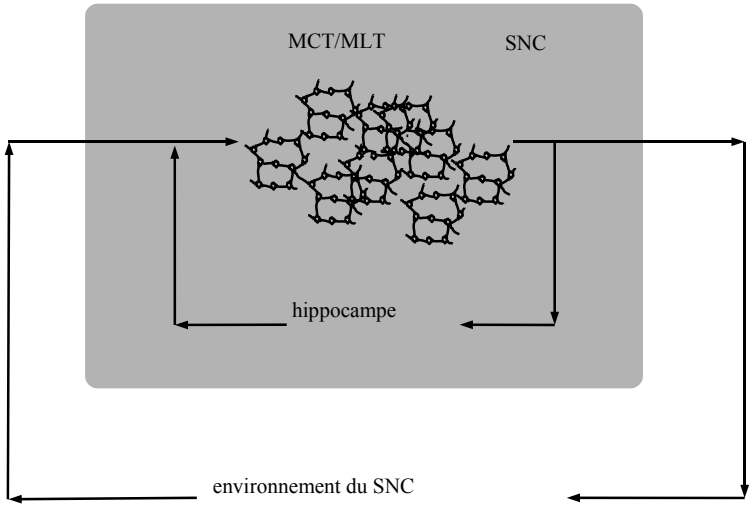


FIG. 5.8 – La double boucle de la mémoire à long terme (issu de ([Lecerf, 1997], p. 139))

présentée à la section précédente.

Au niveau de l'organisme et du monde extérieur La dernière double boucle concerne l'individu et son environnement. L'organisme effectue des échanges avec l'extérieur, son environnement. Il s'alimente en oxygène, en énergie et élimine déchets et chaleur. Parallèlement, l'organisme effectue aussi des échanges internes. L'oxygène est envoyé des poumons aux muscles, le dioxyde de carbone des muscles aux poumons, l'énergie extraite des aliments est acheminée aux muscles, ... Ces échanges de l'organisme à la fois avec lui-même et avec son environnement forment une double boucle (cf. 5.10).

5.1.6.2 La double boucle en apprentissage

Comme nous venons de le voir, la structure en double boucle est un élément structurel invariant que l'on retrouve à différentes échelles dans l'organisme. Dans un sens, cette organisation des flux en boucle permet d'agir sur le comportement tandis que dans le sens opposé, elle permet de transmettre l'adaptation de ce même comportement sur les éléments de la boucle. « *La mesure de l'adaptation se fait sur la régularité des échanges entre la structure et son environnement. Un comportement adapté se traduit par des flux réguliers propres à stabiliser la structure qui lui a donné naissance. Au contraire, un comportement inadapté se traduit par des flux irréguliers qui conduisent à la destruction progressive de la structure initiale.* » ([Lecerf, 1997], p. 178).

C'est cette régularité des échanges, non plus entre structure et environnement de l'organisme mais plutôt entre structure et environnement de la langue que nous cherchons à adopter ici. Nous avons montré dans le chapitre 4 que non seulement une base de vecteurs pouvait être améliorée grâce aux résultats obtenus des fonctions lexicales mais aussi que les résultats de ces mêmes fonctions sont nettement améliorés par l'utilisation d'informations lexicales et des vecteurs correspondants. Ainsi, non seulement les fonctions s'améliorent mais leurs résultats, exploités par la méthode d'apprentissage, servent à la construction de nouveaux vecteurs. Le système global s'enrichit de l'apport des fonctions qui elles-mêmes s'enrichissent de l'apport de l'ensemble du système. Ce principe admet une certaine ressemblance avec le principe de *double boucle* [Lecerf, 1997].

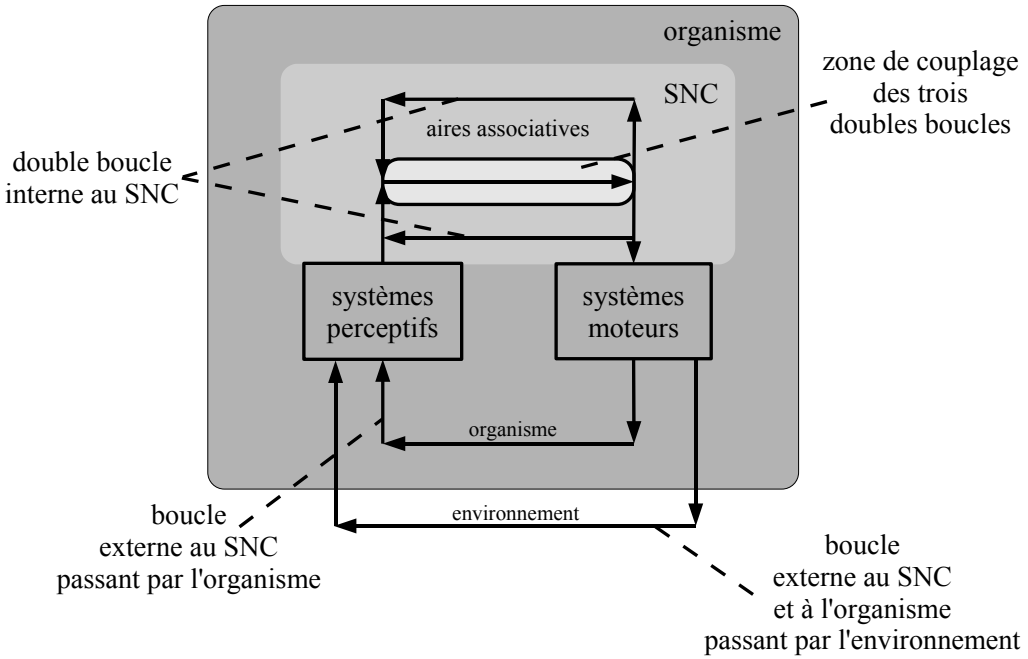


FIG. 5.9 – Les doubles boucles de l'organisme et le couplage sur l'extérieur (issu de ([Lecerf, 1997], p. 173))

Ce principe nous a amené à adopter cette sixième hypothèse forte, l'hypothèse dite de *double boucle*. Ces hypothèses, ainsi que des besoins de robustesse, de génie logiciel, de distribution sur plusieurs machines de gigaoctets de données nous ont conduits à adopter une architecture multi-agent.

5.2 Vers une société d'agents apprenants

Ces hypothèses sont, à nos yeux, fondamentales en ce qu'elles justifient l'architecture conceptuelle et implémentationnelle choisie pour notre système. Le choix d'une architecture à la fois distribuée et multi-agent s'explique aussi pour des raisons purement techniques comme une certaine limitation des systèmes séquentiels vis-à-vis des problématiques du TALN, les possibilités restreintes des ordinateurs actuels, certains bénéfices du point de vue du génie logiciel et une robustesse accrue. Avant d'expliciter en détail les raisons qui nous ont orientés vers cette architecture particulière, nous allons présenter les caractéristiques des systèmes multi-agents, en particulier distribués et les expériences précédemment menées qui ont allié SMA et TALN.

5.2.1 Les systèmes multi-agents

Les SMA sont issus de l'intelligence artificielle distribuée (IAD). L'IAD considère que si un système centralisé d'IA classique est souvent difficile à modéliser, un système basé sur des solutions locales aide à faire émerger plus simplement une solution globale. De fait, dans un SMA, l'intelligence se répartit dans les agents. Tout ou partie de cette intelligence est la conséquence de leurs interactions (phénomène d'émergence). Un agent est *une entité physique ou virtuelle (virtuelle dans notre cas) capable d'agir sur son environnement, qui peut communiquer directement avec d'autres agents, qui possède des ressources propres, qui est capable de percevoir son*

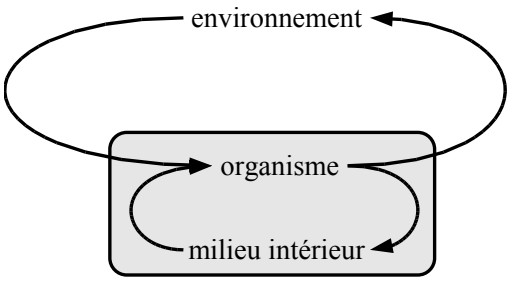


FIG. 5.10 – La double boucle de l'organisme (issu de ([Lecerf, 1997], p. 170))

environnement, qui possède des compétences et offre des services [Ferber, 1995]. Un agent est donc une entité capable d'agir rationnellement et intentionnellement en respectant ses propres buts et l'état courant de ses connaissances [Warren, 1998]. Généralement, certains principes fondateurs concernant la communication entre agents sont adoptés comme la sincérité de l'agent émetteur et la crédulité de l'agent destinataire.

Les SMA se caractérisent essentiellement par le type des agents mis en jeu (réactifs ou cognitifs), les moyens de communication (par mémoire partagée, diffusion de signal ou envoi de message) et par un contrôle centralisé ou distribué [Ferber, 1995, Warren, 1998].

5.2.1.1 Types d'agents

Même si, en pratique, la distinction entre les types d'agents n'est pas forcément très claire, on peut considérer qu'il existe deux types d'agents, les *agents réactifs* et les *agents cognitifs*. Si le mode d'organisation des agents réactifs est souvent comparé à l'organisation biologique, celle des agents cognitifs se rapproche d'une organisation de type social.

Agents réactifs Les systèmes à agents réactifs suivent l'hypothèse que le système peut avoir un comportement global intelligent sans que les agents soient nécessairement intelligents individuellement. La communication des agents réactifs se fait généralement par diffusion d'un signal dans l'environnement. Dans un système composé d'agents réactifs, il n'est pas rare de trouver des centaines voire des milliers d'agents. L'exemple type d'agent réactif est celui des fourmis dont les actions se coordonnent afin de résoudre des problèmes complexes tels que ceux de la recherche de nourriture, construction de nid, ... Nous présenterons les algorithmes à fourmis à la section 7.4.1.

Agents cognitifs Chaque agent possède sa propre base de connaissance, c'est-à-dire l'ensemble des informations et des savoir-faire nécessaires à la réalisation de sa tâche. Il est capable d'effectuer des raisonnements en fonction de la représentation qu'il se fait de son environnement, de ses connaissances et éventuellement de celles des autres agents. Généralement, la taille d'un tel système ne dépasse pas la centaine d'agents. La plupart du temps, les communications entre les agents cognitifs se font par envoi de messages qui peuvent éventuellement entraîner des tractations et des négociations.

5.2.1.2 Modes de Communication

Mémoire partagée La *communication par mémoire partagée* est basée sur l'existence d'une structure de données commune à tous les agents. Ces derniers accèdent, déposent ou modifient dans cette structure commune des informations. On appelle *tableau noir* une architecture dont

la structure commune est accessible directement par tous les agents. Une architecture de type *tableau blanc* est caractérisée par une structure commune cachée des autres composants et à laquelle seul un coordinateur a accès [Boitet & Seligman, 1994, Lafourcade, 1994]. Une architecture tableau blanc offre l'avantage d'éviter les problèmes de concurrence par rapport à une architecture tableau noir sans éviter un certain engorgement dû à cette centralisation autour d'une ressource principale. Contrairement aux autres moyens de communication, la mémoire partagée permet de connaître facilement et à tout moment de l'exécution les états intermédiaires du travail de résolution de la tâche.

Diffusion de signal La *communication par environnement*, par *diffusion de signal* ou par *stygmergie* est basée, comme son nom l'indique, sur des signaux laissés par les agents dans leur environnement. Ces signaux peuvent être perçus par les autres agents. Ce mode de communication est généralement utilisé par les agents réactifs. L'exemple type de communication par diffusion de signal est celui de la phéromone que laissent les fourmis pour indiquer une piste pour se rendre de la fourmilière à une source de nourriture.

Envoi de messages La *communication par envoi de messages* s'établit conformément à un protocole de communication prédéfini. La communication s'établit alors directement d'agent à agent (point à point) ou par diffusion à tous les agents ou une partie d'entre eux (broadcast).

5.2.1.3 Modes de contrôle

Contrôle centralisé Le *contrôle centralisé* est réalisé par un agent spécialisé qui supervise les tâches effectuées par les autres agents. Le principal avantage de ce mode de contrôle est de favoriser la cohésion du système. En revanche, il limite fortement l'autonomie des agents. Toutes les opérations sur les agents (ajout, suppression, modification) doivent être enregistrées par le superviseur. Ces opérations sont, la plupart du temps, transparentes pour les autres agents. Le principal défaut d'un tel système est d'être complètement centralisé et donc à la merci du moindre problème survenant à l'agent superviseur.

Contrôle décentralisé Avec un *contrôle décentralisé*, les agents prennent leurs décisions localement. Contrairement au contrôle centralisé, ce mode assure une meilleure fiabilité et extensibilité du système. En revanche, l'autonomie des agents peut entraîner une certaine incohérence des actions.

Généralement, une architecture de contrôle mixte est avantageuse. La cohérence globale est assurée par un contrôle centralisé tandis que des prises de décisions sont assurées au niveau local.

5.2.1.4 SMA distribués

Dans les SMA distribués les agents se situent sur différentes machines reliées entre elles par un réseau. Les communications entre agents se font alors de façon directe par l'intermédiaire de l'interface réseau. Si on peut estimer le temps d'accès[85] à un emplacement de la RAM d'un ordinateur à quelques nanosecondes (10^{-9}), le temps d'accès à une communication entre deux ordinateurs est, lui, de l'ordre du millième de seconde (10^{-3}). C'est-à-dire qu'une communication entre agents distants est à peu près un million de fois plus lente qu'une communication entre agents locaux.

[85] On peut considérer le débit comme très peu pertinent au regard de la faible taille des messages échangés entre agents.

Le principal défaut des SMA distribués est donc le temps de communication entre deux agents qui semble vraiment prohibitif. Cette considération doit toutefois être largement pondérée pour certains types d'agents. En effet, plus les calculs effectués par les agents sont longs, plus le temps de communication relatif est négligeable. C'est donc seulement à partir de calculs nécessitant des temps de l'ordre du millième de seconde que ce mode de communication n'est plus vraiment un obstacle aux performances globales du système.

5.2.2 Principaux avantages des systèmes multi-agents

5.2.2.1 Facilités dans la conception de systèmes complexes

Lorsque le système à modéliser est trop complexe, une analyse globale révèle rapidement ses limites. Il est souvent plus simple et rapide de chercher des solutions à des problèmes plus locaux selon le principe classique en informatique du « *diviser pour régner* ». Par exemple, pour réguler le trafic aérien, de trop nombreux paramètres et contraintes influent sur le résultat de l'ensemble pour qu'une approche globale permette une résolution fiable [Ferber, 1995]. En revanche, des solutions locales aident à résoudre élégamment et efficacement de telles difficultés. Les résultats obtenus sont souvent acceptables bien qu'ils soient au mieux difficiles et au pire impossibles à prouver théoriquement. L'approche multi-agent permet d'obtenir plus simplement des résultats sur des problèmes complexes en contournant les difficultés qu'auraient des algorithmes globaux pour les gérer.

5.2.2.2 Avantages pour le génie logiciel

Le génie logiciel est l'ensemble des techniques et méthodes mises en œuvre pour la conception de systèmes ou logiciels informatiques. Son évolution a conduit le développement des programmes vers une distribution de composants de plus en plus individualisés et autonomes. Le paradigme objet permet la réalisation de modules autonomes (les objets) capables d'interagir entre eux. L'un des principaux avantages est que ces objets peuvent être conçus par des équipes différentes. Les autres développeurs n'ont qu'à connaître les méthodes des objets pour utiliser celles dont ils ont besoin sans avoir réellement à savoir comment elles sont implémentées. L'objet peut alors être vu comme une *boite noire*.

Une architecture multi-agent offre quelques avantages supplémentaires. Dans le cas où la communication entre agents se fait à un niveau supérieur, indépendamment du langage informatique utilisé, un développeur peut créer des agents dans le langage informatique de son choix en fonction des tâches que cet agent doit effectuer. Par exemple, certains agents peuvent être développés en C dans le cadre d'opérations nécessitant une grande rapidité de calcul et d'autres agents en lisp pour accélérer le développement. Pour la même raison, l'intégration de programmes déjà existants est largement facilitée.

Un autre avantage des systèmes multi-agents est de faciliter le développement et le débogage d'un système. Prenons l'exemple d'un composant qui nécessite une phase de lancement de plusieurs secondes voire minutes pour charger des données. La phase de développement et de débogage des autres composants sera largement ralentie si ce composant doit être relancé à chaque intervention.

5.2.2.3 Robustesse

Les SMA distribués permettent de garantir une certaine robustesse des programmes. En effet, un système centralisé est plus sensible aux pannes, un programme va se dérouler normalement ou bien un bug va arrêter sa progression, il n'y a pas de demi-mesure. En revanche, si un agent connaît un problème technique, un autre peut prendre en charge le calcul grâce à une redondance

des compétences. La meilleure robustesse possible semble pouvoir être atteinte dans un système complètement décentralisé où tout agent peut aussi faire "naître" n'importe quel type d'agents en cas de problème spécifique. Il faut toutefois modérer nos propos en remarquant que mettre en œuvre une telle technologie est extrêmement compliqué en particulier dans le cas d'un système distribué pour ce qui concerne le lancement d'un agent depuis une machine distante (problèmes surtout dus aux systèmes d'exploitation hétérogènes).

5.2.3 SMA et TALN

Les Systèmes Multi-Agents sont utilisés depuis déjà longtemps dans le domaine du TALN. On constate que leur utilisation suit fidèlement leur popularité dans le monde de l'informatique. Au début des années 1970, ce sont des travaux concernant le TALN et en particulier la compréhension automatique de la parole qui sont précurseurs des travaux en IAD par l'invention du concept de tableau noir (Hearsay-II [Erman *et al.*, 1980]). Il faut ensuite attendre la fin des années 1980 avec HÉLÈNE [Zweigenbaum *et al.*, 1989] et CARAMEL [Sabah, 1990], au moment de son avènement comme domaine de recherche à part entière, pour commencer à voir apparaître de nombreux systèmes exploitant la technologie agent. Les années 1990 et le début du nouveau millénaire assistent à une certaine explosion du nombre d'expériences associant les deux domaines pour l'analyse syntaxique du Français avec HACTAR [Lebarbé, 2003] [Lebarbé, 2004] et TALISMAN [Stefanini, 2004], de l'Arabe avec MASPAR [Aloulou, 2003]), la correction d'erreurs avec CÉLINE [Menézo *et al.*, 1996] ou la traduction automatique [Pompidor & Vergnaud, 1995].

Cette section décrit chronologiquement différents systèmes alliant TALN et SMA en s'attachant particulièrement à essayer de comprendre pourquoi les concepteurs ont opté pour une approche orientée agents.

Dans les années 1970, lorsque le système Hearsay-II est mis au point, l'IAD prend son envol et n'a pas encore donné naissance aux SMA. Le vocabulaire utilisé alors, bien que souvent en commun avec la technologie agent, est parfois bien différent. On parle, par exemple, de modules, de sources de connaissances plutôt que d'agents [Ferber, 1995]. La terminologie ne commence à se stabiliser qu'au milieu des années 1990. Pour des soucis de simplification, nous présentons ici les divers systèmes dans le vocabulaire orienté agent exposé dans la section précédente (cf. 5.2.1). Nous préciserons toutefois le vocabulaire employé par l'auteur lorsque celui-ci nous paraît plus explicite.

5.2.3.1 Hearsay II

Hearsay-II est un système d'interrogation de bases de données par l'expression orale de requêtes [Erman & Lesser, 1975, Lesser & Erman, 1977] développé au cours des années 1970 à l'université Carnegie-Mellon (États-Unis d'Amérique). Le problème est de transformer le signal de la voix en question écrite puis de la transformer en requête. Le système comprend 13 agents ayant chacun un rôle bien défini. Chacun des agents a la possibilité d'inscrire ou de modifier sur un tableau noir (cf. 5.2.1.2) les solutions partielles (appelées ici hypothèses) qui permettent de reconstituer la question. Ce tableau est divisé en plusieurs parties relatives aux différents niveaux d'interprétation de la parole (signal, segment, syllabe, mot, séquence de mots et phrase). Afin de lever les ambiguïtés de chacun des niveaux, les agents génèrent, combinent, évaluent et modifient les hypothèses diminuant ainsi l'incertitude. L'intégration de ces expertises multiples et surtout leurs interactions font converger les solutions partielles vers une solution globale dans un véritable processus de coopération. Les rôles de chaque agent sont divers. On peut toutefois les classifier en quatre catégories :

– *agents poseurs d'hypothèses* : Ce sont des agents classiques qui permettent de passer d'un niveau d'interprétation au niveau suivant ou bien des agents qui produisent des solutions

possibles à un même niveau. Par exemple, l'agent MOW pose les hypothèses sur les mots à partir des syllabes possibles tandis que PREDICT évalue les termes qui peuvent suivre ou précéder une séquence de mots.
- *agents contrôleurs des ressources* : Cette catégorie est constituée de deux agents qui veillent à éviter une explosion combinatoire du nombre de solutions possibles pour les mots (WORD-CTL) et pour les séquences de mots (WORD-SEQ-CTL).
- *agents évaluateurs* : Ces agents évaluent des hypothèses posées en fonction des hypothèses fixées au même niveau ou à d'autres niveaux. L'agent RPOL, par exemple, évalue une à une toutes les hypothèses en fonction de l'ensemble des hypothèses du tableau.
- *l'agent STOP* : Cet agent décide d'arrêter le système si celui-ci commence à dépasser les ressources qui lui sont allouées ou s'il estime qu'une solution acceptable a été trouvée.

Le contrôle général du système est effectué par un agent spécialisé, le gestionnaire de tâches. À chaque cycle, il fixe pour chaque agent une priorité à partir d'heuristiques basées sur l'ensemble des informations disponibles. Les agents réalisent alors leurs actions dans l'ordre ainsi défini.

Au niveau résultats de l'expérience, le vocabulaire utilisé dans Hearsay-II est de l'ordre de 1000 mots ce qui est relativement important pour l'époque et fournit des interprétations correctes dans près de 90% des cas. Il est toutefois raisonnable de penser que la faible couverture du lexique ne permet pas de comparer la technique employée aux techniques actuelles.

Le besoin d'intégrer un SMA pour résoudre un problème de reconnaissance de la parole est venu du manque d'interactions entre chaque niveau dans un système séquentiel. Dans ce cas, chaque composant récupère en entrée les informations données par le composant précédent, les traite toutes et rend ses résultats au module suivant. Deux solutions sont alors possibles : soit un module élimine des solutions qui pourraient s'avérer par la suite correctes, soit il renvoie toutes les solutions possibles, ce qui peut grandement complexifier le problème (explosion combinatoire). Une interaction entre les niveaux peut permettre d'éviter de transmettre des solutions correctes hors-contexte mais non attestées en contexte. Au niveau morphologique, par exemple, ‹*livre*› peut être un *nom masculin* ou *féminin* ou la conjugaison d'un *verbe*. Dans la phrase « *Il livre la marchandise* », l'ambiguïté est immédiatement résolue au niveau syntaxique, il s'agit du verbe. Dans le cas d'Hearsay-II, la volonté d'utiliser un SMA est donc animée par l'idée d'utiliser diverses heuristiques partagées entre les agents dans un domaine où une heuristique générale est loin d'être triviale à définir.

5.2.3.2 Caramel

CARAMEL (Compréhension Automatique de Récits, Apprentissage et Modélisation des Échanges Langagiers) [Sabah, 1990] est un système multi-expert s'appuyant à la fois sur le parallélisme et sur une communication par tableaux noirs. Ses objectifs sont divers : analyse, compréhension, génération de textes. Ce système repose sur trois ensembles distincts :
- *les agents*. Cet ensemble, le plus important en nombre d'entités comprend les agents (nommés ici processus) nécessaires à l'analyse d'une phrase comme un analyseur sémantique, un analyseur syntaxique, un résolveur d'ellipses et même des agents agissant au niveau pragmatique comme un système de gestion des personnages ou un système d'interprétation des récits. On peut distinguer les agents *élémentaires* destinés à compléter des représentations existantes et les agents *complexes* qui correspondent à des regroupements d'agents élémentaires, qui construisent de nouvelles représentations et peuvent donner naissance à d'autres agents ;
- *les tableaux noirs*. Chaque processus possède son propre tableau noir qui correspond à la mémoire qu'il partage avec les autres ;
- *le superviseur* nommé SIROP constitue le dernier ensemble. Il réalise une synthèse de tous les tableaux noirs puis, en fonction des résultats, active les agents. La gestion au sein

des agents complexes se fait de la même manière, un sous-superviseur assure des tâches analogues à SIROP.

Le système CARAMEL est sans nul doute un modèle dont nous devons nous inspirer puisqu'il s'attaque lui aussi à de nombreuses thématiques du traitement automatique des langues. Son architecture à plusieurs niveaux (agents élémentaires et complexes) est à la fois originale et intéressante. On peut toutefois se demander si la gestion des communications via les tableaux noirs n'est pas parfois compliquée à réaliser.

L'utilisation d'un SMA est justifiée ici par deux raisons particulières : la première est, comme pour Hearsay-II, d'améliorer la solution globale grâce à la collaboration entre les agents et la deuxième est d'accélérer la résolution par l'utilisation du parallélisme de diverses machines d'un réseau.

5.2.3.3 Talisman

TALISMAN est un projet, mis en route depuis le milieu des années 1990 à Grenoble puis à Marseille par Marie-Hélène Stéfanini, qui porte essentiellement sur l'analyse du français écrit. En un peu plus d'une dizaine d'années, diverses applications ont été abordées comme l'indexation automatique, l'analyse morpho-syntaxique [Warren, 1998], la génération de textes, le traitement informatique de la langue portugaise dans le domaine législatif (projet national brésilien NALAMAS) et plus récemment, la prosodie pour les énoncés oraux [Stefanini, 2004].

L'utilisation de SMA est, ici encore, justifiée par la nécessité de palier certaines lacunes des systèmes séquentiels : « *L'origine de ce travail est la recherche d'une approche permettant une meilleure interaction entre les différents types de connaissances linguistiques afin de réduire la production d'ambiguïtés inhérente à tout système général d'analyse de la langue opérant de façon séquentielle.* » [Stefanini, 2004].

TALISMAN est une architecture multi-agent dont les communications s'effectuent par envoi de messages. Le système est gouverné par des lois qui jouent un rôle semblable à celui des superviseurs dans CARAMEL. On distingue :

- *les lois globales.* Elles permettent la gestion du système. Elles suivent, en particulier, les protocoles d'échanges de messages ;
- *les lois locales* : Elles définissent la manière dont peut se dérouler la coopération entre des agents pour une tâche précise.

Dans le cadre de l'analyse du Français plusieurs agents ont ainsi été mis en place :

- MORPH pour l'analyse morphologique ;
- SYNT pour l'analyse syntaxique ;
- NEG pour la détection des négations aux niveau morphologique et syntagmatique ;
- SEM pour analyser la structure sémantique d'une chaîne donnée.

5.2.3.4 HACTAR

HACTAR est un système conçu pour l'analyse syntaxique [Lebarbé, 2001, Lebarbé, 2003, Lebarbé, 2004]. L'un de ses principaux intérêts est de découper les rôles de chaque agent sur les structures analysées plutôt que sur des tâches d'analyse. Ainsi, les agents correspondent aux mots, aux chunks[86], aux segments et aux phrases. Lorsqu'un groupe d'agents d'un de ces niveaux d'interprétation estime que les informations qu'ils ont échangées leur permettent de fusionner (un groupe de mots pour former un chunk, un groupe de chunks pour former un segment, ...), l'ancien groupe disparaît. Il est à noter que ce point n'est pas habituel dans les SMA pour le TALN puisqu'il empêche toute interaction entre les niveaux contrairement à la philosophie habituellement constatée (Hearsay-II).

[86] Notion proche de celle de syntagme.

Le système HACTAR est basé sur le modèle *APA* (Anticipation par Perception Augmentée) [Girault, 1999] qui a la propriété caractéristique d'être lui-même un système multi-agent. Pour chaque agent, il faut distinguer son environnement externe (la somme des images que les autres agents donnent d'eux-mêmes) de l'environnement interne (une image filtrée en fonction des objectifs de l'agent d'une partie de l'environnement externe). Chaque agent du système est composé de trois modules. Le premier permet à l'agent de percevoir l'environnement externe, le deuxième lui permet d'interpréter cet environnement en fonction de son environnement interne et le dernier lui permet d'agir sur l'environnement externe. L'environnement joue donc un rôle de tableau noir.

Implémentationnellement, le système a été construit pour pouvoir fonctionner sur des machines multiprocesseur et chaque agent peut-être exécuté sur des machines différentes par le biais de liaisons TCP/IP. L'environnement est géré par un serveur, les communications entre lui et les agents (et donc ses modifications) se font par accès distants (sockets). Pour des raisons de performances, le modèle est moins respecté au niveau microscopique, c'est-à-dire à l'intérieur des agents. L'environnement interne n'est donc pas implémenté comme l'environnement externe mais comme une variable accessible par tous les sous-agents, chacun d'entre eux fonctionnant dans un thread JAVA.

Dans le cas du système HACTAR, l'utilisation d'un SMA est justifiée par une volonté de parallélisation des processus calculatoires et d'écarter tout modèle combinatoire.

5.2.3.5 Système de traduction automatique du Chinois dans un but pédagogique

Pierre Pompidor et Jean-François Vergnaud ont collaboré à Montpellier pour la conception d'un système multi-agent pour la traduction automatique du Chinois dans un but pédagogique [Pompidor & Vergnaud, 1995]. Leurs recherches cherchent à mettre en relation des connaissances hétérogènes provenant de plusieurs bases de données. L'application au chinois vise à montrer plus particulièrement aux étudiants comment utiliser les connaissances issues des ressources dont ils disposent (dictionnaires, recueil de grammaire, recueil de phrases types, ...).

Le système est basé à la fois sur des agents de type cognitif et des agents de type réactif. Le contrôle est opéré par un agent cognitif qui a pour rôle d'assigner les tâches à chacun des agents. Les communications entre les agents se font, elles, de façon directe.

Un prototype a été implémenté sur une machine UNIX, les agents sont des processus qui communiquent grâce à une file de messages. Le problème des communications que les auteurs soulèvent eux-mêmes parait crucial. En effet, il faut environ deux cents agents pour une simple phrase.

L'utilisation d'une architecture distribuée pour un système de traduction est novatrice. Cependant comme le soulignent les auteurs, la traduction d'une phrase nécessitant déjà l'action de 16 agents réactifs, quel sera le nombre d'agents créés et les temps de communication pour une traduction plus longue ?

5.2.4 Pourquoi adopter un Système Multi-Agent Distribué ?

Cette section expose les raisons qui nous ont poussées à utiliser une architecture SMA distribuée. Ces raisons sont dues non seulement à nos hypothèses de construction d'une base lexicale sémantique (cf. 5.1) et aux applications finales mais aussi à certaines considérations techniques facilement résolvables par une telle architecture comme le parallélisme et les limitations techniques des machines actuelles.

5.2.4.1 Raisons dues aux hypothèses

Même si une implémentation purement séquentielle d'un système respectant les hypothèses pourrait être possible, une perspective multi-agent facilite largement la tâche non seulement du point de vue implémentationnel mais aussi du point de vue conceptuel. Si les deux premières hypothèses posées, celles concernant plus particulièrement la représentation du sens (*représentation hybride du sens*, *utilisation conjointe des* ACCEPTIONS *et des* ITEMS LEXICAUX), peuvent facilement être compatibles avec n'importe quel outil de TALN et quelle que soit son architecture, les trois hypothèses suivantes (*génération automatique*, *multi-source*, *apprentissage permanent*) plaident largement pour une orientation multi-agent. Cela s'explique essentiellement par la perspective aussi holistique possible que nous souhaitons avoir. Il s'agit d'acquérir autant d'informations lexicales et thématiques qu'il est possible à partir de définitions issues d'un maximum de sources (cf. hypothèse IV, 5.1.6). Recueillir ces informations grâce à des agents indépendants scrutant qui des dictionnaires, qui des listes d'antonymes, qui des listes de synonymes, qui le Web semble être la solution la plus simple pour répondre à cette exigence.

Nous l'avons vu, les SMA distribués sont souvent beaucoup plus robustes (cf. 5.2.2) que les systèmes séquentiels centralisés classiques.

Nous avons retenu une approche totalement holistique, d'un côté en essayant de tenir compte du maximum d'informations qu'il est possible de trouver sur les mots et leurs usages et de l'autre en utilisant les résultats de tous les traitements effectués afin de les auto-améliorer (hypothèse VI, dîte de *double boucle*). La conception et l'implémentation de cette hypothèse est largement facilitée par une orientation multi-agent. Chaque agent modifie sa base de connaissances en fonction des informations lexicales qu'il rencontre ou qu'il déduit (cf. chapitre 4). Les autres agents qui constituent son environnement, bénéficient alors de ses améliorations et il bénéficie en retour des leurs (cf. 5.3.1.1). L'exploitation et l'apprentissage des vecteurs conceptuels sont ainsi fortement liés.

5.2.4.2 Raisons dues aux applications visées

Les applications visées par nos travaux sont variées et hétérogènes. Elles concernent des domaines tels que la traduction automatique, le résumé automatique, la recherche d'informations, l'extraction de connaissances, le suivi d'évènements, ... Ces applications bénéficient des informations de la Base Lexicale Sémantique et doivent donc être construites autour, comme des modules pouvant y être adjoints. Ces modules peuvent, eux aussi, être facilement combinés en vue d'applications mixtes comme la recherche d'informations multilingue ou la synthèse de textes écrits dans diverses langues pour de la veille technologique par exemple. Ainsi, l'exploitation de la base sera grandement facilitée par l'ajout de modules apportant tel ou tel service et facilement connectables entre eux, donc par l'ajout d'agents spécialistes de traduction, d'agents spécialistes de la recherche d'informations ou par un agent résumeur, pouvant éventuellement travailler ensemble. Ce sont ces raisons, dues aux applications visées, qui nous ont aussi amenés à adopter un système multi-agent.

5.2.4.3 Raisons techniques

Possibilité de distribuer sur plusieurs machines Les systèmes TALN ont toujours été consommateurs de ressources systèmes importantes dues aux données à stocker (la taille du lexique est d'au moins 100 000 entrées pour une langue comme le français) et aux calculs souvent lourds. Dans notre cas, par exemple, chacun des agents doit stocker un certain nombre de données (sa base de connaissances) dont la taille est en grande partie fonction de la longueur des vecteurs conceptuels. Pour donner une idée, l'agent correspondant à la base du français

occupe actuellement[87] une taille de l'ordre de 9 Go sur disque et 120 Mo de mémoire vive, on peut considérer que l'ajout d'une source dictionnairique classique (Larousse, Hachette, ...) est de l'ordre de 2 Go sur le disque pour une augmentation de la taille en mémoire vive d'environ 10 Mo. De même, chaque agent d'antonymie occupe environ 50 Mo de Mémoire vive. Même à notre époque, il est difficile d'imaginer l'utilisation de dizaines voire de centaines de tels agents sur une seule et même machine. Chaque agent peut se trouver sur une machine, ainsi, il pourra pleinement utiliser les ressources matérielles disponibles.

Génie Logiciel Nous l'avons vu dans la section 5.2.2.2, la modularité est une caractéristique principale des architectures à agents, en particulier l'ajout de module est facilité. Dans notre application, par exemple, il a été extrêmement facile de créer un agent d'analyse morpho-syntaxique qui n'est qu'une interface à l'analyseur SYGFRAN développé par Jacques Chauché (cf. 1.1.4.2). Un autre avantage important, est de faciliter la réunion de plusieurs systèmes pour n'en obtenir qu'un. Nous verrons dans le chapitre 8 la coexistence et la collaboration d'un système gérant le français et d'un autre gérant l'anglais en vue d'un outil de traduction.

La dimension développement a aussi été prépondérante dans le choix d'un système multi-agent. En effet, il aurait été encore plus compliqué de mettre en œuvre l'apprentissage sur des définitions si nous avions dû attendre à chaque modification de la procédure le rechargement complet de la base lexicale sémantique[88].

5.3 Le système Blexisma

Blexisma (*Base LEXIcale Sémantique Multi-Agent*) est un Système multi-agent dont l'objectif est d'intégrer tout élément lui permettant de créer, d'améliorer ou/et[89] d'exploiter une ou plusieurs Bases Lexicales Sémantiques dont l'architecture à trois niveaux (LEXIE, ACCEPTION, ITEM LEXICAL) a été décrite en 5.1. Le développement du moteur du système (caractéristiques des agents, macro-organisation, communications, ...) a été réalisé au cours de cette thèse. J'ai développé ensuite quelques agents qui ont chacun une fonction particulière (base de données, apprentissage, expert en fonctions lexicales, ...). Nous décrirons dans la section suivante 5.4 les divers agents implémentés en fonction de leur rôle.

Nous présentons ici les caractéristiques conceptuelles du système comme l'organisation interne des agents (agents plutôt cognitifs), leur organisation (par rôle et par langue) et leurs communications (notre vision les considère plutôt comme s'effectuant par envoi de messages bien que l'on puisse voir l'agent `base` comme un tableau noir). Nous nous intéressons enfin aux caractéristiques pratiques des agents (ou caractéristiques techniques), c'est-à-dire à la manière dont les agents sont implémentés et comment sont gérées leurs communications.

5.3.1 Caractéristiques Conceptuelles du système

Les caractéristiques conceptuelles du système décrivent l'organisation interne des agents et l'organisation globale du système. Cette vision est théorique, en ce qu'elle ne considère pas (ou peu) les caractéristiques techniques sous-jacentes que nous aborderons dans la sous-section suivante (cf. 5.3.2).

[87] En Juillet 2005

[88] Le chargement de la base met, suivant les caractéristiques de la machine (Athlon 1,4 GHz avec 768 Mo de RAM ou Sun huit processeurs à 900Mhz, 32 Go de RAM) et la charge supportée entre 30 secondes et plusieurs minutes.

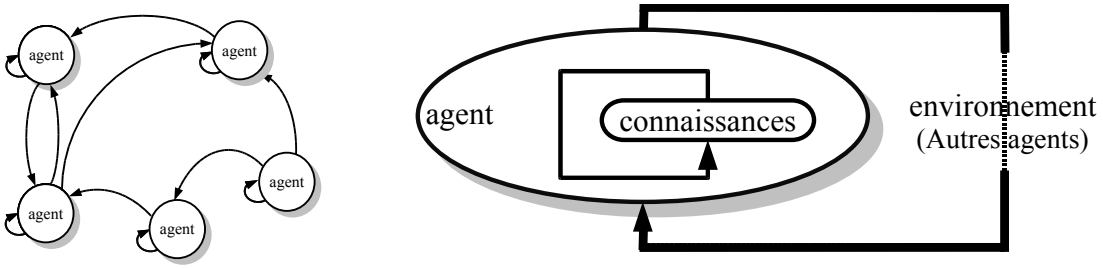
[89] Avec un objectif fort pour le *et* (cf. hypothèse VI, 5.1.6).

5.3.1.1 Agents

Nous examinons ici les caractéristiques (on peut aussi parler d'hypothèses) qui développent le caractère cognitif de nos agents, *i.e.* la manière dont ils "raisonnent". Cette partie présente aussi l'organisation sociale à trois niveaux (agent, rôle, langue) qui permet leurs interactions par leurs communications.

Vision récursive des agents Comme la plupart des Systèmes Multi-Agents (CARAMEL, HACTAR), notre système peut être vu à différentes échelles. Même si nous nous situons clairement dans une approche cognitive au niveau global de notre système, chaque agent peut être lui-même composé d'agents réactifs dont l'effet émergeant fera l'objet de transmission aux autres agents par envoi de messages. Par exemple, pour l'analyse d'un texte, un agent peut utiliser un système de fourmis (agents réactifs) sur l'arbre morpho-syntaxique correspondant afin de désambiguïser les feuilles et faire émerger le vecteur global du texte [Lafourcade, 2003] qui sera envoyé à l'agent demandeur.

Apprentissage par renforcement, double boucle Chacun des agents possède sa propre base de connaissance qu'il modifie au gré de ses expériences et de ses interactions avec les autres agents. À chaque requête, les agents tirent parti des informations reçues pour modifier leurs connaissances avant de répondre à la requête. Par exemple, l'agent d'apprentissage peut extraire d'une définition une liste d'antonymes. Il peut demander aux agents spécialistes de l'antonymie de lui fournir les vecteurs correspondants. Ces agents vont donc permettre une amélioration générale de la cohérence de la base. Parallèlement, ces agents vont utiliser les informations lexicales reçues du système pour modifier leurs méthodes de calcul [Schwab *et al.*, 2002b]. Ainsi, les agents peuvent fournir de meilleurs résultats. Le système global s'enrichit de l'apport des agents qui en retour s'enrichissent du système (cf. fig. 5.11). Cette caractéristique conceptuelle est, en fait, la représentation dans les agents de l'hypothèse VI (cf. 5.1.6).



(a) Organisation macroscopique d'un système et l'interaction des différents agents entre eux.

(b) organisation microscopique d'un système, la "vision du monde" d'un agent. Les données extérieures sont utilisées pour améliorer la propre base de connaissance de l'agent et l'agent améliore les données extérieures (double boucle)

FIG. 5.11 – La double boucle des agents dans blexisma

Expertises en concurrence Lorsqu'un agent est confronté à un problème, il peut demander à d'autres agents de l'aider. Ces agents lui donnent un certain nombre de solutions qu'il lui revient de choisir voire si possible de combiner. Par exemple, si un agent d'analyse sémantique rencontre un schéma particulier, il pourra demander aux agents experts en antonymie, hyperonymie ou méronymie si ce schéma peut les caractériser. De même, il peut demander un service à des agents spécialistes d'un même domaine. Dans les deux cas, l'agent demandeur considèrera toutes les

réponses obtenues et leur attribuera plus ou moins de crédit en fonction de la compétence de l'agent.

Unicité des agents Au cours de sa vie, chaque agent est unique. Plusieurs agents peuvent pourtant avoir un rôle identique, être spécialiste d'un même domaine. Par exemple, plusieurs agents peuvent être chargés de l'analyse du français ou être spécialistes de la même relation sémantique. Ces agents peuvent être conceptuellement différents, c'est-à-dire qu'ils utilisent chacun une méthode particulière de résolution du problème (l'analyse sémantique d'un texte peut se faire, par exemple, par propagation d'un vecteur conceptuel dans l'arbre morpho-syntaxique ou par émergence grâce à des agents réactifs de type fourmis) ou être simplement des clones (exactement le même code source). Dans ce dernier cas, l'unicité des agents est la conséquence de l'expérience différente acquise par chaque agent en fonction des données rencontrées ou des requêtes reçues. Il est par exemple difficile de savoir si un ordre particulier dans l'apprentissage des vecteurs leur assurerait une convergence plus rapide. Un agent d'apprentissage révise donc périodiquement de manière aléatoire chaque item de sa base. Un autre agent de ce type acquiert une expérience distincte puisqu'il analyse les items dans un ordre différent. Il en est de même des agents spécialistes des relations sémantiques puisqu'ils peuvent ne pas rencontrer les mêmes couples de termes ou les rencontrer plus ou moins souvent.

5.3.1.2 Organisation sociale

Dans Blexisma les agents sont connus par trois caractéristiques :
- *nom* : chaque agent est connu sous un nom qui est un identifiant unique ;
- *langue* : indique la "langue maternelle" de l'agent ou son indépendance vis-à-vis de toute langue si l'agent n'utilise aucune information lexicale ;
- *rôle* : le rôle décrit la fonction qu'il a dans le système (base, apprentissage, analyseur morpho-syntaxique, experts en fonctions lexicales, contextualiseur, ...). Le rôle ne présume en rien de la manière dont un agent réalise les actions qu'il est capable de faire (cf. *unicité des agents* 5.3.1.1)

On peut remarquer que cette organisation sous la forme *agent, langue, rôle* se rapproche d'*agent, groupe, rôle* [Gutknecht & Ferber, 1999] à la différence près que pour notre système nous n'avons pas considéré avec intérêt la possibilité pour un agent d'avoir plusieurs rôles ou d'appartenir à plusieurs groupes (plusieurs langues dans Blexisma).

5.3.1.3 Communications : envoi de messages

La communication entre les agents se fait par envoi de messages. Plusieurs types de communications sont possibles :
- *communication directe entre agents* : l'agent envoie directement le message à un autre agent qui exécute une action en fonction de son interprétation du message et envoie une réponse ou signifie son incompétence à y répondre de façon satisfaisante.
- *envoi du message à un ensemble d'agents* : un agent peut envoyer un message à tous les agents ayant un certain rôle, à tous les agents d'une langue donnée, à tous les agents d'une langue ayant un certain rôle et même à tous les agents. Comme pour une communication directe, les agents receveurs effectuent individuellement une action répondant à la requête et envoie éventuellement une réponse à l'agent émetteur.

À une échelle plus locale, on peut aussi considérer chaque agent `base` comme un tableau noir puisqu'il centralise une grande partie de l'information lexicale. Chacun des agents d'apprentissage modifie les objets sémantiques mais aussi indique à quelle date ces derniers ont été modifiés. Tous

ces renseignements peuvent ainsi être vus comme des informations indirectement destinées aux autres agents.

5.3.2 Caractéristiques techniques du système

Le système a été entièrement développé en Java afin d'assurer une relativement bonne portabilité sur des machines et des systèmes d'exploitation hétérogènes. Il repose essentiellement sur un référenceur qui gère l'existence des agents et leurs communications. Les agents, tout comme le référenceur, sont des programmes Java lancés indépendamment les uns des autres sur des machines reliées en réseau. Un fichier de configuration permet de connaître sur chaque machine où se situe le `référenceur`, c'est-à-dire sur quelle machine et sur quel port il écoute.

Dans l'ensemble, le noyau du système est relativement peu important puisqu'il se réduit à deux packages (le package gérant le `référenceur` et celui gérant les agents) et six classes. Le gros du développement s'est fait sur les agents en particulier (cf. 5.4) bien qu'il ait largement influencé l'architecture du noyau.

5.3.2.1 Le référenceur

La gestion des agents se fait par un superviseur, le `référenceur` (Broker). C'est lui qui valide l'existence d'un agent, qui met les agents en relation et qui gère leurs communications. Lors de sa création, chaque agent adresse au superviseur son identifiant, son rôle, et sa langue, ainsi que la machine et le port sur lequel il écoute. Le superviseur accepte la création de l'agent si aucun autre agent encore actif ne présente cet identifiant. Lorsque qu'un agent a besoin d'envoyer un message à un agent dont il ne connaît pas les coordonnées, il demande au référenceur qui les lui indique. Le référenceur a donc un double rôle, un rôle de superviseur qui permet l'ajout d'agents dans le système et un rôle d'annuaire pour mettre en relation les agents.

Le package *cvbroker* comprend toutes les classes implémentant le référenceur et la gestion des communications entre les agents.

5.3.2.2 Agents

Les agents sont développés en Java pour être compatibles avec le système. Le package *agents* comprend les classes permettant de créer des agents et d'assurer leurs communications. Les agents développés doivent hériter d'une classe abstraite *Agent* qui implémente les connexions de base avec le référenceur. La gestion des messages reçus par l'agent est gérée par un module héritant de la classe *RequestProcesing*.

Il est possible sans grande difficulté d'ajouter des programmes écrits dans des langages différents. Si une tâche nécessite une vitesse accrue on peut utiliser le C comme c'est le cas pour l'agent `proximeur` (cf. 5.4.5) ou pour l'analyseur morpho-syntaxique SYGFRAN (cf. 5.4.3) puis les interfacer par un agent.

5.3.2.3 Communications

Les agents pouvant se situer sur des machines distantes, les communications se font par liaisons TCP/IP sur le mode client-serveur. Les messages sont des classes Java expédiées via des sockets.

Deux types de communications sont possibles, la *communication directe entre agents* et la *communication indirecte grâce au référenceur*.

– *communication directe entre agents* : comme pour le *point à point* (*peer to peer*), un agent demande au superviseur l'adresse particulière d'un agent puis communique directement avec lui. Ce type de liaison est intéressant dans le cas de communications fréquentes entre

deux agents comme c'est le cas, par exemple, entre un agent `contextualiseur` qui a besoin de récupérer les objets lexicaux (cf. 5.4.2) que stocke la `base lexicale sémantique`.
- *communication indirecte grâce au référenceur* : le message est envoyé au superviseur qui se charge de l'envoyer à son ou ses destinataires (tous les agents du système ou les agents d'un groupe et/ou d'une langue). Chaque agent qui a reçu le message y répond en apportant une réponse ou en signifiant sa non compétence dans ce domaine.

5.4 Blexisma : agents implémentés et exemple de coopération

Nous présentons dans cette section quelques-uns des principaux agents déjà implémentés et disponibles sur le Web à l'adresse `http://www.lirmm.fr/~schwab` puis un exemple de leur coopération dans le cadre de l'apprentissage. On distingue les agents d'*apprentissage* qui sont les agents dont le rôle[90] est de créer des objets lexicaux à partir d'informations extraites de diverses sources (`agents dictionnairiques`, agents `spécialistes de relations lexicales`, agents `extracteurs de signature lexicale` sur le Web, ...) des autres agents dont le rôle est d'aider les agents d'apprentissage dans leur tâche. Nous donnons des données chiffrées et les caractéristiques techniques sur Blexisma en annexe H.

5.4.1 Gestionnaire de la Base Lexicale Sémantique

Ces agents, que nous appelons parfois aussi `base`, conservent et permettent l'accès et la modification
- de toutes les données brutes extraites des diverses sources (les définitions) ;
- des définitions formatées par les agents d'apprentissage ;
- des objets lexicaux (ITEMS LEXICAUX, ACCEPTIONS, LEXIES) conformément à l'architecture à trois niveaux présentée en 5.1.

5.4.2 Contextualiseur

Cette sorte d'agents implémente la fonction de contextualisation forte présentée en 2.3.6.

5.4.3 Analyseur morpho-syntaxique : SYGFRAN

Cet agent sert d'interface au programme SYGFRAN présenté en 1.1.4.2.

5.4.4 Analyseur Sémantique

Avec l'aide de l'agent d'analyse morpho-syntaxique et l'aide de l'agent contextualiseur, l'agent d'analyse sémantique calcule le vecteur correspondant à un texte. Ce texte, dans le cas d'un apprentissage est une définition de dictionnaire dont l'agent d'apprentissage souhaite le vecteur (cf. fig. 5.12). Deux méthodes de calcul existent : remontée-redescente présentée en 2.1.6 et une méthode par propagation de vecteurs sur l'arbre morpho-syntaxique grâce à des algorithmes "fourmis" (cf. 7.4).

5.4.5 Proximeur

Ce type d'agents sert d'interface à la méthode de voisinage (cf. 2.1.3.2) qui pour des raisons d'efficacité (rapidité) est implémentée en C. En effet, pour calculer le voisinage d'un vecteur V, il faut passer en revue tous les vecteurs des ITEMS LEXICAUX de la base avant de classer les plus

[90] Nous notons habituellement les agents en les décrivant par leur rôle de la manière suivante : `agent`.

proches de V. Cet agent a essentiellement une utilité de vérification et d'évaluation des vecteurs conceptuels d'une base.

5.4.6 Catégoriseur

Ce type d'agents catégorise les LEXIES pour fabriquer les ACCEPTIONS à partir des méthodes exposées dans [Jalabert, 2003, Jalabert & Lafourcade, 2004a] puis les ITEMS LEXICAUX (cf. 5.1.1).

5.4.7 Apprentissage

Les agents d'apprentissage sont les agents qui peuvent modifier la `base lexicale`, ils regroupent, entre autres, les agents apprenants sur les définitions issues de dictionnaires classiques (*agents dictionnairiques*), les agents spécialistes des fonctions lexicales (*agent synonymie, agent antonymie, agent hyperonymie, ...*) et les agents exploitant les signatures thématiques.

5.4.7.1 Agent dictionnairique

Les agents de ce type gèrent l'apprentissage des vecteurs conceptuels. Ils sont directement aidés dans cette tâche par des agents d'analyse sémantique ainsi que par les agents extracteurs de définitions.

5.4.7.2 Agent expert en relations sémantiques

Ces agents sont des experts des relations sémantiques comme la synonymie, l'antonymie, l'hyperonymie ou toute autre fonction lexicale. Ils implémentent les fonctions décrites au chapitre 4 et 6.

5.4.7.3 Agents extracteurs

Agent extracteur de définitions Ces agents ont pour rôle de récupérer les définitions correspondant à des items et de les fournir à l'agent d'apprentissage.

Agent extracteur de relations lexicales Ces agents parcourent des textes divers et variés (liste de synonymes, d'antonymes, textes du Web, définitions issues de dictionnaires, ...) à la recherche de couples en relation lexicale (cf. 4).

5.4.8 Exemple d'interaction entre agents, apprentissage du vecteur d'un item

Dans cette section, nous illustrons la collaboration entre agents par l'exemple de la fabrication des objets lexicaux ITEM LEXICAL, ACCEPTION et LEXIE d'un terme quelconque. Le schéma 5.12 présente les interactions entre quelques-uns des agents présentés dans la section précédente.

Un `agent dictionnairique` demande à un agent `extracteur de définitions` de lui fournir les définitions de cet item qu'il a récupérées en explorant le Web ou des dictionnaires à usage humain *(1)*. Le texte de chacune des définitions est prétraité et envoyé pour stockage à l'agent `base` *(2)*. Chaque définition formatée est donnée à l'`analyseur sémantique` *(3)* qui, à partir de l'arbre morpho-syntaxique obtenu grâce à l'`analyseur morpho-syntaxique` *(4)-(5)*, calcule le vecteur conceptuel correspondant *(9)*. L'agent `contextualiseur` (lui-même aidé par `base` *(7)*) et éventuellement des agents `experts en relations sémantiques` *(10)* collaborent avec lui dans cette tâche *(6)-(8)*. L'agent `apprentissage` récupère chaque vecteur des définitions. Les agents `experts des fonctions lexicales` lui permettent de compléter ses informations *(11)* - *(12)* pour construire les *lexies* qu'il fournit alors à la `base lexicale` *(13)*. Parallèlement

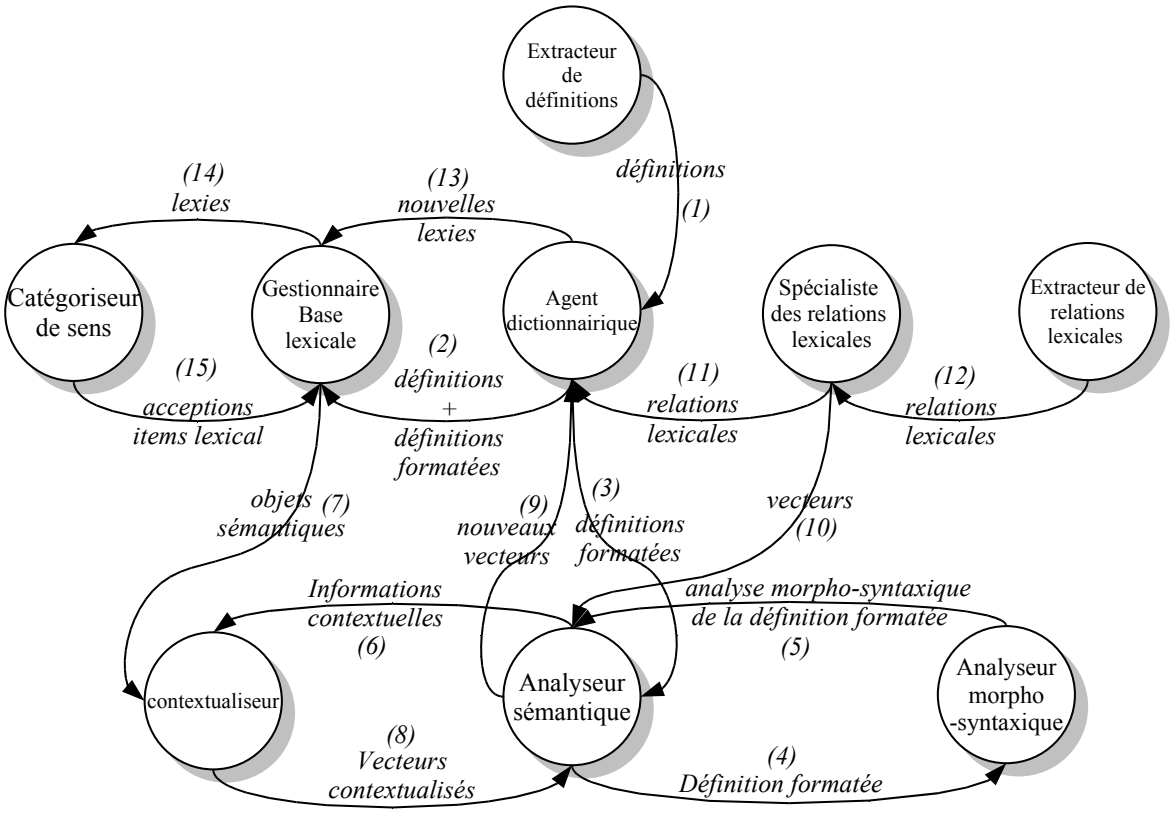


FIG. 5.12 – Organisation macroscopique du système Blexisma au cours d'une analyse sémantique.

à ces opérations, l'agent `catégoriseur` explore la base et fabrique les objets lexicaux ACCEPTION à partir des LEXIES et ITEM LEXICAL à partir de toutes les données dans les ACCEPTIONS *(14)-(15)* (regroupement des informations lexicales et somme des vecteurs conceptuels de chaque ACCEPTION).

5.5 Conclusions du chapitre

Nous avons présenté dans ce chapitre les six hypothèses de départ que nous avons adoptées dans le but de fabriquer une base de données permettant la représentation et l'exploitation du sens des item lexicaux.

La première, *représentation hybride du sens par une approche combinant approche thématique (vectorielle) et approche lexicale (relations sémantiques externes)* a pour but de palier la limitation des vecteurs conceptuels dans la modélisation des fonctions lexicales, de permettre d'allier la précision des réseaux sémantiques au rappel des vecteurs conceptuels et d'assurer une certaine adéquation au modèle cognitif du cerveau humain pour le traitement du langage. Sa première conséquence est que le sens est représenté dans notre base par des objets lexicaux composés d'un vecteur conceptuel et d'informations lexicales comme la morphologie, la fréquence en usage, les Fonctions Lexicales, ... Le sens de chaque terme du lexique est représenté par un tel objet appelé ITEM LEXICAL.

La deuxième hypothèse consiste à tenir compte des *relations sémantiques internes* (polysémie) afin de ne pas être confronté à l'émergence de combinaisons de sens non pertinentes dans le cadre d'une analyse sémantique (hybridation de vecteurs). Cette hypothèse nous a amenés à

créer des objets lexicaux appelés ACCEPTIONS correspondant au sens d'un terme.

L'hypothèse III est l'hypothèse de *génération automatique* des ACCEPTIONS. Nous sommes partis du principe qu'il était impossible de créer autant d'objets ACCEPTIONS manuellement et qu'il fallait donc trouver un moyen pour automatiser cette tâche. Cette automatisation se fait à partir d'un noyau réduit d'ACCEPTIONS indexées manuellement (un millier environ) et d'informations extraites de sources hétérogènes comme des dictionnaires classiques, des dictionnaires de synonymes, d'antonymes, de sites Web, ... Un troisième objet lexical est défini par cette hypothèse : la LEXIE qui regroupe toutes les informations pouvant être extraites d'une définition.

La quatrième hypothèse fondatrice est de réaliser une *analyse multi-source* afin de palier les problèmes dus aux manques définitoires (problèmes quant à la couverture du lexique et au métalangage).

La cinquième dont l'intérêt est de permettre la mise à jour régulière de la base ainsi que la stabilisation des données est l'hypothèse d'*apprentissage permanent*.

Enfin, la dernière hypothèse, l'hypothèse de *double boucle* est une généralisation à l'ensemble des agents des caractéristiques des Fonctions Lexicales présentées au chapitre 4. En effet, nous avons alors montré qu'une base de vecteurs était améliorée par les fonctions lexicales mais aussi que les vecteurs résultats des fonctions sont nettement plus pertinents grâce à l'utilisation d'informations lexicales et des vecteurs correspondants. Ainsi, la fonction s'améliore mais son résultat, exploité par la méthode d'apprentissage, sert à la construction d'un nouveau vecteur correspondant à ce mot. En généralisant, le système global s'enrichit de l'apport de chacun des agents qui eux-mêmes s'enrichissent de l'apport de l'ensemble du système.

Nous avons ensuite présenté comment ces hypothèses, les applications hétérogènes visées ainsi que des caractéristiques techniques nous ont amenées à adopter une architecture multi-agent dont nous avons présenté les caractéristiques conceptuelles (pour les agents *vision récursive* ; *apprentissage par renforcement, double boucle* ; *expertises en concurrence* et *unicité des agents* et pour les communications, *envoi de messages*) et techniques (basé sur un référenceur, distribué, communications par liaisons *TCP/IP*). Le système multi-agent proposé, appelé Blexisma, a pour but d'intégrer tout agent pouvant permettre de créer, d'améliorer ou/et[91] d'exploiter une ou plusieurs Bases Lexicales Sémantiques reposant sur une architecture à trois niveaux (LEXIES, ACCEPTIONS, ITEM LEXICAL). Nous avons enfin présenté dans ce chapitre les différents agents déjà implémentés ainsi qu'un exemple de leur interaction dans le cadre de l'acquisition d'informations sémantiques et de leur exploitation pour fabriquer des objets lexicaux[92].

Dans la suite de notre thèse, nous allons nous attacher à montrer comment nous exploitons Blexisma et ses divers agents dans le cadre du grand réseau lexical (ITEMS LEXICAUX) et infra-lexical (ACCEPTIONS voire LEXIES) induit par les informations lexicales contenues dans les OBJETS LEXICAUX.

[91] Avec un objectif fort pour le *et* (cf. hypothèse VI, 5.1.6).

[92] Une implémentation est accessible en ligne à l'adresse http://www.lirmm.fr/~schwab.

6

Vers un grand réseau lexical et infra-lexical

DANS ce chapitre, nous cherchons à établir quelles relations il serait utile de modéliser dans la base lexicale sémantique (BLS) dans le cadre de l'analyse d'un énoncé. Nous présentons la liste des Fonctions Lexicales d'Analyse (FLA) que nous avons établie afin de bénéficier de connaissances linguistiques et de connaissances du monde. Nous étudions ensuite en détail la question de l'hyperonymie et de l'hyponymie en montrant le rôle fondamental qu'elles jouent dans le cadre de la construction des vecteurs conceptuels. Nous présentons les caractéristiques du grand réseau lexical induit par la structure de la BLS et nous évoquons les pistes à suivre pour le créer le plus automatiquement possible. Nous faisons enfin le bilan des FLA afin de comprendre lesquelles sont à caractère purement lexical (basées sur des relations entre objets lexicaux) et lesquelles sont à caractère à la fois lexical et thématique. Nous montrons alors comment le réseau peut aider à modéliser des fonctions lexicales de construction et des fonctions lexicales d'évaluation pour chacune des FLA.

Sommaire

DANS le chapitre précédent, nous avons présenté les différentes hypothèses que nous considérions pour construire une base lexicale sémantique. La représentation des objets lexicaux qui forment cette dernière est fondée sur une représentation hybride du sens combinant une approche thématique, les vecteurs conceptuels, et une approche lexicale, les relations lexicales externes.

Ces relations, modélisées sous la forme de Fonctions Lexicales d'Analyse, peuvent aider à la résolution de problèmes posés dans le cadre de l'analyse sémantique d'un énoncé comme, par exemple, la désambiguïsation lexicale, la résolution d'anaphores ou de références. Dans ce chapitre, nous cherchons à établir quelles relations il serait utile de modéliser à cette fin dans la base lexicale sémantique. Nous précisons ainsi les notions de connaissances linguistiques et de connaissances du monde pour montrer en quoi elles sont importantes pour cette tâche. Nous présentons la liste des Fonctions Lexicales d'Analyse (FLA) que nous avons établie à partir de relations plutôt de type linguistique, les fonctions lexicales de Mel'čuk et de relations plutôt de type connaissances du monde issues d'un modèle de représentation des connaissances d'un texte, les relations d'UNL (*Universal Networking Language*).

Ces relations forment le grand réseau lexical induit par la structure de la BLS présentée dans le chapitre précédent. Nous en présentons les caractéristiques et nous évoquons les pistes à suivre pour le créer le plus automatiquement possible.

Nous abordons ensuite la question de l'hyperonymie et l'hyponymie en montrant le rôle fondemental qu'elles jouent dans le cadre de la construction des vecteurs conceptuels. Nous montrons en particulier l'existence d'un horizon lexical au-dessous duquel les vecteurs des hyperonymes sont inclus dans les vecteurs de leurs hyponymes et au-dessus duquel le phénomène est inversé. Nous mettons en évidence l'influence des définitions et de la hiérarchie du thésaurus sur cet horizon. Nous nous posons ensuite la question de la possibilité de modéliser des fonctions lexicales de construction d'hyperonymie et d'hyponymie et nous montrons qu'elles existent déjà de façon émergente dans une analyse sémantique de définitions mais qu'en revanche, l'existence de l'horizon lexical empêche de modéliser des fonctions lexicales d'évaluation pour l'hyperonymie et l'hyponymie uniquement à l'aide de vecteurs.

Cette dernière constatation met en évidence les limites, déjà vérifiées par ailleurs, des vecteurs conceptuels à modéliser complètement certaines fonctions lexicales. Nous faisons alors le bilan des Fonctions Lexicales d'Analyse et nous essayons de comprendre lesquelles sont à caractère purement lexical et lesquelles sont à caractère à la fois lexical et thématique. Les premières ne peuvent être modélisées que grâce aux rapports symboliques que les items lexicaux entretiennent entre eux tandis que les secondes peuvent être modélisées en partie à l'aide de vecteurs conceptuels et en partie grâce à des informations de nature lexicale. Nous montrons alors comment le réseau lexical peut aider à modéliser des fonctions lexicales de construction et des fonctions lexicales d'évaluation pour chacune des FLA.

6.1 Retour sur les fonctions lexicales pour l'analyse

Nous avons vu dans les chapitres précédents que les fonctions lexicales pouvaient aider à capturer le sens des items lexicaux. Notre principal objectif est d'améliorer l'analyse sémantique c'est-à-dire d'obtenir une représentation du sens d'un énoncé quelconque qui soit la meilleure possible[93]. Dans cette section, nous cherchons ainsi à faire le point sur les fonctions lexicales d'analyse afin de préciser lesquelles il serait nécessaire de représenter dans notre base.

6.1.1 Relations importantes en vue d'une analyse sémantique de textes

6.1.1.1 Connaissances lexicales et connaissances du monde

L'existence d'une distinction entre connaissances lexicales (CL) et connaissances du monde (CM) fait l'objet d'un grand débat en particulier depuis le début des années 1980. Kornél Bangha qui a fait sa thèse sous la direction d'Alain Polguère, dresse d'ailleurs un état de l'art intéressant sur la question [Bangha, 2003]. Certains, comme John Haiman [Haiman, 1980], considèrent qu'il n'existe aucune distinction tandis que la linguiste Wierzbicka dont les travaux ont été évoqués au premier chapitre (section 1.4.1.2) considère qu'elles devraient pouvoir être complètement séparées. Dans cette thèse, nous adoptons une position intermédiaire proche de celle de Kornél Bangha ([Bangha, 2003], p. 25). Ainsi, nous considérerons que les connaissances peuvent être divisées en trois parties :

1. *une grande partie des CM ne sont pas directement lexicalisées.* Elles ne sont donc pas des CL. Par exemple, un humain peut connaître tel ou tel point de géographie (où se jette le Rhône, où est New York ?), d'Histoire (Comment est mort Ravaillac, quand ?) ou de la vie quotidienne (quel est l'ordre de prix d'une deux-chevaux verte, quelle est la couleur d'un éléphant ?) mais ces informations ne sont pas lexicalisées et seul un énoncé peut les exprimer ;
2. *certaines CM sont directement lexicalisées.* C'est, comme nous l'avons vu dans la section précédente, le cas de la relation d'hyperonymie. La CM « *une chaise est un siège* » est retranscrite en langue par le fait que ‹*siège*› est hyperonyme de ‹*chaise*› qui est une CL ;
3. *certaines CL ne peuvent pas être considérées comme une lexicalisation des CM.* C'est le cas, par exemple pour le français, du genre grammatical ainsi, les items lexicaux ‹*voiture*› et ‹*piscine*› sont féminins ce qui ne correspond à aucune information sur les objets voiture et piscine.

6.1.1.2 Pourquoi ne nous limitons-nous pas aux fonctions lexicales de Mel'čuk ?

Nous avons cherché à savoir quelles relations seraient importantes à modéliser pour permettre une analyse sémantique fine des textes. Il a été rapidement clair que nous ne pouvions pas nous limiter aux fonctions lexicales de Mel'čuk dans ce cadre.

En effet, les fonctions lexicales sont créées selon une visée linguistique, pour permettre de savoir quel est le terme à utiliser avec un autre terme dans un certain contexte. Elles n'ont donc pas été conçues pour décrire le monde lui-même ni les relations de nature plus pragmatique que les termes entretiennent entre eux. On peut ainsi considérer que les fonctions lexicales de Mel'čuk se limitent à des connaissances lexicales et elles ne modélisent donc pas certaines relations entre items qui pourraient permettre de résoudre les différents problèmes d'ambiguïté posés lors d'une analyse sémantique. Par conséquent on ne trouve pas chez Mel'čuk des fonctions

[93] Nous reviendrons dans le prochain chapitre (section 7.2) sur les informations qu'il nous paraît important d'obtenir lors de l'analyse sémantique d'un énoncé (ACCEPTIONS utilisées, anaphores, identité, rattachements prépositionnels, chemins interprétatifs et instanciation des fonctions lexicales).

lexicales modélisant la méronymie, qui permettent de savoir que New York est une ville ou que le ‹*18 brumaire*› a un rapport avec ‹*Napoléon*›[94].

Nous avons cherché à savoir quelles informations il était nécessaire de posséder dans le cadre d'un tel processus. Pour cela, nous avons étudié un modèle de représentation des connaissances d'un texte : UNL.

6.1.1.3 Le projet UNL

Le projet UNL (*Universal Networking Language*)[95] est mené sous l'égide de l'Université des Nations Unies (Tokyo, Japon) par plusieurs équipes dans le monde (Japon, France, Inde, ...). L'acronyme UNL désigne, en plus du projet, un langage artificiel ainsi qu'un format de documents multilingues. Par hypothèse, UNL favorise les relations de type connaissances du monde et c'est à ce titre, afin de compléter les informations des fonctions lexicales de Mel'čuk que nous nous en sommes inspirés.

Caractéristiques d'UNL UNL est un langage artificiel universel et destiné à un usage machinal qui permet d'exprimer les informations contenues dans les phrases sous forme de pseudo-réseaux sémantiques. Il agit comme langage pivot pour permettre la traduction d'une langue A en langue B. L'avantage principal d'une telle architecture est qu'il suffit de créer des méthodes pour passer d'UNL à C (opération de déconversion) pour permettre la traduction de A vers C ou de B vers C et de créer des méthodes pour passer de C à UNL (opération d'enconversion) pour permettre la traduction de C vers A ou de C vers B (cf. figure 6.1).

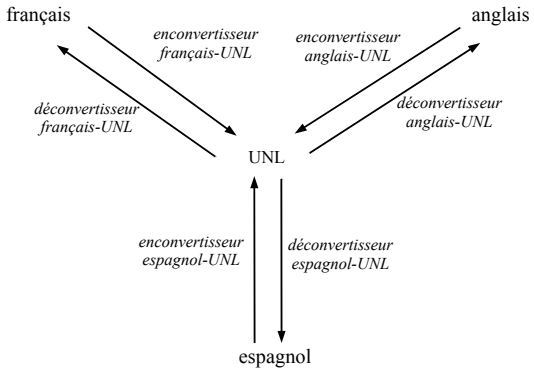


FIG. 6.1 – Principe d'UNL

Les informations des phrases sont représentées par des hypergraphes dont les nœuds sont des UW (Universal Words) et les arcs des relations entre ces UW. Ce sont ces relations dont nous nous sommes inspirés pour représenter les relations entre les mots d'une phrase.

Ce graphe permet de savoir que ‹*Ronaldo*› est une instance de la classe (relation *iof*) ‹*human*› du domaine (relation *fld*) ‹*soccer*›. Cette instance a marqué (‹*score*› est un verbe qui correspond à un ‹*event*› doit avoir comme agent un ‹*human*› d'un domaine (*fld*) lié au ‹*sport*› ; des variables indiquent que ce verbe est conjugué à un temps passé et fini) un but (‹*goal*› est une instance de ‹*thing*›).

Les relations du modèle UNL UNL définit un peu moins d'une cinquantaine de relations présentées en annexe F. Ces relations sont conçues pour lier les UW d'une phrase entre eux,

[94] Le 18 Brumaire de l'An VII du calendrier révolutionnaire (9 novembre 1799), le général Napoléon Bonaparte met fin à la première république française par un coup d'État.

[95] http://www.undl.org/

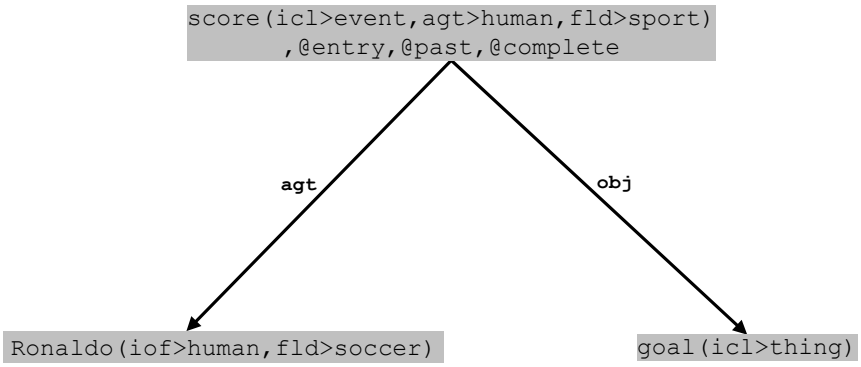


FIG. 6.2 – Graphe UNL de la phrase « *Ronaldo a marqué un but.* ».

elles sont ainsi plus axées sur la représentation de la connaissance de la phrase plutôt que sur la production lexicale comme le sont les fonctions lexicales de Mel'čuk (FLM).

Assez peu de relations sont ainsi présentes à la fois dans UNL et dans les FLM. On pourrait considérer que seule la relation d'hyperonymie se retrouve dans les deux mais elle existe dans un point de vue hyperonymie de classe pour UNL (relation *icl*) et dans un point de vue plus linguistique d'hyperonymie de substitution pour les FLM (relation *Gener*) (cf 6.3.2).

En revanche, UNL ne peut décrire les informations disponibles dans les fonctions lexicales que de façon indirecte. En effet, il n'y rien qui ressemble à des prédicats et des arguments dans UNL or il s'agit de la base fondatrice de la plupart des FLM. De la sorte, si nous considérons les FLM adjectivales (l'intensificateur *Magn* (Magn(peur)=bleu), le confirmateur *Ver* (*Ver*(argument)=valable) ou le laudatif *Bon* (*Bon*(conseil)=précieux)), nous pouvons constater que ces relations n'existent pas en tant que telles. Seule une relation UNL *aoj* (attribut) peut permettre de relier les items impliqués. Ainsi, la transcription UNL de « *peur bleue* » serait *aoj*(strong, fear), il n'y a rien qui qualifie la relation, on ne sait pas si elle est de nature positive ou négative on sait juste que la relation unit un mot à un attribut. Il faudrait imaginer dans un dictionnaire de déconvertion vers le français la présence d'une règle de transfert comme *aoj*(strong, fear) –> *aoj*(bleue, peur). Ainsi, seul le transfert, et non la FL proprement dite, serait alors explicité.

6.1.1.4 Relations importantes en vue d'une analyse sémantique

Parmi les relations d'UNL, nous en avons identifié un certain nombre qu'il nous paraît important de représenter dans la base lexicale sémantique. Il s'agit de relations typiques, c'est-à-dire des relations qui se "répètent régulièrement". Ainsi, ‹*pelle*› est un instrument typique pour ‹*creuser*› et la relation se trouvera dans le graphe, en revanche, ‹*lime*› n'est est pas un et la relation ne devrait pas ainsi se trouver dans le graphe.

Ces relations ont ainsi été fabriquées à partir des fonctions lexicales de Mel'čuk [Mel'čuk, 1988], [Mel'čuk *et al.*, 1995], de [Polguère, 2003] et de la version 2005 des spécifications d'UNL disponible sur la page http://www.undl.org/unlsys/unl/unl2005/. Ces relations sont matérialisées dans le réseau et modélisées sous la forme de Fonctions Lexicales d'Analyse (FLA).

Il existe deux types de FLA : les *FLA pour les Connaissances Linguistiques* (*FLACL*) et les *FLA pour les Connaissances du Monde* (*FLACM*). Il s'agit bien de types, la frontière peut ainsi être considérée comme relativement floue et certaines FLA pourraient être vues comme faisant partie de l'un ou l'autre des types.

Les FLA pour les Connaissances Linguistiques Les Fonctions Lexicales d'Analyse pour les Connaissances Linguistiques (FLACL) sont des fonctions lexicales proches de celles de Mel'čuk. Elles modélisent les fonctions lexicales qui correspondent à des connaissances linguistiques.

Il faut être bien conscient que ces fonctions représentent également un état du monde mais cet état est représenté par un item particulier en langue, un item choisi d'une façon qui semble arbitraire. Ainsi, la phrase « *Jean a eu une peur bleue* » est la représentation dans le monde réel de la grande peur éprouvée par Jean qui est lexicalisée grâce à la fonction lexicale d'intensification *Magn* et sa représentation, l'item ‹*bleu*›.

Parmi les FLACL, on trouve :

- la synonymie (*Syn*) qui caractérise pour un même sens les formes différentes (cf. 3.2.1), formes données uniquement par l'usage donc sans rapport vraiment direct avec le réel ;
- l'antonymie ($Anti_{\alpha}$) ;
- les génériques (*Gener*) qui correspondent aux hyperonymes de substitution c'est-à-dire aux termes de la hiérarchie qui sont préférés à d'autres par l'usage dans un cas de référence (cf 6.3.2) ;
- le singulatif (*Sing*) et sa relation inverse le collectif (*Mult*) ;
- les FLA adjectivales (*Magn*, *Ver*, ...).

Les FLA pour les Connaissances du Monde Les Fonctions Lexicales d'Analyse pour les Connaissances du Monde (FLACM) ont été créées pour relier les objets lexicaux par des connaissances du monde. Parmi ces FLACL, on trouve :

- la relation d'hyperonymie (*Hyper*) qui est ici une hyperonymie de classe contrairement à *Gener* qui est l'hyperonymie de substitution. Comme nous l'avons déjà remarqué, la CM « *une chaise est un siège* » est retranscrite en langue par le fait que ‹*siège*› est hyperonyme de ‹*chaise*› qui est une CL ;
- sa relation inverse l'hyponymie (*Hypo*) ;
- *instance*[*Inst*] : cette relation est inspirée de l'*iof* d'UNL. Il s'agit d'un objet nommé qui est une instance d'une classe. Elle est proche de la notion d'hyponymie que l'on peut considérer comme la transcription en langue de la propriété qu'une classe a d'être une sous-classe d'une autre. Par exemple, *Inst*(‹*écrivain*›)=‹*Émile Zola*›, *Inst*(‹*cheval*›)= ‹*Tornado*› ;
- la relation de méronymie (*Mero*) et sa relation inverse l'holonymie (*Holo*) ;
- la relation instrument (S_{Inst}) qui relie une action (verbe) à son instrument typique. Par exemple, S_{Inst}(‹*creuser*›)= ‹*pelle*› ;
- la relation agent (*agt*) qui relie une action à l'agent typique qui la réalise et patient qui relie une action au patient typique qui la subit (*pt*). Par exemple, *agt*(‹*manger*›)= ‹*être humain*› ; *pt*(‹*manger*›)= ‹*nourriture*› ; *agt*(‹*creuser*›)= ‹*être humain*›. Cette notion devrait sans doute être examinée pour obtenir des informations plus précises sur les verbes dans la lignée des FL verbales de Mel'čuk mais, faute de temps, cette voie n'a pu réellement être explorée au cours de cette thèse. Nous verrons dans le chapitre suivant que cette simplification permet tout de même de fortement désambiguïser les verbes ;

Cette typologie n'est pas seulement théorique et nous allons voir dans la suite de cette thèse les différences entre les types que ce soit pour la modélisation des fonctions à la section 6.4 ou pour leur intérêt dans diverses applications classiques du TALN au chapitre 7.

6.2 Généralités sur le réseau lexical

Comme nous l'avons vu, la représentation du sens des objets lexicaux de la base lexicale sémantique utilise en partie des informations de nature relationnelle. De même, tout ou partie de la modélisation d'une FLA nécessite toujours l'explicitation de sa relation dans la base lexicale sémantique. Ces relations sont donc à double titre stockées dans la base lexicale sémantique (cf 5.1). Toutefois, de part les hypothèses de construction de la BLS, l'acquisition de ces relations explicitées se fait de manière automatique et ne peuvent donc s'y trouver de façon booléenne. C'est la raison pour laquelle nous utilisons des Relations Lexicales Valuées.

6.2.1 Relations Lexicales Valuées

6.2.1.1 Définition

Dans les réseaux sémantiques classiques comme ceux que nous avons présentés en 1.3.2, un arc unit deux nœuds si une relation sémantique existe entre les deux termes qui leur correspondent. Ainsi, on trouvera une relation de méronymie entre ‹*jambe*› et ‹*corps*› ou un rapport d'antonymie entre ‹*frère*› et ‹*soeur*› tandis qu'il ne devrait en exister aucun entre ‹*éléphant*› et ‹*soeur*› ou entre ‹*jambe*› et ‹*voler*›.

Les relations lexicales valuées (RLV) ne sont pas booléennes et ont une valeur qui exprime la probabilité d'existence d'une relation entre deux objets lexicaux (ITEMS LEXICAUX, ACCEPTIONS, LEXIES). Ainsi, une RLV $\mathcal{R}$ est une relation qui donne, pour deux objets lexicaux, une valeur entre 0 et 1 :

$$\mathcal{R} : \sigma^2 \rightarrow [0,1] \tag{6.1}$$

où σ est l'ensemble des objets lexicaux. Plus la valeur est proche de 1, plus la relation entre les deux items est susceptible d'exister, plus la valeur est proche de 0, moins la relation entre les deux items l'est. Si la valeur est de 0, nous pouvons considérer que la relation ne s'applique pas entre les deux termes. Par exemple, on peut considérer que R_{Anti}(‹*éléphant*›,‹*soeur*›) = 0 ou que R_{Mero}(‹*jambe*›,‹*avion*›) = 0 et que R_{Anti}(‹*frère*›,‹*soeur*›) ou que R_{Mero}(‹*jambe*›,‹*corps*›) devraient être proches de 1.

Nous avons choisi d'appeler ces relations Relations Lexicales Valuées (RLV) et non Relations Sémantiques Valuées pour insister sur l'idée que les relations que nous allons représenter unissent des termes du lexique non seulement par des connaissances du monde mais aussi par des connaissances linguistiques. En effet, les réseaux sémantiques tels que ceux que nous avons présentés au chapitre 1 (section 1.3.2) ne relient les termes que par des connaissances du monde et non par d'autre sorte de connaissances infra-lexicales comme c'est le cas avec les RLV.

La figure 6.3 présente un exemple de réseau lexical valué. Il est clair que dans notre base, les liens avec une valeur nulle se sont pas explicités mais celle-ci est présente ici à titre d'exemple.

6.2.1.2 Pourquoi utiliser des Relations lexicales valuées dans notre approche ?

Des relations lexicales valuées entre items lexicaux Selon l'hypothèse IV, dite d'*analyse multi-source*, un maximum de sources sont utilisées pour fabriquer les objets lexicaux de la base lexicale sémantique. Ainsi, on peut utiliser des dictionnaires classiques, des dictionnaires de relations sémantiques ou bien des corpus en particulier le Web.

Les relations extraites de ces différentes sources sont de qualités inégales. Si l'extraction dans les dictionnaires classiques ou les dictionnaires spécialisés dans la synonymie ou l'antonymie est

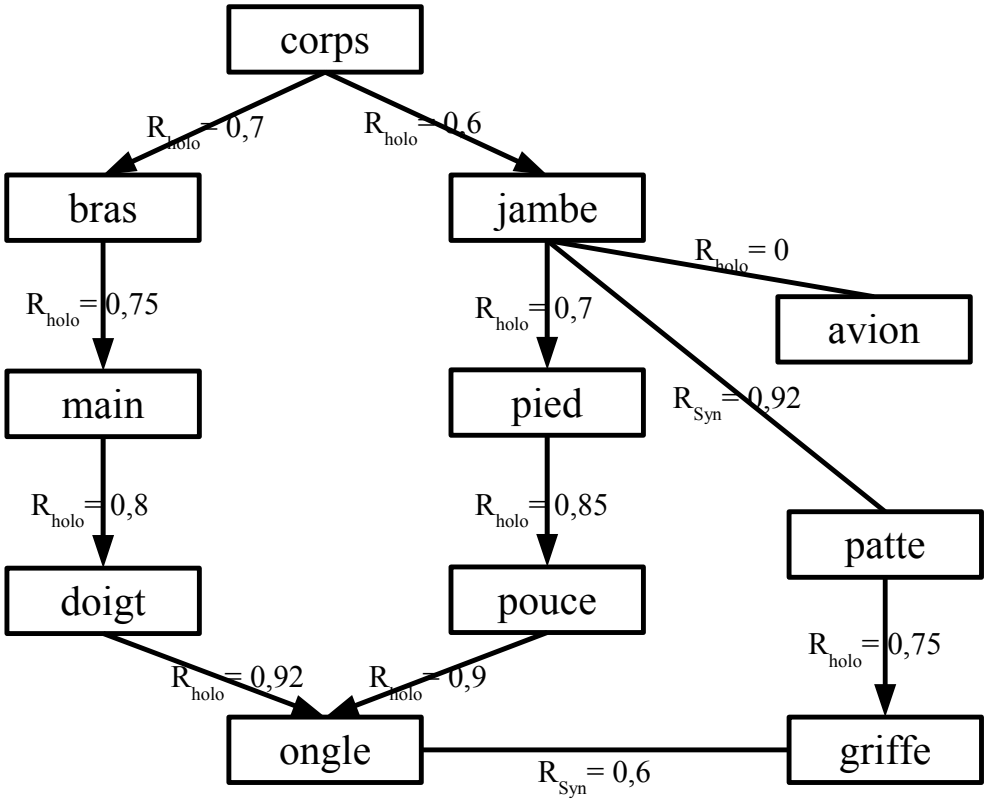


FIG. 6.3 – Exemple de réseau lexical valué.

à la fois trivale et de qualité relativement convenable car attestée (en grande partie[96]) par des lexicographes, l'extraction automatisée en corpus est encore l'objet de bien des recherches [Hearst, 1992, Morin, 1999, Claveau, 2003]. Ainsi, si on peut considérer l'information comme quasi-sûre pour les dictionnaires, on ne peut pas la considérer ainsi dans le cas d'une extraction automatisée en corpus. Une pondération peut être alors nécessaire pour quantifier la pertinence du lien découvert.

Des relations lexicales valuées entre acceptions Comme nous l'avons vu, les relations lexicales s'apprécient toujours en contexte. Ainsi, pour être rigoureusement exact, on ne devrait pas dire que deux termes sont en relation mais que deux de leur sens, que deux de leurs acceptions, sont en relation. Plutôt que les ITEMS LEXICAUX il faudrait donc que les objets lexicaux ACCEPTIONS soient reliés entre eux.

Selon l'hypothèse III, la construction des objets de la base lexicale se fait de manière automatique. Ainsi, c'est de manière automatique que la plupart des liens seront créés. Les incertitudes liées à ces créations automatiques rendent nécessaire l'utilisation de RLV.

Des relations lexicales valuées entre objets lexicaux différents Dans l'approche que nous avons adoptée, nous avons une hiérarchie à trois niveaux : les LEXIES qui correspondent à un sens de terme suivant une source, les ACCEPTIONS qui regroupent les informations des différentes LEXIES qui se rapportent au même sens et enfin les ITEMS LEXICAUX qui regroupent

[96] Une partie de la construction des dictionnaires du CISCO a ainsi été automatisée. Pour les synonymes, il y a eu une symétrisation tandis que celui des antonymes est construit en partie en utilisant les données du dictionnaire des synonymes

toute information des ACCEPTIONS de ce terme. La construction du réseau se fait non seulement de façon automatique (hypothèse III) mais aussi à partir de plusieurs sources (hypothèse IV) et de façon permanente (hypothèse V) afin, entre autres, de permettre de rendre la base cohérente grâce au croisement répété de différentes informations. Si, dans un résultat final très certainement utopique, seules les ACCEPTIONS devraient être reliées entre elles, au cours de la construction du réseau, des RLV doivent pouvoir relier divers objets lexicaux, y compris de type différent. Ainsi, on peut trouver des informations qui permettent de relier telle LEXIE issue de tel dictionnaire à telle autre issue du même dictionnaire ou bien telle LEXIE à tel ITEM LEXICAL, à telle ACCEPTION, etc. Bien entendu tout ceci toujours avec une certaine incertitude qu'il convient de représenter grâce à une RLV.

La figure 6.4 présente un exemple de réseau lexical. Les valeurs des relations ont été cette fois omises pour clarifier le dessin.

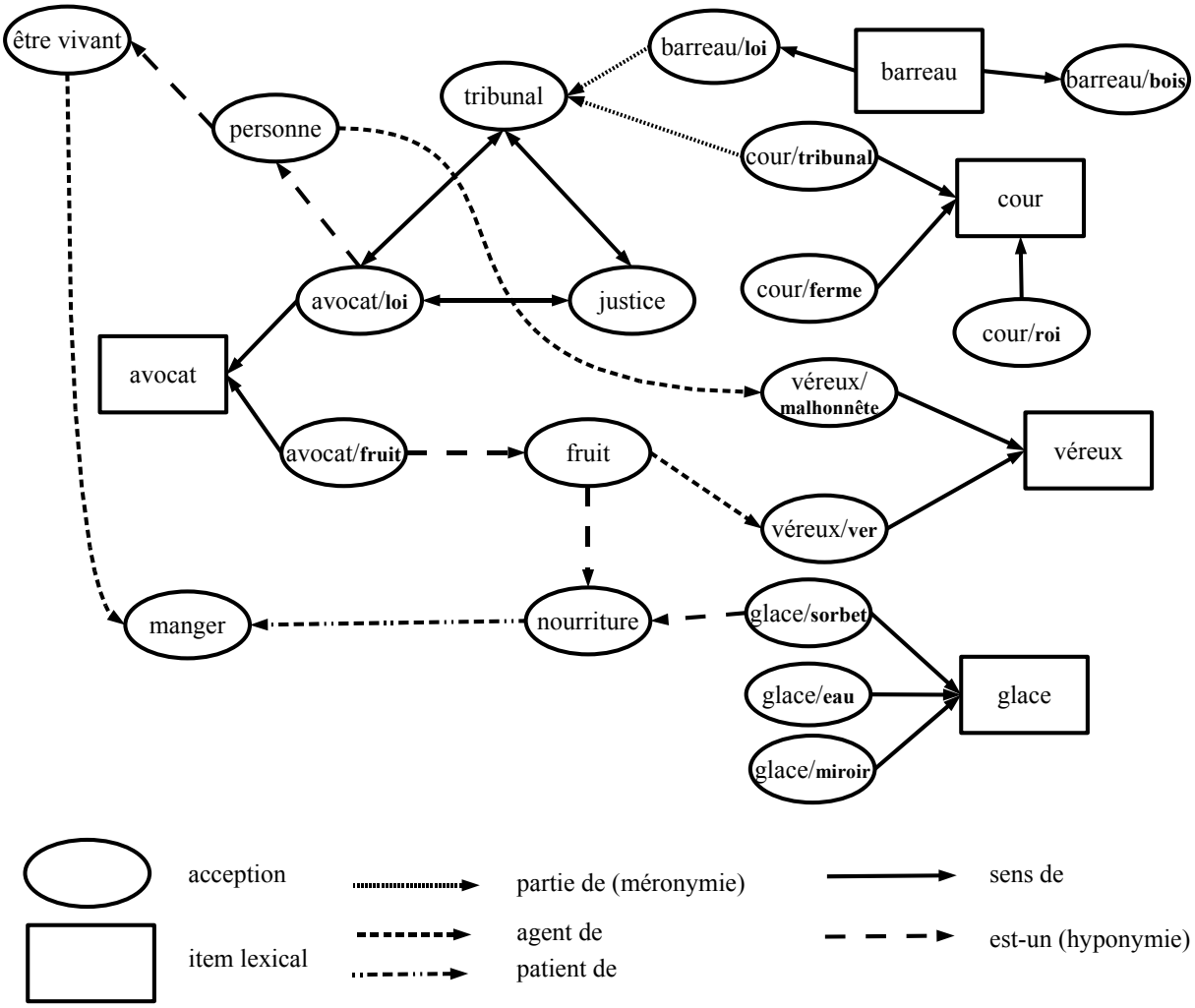


FIG. 6.4 – Exemple de réseau lexical valué.

6.3 L'hyperonymie, l'hyponymie et leurs dérivés

Les vecteurs conceptuels sont principalement construits à partir de définitions issues de dictionnaires à usage humain. La structure de ces définitions est souvent aristotélicienne c'est-à-dire en genre-différences. Ainsi, elles sont généralement présentées sous la forme d'un hyperonyme suivi des différences du terme caractérisé. De même, les noms propres sont présentés en précisant la classe dont fait partie le nom.

Dans cette partie, nous montrons l'intérêt particulier que jouent pour les vecteurs conceptuels les fonctions d'hyperonymie (*Hyper*), d'hyponymie (*Hypo*) et leurs dérivés, la fonction génériques (*Gener*), l'instanciation (*Inst*) et la fonction classe (*classe*) ainsi que les difficultés rencontrées pour les modéliser.

6.3.1 Définition et exemples

Alain Polguère ([Polguère, 2003], p. 120) définit l'hyperonymie comme un cas particulier d'inclusion de sens. Pour lui, l'item lexical I_{hyper} est un hyperonyme de l'item lexical I_{hypo} lorsque la relation sémantique qui les unit possède les caractéristiques suivantes :

1. un des sens de I_{hyper} est inclus dans un des sens de I_{hypo}
2. I_{hypo} peut être considéré comme un cas particulier de I_{hyper}

L'item I_{hypo} est appelé hyponyme de I_{hyper}. Ainsi, ‹*rose*› est un hyponyme de ‹*fleur*›, ‹*voiture*› un hyponyme de ‹*véhicule*› tandis que ‹*fleur*› est un hyperonyme de ‹*rose*› et ‹*véhicule*› un hyperonyme de ‹*voiture*›.

Selon la définition, « *le sens de* I_{hyper} *est inclus dans celui de* I_{hypo} » (cf. figure 6.5). Nous l'avions d'ailleurs vu avec les sèmes de Pottier dans la section 1.4.1.1. Dans l'exemple présenté alors, l'hyperonyme[97] le plus proche des items décrits, ‹‹véhicules de transport››, correspond à l'intersection de tous les sémènes. Ces sémènes possèdent ainsi plus de sèmes et donc un hyponyme a nécessairement un sens plus riche que son hyperonyme.

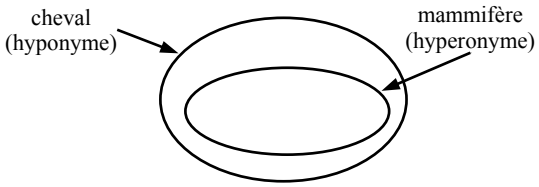


FIG. 6.5 – Inclusion des sens hyperonyme-hyponyme

La relation d'hyperonymie est transitive puisque si A est hyperonyme de B et B est hyperonyme de C alors A est hyperonyme de C. Grâce à cette propriété, la relation d'hyperonymie forme une hiérarchie des termes et c'est à ce titre que nous l'avons classifiée en relation hiérarchique lorsque nous l'avons introduite une première fois à la section 1.3.1.1. De fait, la plupart des articles en représentation des connaissances assimilent l'hyperonymie à la relation *sorte-de* [Lafourcade & Prince, 2004]. De ce point de vue, l'hyperonymie permet de former une classification systématique des termes, une taxinomie. La figure 6.6 présente un extrait de la hiérarchie des termes en français centrée sur ‹*siège*›.

Polguère note qu'on peut parfois considérer que les relations d'hyperonymie/hyponymie ne lient pas forcément des termes de même nature grammaticale ([Polguère, 2003], p. 120). Par exemple, on peut juger que, dans une certaine mesure, le nom ‹*sentiment*› est un hyperonyme du nom ‹*amour*› mais aussi un hyperonyme du verbe ‹*aimer*›. Nous ne retiendrons pas ici cette

[97] L'archisème chez Pottier.

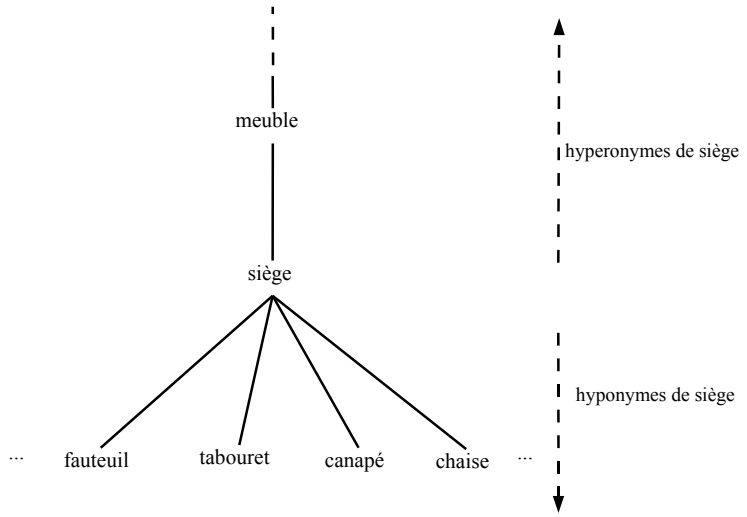


FIG. 6.6 – Extrait de la hiérarchie des items centrée autour de ‹*siège*›

définition et nous considérerons, comme pour les fonctions symétriques, que l'hyperonymie lie des termes issus de la même partie du discours.

6.3.2 Hyperonymie de substitution et hyperonymie de classe

En langue, il est habituel d'utiliser des termes pour substituer un mot par un autre pour des raison de style ou pour éviter les répétitions. Ainsi, on emploie pour désigner une chaise l'item ‹*chaise*› ou bien son hyperonyme ‹*siège*›. Il existe toutefois des cas où un hyperonyme n'est pas le terme le plus adéquat à utiliser selon l'usage. Si nous considérons un tractopelle, l'usage n'est pas d'employer le terme de ‹*véhicule*› mais plutôt celui d'‹*engin*›. On parle alors d'*hyperonyme de substitution* qui est en langue l'un des termes par lequel on peut remplacer l'item cible et l'*hyperonyme de classe* l'un des termes correspondant à une sur-classe de la classe représentée par l'item cible.

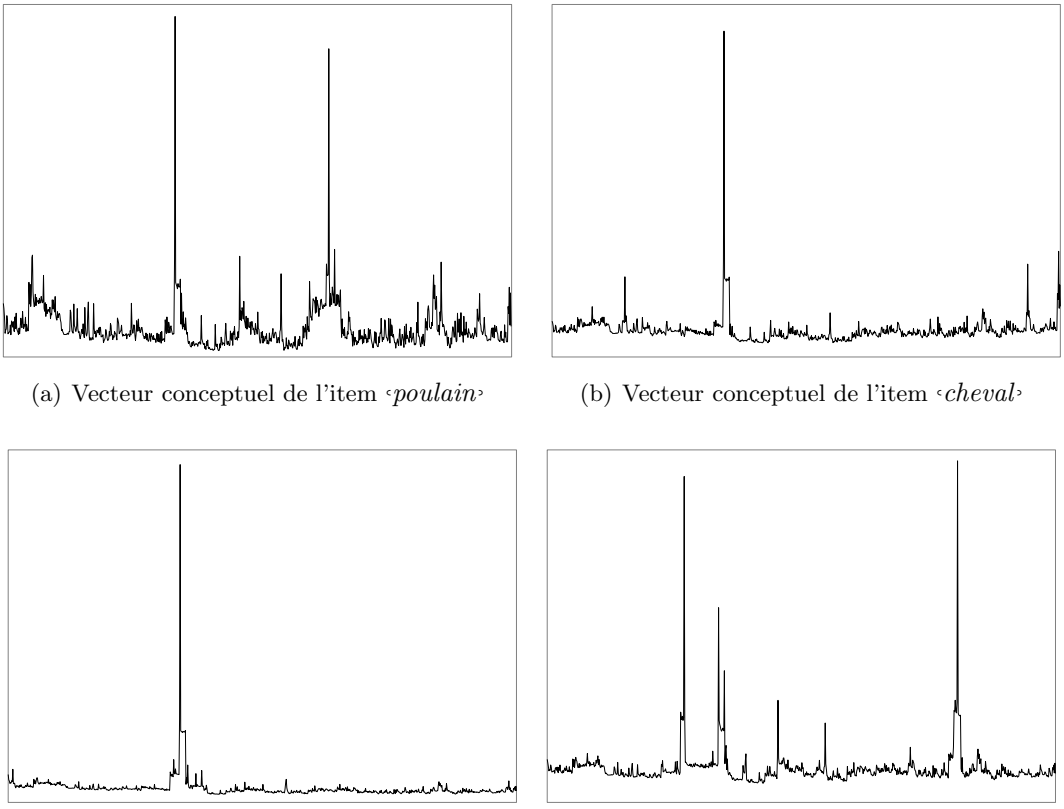
Cette distinction se retrouve dans les FLA et nous avons l'hyperonymie de substitution qui est ainsi modélisée par la FLA pour les connaissances linguistiques *Gener* et l'hyperonymie de classe par la FLA pour les connaissances du monde *Hyper*.

6.3.3 Des relations essentielles dans le cadre des vecteurs conceptuels

6.3.3.1 L'horizon lexical

Observons quatre vecteurs correspondant à la taxinomie centrée sur ‹*cheval*›. On peut remarquer que le vecteur de ‹*poulain*› est le plus contrasté, il est nettement moins plat que les trois autres. En terme mathématique, on dira que son coefficient de variation c, défini en 2.1.5.4, est inférieur à celui des autres. Rappelons ici que le coefficient de variation est un outil statistique qui permet d'évaluer la "conceptualité" du vecteur. S'il est proche d'un vecteur génératif, $c(V)$ est maximum tandis que si le vecteur est plat (ie. toutes ses composantes ont la même valeur), il est nul.

Le coefficient de variation est plus important pour le 3ème vecteur, celui de ‹*mammifères*›, que pour les autres. Il est ainsi celui qui se rapproche le plus d'un vecteur génératif. Si on part du bas de la hiérarchie et en remontant vers le haut, le coefficient de variation est de plus en plus important ($c(V(‹poulain›)) > c(V(‹cheval›)) > c(V(‹mammifère›))$) puis l'est de moins en moins à partir de cet item ($c(V(‹mammifère›)) < c(V(‹animal›))$).



(a) Vecteur conceptuel de l'item ‹*poulain*›

(b) Vecteur conceptuel de l'item ‹*cheval*›

(c) Vecteur conceptuel de l'item ‹*mammifères*›

(d) Vecteur conceptuel de l'item ‹*animal*›

FIG. 6.7 – Vecteurs conceptuels de la hiérarchie sémantique centrée autour de l'item ‹*cheval*›

Le même type d'observation peut être renouvelé à d'autres endroits de la taxonomie générale. On peut toujours trouver une zone où les coefficients de variation des vecteurs des termes de la hiérarchie est plus important que celui de son hyponyme mais aussi une zone où c'est l'inverse. Entre les deux, il existe donc un point (probablement non lexicalisé) où le coefficient de variation est maximum[98], l'*horizon lexical*. La figure 6.8 présente en deux dimensions cet horizon lexical.

Pourquoi existe-t-il cet horizon lexical et quelles sont les caractéristiques qui le placent à tel ou tel endroit de la hiérarchie ? L'apparition de ce phénomène est une conséquence de la construction des vecteurs. Cette dernière est basée sur des définitions issues de dictionnaires et sur une hiérarchie de concepts. Étudions le rôle de chacun dans la construction et leur conséquence sur les composantes des vecteurs.

6.3.3.2 Le rôle des définitions dans la construction des vecteurs conceptuels du point de vue de l'hyperonymie

Définitions hyperonymiques et définitions hyponymiques Examinons plusieurs extraits de définitions issues de [Larousse, 2004] :

arc : Arme formée d'une tige flexible (...)

cheval : Grand mammifère ongulé domestique (...)

fourmi : Insecte vivant en société (fourmilières) regroupant (...)

[98] Le coefficient de variation maximum est de 29 pour nos vecteurs à 873 dimensions (cf 2.1.5.4).

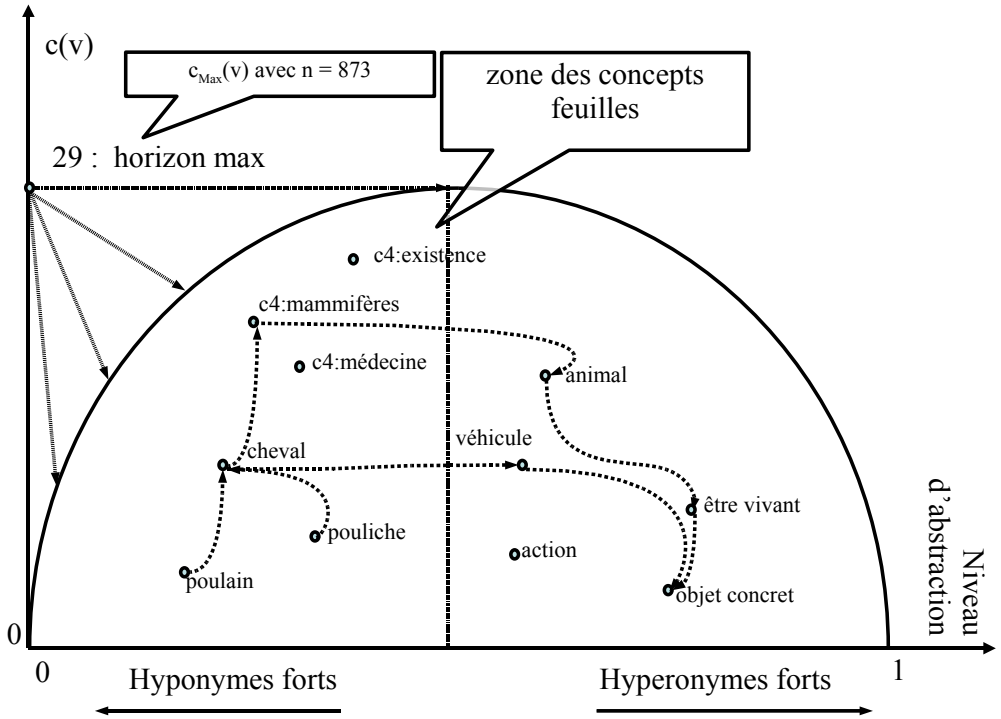


FIG. 6.8 – Représentation en deux dimensions de l'horizon lexical pour les vecteurs conceptuels construits grâce aux thésaurus Larousse [Larousse, 1992] - (figure issue de [Lafourcade & Prince, 2004])

> **paréo** : Vêtement traditionnel tahitien (...)
>
> **schiste** : Roche sédimentaire (...)

La structure de chacune est identique, elles définissent les termes par *genre* et *différences*. Par exemple, ‹*paréo*› est de genre ‹*vêtement*› et se différencie par le fait d'être à la fois ‹*traditionnel*› et ‹*tahitien*›. De même, un ‹*cheval*› est un ‹*mammifère*› qui a, en plus, les caractéristiques d'être ‹*grand*›, ‹*ongulé*› et ‹*domestique*›. Les définitions qui correspondent à ce type de structures sont fréquentes dans les dictionnaires classiques et sont appelées des *définitions aristotéliciennes* (ou hyperonymiques).

En revanche, certains termes se situent trop haut dans la hiérarchie et les dictionnaires ont alors tendance à les définir en donnant pour exemple leurs hyponymes comme en témoigne une des définitions de ‹*véhicule*› dans [Robert, 2000].

> **véhicule** : Engin à roue(s) ou à moyen de propulsion, servant à transporter des personnes ou des marchandises (ex. autobus, autocar, autochenille, automobile, (...), tramway, voiture, wagon)
>
> **engin** : Tout objet servant à faire une opération précise. (voir appareil, instrument, outil)

On parle dans ce cas de *définition hyponymique*.

Le rôle de l'hyperonymie et de l'hyponymie dans la construction des vecteurs Les vecteurs conceptuels sont construits en partie grâce à des dictionnaires à usage humain (cf. hypothèse III dite de *génération automatique*, section 5.1.3). Ainsi, certains vecteurs ont été construits à partir de définitions plutôt hyperonymiques tandis que d'autres l'ont été à partir de définitions plutôt hyponymiques.

Quelles sont les conséquences sur les vecteurs ? Examinons deux cas extrêmes. S'il s'agit d'un terme :

- *en bas de la taxinomie* : le vecteur a été construit par un enchaînement de définitions hyperonymiques et à chaque niveau de la hiérarchie le vecteur se sera enrichi de nouvelles idées, il s'en trouve ainsi relativement aplati, son coefficient de variation se rapproche de zéro.
- *en haut de la taxinomie* : le vecteur a été construit à partir de certains de ses hyponymes. Plus le terme est haut dans la hiérarchie, plus ses hyponymes sont susceptibles de provenir de domaines différents. Par exemple, ‹*moyen de transport*› peut être employé non seulement pour parler de ceux utilisés par des humains ou des animaux (‹*voitures*›, ‹*bétaillère*›, ...) mais aussi pour ceux utilisés par l'oxygène dans l'organisme (‹*globules rouges*›) ou par l'électricité (‹*fil électrique*›, ‹*ligne haute-tension*›, ...). Les idées contenues alors dans le vecteur sont d'autant plus nombreuses que les hyponymes du terme couvrent des domaines hétérogènes. Dans ce cas aussi, le vecteur s'en trouve relativement aplati, son coefficient de variation se rapproche de zéro.

Entre ces deux cas extrêmes, se trouve donc une zone où les vecteurs sont beaucoup plus contrastés, où peu de composantes sont importantes où les coefficients de variation sont particulièrement élevés et où on trouve en particulier les vecteurs génératifs, les vecteurs à partir desquels les vecteurs conceptuels sont construits (cf. section 2.1.2).

Cette zone et l'horizon lexical en son sein ne semble pas seulement tributaires des définitions. En fait, dans le cas des définitions hyponymiques, même si ces dernières couvrent des domaines hétérogènes, l'incidence sera pratiquement nulle si la hiérarchie employée ne considère pas ces idées ou sera très importante dans le cas contraire.

6.3.3.3 Le rôle de la hiérarchie dans la construction des vecteurs conceptuels du point de vue de l'hyperonymie

La zone intermédiaire où est située l'horizon lexical est tributaire des concepts de la hiérarchie utilisée que les définitions soient hyperonymiques ou qu'elles soient hyponymiques. Reprenons l'exemple des animaux. Cette partie de la hiérarchie Larousse est composée d'un concept de niveau 3 *C3 :LES ANIMAUX* et de plusieurs concepts de niveau 4 comme le présente l'extrait de la hiérarchie[99] présenté par la figure 6.9.

Quelles seraient les conséquences sur les vecteurs si la hiérarchie était différente ? Deux cas sont possibles :

- *la hiérarchie ne considère qu'une composante pour les animaux* : à ce moment-là, le vecteur conceptuel le plus proche d'un vecteur génératif ne pourrait être que celui d'‹*animal*›. En effet, tous ses hyponymes (‹*mammifères*›, ‹*oiseaux*›, ‹*reptiles*›, ...) n'auraient pour composante principale que celle qui correspondrait à *ANIMAL* ainsi, ‹*animal*› ne pourrait avoir que cette dernière comme composante importante. L'horizon lexical se situerait ainsi plus haut dans la taxinomie.
- *la hiérarchie considère une ramification plus importante* : On aurait dans ce cas un niveau 4 qui serait constitué non plus par les différentes classes animales mais par une distinction plus précise. Nous aurions pu ainsi avoir, par exemple, les différentes espèces d'animaux comme les *CHEVAUX*, les *CHIENS*, les *CHAUVES-SOURIS*, les *HOMINIDÉS*, les *GRENOUILLES*, ...

[99] Le lecteur trouvera la hiérarchie complète en annexe B

0 UNIVERS
1 LE MONDE
2 LA VIE
3 LES ANIMAUX
4 ZOOLOGIE
4 MAMMIFÈRES
4 OISEAUX
4 POISSONS
4 REPTILES
4 BATRACIENS
4 INSECTES ET ARACHNIDES
4 CRUSTACÉS
4 MOLLUSQUES ET PETITS ANIMAUX MARINS
4 VERS
4 CRIS ET BRUITS D'ANIMAUX

FIG. 6.9 – Extrait de la hiérarchie du thésaurus Larousse [Larousse, 1992] centrée autour du concept de niveau 3 *LES ANIMAUX*

Dans ce cas, le vecteur de ‹*cheval*› serait plus proche d'un vecteur génératif et le vecteur de ‹*mammifère*› ne pourrait avoir que ces composantes de niveaux 5 prépondérantes par rapport aux autres. La zone intermédiaire se situerait ainsi plus bas dans la taxinomie.

Dans l'expérience sur les vecteurs conceptuels menée ici, à savoir une méthode basée sur une hiérarchie de concepts pré-établie, les définitions mais aussi la hiérarchie a une influence déterminante sur les propriétés des vecteurs au moins du point de vue de l'hyperonymie.

6.3.4 Fonctions lexicales de construction et d'évaluation de l'hyperonymie et de l'hyponymie

Nous montrons ici en quoi les fonctions lexicales de construction d'hyperonymes, d'hyponymes et celles de leurs dérivés existent déjà dans l'analyse sémantique des textes.

6.3.4.1 Fonction lexicale de construction d'hyperonymie et d'hyponymie

Les cas où nous aurions besoin de fonctions lexicales de construction d'hyperonymes et d'hyponymes sont assez peu nombreux. La première fonction servirait, pour l'analyse d'une définition hyperonymique, à calculer un hyponyme tandis que la seconde servirait, pour celle d'une définition hyponymique, à calculer un hyperonyme.

Analyse de définitions hyperonymiques Une fonction lexicale de construction d'hyponymes devrait permettre, à partir d'un item hyperonyme et d'items décrivant les propriétés supplémentaires que possède le référent de l'hyponyme, de construire un vecteur conceptuel correspondant à cet hyponyme (cf.figure 6.10).

$$\omega \times \omega^n \rightarrow \vartheta \quad : \quad g, d_1, d_2, \ldots, d_n \rightarrow Z = Chyper_R(g, d_1, d_2, \ldots, d_n)$$

FIG. 6.10 – Construction d'un vecteur hyponyme : fonction $Chyper_R$

On peut ainsi considérer que cette fonction serait une somme pondérée du vecteur de l'hyperonyme et des vecteurs correspondant aux différences. La principale difficulté qu'il faudrait gérer

dans cette fonction est de savoir quelle intensité donner à chacune des propriétés supplémentaires que possède l'hyponyme par rapport à l'hyperonyme.

Une définition hyperonymique est constituée en genre et en différence. Lors d'une analyse sémantique, le gouverneur de la phrase a un poids prépondérant sur les autres constituants. Or ce gouverneur dans une définition hyperonymique est l'hyperonyme. Nous avons donc ainsi le genre qui a une importance supérieure à celle des différences. De même, les différences ont un poids fonction de leur rôle syntaxique dans la définition, rôle qui est plus ou moins important suivant l'influence de la propriété. Les différents exemples de définitions hyperonymiques présentés précédemment semblent étayer cette hypothèse.

La fonction d'hyperonymie existe donc déjà dans l'apprentissage mais de façon émergente et non en tant que tel. Il est ainsi inutile d'essayer d'en construire une autre dont l'intérêt serait non seulement limité mais aussi sans doute peu fiable à cause de la difficulté à connaître le poids des différences dans l'item.

Ce raisonnement peut être renouvelé avec les dérivés de l'hyperonymie, la fonction *classe* et la fonction *Gener*.

Analyse de définitions hyponymiques Une fonction lexicale de construction d'hyperonyme devrait permettre, à partir d'items hyponymes de construire le vecteur conceptuel qui lui correspondrait le mieux (cf.figure 6.11).

$$\omega^n \rightarrow \vartheta \quad : \quad h_1, h_2, \ldots, h_n \rightarrow Z = Chypo_R(h_1, h_2, \ldots, h_n)$$

FIG. 6.11 – Construction d'un vecteur hyperonyme : fonction $Chypo_R$

Comme nous l'avons vu dans la définition de l'hyperonymie, le sens de l'hyperonyme est inclus dans celui de l'hyponyme. Dans une vision ensembliste, on peut ainsi considérer que calculer les intersections des sens des hyponymes donnerait un ensemble qui devrait relativement correspondre au sens de l'hyperonyme. Du point de vue de nos vecteurs, faire une intersection d'idées équivaut en fait au produit terme à terme des différents vecteurs (cf 2.1.4.3).

Cette solution semble pourtant difficile à mettre en œuvre. Rappelons-le encore une fois, les fonctions lexicales de construction sont mises au point pour rendre une base lexicale sémantique plus cohérente. Les vecteurs utilisés lors d'une analyse quelconque ne sont donc pas forcément pertinents et leur utilisation est, pour cette raison, parfois délicate. En particulier, l'intersection est extrêmement risquée à employer. En effet, avec cette solution, si un seul des vecteurs est mal indexé, les idées qui ressortent dans le vecteur final ne sont pas celles qui devraient être les plus pertinentes. Le produit terme à terme entraîne ainsi une élimination de certaines idées au profit d'autres et le vecteur obtenu est relativement aléatoire.

Une solution qui semble plus réaliste en pratique consiste à utiliser la somme vectorielle. En effet, que se passe-t-il dans ce cas ? Nous avons toutes les idées des hyponymes dans le vecteur résultant, c'est-à-dire non seulement celles qui nous intéressent plus particulièrement, celles de l'hyperonyme mais aussi celles qui marquent les différences de chacun des hyponymes et qui, si on suit rigoureusement la définition de l'hyperonymie/hyponymie, ne devraient pas se trouver dans le vecteur de l'hyperonyme. La présence de ces idées devrait *a priori* sembler problématique mais il faut toutefois se souvenir que, dans le modèle des vecteurs d'idées, ce ne sont pas les idées prises de façon individuelle qui ont une interprétation particulière mais plutôt les importances relatives des idées entre elles. Les vecteurs sont ainsi normés. Utiliser la somme vectorielle plutôt que le produit terme à terme normalisé, qui pourtant est l'opération sur les vecteurs d'idées qui correspond le mieux au modèle linguistique de modélisation de la fonction lexicale de construction

d'hyperonymes, permet de regrouper les idées de l'hyperonyme qui se retrouvent dans un certain nombre de vecteurs correspondants à ses hyponymes et d'atténuer fortement les autres grâce à la normalisation. Nous obtenons ainsi un vecteur dont les composantes principales correspondent à celles de l'hyperonyme tandis que les différences des hyponymes s'éliminent les unes les autres.

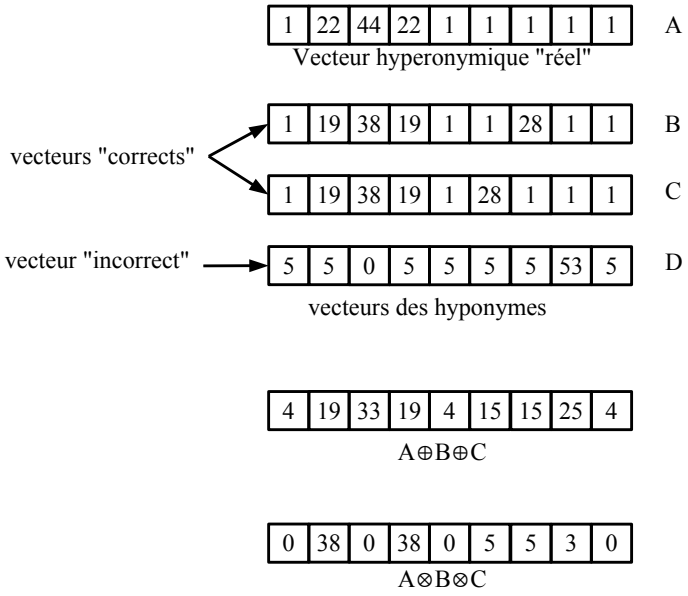


FIG. 6.12 – Exemple de comparaison *somme normée-produit terme à terme normalisé* dans le cadre du calcul d'un vecteur hyperonyme (les composantes indiquées 0 sont proches de zéros et non réellement nulles).

La figure 6.12 présente un exemple très simplifié de comparaison entre somme vectorielle normée et produit terme à terme normalisé. Nous avons construit cet exemple en fabriquant le vecteur hyperonymique "réel" puis construit deux hyponymes en augmentant une de leurs composantes (la différence). Enfin, nous avons pris un troisième vecteur qui a un coefficient de variation particulièrement élevé et une composante proche de zéro (indiquée comme nulle). Comme nous pouvons le constater, le produit terme à terme "élimine" l'idée principale de l'hyperonyme puisqu'elle est négligeable dans le vecteur incorrect. La somme normée, en revanche, conserve les idées principales. Dans notre exemple, une des idées non présente dans l'hyperonyme "réel" est fortement présente dans l'hyperonyme calculé mais c'est uniquement à cause du vecteur "incorrect". Au cours de l'apprentissage, ce vecteur se modifiera petit à petit pour atteindre une position plus pertinente, ce qui aurait été difficile si nous avions utilisé le produit dont l'influence néfaste semble trop importante pour pouvoir être corrigée.

L'utilisation de la somme vectorielle semble donc la manière la plus réaliste pour modéliser la fonction lexicale de construction d'hyperonymes. On pourrait s'étonner de l'usage de cette somme alors que nous avions fait de même pour la fonction lexicale de construction de synonymes (cf. 4.1.1.1). La différence vient des relations qui unissent les termes entre eux. La somme vectorielle réalise une union des idées des vecteurs. Dans un cas de synonymie, les idées des différents vecteurs sont relativement identiques donc la somme donnera un vecteur aux idées assez semblables. En revanche, dans le cas de l'hyperonymie, certaines idées, celles liées au genre, sont identiques dans tous les vecteurs tandis que d'autres, les différences, ne se retrouveront pas et seront ainsi atténuées dans la somme.

Dans l'analyse d'une définition hyponymique, chacun des hyponymes joue le même rôle syntaxique dans la phrase et a ainsi un poids similaire. Nous avons donc aussi dans l'analyse

l'équivalent de la fonction lexicale de construction d'hyperonymes.

Ce raisonnement peut être renouvelé avec le dérivé de l'hyponymie, la fonction d'instanciation (*inst*).

6.3.4.2 Fonction lexicale d'évaluation de l'hyperonymie et de l'hyponymie

Comme nous l'avons déjà constaté plusieurs fois dans ce mémoire, les vecteurs conceptuels se prêtent mal à la modélisation des fonctions lexicales d'évaluation de l'hyperonymie. Nous avons montré, par exemple, dès le deuxième chapitre que la distance angulaire entre vecteurs évaluait les rapports thématiques plutôt que les rapports ontologiques (cf. 2.1.3.1). De même, l'existence d'un l'horizon lexical nous a montré qu'il était impossible de réellement savoir uniquement à partir de vecteurs lequel correspondait à l'hyperonyme et lequel à l'hyponyme. Comme l'a montré [Lafourcade & Prince, 2004], il ne semble donc pas possible de créer une fonction d'évaluation pour ces relations répondant aux caractéristiques présentées à la section 3.1.2, à savoir évaluer la pertinence de la relation entre plusieurs termes.

Modèles d'inclusion La définition de l'hyperonymie est basée sur la notion d'inclusion des idées de l'hyperonyme dans celles de l'hyponyme. Les modèles d'inclusion sont basés sur cette même notion toutefois modifiée du fait de l'horizon lexical. Ainsi, [Lafourcade & Prince, 2003] présente le modèle d'inclusion pour l'hyperonymie basé sur le même principe que celui de la fonction lexicale de construction que nous venons de présenter. Si A est un hyperonyme de B, alors les propriétés de A sont incluses dans celles de B. Cette inclusion ne se fait pas grâce au produit terme à terme mais grâce à la contextualisation faible. (cf. 6.3.4.1) :

$$Hyper(A,B) \quad \Rightarrow \quad D_A(V(A),\, \gamma(V(A)\ V(B))) \quad \leq \quad D_A(V(B),\, \gamma(V(A),V(B))) \tag{6.2}$$

Cette formule est vraie en dessous de l'horizon lexical, là où les termes ne sont trop généraux, en revanche au-delà, l'inclusion est inversée.

$$Hyper(A,B) \quad \Rightarrow \quad D_A(V(A),\, \gamma(V(A)\ V(B))) \quad \geq \quad D_A(V(B),\, \gamma(V(A),V(B))) \tag{6.3}$$

Par exemple, nous obtenons les mesures présentées à la figure 6.13. Les valeurs sont ordonnées des items les plus spécifiques aux items les plus généraux. On peut constater le passage au-delà de l'horizon conceptuel que ce soit pour la sous-hiérarchie des chiens, des chats ou des serpents. Les propriétés de ‹*cheval*› sont incluses dans celles de ‹*poulain*›, celles de ‹*mammifère*› dans celles de ‹*cheval*› en-dessous de l'horizon mais celles de ‹*mammifère*› sont incluses dans celles d'‹*animal*› au-dessus de ce même horizon. De même, ‹*serpent*› est inclus dans ‹*boa*› et ‹*reptile*› dans ‹*serpent*› mais ‹*reptile*› est inclus dans ‹*animal*›.

6.3.5 Conclusions sur la représentation de l'hyperonymie

Comme nous le laissions sous-entendre dès notre première présentation des vecteurs d'idées au chapitre 2, ces vecteurs ne semblent pas permettre de modéliser complètement les relations d'hyperonymie/hyponymie. Du point de vue de la construction de vecteurs, nous avons montré que l'analyse sémantique telle que nous la pratiquions permettait de modéliser ces relations. En revanche, du point de vue de l'évaluation, le seul phénomène caractéristique que les vecteurs permettent de modéliser est donné par le modèle d'inclusion. Si on sait que deux items sont en situation d'hyperonymie, alors on peut se baser sur ce modèle pour vérifier la cohérence des vecteurs. Pour la synonymie et l'antonymie, lorsque nous prenons deux items dans la base, nous pouvons évaluer dans quelle mesure ils sont liés par cette relation grâce aux différentes relations d'évaluation de synonymie et à celles de l'antonymie. Pour l'hyperonymie/hyponymie, en revanche, même en ne considérant pas le problème de l'horizon lexical, nous n'avons pas la même

A	B	D_A(A, $\gamma(A,B)$)	D_A(B, $\gamma(B,A)$)
‹*poulain*›	‹*cheval*›	0,42	**0,28**
‹*étalon*›	‹*cheval*›	0,44	**0,23**
‹*cheval*›	‹*mammifère*›	0,46	**0,18**
‹*mammifère*›	‹*animal*›	**0,24**	0,52
‹*siamois*›	‹*chat*›	0,36	**0,27**
‹*persan*›	‹*chat*›	0,38	**0,21**
‹*chat*›	‹*félins*›	0,4	**0,25**
‹*félins*›	‹*mammifères*›	0,35	**0,19**
‹*boa*›	‹*serpent*›	0,3	**0,24**
‹*crotale*›	‹*serpent*›	0,34	**0,22**
‹*serpent*›	‹*reptile*›	0,45	**0,26**
‹*reptile*›	‹*animal*›	**0,17**	0,36

FIG. 6.13 – Mesures sur le modèle d'inclusion sur des items représentants des animaux

facilité. Nous n'avons pas réussi, et il ne semble pas possible, de modéliser une fonction lexicale d'évaluation de l'hyperonymie ou de l'hyponymie uniquement basée sur les vecteurs. En effet, cette fonction devrait être basée sur la notion d'inclusion de la définition de l'hyperonymie (cf. 6.3.1). Pour deux items pris au hasard il y en a toujours un qui est plus inclus dans l'autre que l'inverse[100] et ceci même si les deux items n'ont pratiquement rien en commun. L'implication du modèle d'inclusion est inverse. Il faut d'abord savoir que deux items sont en relation d'hyperonymie/hyponymie dans le réseau lexical avant de les tester pour vérifier que l'inclusion des deux vecteurs se fait bien dans le bon sens.

Ainsi, la relation d'hyperonymie/hyponymie doit être modélisée à la fois grâce à des informations thématiques de type vectoriel et à la fois grâce à des informations de type lexical.

6.4 Modélisation des Fonctions Lexicales d'Analyse

6.4.1 Caractère thématique et lexical des fonctions lexicales d'analyse

Nous revenons dans cette section sur la modélisation des Fonctions Lexicales d'Analyse que nous avons en partie déjà évoquée dans cette thèse. Rappelons que les FLA modélisent les relations lexicales qui peuvent exister entre les objets du lexique et qu'elles peuvent être de deux catégories :

- les *fonctions lexicales de construction* qui permettent de fabriquer des vecteurs conceptuels à partir d'informations lexicales ;
- les *fonctions lexicales d'évaluation* qui permettent, elles, de mesurer la pertinence d'une relation entre deux objets lexicaux suivant certaines informations lexicales.

Comme nous l'avons déjà constaté à plusieurs reprises, parmi les relations modélisées par ces fonctions, certaines ne peuvent l'être que grâce à des informations lexicales tandis que d'autres le sont grâce à des informations à la fois thématiques et lexicales. Nous faisons ici le bilan global de ces caractéristiques pour chacune des FLA.

[100] Sauf des vrais synonymes.

6.4.1.1 Relations à caractère à la fois thématique et lexical

Ce type de relations met en partie en jeu des informations thématiques (vecteurs conceptuels) mais nécessite aussi d'être complété par des informations lexicales comme nous l'avons vu avec l'antonymie et dans une moindre mesure avec la synonymie. Parmi ces relations on trouve aussi l'hyperonymie, au moins du point de vue des vecteurs d'idées, comme nous l'avons vu lorsque nous avons tenté de modéliser les fonctions lexicales de construction et d'évaluation de l'hyperonymie dans la section 6.3.

Dans la typologie proposée dans la section précédente, on peut constater que les relations à caractère à la fois thématique et lexical existent dans les deux types :

- *FLA pour les Connaissances Linguistiques* : il s'agit de celles correspondant aux paradigmatiques de Mel'čuk. Ce sont les synonymes, les antonymes et les génériques dont la modélisation est semblable en termes de vecteurs à celle des hyperonymes ;
- *FLA pour les Connaissances du Monde* : ce sont l'hyperonymie, l'hyponymie, l'instanciation et la fonction *classe*.

6.4.1.2 Relations à caractère purement lexical

Ces relations ne peuvent pas être représentées grâce à des informations thématiques de type vecteurs d'idées. On distingue :

- *FLA pour les Connaissances du Monde* : une majorité des FLACM sont à caractère purement lexical. Par exemple, si on considère la relation de méronymie, rien dans le thème des items ‹*main*› et ‹*doigt*› ni dans celui les items ‹*mât*› et ‹*bateau*› ne permet de savoir que le doigt est une partie de la main et le mât une partie du bateau. De même, aucune information linguistique ne permet de savoir que ‹*pelle*› est un instrument typique de ‹*creuser*› (relation S_{inst}) ou que le lieu où on fait du sport est un ‹*stade*› ou un ‹*gymnase*› (relation S_{loc}).
- *FLA pour les Connaissances Linguistiques* : à part la synonymie, l'antonymie et les génériques, toutes les FLACL sont à caractère purement lexical. Il s'agit, selon la typologie des FLM de [Polguère, 2003], des FLA syntagmatiques qui correspondent aux collocations c'est-à-dire, comme nous le notions en 1.3.1.2, à des « *combinaisons d'items lexicaux qui prévalent sur d'autres sans qu'il ne semble n'y avoir de raison logique.* ». Comme il ne semble y avoir de raison logique à ces relations, elles ne peuvent être que de nature purement lexicale.

6.4.2 Fonctions Lexicales de construction et d'évaluation des FLA

Nous avons présenté dans les chapitres 3 et 4, les fonctions lexicales de construction et d'évaluation de la synonymie puis de l'antonymie. Dans quelle mesure peut-on faire de même avec les autres FLA et comment ?

6.4.2.1 Fonctions lexicales de construction

Nous avons montré dans la section 6.1 le caractère thématique et lexical des FLA. La possibilité de créer une fonction lexicale de construction de vecteurs conceptuels dépend de cette caractéristique. Ainsi :

- *pour les relations à caractère à la fois thématique et lexical*, nous avons vu qu'il était possible de créer de telles fonctions pour la synonymie et l'antonymie (cf. chapitres 3 et 4) mais, en revanche, pour l'hyperonymie et l'holonymie, il s'agit une opération à la fois difficile et inutile (cf. section 6.3.4).

– *pour les relations à caractère purement lexical*, une telle fonction est à la fois impossible et inutile à créer.

6.4.2.2 Fonctions lexicales d'évaluation

Une fonction lexicale d'évaluation (FLE) est une fonction qui mesure la pertinence de la relation correspondante entre deux objets lexicaux. Le domaine image est compris entre 0 et $\frac{\pi}{2}$ pour rester compatible avec les FLE déjà présentées (synonymie et antonymie) ainsi qu'avec la proximité thématique et de faciliter ainsi les calculs combinant ces informations.

Une fonction lexicale f qui évalue la pertinence d'une relation entre les objets lexicaux x et y en fonction des objets lexicaux $z_1,\ldots,z_m$ a les caractéristiques suivantes :

$$\sigma^2 \times \sigma^m \rightarrow [0,\frac{\pi}{2}] : x,y,z_1,\ldots,z_m \rightarrow f = F(x,y,z_1,\ldots,z_m) \tag{6.4}$$

Pour les relations à caractère purement lexical, la seule information que nous sommes susceptible d'avoir est la probabilité d'existence des relations par lesquelles l'objet lexical est lié. Nous considérerons ainsi que l'évaluation est fonction de la probabilité de la relation si elle existe.

Le cas des relations à caractère à la fois thématique et lexical est différent suivant les relations. Nous ne faisons ici que les reprendre puisque nous les avons déjà examinés précédemment. Pour la synonymie et l'antonymie, nous avons ainsi montré que des fonctions d'évaluations basées sur les vecteurs et sur les objets lexicaux existent. En revanche pour l'hyperonymie, l'hyponymie mais aussi l'instanciation et la fonction génériques *Gener* qui sont proches de la première, la création d'une telle fonction est impossible (cf. 6.3.4.2). Ici aussi la solution apportée est de considérer, comme pour les relations à caractère purement lexical, que l'évaluation est fonction de la probabilité de la relation si elle existe.

Ainsi, nous considérons pour toute autre FLA que la synonymie ou l'antonymie que la FLE correspondante se calcule par :

$$f = \frac{\pi}{2} R_f \tag{6.5}$$

Il s'agit du passage linéaire de l'intervalle $[0,1]$, celui des RLV, à l'intervalle $[0,\frac{\pi}{2}]$, celui des FLE. Ce passage est linéaire et repose sur l'hypothèse que plus on est certain de l'existence d'une relation plus la RLV qui lui correspond doit être importante.

6.4.2.3 Généralisation de la notion de voisinage

Dans les chapitres précédents nous avons déjà présenté la notion de voisinage pour le thème (section 2.1.3.2) ainsi que pour les fonctions lexicales symétriques (synonymie section 3.3.5.3 et antonymie section 3.3.5.3). Dans cette section, nous généralisation cette notion à toute fonction lexicale d'analyse.

6.4.3 Définition

La fonction de voisinage $\mathcal{V}$ est la fonction qui renvoie les n ITEMS LEXICAUX les plus proches de l'item lexical x suivant une FLE f et les objets lexicaux $u_1,\ldots,u_m$:

$$\mathcal{F} \times \sigma^m \times \mathbb{N} \rightarrow \sigma^n : f,x,u_1\ldots,u_m,n \rightarrow E = \mathcal{V}(f,x,u_1,\ldots,u_m) \tag{6.6}$$

où $\mathcal{F}$ est l'ensemble des fonctions lexicales d'évaluation et σ l'ensemble des objets lexicaux. La fonction $\mathcal{V}$ est définie par :

$$\begin{array}{c} \left|\mathcal{V}(f,x,u_1,\ldots,u_m)\right| = n,\ \forall y \in \mathcal{V}(f,x,u_1\ldots,u_m),\ \forall z \notin \mathcal{V}(f,x,u_1,\ldots,u_m), \\ f(x,y,\ldots,u_m) \leq f(x,z,u_1,\ldots,u_m) \end{array} \tag{6.7}$$

6.4.4 Exemples

Voici quelques exemples de voisinage. Les deux premiers sont extraits de la section 3.3.5.3. On remarquera que par souci de simplification du discours, nous considérons dans cette thèse que la généralisation de la fonction de voisinage peut prendre en argument la fonction de distance thématique D_A qui n'est pas une FLA :

$\mathcal{V}(D_A$, ‹*destin*›, 10) = ‹*destin*› ; ‹*destinée*› ;‹*sort*› ; ‹*détermination*› ; ‹*déterminer*› ; ‹*être*› ; ‹*fatidique*› ; ‹*fatalité*› ; ‹*déterminisme*› ; ‹*constant*› ;

$\mathcal{V}(Syn_R$, ‹*destin*›, ‹*vie*›, 10) = ‹*destin*› ; ‹*destinée*› ; ‹*sort*› ; ‹*vivifier*› ; ‹*détermination*› ; ‹*accident*› ; ‹*déterminer*› ; ‹*vital*› ; ‹*déterminisme*› ; ‹*existence*› ;

$\mathcal{V}(Mero$, ‹*navire*›, 10) = ‹*coque*› ; ‹*pont*› ; ‹*mât*› ; ‹*hélice*› ; ‹*gouvernail*› ; ‹*équipage*› ; ‹*ancre*› ; ‹*cale*› ; ‹*proue*› ; ‹*poupe*› ;

En pratique, le système renvoie la valeur de la FLE pour chacun des voisins, nous ne l'avons pas mis ici par souci de simplification. Cette distance est utilisée par les agents exploitant ce voisinage comme nous le verrons à la section 8.3.2.2 pour une autre application que l'évaluation de cette méthode. Il s'agira alors pour deux bases de se communiquer des voisinages afin d'affiner leur données.

6.5 Conclusions du Chapitre

Dans ce chapitre, nous avons présenté les Fonctions Lexicales d'Analyse dont l'objectif est d'aider à la résolution de problèmes posés dans le cadre de l'analyse sémantique d'un énoncé comme la désambiguïsation lexicale, la résolution d'anaphore ou les problèmes de référence. Nous avons établi la liste des relations utiles à modéliser à cette fin dans la base lexicale sémantique. Cette liste est composée de Fonctions Lexicales permettant de représenter à la fois les connaissances du monde et les connaissances linguistiques qui jouent toutes deux un rôle dans le processus de compréhension.

Ces relations forment le grand réseau lexical induit par la structure de la BLS et nous en avons présenté les caractéristiques (utilisation de relations lexicales valuées entre les objets lexicaux) puis évoqué les pistes à suivre pour le créer le plus automatiquement possible.

Nous avons ensuite abordé la question de l'hyperonymie, de l'hyponymie et de leurs dérivés, la fonction générique (*Gener*), l'instanciation (*Inst*) et la fonction classe (*classe*). Nous avons montré le rôle fondamental qu'elles jouent dans le cadre de la construction des vecteurs conceptuels à cause de leur utilisation dans les dictionnaires à usage humain. Nous appuyant sur les travaux de [Lafourcade & Prince, 2004], nous avons établi l'existence d'un horizon lexical au-dessous duquel les vecteurs des hyperonymes sont inclus dans les vecteurs de leurs hyponymes et au-dessus duquel le phénomène est inversé. Nous avons par ailleurs montré que cet horizon est influencé à la fois par les définitions et la hiérarchie du thésaurus.

Revenant ensuite sur notre objectif premier de modéliser les fonctions lexicales nous avons alors étudié la possibilité de construire des fonctions lexicales de construction et d'évaluation pour l'hyperonymie, l'hyponymie et leurs dérivées. Nous avons ainsi montré qu'elles existent déjà de façon émergente dans une analyse sémantique de définitions. En revanche, nous avons constaté que l'existence de l'horizon lexical empêche de modéliser des fonctions lexicales d'évaluation pour l'hyperonymie et l'hyponymie uniquement à l'aide de vecteurs.

Enfin, dans la dernière partie de ce chapitre, nous avons tiré un bilan complet des Fonctions Lexicales d'Analyse pour ce qui concerne leur modélisation. Nous avons ainsi mis en évidence que certaines doivent être modélisées en partie avec des vecteurs conceptuels et en partie à l'aide des rapports symboliques que les items lexicaux entretiennent entre eux tandis que d'autres ne peuvent l'être qu'avec des informations purement lexicales. Nous avons ainsi pu modéliser des

fonctions lexicales d'évaluation pour l'ensemble des FLA puis généraliser la notion de voisinage à l'ensemble de ces fonctions lexicales d'évaluation.

Conclusions de la partie II

Dans la conclusion de la première partie, nous avions mis l'accent sur la difficulté à se limiter aux vecteurs d'idées pour représenter le sens des termes et les fonctions lexicales. Ce sont ces deux problématiques auxquelles nous nous sommes plus particulièrement attaqués dans cette partie.

Nous avons ainsi fait le point sur les fonctions lexicales en essayant de comprendre lesquelles sont à caractère purement lexical et lesquelles sont à caractère à la fois lexical et thématique. Les premières, toutes les syntagmatiques ainsi que celles qui relèvent le plus de la pragmatique comme la méronymie, ne peuvent être modélisées que grâce aux rapports symboliques que les items lexicaux entretiennent entre eux tandis que les secondes peuvent être modélisées, en partie à l'aide de vecteurs conceptuels et en partie grâce à des informations de nature lexicale. Dans cette seconde classe, on trouve les relations symétriques, synonymie et antonymie pour lesquelles nous avons redéfinies les fonctions lexicales d'évaluation et de construction ainsi que l'hyper/hyponymie pour lesquelles ces fonctions n'ont pu être mises au point aussi finement à cause de l'existence d'un horizon lexical dû à la construction des vecteurs.

Toujours dans cette partie, nous avons tiré le bilan de l'expérience acquise pour revoir en partie le modèle de Base Lexicale Sémantique. Nous avons présenté les six hypothèses de départ qu'il nous paraît nécessaire d'adopter dans le but de construire une telle base. La première consiste à tenir compte des idées sur la représentation du sens des items lexicaux que nous avons déjà évoquée en utilisant une *représentation hybride du sens par une approche combinant approche thématique (vectorielle) et approche lexicale (relations sémantiques externes)*. La deuxième hypothèse, dite de *relations lexicales internes*, consiste à tenir compte de la polysémie dans la base. La troisième, l'hypothèse de *génération automatique*, part du constat de la difficulté à fabriquer des objets numériques et symboliques à la fois nombreux et de taille importante de manière manuelle.La quatrième, l'*analyse multi-source* cherche à palier les éventuels manques définitoires des dictionnaires en ce qui concerne la couverture du lexique ou le métalangage. La mise à jour régulière de la base ainsi que la stabilisation des données nous a imposé de choisir comme cinquième hypothèse d'avoir un *apprentissage permanent*. Enfin la sixième et dernière hypothèse, dite de *double boucle* est d'enrichir le système global par l'apport de chacun des agents qui eux-mêmes s'enrichissent de l'apport de l'ensemble du système.

Ces hypothèses nous ont conduit à choisir une architecture à trois niveaux d'objets lexicaux (LEXIE, ACCEPTION, ITEM LEXICAL).Nous avons montré comment ces hypothèses, les applications hétérogènes visées ainsi que des caractéristiques techniques nous ont amenés à adopter une architecture multi-agent dont nous avons présenté les caractéristiques conceptuelles et techniques. Le système proposé, appelé Blexisma, a pour but d'intégrer tout agent pouvant permettre de créer, d'améliorer et/ou d'exploiter une ou plusieurs Bases Lexicales Sémantiques. Nous avons enfin exposé les différents agents déjà implémentés ainsi qu'un exemple de leur interaction dans le cadre de l'acquisition d'informations sémantiques et de leur exploitation pour fabriquer des objets lexicaux.

Dans la dernière partie de cette thèse, nous allons plus particulièrement étudier comment utiliser toutes ces informations pour améliorer l'analyse sémantique. Nous allons montrer les

limites de l'analyse en remontée-descente utilisée précédemment pour capturer certains phénomènes sémantiques, l'instanciation des fonctions lexicales en particulier. Enfin nous présenterons des recherches sur la collaboration entre plusieurs bases de vecteurs notamment dans le cadre du multilinguisme et de la Traduction Automatique.

Troisième partie

Vers la Création d'Outils Sémantiques

7

Fonctions Lexicales et Analyse Sémantique

DANS ce chapitre, nous revenons sur l'analyse sémantique en essayant de mettre en relief les problèmes d'ambiguïtés qu'il est important de résoudre dans le cadre des différentes applications visées par l'équipe : le problème de l'ambiguïté lexicale, les problèmes de références, les rattachements prépositionnels, les chemins interprétatifs possibles et enfin le dernier, qui nous intéresse plus particulièrement dans cette thèse, celui de l'instanciation des fonctions lexicales. Nous montrons comment ces dernières peuvent aider non seulement à résoudre les autres ambiguïtés mais surtout en quoi leur instanciation peut être utilisée dans le cadre de la traduction automatique, de la recherche d'informations ou du résumé automatique. Nous montrons que l'analyse sémantique en remontée-redescente présentée précédemment est clairement insuffisante pour un tel objectif et nous présentons différents modèles d'analyse sémantique basés sur des algorithmes à fourmis. Le dernier, mis au point au cours de cette thèse, montre l'efficacité de cette méthode pour résoudre les problèmes posés dans une analyse sémantique en particulier celui qui concerne l'instanciation des fonctions lexicales.

Sommaire

DANS les chapitres précédents, nous avons présenté l'architecture de notre base sémantique lexicale et le réseau lexical sous-jacent qu'elle décrit. Ce réseau relie via des fonctions lexicales les objets lexicaux ITEM LEXICAL, ACCEPTION et LEXIES qui composent cette base. Dans ce chapitre, nous revenons sur la manière d'exploiter cette base en vue d'applications concrètes comme le résumé automatique ou la traduction automatique. Dans le cadre de notre projet, cette exploitation se fait toujours à partir d'une analyse sémantique de textes c'est-à-dire par le calcul, entre autres choses, d'une représentation thématique de l'ensemble du texte et de ses sous-parties.

Nous revenons ainsi sur l'analyse sémantique en essayant de mettre en relief les différents problèmes d'ambiguïtés qu'il est important de résoudre dans le cadre des différentes applications visées par l'équipe : le problème de l'ambiguïté lexicale, les problèmes de références, les rattachements prépositionnels, les chemins interprétatifs possibles et enfin le dernier, qui nous intéresse plus particulièrement dans cette thèse, celui de l'instanciation des fonctions lexicales. Nous montrons comment ces dernières peuvent aider non seulement à résoudre les autres ambiguïtés mais surtout en quoi leur instanciation peut être utilisée dans le cadre de la traduction automatique, de la recherche d'informations ou du résumé automatique. Nous montrons que l'analyse sémantique en remontée-redescente présentée précédemment est clairement insuffisante pour un tel objectif et nous présentons trois modèles d'analyse sémantiques basés sur des algorithmes à fourmis.

Le premier exploite un algorithme à fourmis mono-caste et mono-environnement (l'arbre morpho-syntaxique). Inventé par Mathieu Lafourcade, il pose les bases de l'exploitation du paradigme des fourmis pour une telle tâche.

Le deuxième, développé par Thibault Zamora au cours de son DEA, est un modèle d'analyse sémantique par algorithmes à fourmis multi-caste à environnements séparés (arbre morpho-syntaxique et réseau lexical). Il est le premier à bénéficier des avancées sur la représentation du sens à partir de vecteurs conceptuels présentées dans cette thèse.

Enfin, le dernier, mis au point au cours de cette thèse est, lui, un modèle multi-caste (une par fonction lexicale d'analyse) et à environnements partagés. Une première étude, présentée ici, montre que ce modèle est le plus efficace voire le seul des trois à pouvoir résoudre l'ensemble des problèmes posés lors d'une analyse sémantique en particulier en ce qui concerne l'instanciation des fonctions lexicales.

7.1 Généricité ou spécialisation ?

Dans le domaine du Traitement Automatique du Langage Naturel, on peut considérer qu'il existe deux sortes d'outils : les outils spécialisés et les outils génériques.

7.1.1 Outils spécialisés

Les tenants de la spécialisation cherchent à développer une application particulière et à ne résoudre que les problèmes auxquels ils sont confrontés dans ce cadre. On entend par application particulière non seulement un certain outil qui viserait la traduction ou un autre qui viserait la recherche d'informations mais aussi un outil couplé à une certaine langue. Ainsi, pour le traitement de l'anglais, on aura un outil spécialisé dans la traduction vers le français et donc l'analyse du texte anglais sera spécifique aux problèmes posés dans ce cas particulier de traduction. Dans la phrase « *The man was going to the river.* », il faut, par exemple, se poser la question de la destination finale du cours d'eau (un autre cours d'eau ou en mer) pour traduire ‹*river*› en ‹*rivière*› ou ‹*fleuve*› (cf. raffinement de sens section 1.17). En revanche, pour d'autres langues comme l'espagnol, la question ne se pose pas et il n'y aura pas de distinction à effectuer pour traduire le terme en ‹*rio*› pour donner « *El hombre iba al rio.* ».

Le principal avantage des outils spécialisés est de fortement réduire les problèmes posés à ses concepteurs ce qui, en revanche, circonscrit les fonctionnalités et en limite donc la réutilisabilité.

7.1.2 Outils génériques

Les tenants de la généricité cherchent à développer des outils qui seraient les plus génériques possibles en vue d'une réutilisation ultérieure. Dans le cadre de cette thèse, nous nous situons assez clairement dans cette deuxième école. La principale difficulté entraînée par la généricité est de parfois chercher à résoudre des problèmes que la tâche particulière qui s'effectuera ensuite n'aurait pas à traiter. Ainsi, dans le cas de la TA, la traduction du français au malais du pronom personnel ‹*nous*› est différente si on inclut la personne à qui est adressé l'énoncé ou si on ne le fait pas. Le raffinement de sens pour les cours d'eau en français en constitue aussi un bon exemple (cf 1.3.3.2). Un outil générique se doit ainsi de tenir compte de ces possibilités et tenter de les résoudre.

Il semble ainsi extrêment difficile d'obtenir une généricité totale des outils qui permettrait le passage d'une langue à une autre par simple connexion entre agents. Une généricité plus acceptable serait de se circonscrire à un certain nombre d'applications quitte à améliorer l'outil pour en permettre d'autres le cas échéant. C'est dans ce cadre que nous allons maintenant reconsidérer l'analyse sémantique.

7.2 Retour sur l'analyse sémantique

Une analyse sémantique telle que celle que nous visons consiste, entre autre, à calculer une représentation thématique de l'ensemble du texte et éventuellement de chacune de ses sous-parties. Cette représentation sous forme de vecteurs conceptuels peut se révéler particulièrement utile lors d'une phase d'indexation en recherche d'informations, pour une tâche de segmentation de textes qui a des applications, par exemple, en résumé automatique, en recherche d'informations, en classification et en catégorisation [Reynar, 1998] [Bellot & El-Bèze, 2001] [Bestgen, 2004] ou enfin, comme boucle interne de notre système, pour permettre le calcul d'autres vecteurs conceptuels si le texte est une définition issue d'un dictionnaire (cf. 5.1).

Pour permettre ce calcul, il est nécessaire de résoudre un certain nombre de phénomènes d'ambiguïtés comme l'ambiguïté lexicale, le rattachement propositionnel ou l'instanciation de

fonctions lexicales. Ces ambiguïtés nécessitent, de plus, d'être levées dans le cadre spécifique d'une application. Nous avons déjà présenté et montré l'utilisation de l'analyse sémantique en remontée-descente à plusieurs reprises dans ce mémoire. La méthode utilisée, bien que séduisante à plusieurs niveaux, peut-elle rendre compte de l'ensemble des phénomènes mis en jeu dans la construction du sens ? Est-elle capable de résoudre simplement l'ensemble des problèmes d'ambiguïtés qui peuvent se poser dans le cadre des applications visées par l'équipe que sont la Traduction Automatique (TA), la Recherche d'Informations (RI) ou le Résumé Automatique (RA) ?

7.2.1 Les différents problèmes d'ambiguïtés à résoudre lors d'une analyse sémantique

On peut distinguer cinq phénomènes sémantiques à résoudre lors d'une analyse sémantique : le *problème de l'ambiguïté lexicale*, les *problèmes de références*, les *rattachements prépositionnels*, les *chemins interprétatifs possibles* et enfin le dernier, qui nous intéresse plus particulièrement dans cette thèse, celui de *l'instanciation des fonctions lexicales*.

7.2.1.1 Problème de l'ambiguïté lexicale

La désambiguïsation lexicale est, sans conteste, le principal problème posé lors d'une analyse sémantique. Nous en avons déjà suffisamment présenté les tenants et les aboutissants dans les précédents chapitres de cette thèse pour y revenir ici (voir, par exemple, les sections 1.1.4.3, 2.1.6 et 5.1.2).

7.2.1.2 Problème de la référence

Résolution d'anaphore Un mot à valeur anaphorique ne peut être interprêté que lorsqu'il est mis en relation avec un autre élément de l'énoncé. Par exemple, dans « *En ce moment, le second attira de nouveau l'attention du capitaine. Celui-ci suspendit sa promenade et dirigea sa lunette vers le point indiqué.* », ‹‹celui-ci›› est une anaphore de ‹‹capitaine››. Il en est de même pour ‹‹il›› dans la phrase « *L'homme marcha sur la queue du chien, il aboya* ». Cet exemple montre bien l'intérêt de la résolution des anaphores dans le cadre de la TA. En effet les genres diffèrent souvent en fonction de la langue et résoudre l'anaphore peut permettre la traduction du mot qui la supporte. Ainsi pour l'anglais, le pronom ‹*il*› peut se traduire par ‹*he*› ou ‹*it*› suivant les cas (‹*it*› pour notre exemple) et en allemand, où il existe un troisième genre pour les noms, le *neutre*, on pourra avoir respectivement ‹*er*›, ‹*sie*› ou ‹*es*› (‹*er*› dans notre exemple). De plus, le calcul d'une représentation peut être améliorée par la recopie du vecteur sur le terme support.

Relation d'identité Nous appelons relation d'identité la relation qui unit les mots du texte qui font référence à une même entité. Ainsi, dans la phrase « *Le chat est monté sur la chaise (...) L'animal s'assoupit.* », ‹‹chat›› et ‹‹animal›› font référence à la même entité. Nous verrons dans la partie 7.2.1.5 les exemples de relation d'identité que les fonctions lexicales, en particulier la synonymie et l'hyperonymie, peuvent aider à résoudre.

7.2.1.3 Rattachement des groupes prépositionnels

Le problème du rattachement des groupes prépositionnels consiste à trouver le lien de dépendance qu'il existe entre un syntagme prépositionnel et une tête syntaxique (verbe, nom, adjectif) [Gala Pavia, 2003a, Gala Pavia, 2003b]. Par exemple, dans la phrase « *Il voit la fille avec un télescope.* » le groupe prépositionnel « *avec un télescope* » peut être rattaché au groupe nominal « *la fille* » ou au groupe verbal constitué par ‹*voir*›.

Le rattachement prépositionnel est, de façon assez nette, intéressant dans le domaine de la recherche d'informations mais aussi dans celui de la TA. En effet, prenons l'exemple d'une langue comme l'anglais qui utilise fréquemment des tournures verbales comprenant des prépositions et qui en modifient le sens. Si nous considérons la phrase « *The man took a ferry across the river.* », le rattachement le plus logique pour ‹*across*› serait le verbe ‹*to take*›. Nous aurions alors comme traduction en français « *L'homme a traversé la rivière en ferry.* ». Le rattachement à ‹*ferry*› change complètement le sens et donnerait comme traduction « *L'homme a pris un ferry à travers la rivière.* ».

7.2.1.4 Chemins interprétatifs possibles

La figure 7.1 présente l'analyse morpho-syntaxique de la phrase « *L'avocat est véreux.* ». Cette phrase a deux interprétations raisonnablement possibles : d'un côté il s'agit d'un « *auxiliaire de justice crapuleux* » et de l'autre d'un « *fruit pourri par les vers* ». Une analyse sémantique devrait pouvoir permettre de connaître ces deux possibilités, en d'autres termes d'extraire plusieurs chemins d'interprétations possibles.

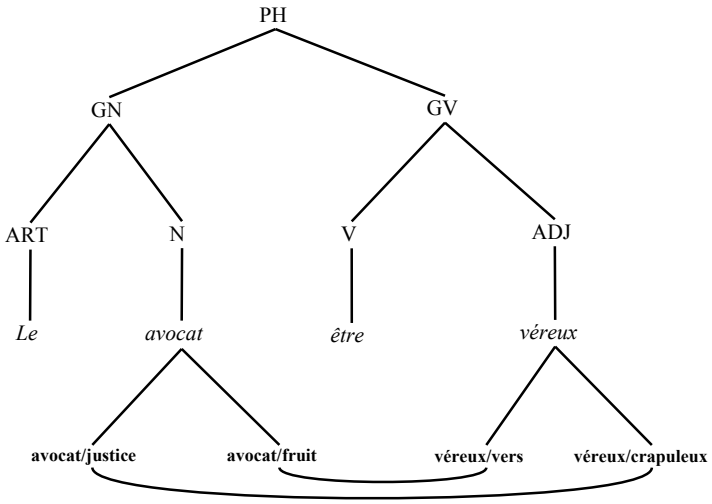


FIG. 7.1 – Analyse morpho-syntaxique de la phrase « *L'avocat est véreux.* » et ses deux interprétations raisonnables possibles.

7.2.1.5 Instanciation des fonctions lexicales

Le dernier phénomène sémantique qu'il nous paraît important d'étudier dans les textes est l'instanciation des fonctions lexicales. En effet, ces fonctions peuvent avoir leur importance non seulement dans le cadre des applications que nous visons mais aussi pour aider à la résolution des problèmes de l'analyse sémantique elle-même.

Utilisation des FLA pour la résolution des problèmes de l'analyse sémantique Dans l'analyse sémantique elle-même, les FLA peuvent apporter un certain nombre d'indices qui peuvent aider dans les différentes tâches que nous venons d'évoquer.

Aide à la désambiguïsation lexicale Les deux types de fonctions lexicales peuvent alors nous aider :

– *les FLACL* : Identifier une des relations syntagmatiques entre deux mots d'une phrase ou au moins supputer son existence peut aider à identifier les acceptions possibles pour l'item lexical correspondant. Ainsi, dans la phrase « *Lors de sa récente élection au sénat, monsieur Smith a obtenu une majorité écrasante.* » ‹*majorité*› peut être désambiguïsé, en partie, grâce à la fonction lexicale *Magn*. En effet, on peut considérer que ‹*majorité*› peut avoir comme acceptions possibles celles qui concerne l'**âge**, le **bureau**, le **vote** ou celle qui concerne l'**assemblée** mais seule *Magn*(*majorité*/**vote**) = ‹*écrasante*› et *Magn*(*majorité*/**assemblée**) = ‹*écrasante*› existent. De même la synonymie ou la fonction générique peuvent indirectement aider à la désambiguïsation via la relation d'identité.
– *les FLACM* : Ces fonctions formalisent les relations du monde qu'il peut exister entre les termes. Ainsi les informations comme « *Renault est en rapport avec les automobiles* » ou que « *Napoléon était empereur* » (le chef d'état et pas le poisson) peuvent aider à désambiguïser lexicalement le texte. La désambiguïsation peut se faire de nouveau ici de manière indirecte en particulier en identifiant les relations d'identité grâce à l'hyperonymie ou l'instanciation.

Aide à l'identification des relations d'identité Les relations d'identités sont, en partie, assurées par des termes équivalents dans le contexte. Ainsi, parmi les termes que l'on peut substituer à un autre, il y a bien entendu les synonymes mais aussi les hyperonymes. Connaître et identifier ces relations dans un texte peut ainsi être un élément déterminant pour la reconstitution du sens.

Aide aux rattachements prépositionnels Disposer d'informations de collocations, c'est-à-dire de fonctions lexicales syntagmatiques, peut aider au rattachement prépositionnel. Une méthode utilisant le Web a d'ailleurs été testée dans [Gala Pavia, 2003a]. Il s'agissait de créer un grand corpus pour extraire automatiquement des informations lexicales et statistiques sur les rattachements pour trouver les plus vraisemblables ensuite dans une analyse syntaxique en dépendance.

Modélisation du sens du texte pour une utilisation dans une application

Traduction automatique La traduction automatique est, sans conteste, l'application pour laquelle disposer de fonctions lexicales semble la plus intéressante. En effet, Igor Mel'čuk les a introduite, au début des années 1960, dans le but de résoudre une partie des problèmes posés par la TA. À l'époque, il cherchait « *une méthode simple permettant d'éviter les milliers de tests ennuyeux nécessaires pour permettre à l'ordinateur de trouver les équivalents russes de lexèmes anglais...* » [Mel'čuk, 1994]. Il remarque alors un fait bien connu des traducteurs, et qui se retrouve dans la plupart des langues, on associe certains termes à d'autres dans une alors qu'on n'utilise pas leurs équivalents directs pour marquer pourtant une idée semblable dans une autre. Ainsi, on parle de « *grosse fièvre* » en français mais pas de *« *big fever* » en anglais où on utilisera plutôt « *high fever* ». De même, en espagnol on utilise « *alta fiebre* » ou « *mucha fiebre* » et pas « *gran fiebre* ». Ces phénomènes ont été ainsi modélisés sous forme de fonctions lexicales considérées universelles car pouvant s'appliquer à n'importe quelle langue avec des moyens identiques. L'utilisation dans la TA peut alors se faire en considérant les FLA comme une sorte *interlingua*, c'est-à-dire comme une langue intermédiaire, comme le présente [Heylen *et al.*, 1994].

Résumé automatique Mehdi Yousfi-Monod et Violaine Prince travaillent au sein de l'équipe sur une technique de résumé automatique de textes par compression de phrases. Cette

approche utilise l'arbre morpho-syntaxique fourni par SYGFRAN et supprime les constituants de la phrase qui sont jugés peu importants pour conserver la pertinence du discours original [Yousfi-Monod & Prince, 2005b]. Ces suppressions se font sur la base de règles symboliques. Ainsi les gouverneurs ne sont jamais supprimés alors qu'en revanche certains compléments circonstanciels le sont. De même, l'adjectif épithète est supprimé lorsque le nom est précédé d'un pronom indéfini alors qu'il ne l'est pas s'il l'est par un pronom défini. Par exemple, la phrase « *Jean mange une pomme verte.* » sera résumée en « *Jean mange une pomme.* » puisque la règle sur l'épithète s'applique. En revanche, la même méthode ne supprimera pas l'adjectif dans la phrase « *Jean mange la pomme verte.* » puisque l'adjectif défini laisse entendre qu'il y a au moins une autre pomme d'une autre couleur.

Ce genre de règle compresse les textes de type narratif à hauteur d'environ 30% et fonctionnent très bien sauf pour certain cas de fonctions lexicales. En effet, sur une phrase comme « *Jean a eu une peur bleue.* », l'adjectif est supprimé (« *Jean a eu une peur.* ») ce que l'instanciation de la fonction lexicale *Magn* et une règle adéquate permettrait d'éviter.

Recherche d'informations On peut diviser le processus de recherche d'informations en deux phases. La première, l'*indexation des documents* consiste à fabriquer pour chacun d'eux une représentation calculatoire. La seconde, *la recherche des documents* consiste, à utiliser une requête, à la transformer en une représentation de même nature puis à extraire les documents les plus proches en fonction de critères donnés (cf. système SMART section 1.2.2.1).

Les fonctions lexicales peuvent être intéressantes dans le cas de la RI pour chercher les éventuelles synonymies de valeurs. Par exemple, l'indexation de documents qui consiste à faire une analyse sémantique instanciera les fonctions lexicales. On peut imaginer que la représentation du texte ne correspondra plus directement à des extraits de phrases comme « *une grande peur* » ou « *une majorité écrasante* » mais plutôt à *Magn*(‹*peur*›) et *Magn*(‹*majorité*›) ce qui permettra de les retrouver quelle que soit la valeur de la fonction. Ainsi des textes comportant « *peur bleue* » ou « *belle peur* » d'un côté et « *forte majorité* » ou « *majorité écrasante* » de l'autre pourront être retrouvés plus facilement qu'avec de simples systèmes distributionnels voire même des expansions de requêtes synonymiques.

7.3 Les limites d'une analyse sémantique en remontée-redescente

Nous venons de voir les différents points qu'il semblait nécessaire de résoudre dans le cadre d'une analyse sémantique. Ces points se résument techniquement à trois questions :

- Comment utiliser les informations du réseau lexical ?
- Comment faire des rattachements prépositionnels ?
- Comment identifier plusieurs interprétations possibles ?

Nous allons tenter d'apporter une solution à chacune de ces méthodes en essayant d'adapter l'analyse en remontée-descente pour qu'elle réponde à ces problèmes techniques.

7.3.1 Comment utiliser les informations du réseau lexical ?

Le réseau lexical correspond à la représentation dans le système de base lexicale sémantique des fonctions lexicales. Une méthode pour l'exploiter dans une analyse sémantique de type remontée-resdescente pourrait utiliser des règles sur l'arbre morpho-syntaxique antérieurement à toute autre opération. Il s'agirait de repérer les occurrences possibles de fonctions lexicales pour ensuite les utiliser dans l'analyse. Or, cette approche pose un problème important. Si la remontée-redescente et sa récurrence se prêtent bien à un arbre, elles ne sont absolument pas adaptées à un graphe où des cycles sont possibles.

7.3.2 Comment faire des rattachements prépositionnels ?

Les rattachements prépositionnels posent ici aussi les mêmes problèmes que l'utilisation des informations du réseau lexical. Ici aussi nous aurions un graphe au lieu d'un arbre ce qui empêche l'utilisation de la méthode en remontée-redescente.

7.3.3 Comment identifier plusieurs interprétations possibles ?

Considérons la phrase déjà donnée en exemple « *L'avocat est véreux.* ». Que se passe-t-il avec une analyse sémantique de type remontée-descente ? Nous obtiendrons un vecteur global qui regroupera l'ensemble des idées des items lexicaux pris individuellement. Ainsi nous aurons les idées de JUSTICE, FRUIT, ANIMAUX et MALHONNÊTETÉ qui ressortiront, c'est-à-dire les idées issues des deux interprétations acceptables sans qu'il ne soit possible de les distinguer l'une de l'autre.

Aucune solution pour pallier ce problème ne nous semble possible.

Tous ces problèmes semblent donc impossibles à résoudre avec une méthode de type remontée-redescente. Il nous a donc fallu nous tourner vers une méthode plus simple du moins du point de vue de la mise en place, une méthode basée sur des algorithmes à fourmis.

7.4 Algorithmes à fourmis et Analyse sémantique

7.4.1 Algorithmes à fourmis

7.4.1.1 Principe

Les algorithmes à fourmis ont pour origine la biologie et les observations réalisées sur le comportement social des fourmis. En effet, ces insectes ont collectivement la capacité de trouver le plus court chemin entre leur nid et une source de nourriture. Leur étude a ainsi soulevé plusieurs questions :

- pourquoi le groupe est-il cohérent alors que chaque individu semble autonome ?
- Comment les activités de tous les individus sont-elles coordonnées sans supervision ?

Il a pu être démontré que la coopération au sein de la colonie est auto-organisée et résulte d'interactions entre les individus. Ces interactions, souvent très simples, permettent à la colonie de résoudre des problèmes compliqués. Ce phénomène est appelé intelligence en essaim [Bonabeau & Théraulaz, 2000]. Il est de plus en plus utilisé en informatique où des systèmes de contrôle centralisés gagnent souvent à être remplacés par d'autres, fondés sur les interactions d'éléments simples.

7.4.1.2 Utilisations des algorithmes à fourmis en informatique

En 1989, Jean-Louis Deneubourg étudie le comportement des fourmis biologiques dans le but de comprendre la méthode avec laquelle elles choisissent le plus court chemin et le retrouvent en cas d'obstacle. Il élabore ainsi le modèle stochastique dit de Deneubourg [Deneubourg *et al.*, 1989], conforme à ce qui est observé statistiquement sur les fourmis réelles quant à leur partage entre les chemins. Ce modèle stochastique est à l'origine des travaux sur les algorithmes à fourmis.

Le concept principal de l'intelligence en essaim est la stygmergie, c'est-à-dire l'interaction entre agents par modification de l'environnement (cf. 5.2.1.2). Une des premières méthodes que l'on peut apparenter aux algorithmes à fourmis est l'écorésolution qui a montré la puissance d'une heuristique de résolution collective basée sur la perception locale, évitant tout parcours

explicite de graphe d'états [Drogoul, 1993]. Les agents sont, dans ce cas, tout à fait autonomes et n'ont aucune stratégie prédéfinie. Informellement on peut dire qu'ils cherchent seulement à être dans une bonne situation et à éviter les mauvaises.

En 1992, Marco Dorigo est un pionnier dans le développement des fourmis artificielles par la conception du premier algorithme basé sur ce paradigme et appliqué au célèbre problème combinatoire du voyageur de commerce [Dorigo & Gambardella, 1997]. Dans les algorithmes à base de fourmis artificielles, l'environnement est généralement représenté par un graphe et les fourmis virtuelles utilisent l'information accumulée sous la forme de chemins de phéromone déposée sur les arcs du graphe. De façon simple, une fourmi se contente de suivre les traces de phéromones déposées précédemment ou explore au hasard, le cas échéant dans le but de trouver un chemin optimal dans le graphe. La quantité de phéromone joue ainsi le rôle d'heuristique [Hao *et al.*, 1999].

Ces algorithmes offrent une bonne alternative à tout type de résolution de problèmes modélisables sous forme d'un graphe. Ils permettent un parcours rapide et efficace et offrent des résultats comparables à ceux obtenus par les différentes méthodes de résolutions. Leur grand intérêt réside dans leur capacité à s'adapter à un changement de l'environnement. Un ouvrage récent [Dorigo & Stützle, 2004] fait le point sur cette question.

7.5 Analyse sémantique par algorithme à fourmis mono-caste et mono-environnement

7.5.1 Précisions historiques

Mathieu Lafourcade a, pour la première fois, exploité le paradigme des fourmis dans le cadre de l'analyse sémantique dans le but d'apporter une solution à trois des problèmes de l'analyse sémantique présentés en 7.2.1 : l'ambiguïté lexicale, le rattachement des groupes prépositionnels et la découverte de chemins interprétatifs possibles. À l'époque de ces premières expériences, le modèle des vecteurs conceptuels utilisé est celui présenté au chapitre 2. Il n'y a ainsi aucune information sur les fonctions lexicales et les seules données disponibles sont issues des LEXIES, des ITEMS LEXICAUX et de l'arbre morpho-syntaxique fourni par SYGFRAN. [Lafourcade, 2003] et [Lafourcade & Guinand, 2005] présentent ce modèle qui est à la base des expériences sur les algorithmes à fourmis menées dans cette thèse. Nous en donnons les principes généraux qui sont repris dans la suite mais pas les heuristiques de choix dans les déplacements ni les valeurs numériques utilisées. En effet, nous avons jugé non seulement que l'intérêt était relativement faible puisque ce modèle est aujourd'hui en partie dépassé mais aussi qu'ils amèneraient plus de bruit que d'éclaircissements sur notre discours. Une partie du vocabulaire a été revue pour être plus compatible avec le modèle proposé dans cette thèse.

7.5.2 Principe et définitions

7.5.2.1 Amorçage

Sur l'arbre morpho-syntaxique obtenu à partir de SYGFRAN, on associe à l'ensemble des sens du terme une fourmilière. Les sens sont fabriqués ici par regroupement des LEXIES des mots du texte effectué par un expert afin de tester la faisabilité de l'expérience sur des données relativement sûres. On remarquera que ces regroupements correspondent aux ACCEPTIONS de BLEXISMA qui, elles, sont fabriquées automatiquement (cf. chapitre 5).

Pour parfaire l'amorce, on place sur chaque nœud une quantité d'énergie (ou nourriture selon la métaphore biologique) qui correspond à la récompense des fourmis et un vecteur conceptuel unitaire (toutes les composantes égales), l'*odeur* du nœud.

7.5.2.2 Simulation

La simulation consiste en une itération potentiellement infinie de cycles. À tout moment, la simulation peut être interrompue et l'état courant observé. Durant un cycle, on effectue les tâches suivantes :

- éliminer les fourmis trop vieilles (nombre de cycles fixé) ;
- pour chaque fourmilière, solliciter la production d'une fourmi (une fourmi peut ou non voir le jour, de façon probabiliste) ;
- pour chaque arc, diminuer le taux de phéromone (évaporation des traces) ;
- pour chaque fourmi : déterminer son mode (recherche de nourriture, retour à la fourmilière) et la déplacer. Créer un pont interprétatif le cas échéant ;
- calculer les conséquences du déplacement des fourmis (sur l'activation des arcs et l'énergie des nœuds) ;

D'une façon informelle, on peut résumer le déplacement d'une fourmi comme suit. Une fourmi qui vient de naître (ie. être produite par sa fourmilière) part à la recherche de nourriture. Elle est attirée par les nœuds qui portent beaucoup d'énergie. Elle ramasse autant d'énergie qu'elle peut en porter (et de ce qui est disponible) et se promène ainsi sur l'arbre. Au fur et et à mesure, elle transporte de plus en plus de nourriture et la probabilité de souhaiter rentrer à la fourmilière augmente. Lorsqu'elle veut rentrer, elle se déplace en suivant (statistiquement) les chemins qui contiennent la phéromone de sa fourmilière, et y retourne ainsi pour y déposer son chargement. Si elle rencontre une autre fourmilière (non concurrente) elle dépose également une petite partie de son chargement.

7.5.3 Les fourmilières

7.5.3.1 Caractéristiques

On associe à chaque ACCEPTION des mots du texte une fourmilière F_A caractérisée par :

- un *niveau d'énergie* : la quantité de nourriture sur le nœud.
- le *vecteur conceptuel de l'acception,* $V(F_A)$: ce vecteur est fixe et provient de la base de données vectorielles.

7.5.3.2 Production de fourmis

Les fourmilières produisent des fourmis de façon pseudo-aléatoire à chaque cycle de la simulation. Cette production se fait selon une fonction de profil sigmoïde en fonction de la quantité de nourriture qui se trouve dans la fourmilière.

$$[-\infty,+\infty] \rightarrow [0,1] : Sigmo(x) = \frac{arctan(x)}{\pi} + \frac{1}{2} \tag{7.1}$$

Lorsque la fourmilière produit une fourmi, elle dépense une partie de son énergie. Cette quantité, notée E_f, sera déposée sur le nœud où mourra la fourmi (cf. 7.5.6). Une fourmilière peut produire une fourmi même si son niveau d'énergie arrive en dessous de 0. Dans ce cas, la fourmilière "emprunte à la nature" pour tenter de s'imposer face aux autres. On dit qu'elle est *inhibée*. Cet état se prolongera et s'aggravera si effectivement rien ne permet à la fourmilière de combler son retard tandis que dans le cas inverse, il lui permettra de se revigorer. Ainsi, la fonction sigmoïde permet, même pour une fourmilière fortement inhibée, de conserver des chances de faire naître une fourmi (cf. figure 7.2). L'ACCEPTION associée à la fourmilière conserve donc des chances d'émergence même dans un milieu hostile.

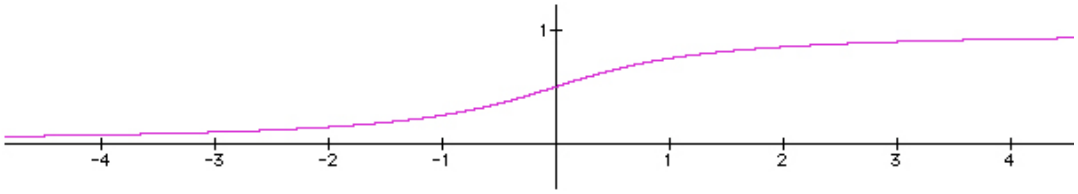


FIG. 7.2 – La fonction utilisée pour la production des fourmis. $\frac{arctan(x)}{\pi} + \frac{1}{2}$

7.5.4 Les fourmis

7.5.4.1 Caractéristiques des fourmis

Chaque fourmi f_{A_i} a les caractéristiques suivantes :
- *une durée de vie* : Il s'agit d'un nombre de cycles prédéterminés qui correspond au nombre de déplacements qu'une fourmi peut effectuer avant de mourir ;
- *un niveau d'énergie E(f)* : lors de ses déplacements, la fourmi ramasse de la nourriture. La quantité maximale qu'elle peut transporter est bornée par $E_{\max}(f)$. À sa mort, la fourmi restitue l'ensemble de l'énergie portée au nœud où elle se trouve plus l'énergie E_f nécessaires à sa fourmilière mère pour la produire. Le niveau d'énergie influe sur le mode de déplacement de la fourmi (recherche ou retour) ;
- *une référence à la fourmilière maison* : la fourmilière F_A où la fourmi f_{A_i} est née ;
- *un mode* : il détermine le comportement de la fourmi. Il peut être *recherche de nourriture* ou *retour à la fourmilière maison*.
- *un vecteur conceptuel, $V(f_{A_i})$* correspondant à celui de la fourmilière maison. On a ainsi pour chaque fourmi f_{A_i} issue de la fourmilière F_A, $V(f_{A_i}) = V(F_A)$.

7.5.4.2 Types de nœud du point de vue d'une fourmi

Pour une fourmi, un nœud peut être :
- *la fourmilière maison* : la fourmilière où la fourmi est née ;
- *une fourmilière ennemie* : elle correspond à une autre ACCEPTION du même l'item lexical. Elle a le même père que la fourmilière maison ;
- *une fourmilière potentiellement amie* : elle correspond à une des ACCEPTIONS d'un autre item lexical que celui de la fourmilière maison ;
- *un nœud qui n'est pas une fourmilière.* Dans le modèle présenté dans cette section, il s'agit d'un nœud de l'arbre morpho-syntaxique.

7.5.5 Déplacements, création et suppression de ponts

7.5.5.1 Traces influençant le déplacement

Phéromone Les fourmis laissent des traces sur les arcs où elles passent sous la forme de phéromone. Ainsi, au temps t, un arc A porte une quantité de phéromone notée $\varphi_t(A)$ telle que $\varphi_t(A) \in \mathbb{R}^+$.

Dépôt de phéromone Lors d'un déplacement, une fourmi laisse une trace en déposant sur l'arc traversé une quantité de phéromone $\theta \in \mathbb{R}^+$ telle que :

$$\varphi_{t+1}(A) = \varphi_t(A) + \theta \tag{7.2}$$

Évaporation de la phéromone À chaque cycle, il y a une légère évaporation de la phéromone. Cette baisse se fait de façon linéaire jusqu'à la disparition totale de la phéromone :

$$\varphi_{t+1}(A) = \min(\varphi_t(A) - \delta, 0) \tag{7.3}$$

où $\delta \in \mathbb{R}^+$ est la quantité de phéromone qui s'évapore à chaque cycle.

Propagation de vecteurs Lors de son déplacement sur les nœuds internes de l'arbre, une fourmi propage son vecteur conceptuel. Le vecteur V(N) porté par un nœud N est modifié lors du passage d'une fourmi. Les fourmilières sont les seuls nœuds dont le vecteur ne peut pas être modifié par le passage d'autres fourmis.

Le déplacement des fourmis implique une propagation des vecteurs et participe à l'effet stygmergétique (traces). Cependant, l'effet est ici rationnel pour les fourmis car elle ont tendance, selon leur type, à choisir leur destination selon les vecteurs porté par les nœuds. Ce phénomène permet aux fourmis de revenir à leur fourmilière, ou éventuellement de se tromper et de se diriger vers des fourmilières amies. Cette erreur est potentiellement bénéfique puisqu'elle peut permettre de créer un pont entre les deux fourmilières (cf 7.5.5.3).

À tout moment d'une analyse sémantique, le vecteur conceptuel du texte est donné par le vecteur du sommet de l'arbre.

7.5.5.2 Déplacements

Modes de déplacement des fourmis Les fourmis peuvent être dans deux modes de déplacement différents le mode recherche et le mode retour. Le passage d'un mode à l'autre se fait aléatoirement en fonction de la quantité de nourriture transportée.

$$P(retour) = \frac{E(f)}{E_{\max}(f)} \tag{7.4}$$

où E est la quantité de nourriture transportée par la fourmi et $E_{\max}$ la quantité de nourriture maximale qu'elle peut transporter.

À chaque tour, une fourmi se déplace. Les destinations possibles N_i sont les nœuds accessibles via les arcs A_i selon la géométrie de l'arbre.

$$P(N_i, A_j) = \max \begin{cases} \frac{Eval(N_i, A_j)}{\sum_{k=1,l=1}^{k=n,l=m} Eval(N_k, A_l)} \\ \varepsilon \end{cases} \tag{7.5}$$

Les destinations possibles sont l'ensemble des nœuds connectés au nœud courant par un arc auquel on enlève celui d'où vient la fourmi. Cette contrainte empêche les fourmis de faire des allers-retours constants sur un arc. En revanche, en cas de nœud terminal (ie. le seul nœud reliant celui où la fourmi se situe est celui d'où elle vient), la fourmi peut revenir.

L'évaluation en mode recherche En mode recherche, les fourmis sont à la recherche de nourriture. Elles sont ainsi attirées par les nœuds qui ont beaucoup d'énergie mais elles évitent de passer par des arcs sur lesquels trop de fourmis seraient déjà passées afin d'explorer le plus grand nombre de solutions possibles. L'évaluation en *mode recherche* d'une destination est donc

inversement proportionnelle à l'excitation de l'arc et proportionnelle à la quantité d'énergie du nœud.

L'évaluation en mode retour En *mode retour*, les fourmis cherchent à retourner à leur fourmilière pour y déposer la nourriture trouvée. Elles vont ainsi plutôt prendre les arcs très excités et se diriger vers les nœuds dont le vecteur est le plus proche possible du leur. L'évaluation des destinations possibles est donc proportionnelle à l'excitation de l'arc et inversement proportionnelle à l'angle entre son vecteur conceptuel $V(f)$ et l'odeur du nœud destination $V(N)$.

7.5.5.3 Création et suppression de ponts

Un pont peut être créé lorsqu'une fourmi atteint une fourmilière potentiellement amie, c'est-à-dire lorsqu'elle arrive sur un nœud qui correspond à une des ACCEPTIONS d'un autre item lexical que celui de la fourmilière mère. Dans ce cas, la fourmi évalue le nœud correspondant à sa fourmilière maison comme les nœuds géométriquement liés à cette fourmilière potentiellement amie. Si ce nœud est sélectionné, il y a création d'un pont entre les deux fourmilières. Ce pont est ensuite considéré comme un arc standard par les fourmis. Un pont est un arc standard et dispose d'un niveau d'excitation. Si ce niveau d'excitation atteint une valeur trop proche de zéro, le pont est alors détruit. Un pont peut être vu comme une compatibilité entre deux fourmilières, un chemin interprétatif possible.

7.5.5.4 Exemple

Prenons l'exemple de la phrase « *L'avocat plaide à la cour.* ». Comme le montre la figure 7.3, on considère ici que ‹*plaider*› n'a qu'un sens, et que ‹*avocat*› et ‹*cour*› en ont deux. Les fourmilières vont commencer à produire des fourmis. Lorsqu'une fourmi de *avocat*/**justice** arrive sur la fourmilière ‹*plaider*›, elle tente d'établir un pont avec sa fourmilière de départ ce qui sera relativement probable vu que les idées entre les deux vecteurs sont relativement proches. D'autres fourmis seront susceptibles d'en faire de même particulièrement en mode retour où la phéromone attire les fourmis ce qui renforcera d'autant plus le pont. Les fourmis de ‹*plaider*› agissent sur ce pont dans la direction opposée. La création de pont entre fourmilières amies (compatibilité de sens) va leur permettre de s'allier puisque les fourmis de ces fourmilières vont exciter particulièrement les arcs concernés ce qui attirera encore plus de fourmis susceptibles de rapporter de l'énergie à ce sous-système et l'entretenir. Il en sera de même entre ‹*plaider*› et *cour*/**justice** ainsi qu'entre *avocat*/**justice** et *cour*/**justice**.

En revanche, une fourmi d'*avocat*/**fruit** n'aurait pas plus de chance de construire un pont entre ‹*plaider*› et sa fourmilière mère que de revenir en arrière. Ainsi même s'il est construit, il ne sera pas maintenu par le passage de fourmis (phéromone) et finira par disparaître.

7.5.6 Énergie

Au début de la simulation, le système possède une certaine énergie qui est répartie équitablement sur chacun des nœuds. Les fourmilières utilisent celle qu'elles possèdent pour fabriquer des fourmis. Ces dernières se déplacent dans l'environnement et ramènent de l'énergie aux fourmilières qui l'utiliseront pour produire d'autres fourmis. À la mort d'une fourmi, l'énergie qu'elle porte et l'énergie qu'il a fallu pour la produire se déposent sur le nœud où elle se trouve. Il n'y a donc ni perte ni apport d'énergie à aucun moment que ce soit (si on excepte l'emprunt à la nature que peuvent faire de façon très limitée les fourmilières). Le système fonctionne complètement en vase clos.

La quantité d'énergie est un élément fondamental de la convergence du système vers une solution. En effet, puisque l'énergie globale est limitée, les fourmilières sont en concurrence les

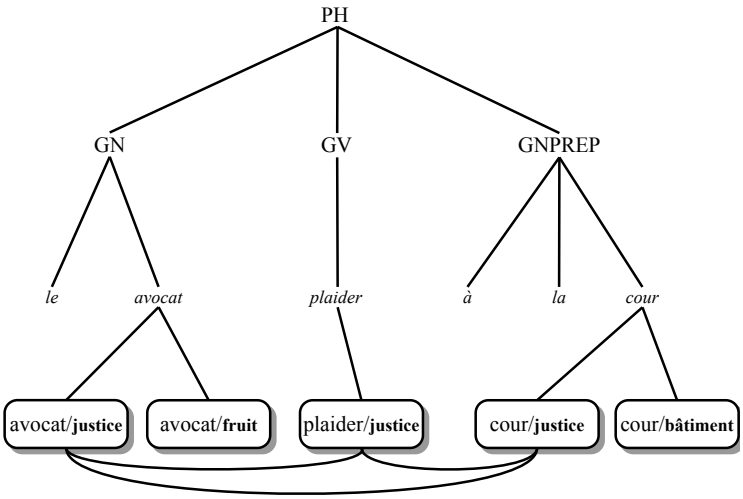


FIG. 7.3 – Exemple d'analyse sémantique avec la phrase « *L'avocat plaide à la cour.* »

unes avec les autres et seules des alliances peuvent permettre de faire émerger des solutions. Si nous avions une énergie non bornée, toutes les fourmilières recevraient de l'énergie et toutes seraient fortement activées et aucune ne serait inhibée.

7.5.7 Mode d'évaluation

L'évaluation du modèle en termes d'analyse linguistique constitue en lui-même un problème difficile. Examiner manuellement les populations de fourmis, les activations des nœuds, et les ponts, pour un texte donné, prend trop de temps pour être effectué à grande échelle. Pour avoir une idée plus large des performances du modèle, on préfère l'évaluer en fonction de la résolution des problèmes d'ambiguïtés posés dans un corpus.

Un ensemble d'une quarantaine de textes courts a ainsi été constitué pour permettre de comparer les différents modèles présentés dans ce chapitre. Ces textes ont été sélectionnés pour leur représentativité des phénomènes sémantiques que nous cherchons à résoudre (cf. 7.2.1). Dans ce corpus, chaque phrase a été annotée manuellement de manière à décrire, dans l'idéal, son analyse sémantique complète. Ce corpus se trouve en annexe D.

7.5.7.1 Annotation du corpus

Formalisme d'annotation Pour chacune des phrases, il s'agit de décrire chaque interprétation possible :

- *l'*ACCEPTION *utilisée pour chaque mot* ;
- *les références* que ce soit des relations anaphoriques ou des relations d'identité ;
- *le rattachement des groupes prépositionnels* ;
- *l'instanciation des fonctions lexicales.*

La méthode la plus simple pour décrire de telles relations entre les mots semble être l'utilisation d'un graphe où les nœuds correspondent aux acceptions des mots de la phrase et les arcs étiquetés aux relations entre les mots ; en d'autres termes, sur l'environnement utilisé par les fourmis ne considérer que les fourmilières et les arcs qui les lient entre elles. Nous avons ainsi plusieurs types d'arcs pour une évaluation, chacun correspondant à un phénomène particulier :

- un *arc d'interprétation* ;

– un *arc de référence* ;
– un *arc de rattachement de groupe prépositionnel* :
– un *arc d'instanciation de fonction lexicale d'analyse*

Il est important de comprendre que dans les deux premiers modèles présentés ici, les arcs issus du système ne sont pas étiquetés, les arcs ont donc été nommés manuellement.

Si nous prenons comme exemple la phrase « *L'avocat est véreux.* », nous aurions le graphe :

nœuds : ‹*Le*›, *avocat*/**justice**, *avocat*/**fruit**, ‹*être*›, *véreux*/**crapuleux**, *véreux*/**vers**
arcs : interp(*avocat*/**justice**, *véreux*/**crapuleux**) ; interp(*avocat*/**fruit** , *véreux*/**vers**)

La figure 7.4 présente le dessin de ce graphe (à rapprocher de la figure 7.1). Par convention, en mode écrit ou en mode dessin on respecte l'ordre de l'analyse syntaxique obtenue de SYGFRAN.

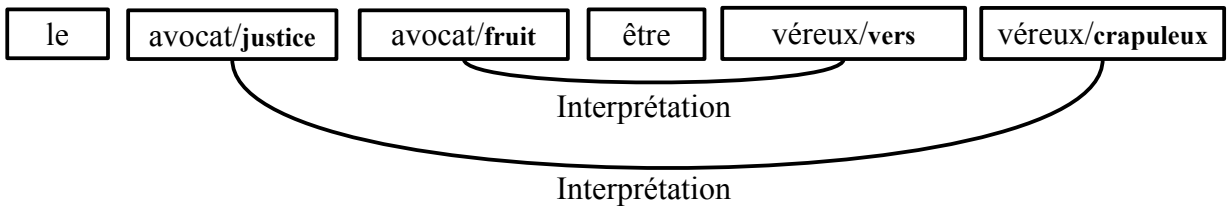


FIG. 7.4 – Graphe d'évaluation de l'analyse de la phrase « *L'avocat est véreux.* »

De même un exemple, mettant en jeu une FLA serait la phrase « *Jean a eu une peur bleue.* » :

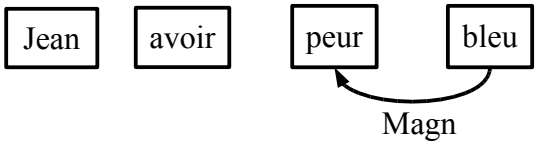


FIG. 7.5 – Le graphe interprétatif de la phrase « *Jean a eut une peur bleue.* ».

Par la suite on appelle *graphes référents* les graphes obtenus manuellement.

7.5.7.2 Annotation du corpus par le système et principe d'évaluation

Au bout de cinq minutes[101] d'analyse de chaque énoncé, le calcul est interrompu. Seules les fourmilières dont le niveau d'activation est supérieur à 0 sont conservées. En d'autres termes, les sens inhibés sont ignorés ainsi que les éventuels arcs dont ils seraient un sommet (ce qui est fort peu probable).

Habituellement, on compare des résultats selon la méthode classique de rappel-précision. La précision est le nombre d'étiquetages corrects sur le nombre d'étiquetages effectués tandis que le rappel est le nombre d'étiquetages corrects sur le nombre d'occurrences à étiqueter. Pour chaque phénomène que nous cherchons à désambiguïser, on comparera ainsi le graphe issu du système au graphe référent. Ensuite :

– *pour les acceptions* : on précisera les résultats suivant la catégorie morphologique du terme (nom, adjectif, verbe et adverbe) ;

[101] Cette durée a été choisie car nous la considérons comme suffisamment longue puisque dans tous nos tests, nous n'avons pas trouvé de convergence dépassant les deux minutes.

– *pour les arcs* : ces calculs seront effectués pour chacun des types d'arc (interprétation, référence, rattachement, instanciation).

7.5.8 Résultats

acceptions	global	noms	adjectifs	verbes	adverbes
rappel	0,64	0,66	0,69	0,48	0,72
précision	0,7	0,72	0,75	0,5	0,75
arcs	global	interprétation	rattachement prep	références	instanciation FLA
rappel	0,28	0,64	0,3	0	0
précision	0,29	0,6	0,41	0	0

FIG. 7.6 – Evaluation du système mono-environnement et mono-caste.

Il est important de noter que les résultats de cette expérience, donnés dans la figure 7.6, ne peuvent pas être comparés aux résultats d'évaluation d'autres approches (évalués dans le cadre de la campagne SENSEVAL[102]). En effet, notre expérience a porté sur des textes en français dont les résultats ne sont pas disponibles pour Senseval-2[103]. La comparaison est donc relativement indirecte, car ne portant pas sur les mêmes langues ni sur les mêmes corpus. Il faut ainsi considérer ces résultats uniquement comme un moyen de comparaison des modèles à fourmis présentés dans ce chapitre.

Pour le moment, faute de pouvoir comparer avec un autre modèle, on ne peut que commencer à dégager certains phénomènes en désambiguïsation lexicale. On constate en particulier, et ce n'est pas une surprise, que la catégorie la plus difficile à désambiguïser sont les verbes qui sont très polysémiques (avec des distinctions de sens souvent subtiles). Les adverbes, d'un autre côté, sont plus faciles à traiter car beaucoup moins polysémiques.

On peut noter que les références et l'instanciation de fonctions lexicales sont nulles ce qui est logique puisque ce n'est à la fois pas possible et pas prévu dans ce modèle.

7.5.9 Critique du modèle

Ce modèle est une bonne base de départ en vue d'une analyse sémantique, toutefois, il ne tient pas compte (vu la date des premières expériences[104]) des avancés réalisées par l'équipe à propos de la représentation du sens et l'utilisation des fonctions lexicales. Il ne peut donc pas analyser correctement une phrase comme « *L'avocat mange une glace.* » qui clairement ne peut être considérée par le système que comme « *L'*avocat*/**fruit** mange une glace.* » puisque les informations thématiques mutuelles indiquent fortement l'idée de nourriture. De même dans la phrase « *La pelle se casse.* », strictement aucune information thématique ne permet de favoriser une ou l'autre des acceptions de ‹*pelle*›.

La solution est d'utiliser le réseau lexical évoqué au chapitre précédent, solution utilisée une première fois dans FOETAL.

[102] http://www.itri.brighton.ac.uk/events/senseval/
[103] http://www.sle.sharp.co.uk/senseval2/
[104] En 2002-2003

7.6 Analyse sémantique par algorithme à fourmis multi-caste à environnements séparés

7.6.1 Analyse sémantique par algorithmes à fourmis multi-caste à environnements séparés

L'objectif d'une analyse sémantique serait dans l'idéal de permettre de résoudre l'ensemble des phénomènes sémantiques présentés dans la section 7.2.1. La résolution de ces tâches fait appel à des compétences et des informations de nature diverse. Non seulement, il est, dans la plupart des cas, difficile de modéliser la résolution de chacun de ces problèmes, mais il est encore plus problématique de combiner à l'aide d'un coordinateur chacune des expertises. Les critères sont souvent contradictoires et leurs éventuelles pondérations sont souvent fonction des autres critères. Au final, la pierre d'achoppement n'est pas la conception d'agents experts dans tel ou tel phénomène linguistique, mais plutôt la définition précise de la fonction d'agrégation des réponses retournées par les agents.

C'est pour apporter une solution à ce problème que Mathieu Lafourcade a étendu le modèle des algorithmes à fourmis grâce à un système de castes. Chaque caste correspond à une heuristique destinée à résoudre un problème particulier et a ainsi un comportement propre influencé en partie par celui des autres.

L'un des principaux avantages des castes est ainsi de permettre une relativement simple extensibilité puisqu'il suffit de rajouter une caste pour tenir compte de tel ou tel phénomène.

7.6.2 Le modèle FOETAL

Le modèle FOETAL (FOurmis et Émergence pour le Traitement Automatique des Langues) a été mis au point par Thibault Zamora au cours de son DEA achevé en Juin 2005 [Zamora, 2005]. Contrairement à la section précédente dont les expériences ont été réalisées antérieurement à la création de BLEXISMA, il a pu s'appuyer sur ce prototype et les avancées en matière de représentation du sens présentées dans cette thèse. Il se base ainsi sur les trois types de sources de données que nous avons présentées dans les chapitres précédents :

1. l'arbre d'analyse morphosyntaxique du texte à analyser,
2. une représentation thématique des sens de termes basée sur des vecteurs,
3. une représentation des relations prototypiques (prédicats, arguments) basée sur un réseau lexical.

L'arbre d'analyse fournit la structure sur laquelle les fourmis vont extraire des informations syntaxiques et lexicales présentes dans le texte. Ces informations vont être exploitées, soit dans le registre de l'information mutuelle à l'aide des vecteurs, soit dans le registre de la concordance d'arguments à l'aide du réseau lexical.

7.6.3 Précisions sur le réseau lexical utilisé dans ces expériences

Dans le chapitre précédent, nous avons présenté les informations qu'un réseau lexical devrait contenir pour permettre à la fois la modélisation des fonctions lexicales d'analyse mais aussi pour permettre leur utilisation dans le cadre d'une analyse sémantique. Des techniques ont commencé à être exploitées pour le mettre au point automatiquement mais il s'agit d'un travail à la fois long et encore tributaire de bien des recherches. Nous avons ainsi dû concevoir un réseau lexical manuellement pour l'étude de ce corpus. Il s'agit d'un réseau semblable à celui de la figure 6.4 et qui a été mis au point à partir des relations contenues dans les phrases. On peut ainsi considérer qu'il s'agit du réseau le plus précis que l'on pourrait avoir à disposition, l'objectif étant ici d'essayer de constater la faisabilité d'une telle analyse.

7.6.4 Fourmilières, castes et fourmis

L'environnement de FOETAL est constitué de deux parties : l'arbre morpho-syntaxique et un réseau lexical. Tout comme dans l'exemple précédent, les fourmilières se trouvent sur les ACCEPTIONS qui elles-mêmes sont reliées au réseau lexical.

Dans ce modèle, il y a deux castes principales dont l'environnement est séparé. La première se déplace sur l'arbre morpho-syntaxique tandis que la seconde se déplace sur le réseau lexical. Les fourmis de ces deux castes ne peuvent ainsi se croiser que sur les fourmilières et les ponts.

7.6.4.1 Caste de fourmis à vecteurs (T).

L'environnement pour les fourmis issues de la caste ***T*** (avec ***T*** comme thématique) a un comportement pratiquement identique à celui présenté dans le cadre de l'environnement unique et mono-caste présenté dans la section précédente (cf. 7.5).

7.6.4.2 Caste de fourmis à réseau lexical (R).

La caste ***R*** rassemble les fourmis ayant comme environnement un réseau lexical. Comme pour les fourmis de la caste ***T***, les mêmes fourmilières sont associées à chaque ACCEPTION de chaque terme du segment textuel étudié. Chaque fourmilière est ensuite reliée à un nœud du réseau lexical (dans lequel chaque ACCEPTION est aussi un nœud). Les fourmis se déplacent dans le réseau lexical comme s'il s'agissait d'un labyrinthe. Leur objectif ici n'est pas une recherche de nourriture (même si on pourrait le modéliser ainsi) mais de chercher une sortie acceptable dans le réseau lexical en ne revenant pas (ou peu) sur ses pas. Les seules sorties possibles du labyrinthe étant les *fourmilières amies* telles que définies en 7.7.3.3. Comme les fourmis ***T***, les fourmis ***R*** arrivant sur une fourmilière ont une forte probabilité de créer ou d'emprunter un *pont* pour rentrer dans leur fourmilière (cf. 7.5.5.3).

Certaines restrictions ont lieu dans le choix des destinations possibles, la principale étant de ne pas pouvoir prendre deux fois de suite un arc différent représentant une relation *sens-de* car cela peut court-circuiter le réseau dans des phrases telles que *L'avocat mange un avocat.*

Les fourmis ***R*** sont spécialisées en fonction d'un prédicat. Le prédicat étant dans un segment l'élément articulatoire principal. Trois sous-castes (avec leur symétrique) ont été étudiées :

- le prédicat cherche l'agent (***PA***) et (l'agent cherche le prédicat (***AP***)) ;
- le prédicat cherche le patient (***PC***) et (le patient cherche le prédicat (***CP***)) ;
- le prédicat cherche l'instrument (***PI***) et (l'instrument cherche le prédicat (***IP***)).

Les fourmis ***R/PA*** favorisent les arcs représentant les relations de type *patient-de*, *est-un* (et son inverse), *sens-de* (et son inverse) au détriment principalement des relations de type *agent-de* et *instrument-de*.

Les fourmis ***R/PC*** favorisent quant à elles les relations *agent-de*, *est-un* (et son inverse), *sens-de* (et son inverse) alors que les relations de type *objet-de* et *instrument-de* auront peu de poids.

Enfin les fourmis ***R/PI*** seront plus attirées par les relations de type *instrument-de*, *est-un* (et son inverse), *sens-de* (et son inverse) que par les relations *agent-de* et *objet-de*.

Cette spécialisation des fourmis ***R*** permet de trouver des ponts reliant la ou les ACCEPTIONS du prédicat aux ACCEPTIONS des autres termes du segment textuel étudié. Il s'agit en fait d'un appariement entre les relations syntaxiques du texte disponibles sur l'arbre d'analyse morpho-syntaxique et les informations typologiques (de types connaissance du monde) disponible dans le réseau lexical.

7.6.4.3 Ponts

Dans FOETAL, les ponts sont créés, comme dans le modèle mCmE, lorsqu'une fourmi arrive sur une fourmilière potentiellement amie. La fourmilière mère est alors considérée comme un chemin possible pour la fourmi. Les ponts sont les seuls arcs que peuvent utiliser les fourmis des deux castes.

7.6.4.4 Résultats

Le tableau de la figure 7.7 présente l'évaluation de FOETAL pour les deux castes séparément et les deux castes combinées.

T :

acceptions	global	noms	adjectifs	verbes	adverbes
rappel	0,64	0,66	0,69	0,48	0,72
précision	0,7	0,72	0,75	0,5	0,75
arcs	global	interprétation	rattachement prep	références	instanciation FLA
rappel	0,28	0,64	0,3	0	0
précision	0,29	0,6	0,41	0	0

R :

acceptions	global	noms	adjectifs	verbes	adverbes
rappel	0,54	0,53	0,55	0,65	0,3
précision	0,58	0,55	0,57	0,68	0,4
arcs	global	interprétation	rattachement prep	références	instanciation FLA
rappel	0,31	0,47	0,68	0	0
précision	0,35	0,5	0,81	0	0

R* + *T :

acceptions	global	noms	adjectifs	verbes	adverbes
rappel	0,75	0,75	0,76	0,7	0,82
précision	0,77	0,77	0,79	0,73	0,78
arcs	global	interprétation	rattachement prep	références	instanciation FLA
rappel	0,41	0,71	0,81	0	0
précision	0,43	0,73	0,84	0	0

FIG. 7.7 – Évaluation du système avec une caste de chaque type et deux castes combinées.

On peut constater :
- la caste ***T*** considérée seule équivaut au modèle mono caste mono environnement présenté dans la section 7.5. Nous n'y revenons pas ici ;
- la caste ***R*** seule est, dans l'ensemble, moins efficace que celles de ***T*** sauf en ce qui concerne les verbes. En effet, les informations permettant de désambiguïser cette catégorie morphologique sont rarement thématiques particulièrement, semble-t-il, sur le corpus choisi. Ainsi, le réseau possède, contrairement aux vecteurs, des informations sur les arguments des verbes (agent, patient, ...) ce qui permet avantageusement de les désambiguïser. On remarquera que ce n'est pas le cas pour les autres catégories en particulier chez les adverbes.

En ce qui concerne les arcs, ceux d'interprétation souffrent de la baisse du niveau de désambiguïsation. En revanche en ce qui concerne les rattachements prépositionnels, le taux est largement supérieur ce qui s'explique ici aussi par le rôle central joué par les verbes. En effet, si on observe le réseau lexical test, la plupart des rattachements retrouvés grâce au thème sont contenus directement ou indirectement dans le réseau ;
– la combinaison des deux castes accroît nettement les résultats. Le constat sur les différences que nous notions entre les informations sur le thème et celles sur les rapports entre les termes dans le lexique qui nous semblaient relativement complémentaires sont ici confirmées. Tous les taux des castes combinées sont supérieurs à ceux des castes prisent séparément.

7.6.4.5 Critique du modèle FOETAL

Le modèle FOETAL, bien qu'original car utilisant pour la première fois les idées sur la représentation du sens présentées dans cette thèse, souffre de plusieurs défauts.

Difficulté à résoudre certains phénomènes sémantiques Rien n'est prévu dans le modèle pour permettre la résolution des références ainsi que de l'instanciation des fonctions lexicales d'analyse.

L'environnement séparé Dans ce modèle, l'environnement est séparé. Ainsi une fourmi de $\boldsymbol{T}$ ne sortira de l'arbre morpho-syntaxique que pour franchir les ponts. Il en est de même avec les fourmis de $\boldsymbol{R}$. Ainsi, les fourmis ne collaborent réellement directement que lorsqu'elles sont sur les fourmilières ou sur les ponts. Pour le reste, la collaboration ne se fait qu'indirectement.

Utilisation directe du réseau lexical Il s'agit du principal problème dont souffre ce modèle puisqu'il empêche son utilisation dans des cas réels. En effet, nous sommes alors confrontés à deux problèmes :

1. *comment réaliser plusieurs analyses simultanément ?* Il faudrait différencier les fourmis et la phéromone suivant les analyses ce qui est possible techniquement mais relativement lourd à mettre en œuvre et surtout facteur important de ralentissement du système ;
2. *comment gérer la présence de plusieurs occurrences d'un même item lexical se trouvant dans le texte à analyser ?* Elles seraient toutes reliées aux mêmes sommets ce qui entraînerait deux conséquences néfastes à la qualité de l'analyse :
   – les fourmis pourraient ainsi passer d'une partie du texte à analyser à une partie bien plus éloignée, et ce de façon extrêmement simple. Des informations lointaines syntaxiquement viendraient alors à avoir autant d'importance que des informations proches ;
   – on favoriserait les acceptions identiques puisque l'alliance des fourmilières serait alors grandement favorisée. Ainsi, considérons la phrase « *Il cliqua sur la souris pour enregistrer son programme au moment où le chat, trop occupé à poursuivre le morceau de polystyrène poussé par le courant d'air et qu'il avait pris pour une souris, lui heurta le bras.* ». Les deux acceptions de ‹*souris*› employées sont différentes et l'utilisation d'un réseau lexical où on trouverait que *souris*/**ordinateur** et *programme*/**ordinateur** sont liés ensemble influerait, bien entendu, sur le choix de la première occurrence mais aussi de façon disproportionnée sur le choix de la seconde. En effet, les fourmis issues de la fourmilière correspondant à cette dernière pourraient emprunter trop facilement les chemins déjà suivis par les fourmis de la première ce qui entraînerait nécessairement le choix de la même ACCEPTION.

C'est en tenant compte de tous les enseignements tirés du modèle mono-caste mono-environnement et du modèle FOETAL que nous avons mis au point, au cours de cette thèse, le modèle d'analyse sémantique par algorithmes à fourmis multi-caste et à environnements partagés.

7.7 Analyse sémantique par algorithme à fourmis multi-caste et à environnements partagés

Nous présentons ici le modèle d'analyse sémantique mis au point au cours de cette thèse. Il s'agit d'un système à fourmis multi-caste et à environnements partagés. Ici, une caste n'est donc pas bloquée artificiellement sur son environnement. Quelle que soit sa caste, une fourmi a un certain nombre de possibilités et la caste ne fait qu'influer sur les choix de déplacements. Ainsi, une fourmi de la caste ***cherche_hyperonymie*** aura tendance à privilégier l'utilisation de ce type d'arc. Quel que soit l'environnement où elle se trouve la fourmi est à la recherche de nourriture puis, pour la rapporter, elle souhaite revenir à son point de départ. Ainsi, on peut globalement considérer que nous avons adapté les heuristiques du modèle mono-caste et mono-environnement (dorénavant noté mCmE) au modèle multi-caste à environnements partagés (dorénavant noté MCEP).

7.7.1 Principe général et définitions

7.7.1.1 Amorçage

Il y a peu de différences avec l'amorçage du modèle mono-caste et mono-environnement (cf. 7.5). Sur l'arbre morpho-syntaxique obtenu à partir de SYGFRAN, on place :

- une fourmilière sur chaque ACCEPTION de l'item lexical ;
- sur chaque nœud une quantité d'énergie qui correspond à la récompense des fourmis. À chaque découverte d'un nœud du réseau lexical par une fourmi, on y placera aussi une quantité d'énergie égale. On remarquera que la quantité d'énergie reste constante pour le système global au maximum de son développement (cf 7.5.6) ;
- un vecteur conceptuel unitaire, l'*odeur* du nœud, sur chacun des nœuds de l'arbre.

Enfin, et c'est la principale différence avec les précédents modèles, on modifie l'arbre en créant des liens interphrases. Ces liens relient les nœuds correspondants aux phrases et permettent de modéliser leur ordre dans un texte comme le montre la figure 7.8. Ces liens interphrases nous serviront pour l'évaporation des phéromones.

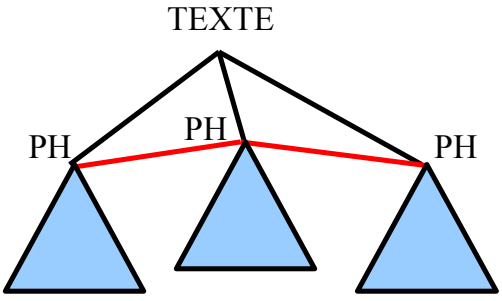


FIG. 7.8 – Liens interphrases.

7.7.1.2 Arcs

On distingue 4 types d'arcs :
- *les arcs de structure* : ce sont les arcs de l'arbre morpho-syntaxique ;
- *les arcs de réseau* : ce sont des arcs copiés du réseau lexical global.
- *les ponts* : ce sont les arcs créés dynamiquement par les fourmis entre deux ACCEPTIONS compatibles ;
- *les arcs interphrases* : ce sont les arcs rajoutés entre les nœuds de deux phrases consécutives.

Tous ces arcs sont orientés mais **les fourmis peuvent les franchir dans n'importe quel sens**.

7.7.1.3 Simulation

La simulation se déroule comme pour le mCmE à la différence près que lorsqu'une fourmi se trouve sur une fourmilière, elle a le choix entre un arc de l'arbre morpho-syntaxique ou un arc du réseau lexical. Si elle choisi de prendre un arc du réseau, on recopie le nœud d'arrivée et on y place une quantité d'énergie égale à celle que l'on a placée sur chacun des nœuds de l'arbre morpho-syntaxique durant la phase d'amorce. Tout comme une fourmilière, ce nœud correspond à une ACCEPTION, son vecteur ne peut donc être modifié. La fourmi cherchera ensuite à explorer de proche en proche d'autre nœuds et sera susceptible de construire un pont vers sa fourmilière mère si elle retrouve un nœud proche thématiquement.

7.7.2 Les fourmilières

7.7.2.1 Caractéristiques

Les caractéristiques des fourmilières sont strictement identiques à celles du mCmE (cf. 7.5).

7.7.2.2 Production de fourmis

Comme pour le mCmE, à chaque cycle de la simulation, les fourmilières produisent des fourmis d'une façon pseudo-aléatoire qui varie suivant la quantité de nourriture se trouvant dans la fourmilière (cf. 7.5.3.2). La fourmilière produit aléatoirement une fourmi parmi les castes existantes. Ce tirage est influencé par les fourmis qui rapportent de la nourriture. Ainsi, plus il y aura de fourmis de la caste c qui rapporteront de la nourriture plus la fourmilière produira des fourmis de cette classe.

7.7.3 Les fourmis

7.7.3.1 Caractéristiques des fourmis

Les caractéristiques des fourmis sont identiques à celles du mCmE (cf 7.5.4.1) à la différence qu'elles ont une *caste* qui influe sur leur mode de déplacement.

7.7.3.2 Castes

Dans le modèle MCEP, une caste n'est pas bloquée sur un environnement comme dans le modèle FOETAL. Parmi les castes, il y a la caste ***standard*** dont les membres ont un comportement similaire à celui des fourmis du modèle mCmE. Les membres des autres castes ne sont que des fourmis de ***standard*** pour lesquelles la probabilité d'emprunter un arc correspondant à leur relation privilégiée est plus importante.

Suivant leur caste, les fourmis déposent en plus de la phéromone de passage une phéromone spécifique. Cette phéromone influe aussi sur les choix de déplacement des fourmis. Les fourmis de ***standard*** n'ont pas une telle phéromone.

7.7.3.3 Types de nœud du point de vue d'une fourmi

Du point de vue d'une fourmi, les types de nœuds sont identiques à ceux du mCmE.

7.7.4 Phéromone

7.7.4.1 Type de phéromone

On distingue deux types de phéromone, la *phéromone de passage* et la *phéromone de caste*.

Phéromone de passage La phéromone de passage est la phéromone que laissent toutes les fourmis lorsqu'elles passent sur un arc. Elle influe sur les déplacements des fourmis qui préfèrent l'éviter en mode recherche et préfèrent la suivre en mode retour.

Phéromone de caste La phéromone de caste est la phéromone que laissent les fourmis d'une caste. Il y a ainsi une phéromone pour la caste ***cherche_hyper***, une pour la caste ***cherche_hypo*** ou une pour la caste ***cherche_magn***. Seule la caste ***standard*** n'en a pas.

Cette phéromone agit à deux niveaux différents :
- sur les déplacements des fourmis de la caste ;
- pour permettre de qualifier un pont. Ainsi si un pont a une phéromone ***Magn*** très élevée, on pourra considérer qu'il relie deux termes qui sont unis par cette relation.

7.7.4.2 Dépôt de phéromone

Le dépôt de la phéromone de passage est identique à celui du mCmE (cf 7.5.5.1). Une formule similaire est utilisée pour le dépôt d'une phéromone de caste :

$$\varphi_{\boldsymbol{c}_{t+1}}(A) = \varphi_{\boldsymbol{c}_t}(A) + \theta \tag{7.6}$$

où $\varphi_{\boldsymbol{c}_t(A)}$ est le taux de phéromone de la caste $\boldsymbol{c}$ sur l'arc A à l'instant t et θ la quantité de phéromone déposée par la fourmi.

7.7.4.3 Évaporation de la phéromone

Distance syntaxique Soit G l'arbre correspondant à l'arbre morpho-syntaxique donné par SYGFRAN dont on a supprimé la racine (notée *TEXTE* dans la figure 7.8) et établi des liens interphrases. On appelle *distance syntaxique* la fonction qui donne la distance minimale entre deux nœuds suivant la pondération des arcs. Soit :

$$D_s(N,N) = 0 \tag{7.7}$$

$$D_s(N_1,N_2) = \min \begin{bmatrix} D_s(Sup(N_1),N_2) + \vartheta(N_1,Sup(N_1)) \\ D_s(N_1,Sup(N_2)) + \vartheta(N_2,Sup(N_2)) \end{bmatrix} \tag{7.8}$$

où N, N_1 et N_2 sont des nœuds de G, $sup(N)$ le père de N dans G et $\vartheta(A_i,A_j)$ la pondération de l'arc reliant A_i et A_j. Chaque arc de structure (ou arc interphrase) A est pondéré pour le calcul de la distance syntaxique. Cette pondération est, par défaut, de 1. Dans ce cas, on remarquera que la distance syntaxique est une distance ultramétrique telle que celle présentée en 2.3.2.

Évaporation des phéromones et ponts À chaque cycle, il y a une légère évaporation des phéromones. Cette baisse se fait de façon linéaire en fonction de la distance syntaxique entre deux noeuds et jusqu'à la disparition totale des phéromones. Pour chaque phéromone :

$$\varphi_{t+1}(A) = \min(\varphi_t(A) - D_s(A_i, A_j) \times \delta, 0) \tag{7.9}$$

où $\delta \in \mathbb{R}^+$ est la quantité de phéromone d'un certain type qui s'évapore à chaque cycle et A_i, A_j les nœuds départ et arrivée de A.

En ce qui concerne les arcs de structure, interphrases ou les arcs de réseau on remarquera que l'évaporation de phéromone de passage se fait de manière strictement identique à celle du mCmE (cf. 7.5.5.1).

La distance syntaxique est une distance entre les mots qui tient compte de l'ordre des phrases dans le texte. Son introduction est rendue nécessaire pour le maintient ou non des ponts. Plus un pont relie des termes éloignés dans le texte plus sa phéromone de passage s'évaporera. Si nous n'utilisions pas une telle stratégie, on pourrait avoir trop facilement des ponts unissant des termes très éloignés au détriment de termes candidats plus proches et souvent plus pertinents. Prenons un exemple, considérons l'extrait suivant : « *Il creuse avec une pelle. La pelle s'est cassée.* ». Si nous savons que l'ACCEPTION *pelle*/**outil** est un *instrument_de creuser*/**trou**, les fourmis seront susceptibles de fabriquer deux ponts comme l'illustre la figure 7.9. Le second pont sera plus fragile puisque la distance syntaxique entre ses deux sommets est de 7 contre 6 pour le premier. En conséquence de quoi, il aura une probabilité plus forte de disparaître ce qui est vraisemblable dans notre exemple vu qu'aucune autre information ne serait susceptible de le renforcer.

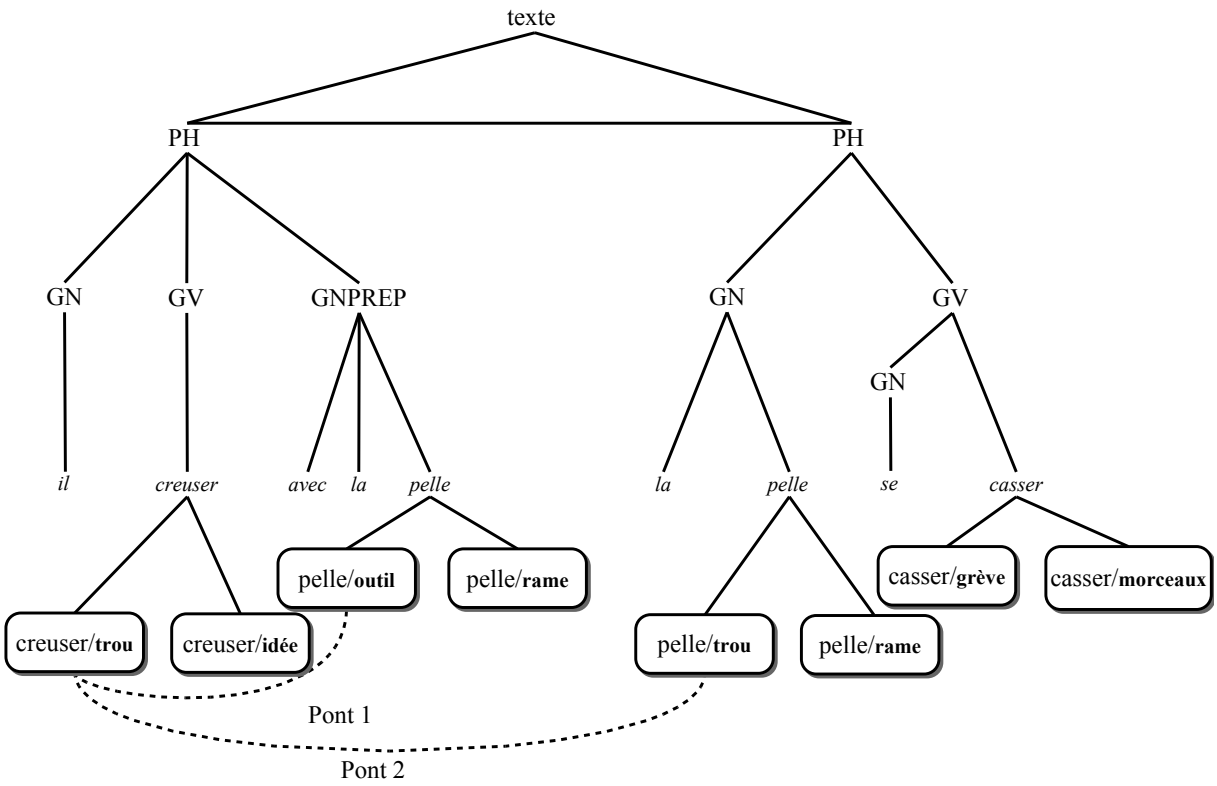


FIG. 7.9 – Exemple d'évaporation de la phéromone de passage.

7.7.5 Déplacements

7.7.5.1 Changement de mode

Le passage du mode recherche au mode retour se passe de la même manière que pour le modèle mCmE (cf. 7.5.5.2).

7.7.5.2 Évaluation des arcs et évaluation des nœuds

Dans l'objectif d'avoir un modèle d'évaluation qui soit le plus générique possible, nous ramenons les intervalles d'évaluation d'un chemin (N_i,A_j) à l'intervalle $[0,1]$. Ainsi, la fonction d'évaluation $E_f(N)$ de l'énergie du nœud N selon une fourmi f est :

$$E_f(N) = \min \begin{cases} \frac{E(N)}{E_{max}} \\ 1 \end{cases} \tag{7.10}$$

De même, la fonction d'évaluation $\varphi_f(A)$ de la phéromone de passage sur l'arc A selon une fourmi f est :

$$\varphi_f(A) = \min \begin{cases} \frac{\varphi(A)}{\varphi_{max}} \\ 1 \end{cases} \tag{7.11}$$

À l'identique, la fonction d'évaluation $\varphi_{\boldsymbol{c}_f}(A)$ de la phéromone de la caste $\boldsymbol{c}$ sur l'arc A selon une fourmi f est :

$$\varphi_{\boldsymbol{c}_f}(A) = \min \begin{cases} \frac{\varphi_{\boldsymbol{c}}(A)}{\varphi_{max}} \\ 1 \end{cases} \tag{7.12}$$

7.7.5.3 Modes de déplacement des fourmis

L'évaluation en mode recherche En mode recherche, les fourmis sont en quête de nourriture. Elles sont ainsi attirées par les nœuds qui ont beaucoup d'énergie mais elles évitent de passer par des arcs sur lesquels trop de fourmis seraient déjà passées afin d'explorer le plus grand nombre de solutions possibles. L'évaluation en *mode recherche* d'une destination est donc inversement proportionnelle à l'excitation de l'arc et proportionnelle à la quantité d'énergie du nœud. Les fourmis sont donc plus attirées par un lieu peu visité (excitation de l'arc faible) et une grande quantité d'énergie.

Les deux derniers critères sont ceux influencés par la caste de la fourmi. Le premier est celui de la phéromone de caste qui attire les fourmis de cette caste. Le second est le type de la relation correspondant à l'arc. Si l'arc à évaluer correspond à la caste de la fourmi, son évaluation est forcément supérieure à la moyenne. L'évaluation $Eval(N_i,A_j)$ du nœud N_i par l'arc A_j en mode recherche est donc donnée par les quatre critères suivants :

- l'évaluation de la quantité d'énergie $Eval_E(N_i,A_j)$ proportionnelle à la quantité d'énergie du nœud :

$$Eval_E(N_i,A_j) = E_f(N_i) \tag{7.13}$$

- l'évaluation de la quantité de phéromone de passage $Eval_\varphi(N_i,A_j)$ inversement proportionnelle à la quantité de phéromone sur l'arc :

$$Eval_\varphi(N_i,A_j) = 1 - \varphi_f(A_j) \tag{7.14}$$

– l'évaluation de la quantité de phéromone de la caste $\boldsymbol{c}$ $Eval_{\varphi_{\boldsymbol{C}}}(N_i,A_j)$ proportionnelle à la quantité de phéromone sur l'arc :

$$Eval_{\varphi_{\boldsymbol{C}}}(N_i,A_j) = \varphi_{\boldsymbol{C}f}(A_j) \tag{7.15}$$

– l'évaluation de la relation correspondant à l'arc A_j en fonction de la caste de la fourmi $Eval_{caste}(N_i,A_j)$:

$$Eval_{caste}(N_i,A_j) = \begin{cases} 3 \text{ si caste}(f) = \text{relation}(A_j) \\ 0 \text{ sinon} \end{cases} \tag{7.16}$$

La valeur 3 est utilisée pour contrebalancer les trois autres valeurs qui, au maximum, valent 1 chacune.

L'évaluation globale du nœud N_i en passant par l'arc A_j en mode recherche est donc :

$$Eval(N_i,A_j) = Eval_E(N_i,A_j) + Eval_{\varphi}(N_i,A_j) + Eval_{\varphi_{\boldsymbol{C}}}(N_i,A_j) + Eval_{caste}(N_i,A_j) \tag{7.17}$$

L'évaluation en mode retour En *mode retour*, les fourmis cherchent à retourner à leur fourmilière pour y déposer la nourriture trouvée. Elles vont ainsi plutôt prendre les arcs très excités et se diriger vers les nœuds dont le vecteur est le plus proche possible du leur. L'évaluation des destinations possibles est donc proportionnelle à l'excitation de l'arc et inversement proportionnelle à l'angle entre son vecteur conceptuel et l'odeur du nœud destination.

Les deux derniers critères sont ceux influencés par la caste de la fourmi. Ils sont identiques au mode recherche. L'évaluation $Eval(N_i,A_j)$ du nœud N_i par l'arc A_j en mode retour est donc donné par les trois critères suivants :

– L'évaluation de la quantité de phéromone $Eval_{\varphi}(N_i,A_j)$ est proportionnelle à l'excitation de l'arc :

$$Eval_{\varphi}(N_i,A_j) = \varphi_f(A_j) \tag{7.18}$$

– L'évaluation de l'odeur du nœud $Eval_{odeur}(N_i,A_j)$ est inversement proportionnelle à l'angle entre son vecteur conceptuel et l'odeur du nœud destination :

$$Eval_{odeur}(N_i,A_j) = 1 - \frac{2}{\pi} D_A(\mathrm{V(f)}, \mathrm{V(N)}) \tag{7.19}$$

– l'évaluation de la quantité de phéromone de la caste $\boldsymbol{c}$ $Eval_{\varphi_{\boldsymbol{C}}}(N_i,A_j)$ proportionnelle à la quantité de phéromone sur l'arc est identique à celle du mode recherche.

– L'évaluation de la relation correspondant à l'arc A_j en fonction de la caste $\boldsymbol{c}$ de la fourmi $Eval_{caste}(N_i,A_j)$ est identique à celle du mode recherche.

L'évaluation globale du nœud N_i en passant par l'arc A_j en mode retour est donc :

$$Eval(N_i,A_j) = Eval_{\varphi}(N_i,A_j) + Eval_{odeur}(N_i,A_j) + Eval_{\varphi_{\boldsymbol{C}}}(N_i,A_j) + Eval_{caste}(N_i,A_j) \tag{7.20}$$

7.7.5.4 Propagation de vecteurs

En ce qui concerne la propagation des vecteurs, on distingue les deux environnements. Sur l'arbre morpho-syntaxique, les fourmis modifient légérement le nœud où elles arrivent comme dans le modèle mCmE ou le modèle FOETAL. Ainsi, dans ce modèle encore, à tout moment d'une analyse sémantique, le vecteur conceptuel du texte est donné par le vecteur du sommet de l'arbre et plus généralement le vecteur d'une partie du texte par le sous-arbre qui lui correspond.

En revanche, lorsqu'une fourmi se trouve sur une fourmilière ou sur toute autre ACCEPTION recopiée dans le réseau lexical, le vecteur n'est pas modifié. Ces nœuds conservent ainsi un vecteur constant tout au long de la simulation.

7.7.5.5 Création, suppression et type de ponts

Dès qu'une fourmi se trouve sur un nœud correspondant à une ACCEPTION, c'est-à-dire une fourmilière ou un nœud recopié du réseau lexical, elle peut construire un pont.

Un pont peut être créé lorsqu'une fourmi atteint une fourmilière potentiellement amie. Dans ce cas, la fourmi évalue le nœud correspondant à sa fourmilière maison comme les nœuds géométriquement liés à cette fourmilière. Si ce nœud est sélectionné, il y a création d'un pont entre les deux fourmilières. Ce pont est ensuite considéré comme un arc standard par les fourmis, c'est-à-dire que les nœuds qu'il lie sont considérés comme voisins. Un pont peut être vu comme une compatibilité entre deux fourmilières, un chemin interprétatif possible.

Ce pont est recouvert à la fois par la phéromone de passage déposée par chacune des fourmis qui l'emprunte mais aussi par la phéromone spécifique à chaque classe. Si toute la phéromone du pont s'est évaporée, le pont est supprimé.

Les ponts ont une importance fondamentale encore plus critique dans ce modèle que dans les précédents. En effet, non seulement ils permettent de connaître les différents chemins interprétatifs possibles mais ils permettent aussi de qualifier parfois ces chemins. Ainsi, si un pont entre deux fourmilières est souvent emprunté par des fourmis d'une caste ***c***, on pourra en déduire un certain nombre d'informations plus ou moins directes en fonction de la caste. Par exemple, un pont fortement emprunté par des fourmis de la caste ***cherche_magn*** représentera vraisemblablement cette relation, il en sera de même avec les fourmis ***cherche_prédicat*** ou les fourmis ***cherche_patient***.

D'autres en revanche sont moins facilement interprétables comme la synonymie ou l'hyperonymie qui peuvent aider à la découverte des relations d'identité si les fourmilières joignent des termes de même morphologie.

7.7.5.6 Découverte de nœuds du réseau lexical

Lorsqu'une fourmi se trouve sur une fourmilière, trois possibilités de déplacement s'offrent à elle :

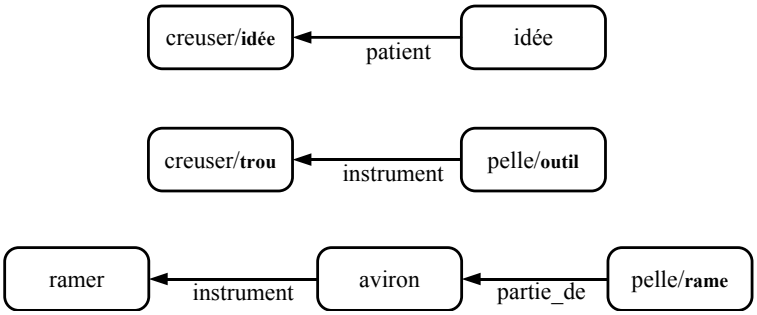
- *remonter sur l'arbre morpho-syntaxique* ;
- *emprunter un pont (créé par elle ou non)* ;
- *explorer le réseau lexical* ;

Dans ce troisième cas, si l'arc qu'elle a choisi n'a pas encore été emprunté par une fourmi au cours de l'analyse, on effectue une recopie locale de cet arc et du nœud destination. Cette recopie est indispensable pour garder la cohérence du système comme nous l'avons noté dans les critiques du modèle FOETAL (cf. 7.6.4.5).

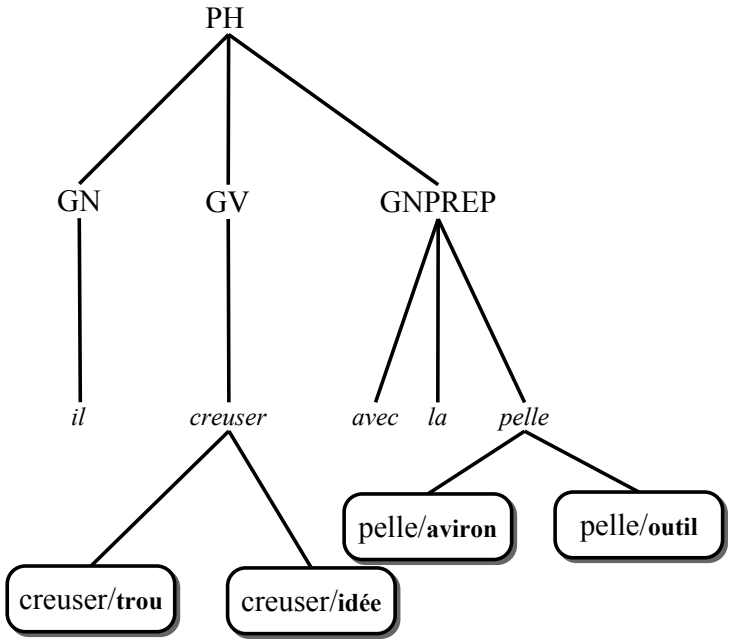
7.7.6 Exemple d'analyse sémantique dans le modèle MCEP

Prenons un exemple ultra simplifié pour permettre de mieux comprendre comment se déroule une analyse dans le modèle MCEP. Considérons la phrase « *Il creuse avec la pelle.* » et le mini réseau lexical présenté à la figure 7.10(a).

Ce qu'il est vraiment nécessaire de comprendre ici est la dynamique globale du système. À partir de l'ensemble d'heuristiques relativement simples présentées dans les sections précédentes, on assiste, par simple émergence, à la résolution des divers problèmes d'analyse posés par le texte (cf. 7.2.1). Dans notre exemple, les seules difficultés se situent au niveau de l'ambiguïté lexicale : ‹*pelle*› est-il ici dans son acception d'**outil** ou dans son acception de **rame** (pour l'aviron) et ‹*creuser*› est-il dans son sens métaphorique de « *creuser une idée* » ou dans son sens premier de « *creuser un trou* » ? Il est ainsi vraisemblable que de comprendre comment vont se former les ponts (F), (G) et (H) de la figure 7.11, donc comment le système choisit cette interprétation plutôt que les autres, peut aider à la compréhension de cette dynamique.



(a) Mini réseau lexical pour l'expérience



(b) Analyse morpho-syntaxique de la phrase « *Il creuse avec la pelle.* »

FIG. 7.10 – Exemple d'analyse sémantique avec le modèle MCEP : état avant l'analyse sémantique

Dans cet exemple simple, les fourmilières de *pelle*/**rame** et celles de *creuser*/**idée** ne peuvent raisonnablement pas s'allier pour faire émerger un chemin interprétatif. En effet, le réseau lexical donné ne les relie pas et les thèmes donnés par chacun sont relativement éloignés. Ceci est d'ailleurs vrai pour chacune des ACCEPTIONS de ce texte et cette phrase ne pourrait ainsi pas être correctement analysée avec le modèle mCmE (cf. critiques du modèle mCmE, section 7.5.9). Cet état de fait a une conséquence importante sur les déplacements des fourmis dans l'arbre morpho-syntaxique. Dans cet environnement, il ne peut ainsi qu'être chaotique au début de l'expérience et presque uniquement influencé par le réseau ensuite.

Considérons chaque fourmilière et le comportement des fourmis qui en sont issues.

7.7.6.1 Les fourmilières *creuser*/idée (2) et *pelle*/rame (4)

Les fourmis de *pelle*/**rame** (4), tout comme les fourmis de *creuser*/**idée** (2), explorent le réseau lexical ou l'arbre et s'y perdent puisqu'elles ne peuvent rien trouver de suffisamment tangible pour les ramener vers leur fourmilière mère. Ainsi, elles meurent souvent, construisent rarement

des ponts qui, s'ils le sont, ne sont que peu empruntés et disparaissent rapidement.

7.7.6.2 La fourmilière *pelle*/outil (3)

Les fourmis de *pelle*/**outil** (3), en particulier celles de la caste ***cherche_instrument*** empruntent l'arc éponyme (C) pour arriver sur l'ACCEPTION *creuser*/**trou** (7). Les fourmis qui sont montées dans l'arbre morpho-syntaxique redescendent vers les feuilles et en particulier elles atteignent la fourmilière *creuser*/**trou** (1). Statistiquement, un pont stable (H) ne peut être directement envisagé ici dès maintenant puisque l'arrivée des fourmis est trop peu probable car uniquement possible depuis l'arbre. Ces fourmis commencent donc à aller en grande majorité sur le réseau lexical par l'arc (B), sur des nœuds très probablement déjà copiés par les fourmis issues de la fourmilière *creuser*/**trou** (1). Arrivées sur *pelle*/**outil** (9), elles créent vraisemblablement un pont vers leur fourmilière mère (3) puisque le critère d'odeur sera alors maximal.

7.7.6.3 La fourmilière *creuser*/trou (1)

Les fourmis issues de *creuser*/**trou** (1) agissent de manière symétrique à celles de *pelle*/**outil** (3). Certaines fourmis choisissent d'emprunter l'arc (B) vers *pelle*/**outil** (9). Parallèlement, celles qui ont choisi d'aller dans l'arbre redescendent vers les feuilles et en particulier vers la fourmilière *pelle*/**outil** (3). Statistiquement, à ce moment là, le pont (H) peut être crée mais sa conservation est peu probable vu le flux relativement faible de fourmis de (1) arrivant alors en (3). La plupart de ces fourmis vont donc aller explorer le réseau lexical par l'arc (C) vers *creuser*/**trou** (7). Arrivées à ce nœud, elles ont une probabilité assez forte de créer un pont (G) vers leur fourmilière mère (1).

7.7.6.4 Collaboration entre les fourmis issues de *creuser*/trou (1) et celles issues de *pelle*/outil(3)

Observons maintenant le comportement collaboratif de *pelle*/**outil** (3) et de *creuser*/**trou** (1). Les fourmis de (1) ont créé le pont (G). Les fourmis de (3) peuvent ainsi l'emprunter et se retrouver sur la fourmilière (1). De là, elles peuvent fabriquer un pont (H) qui, cette fois, aura statistiquement plus de chance d'être conservé puisque compatible avec les informations disponibles réunies dans le circuit HCG. De même, ce pont sera renforcé par les fourmis de *creuser*/**trou** (1) qui, elles, utiliseront un circuit BFH.

7.7.7 Expérience

7.7.7.1 Castes

Dans les expérimentations les plus poussées, nous avons testé avec les castes suivantes :
- ***cherche_synonymie***, ***cherche_hyperonymie*** et ***cherche_hyponymie*** qui permettent de rechercher les références ;
- ***cherche_Magn***, ***cherche_Ver***, ***cherche_Bon*** et leurs inverses ;
- ***cherche_méronymie***, ***cherche_holonymie*** ;
- ***cherche_instrument*** ;
- ***cherche_agent***, ***cherche_patient cherche_destinataire*** et leur inverse ***cherche_prédicat*** ;

7.7.7.2 Résultats

Ce modèle donne automatiquement un type aux arcs contrairement aux deux précédents modèles dont le type était donné manuellement postérieurement aux annotations du système. Afin de pouvoir vérifier à la fois la désambiguïsation et cette annotation automatique, nous

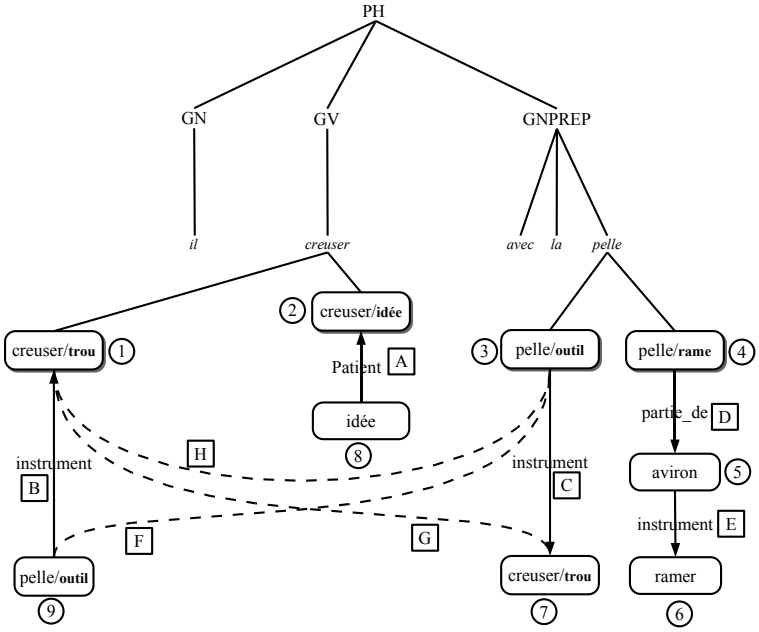


FIG. 7.11 – Exemple d'analyse sémantique avec le modèle MCME sur la phrase « *La pelle s'est cassée.* »

produisons trois tableaux pour les arcs. Le premier a été annoté manuellement comme pour les précédents modèles (les valeurs entre parenthèses indiquent le gain obtenu par rapport au modèle FORTAL). Le deuxième tableau considère les mêmes résultats annotés automatiquement. Le dernier exprime en fonction de la FLA le rappel et la précision obtenus.

La première remarque à faire sur ces résultats concerne les arcs de référence et d'instanciation des FLA. Le modèle, contrairement aux deux précédents, permet en effet leur détection. Celle-ci, bien que n'étant pas parfaite puisque mal interprétée dans un certain nombre de cas, semble, en revanche, avoir une action bénéfique indirecte sur les arcs d'interprétation et sur la désambiguïsation des termes. On remarque ainsi une amélioration nette pour chacune des catégories. On peut, en particulier, expliquer l'augmentation des taux de résultats corrects pour les adjectifs et les noms par l'instanciation importante des FLA adjectivales. On retrouve ce phénomène chez les verbes bien que le taux d'instanciation des FLA verbales soit moindre sauf pour la relation *agent*.

7.8 Principaux problèmes non encore réglés par l'analyse sémantique par fourmis

Nous avons donc montré la faisabilité d'une analyse sémantique par fourmis qui permet de résoudre les problèmes enoncés dans la section 7.2.1. Cette analyse semble ainsi beaucoup plus séduisante que l'analyse en remontée-redescente utilisée habituellement par l'équipe. Toutefois, de nombreux problèmes restent encore à régler.

acceptions	global	noms	adjectifs	verbes	adverbes
rappel	0,79 (+4%)	0,79 (+5%)	0,79 (+3%)	0,74 (+6%)	0,82 (+0%)
précision	0,78 (+3%)	0,78 (+2%)	0,82 (+4%)	0,76 (+4%)	0,8 (+2%)

arcs manuels	global	interprétation	rattachement prep	références	instanciation FLA
rappel	0,82 (+100%)	0,81 (+14%)	0,83 (+2%)	0,78 (+∞%)	0,83 (+∞%)
précision	0,85 (+98%)	0,82 (+12%)	0,89 (+6%)	0,81 (+∞%)	0,87 (+∞%)

arcs auto	global	interprétation	rattachement prep	références	instanciation FLA
rappel	0,73	0,81	0,67	0,68	0,71
précision	0,75	0,82	0,7	0,69	0,72

instanciation FLA	syn	hyper	Magn	Ver	Bon	mero	holo	instr	agent	patient	dest
rappel	0,85	0,77	0,7	0,72	0,73	0,62	0,7	0,72	0,72	0,6	0,61
précision	0,86	0,78	0,72	0,74	0,75	0,66	0,69	0,7	0,74	0,62	0,62

FIG. 7.12 – Evaluation du système multi-caste, Multi-environnement.

7.8.1 Problèmes techniques

7.8.1.1 Comment gérer l'antonymie ?

Comme nous l'avons vu à la section 3.3.5.2, l'exécution de la fonction naïve d'antonymie est encore longue à exécuter (en utilisant le parallélisme pour calculer chaque antonymie sur un processeur, de l'ordre de dix secondes sur un Sun à 8 processeurs 800 Mhz, de l'ordre de 25 sur un monoprocesseur Athlon AMD 1,4 Ghz). Ce qui est ainsi acceptable pour un algorithme en remontée-redescente, puisque l'appel à cette fonction ne s'y fera que de façon très limitée, est, en revanche, incompatible avec le concept d'agents réactifs où ce calcul serait nécessaire à de trop nombreuses reprises lors d'une analyse.

7.8.1.2 Arrêt du système

L'arrêt du système est un problème important dans les algorithmes à fourmis. La question de la convergence est un problème totalement ouvert. Ainsi, rien ne prouve d'un point de vue formel que le système va converger vers une solution et qu'il ne s'engage pas dans un cycle ou bien que les fourmilières auront toujours un comportement chaotique. Il semble pourtant très exceptionnel que cela ne soit pas le cas.

Quoi qu'il en soit, l'exécution ne peut, pour l'instant, s'arrêter toute seule. Dans nos expériences, nous avons toujours arrêté son déroulement au bout de quelques minutes. La perspective d'une approche endogène (fourmis gelantes) forçant la convergence et l'arrêt du système pourrait être une solution à ce problème et doit être étudiée.

7.8.2 Autres problèmes

Parmi les autres problèmes non encore réglés par l'analyse par fourmis, on peut citer celui des formes passives. Ainsi, ni une analyse en remontée-descente ni une analyse basée sur un des trois modèles présentés n'est capable de produire un résultat correct pour une phrase comme « *L'avocat a été mangé.* ». En effet, l'utilisation d'un réseau comme celui présenté dans la figure

6.4 considérerait inévitablement qu'il s'agit de l'**être humain** plutôt que du **fruit**. Il s'agit ici du problème des relations agent-patient qui sont, dans ce cas, mal repérées.

Une autre voie serait d'utiliser les fonctions lexicales d'évaluation et non pas uniquement l'arc, en d'autres termes passer l'utilisation des arcs du domaine du discret au domaine du continu.

7.9 Conclusions et perspectives

Dans ce chapitre, nous sommes revenus sur l'analyse sémantique en montrant quels sont les différents problèmes d'ambiguïtés qu'il serait nécessaire de résoudre dans le cadre des applications visées par l'équipe. Ces phénomènes linguistiques, à savoir l'ambiguïté lexicale, les problèmes de références, les rattachements prépositionnels, les chemins interprétatifs possibles et celui de l'instanciation des fonctions lexicales ne peuvent être résolus qu'en utilisant, conjointement aux vecteurs conceptuels, les informations issues du réseau lexical. Nous avons ainsi montré qu'une telle analyse sémantique ne se prêtait pas à une résolution par la méthode de remontée-descente et présenté une méthode qui peut satisfaire nos besoins, une méthode basée sur les algorithmes à fourmis.

Nous avons ainsi présenté trois modèles. Le premier pose les bases de l'exploitation du paradigme des algorithmes à fourmis pour une tâche d'analyse sémantique basée sur les vecteurs conceptuels simples et en montre la faisabilité. Il exploite un algorithme à fourmis mono-caste et mono-environnement (l'arbre morpho-syntaxique). Le deuxième est un modèle d'analyse sémantique par algorithmes à fourmis multi-caste à environnements séparés (arbre morpho-syntaxique et réseau lexical). Il est le premier à bénéficier des avancées sur la représentation du sens à partir de vecteurs conceptuels présentées dans cette thèse. Enfin, le dernier, mis au point au cours de cette thèse est, lui, un modèle multi-caste et à environnements partagés. Une première étude que nous avons présentée ici, montre que ce modèle est le plus efficace des trois. En particulier, il peut permettre l'instanciation des fonctions lexicales et la résolution des problèmes de références.

Nous pensons fortement que notre modèle est, au moins dans son principe si ce n'est dans sa mise en oeuvre, porteur de nombreuses pistes de recherche intéressantes. En particulier, la généricité de l'approche permet d'entrevoir aisément la définition de nouvelles castes de fourmis correspondant à de nouvelles heuristiques ou l'exploitation de nouveaux types d'informations. Il est clair que nous n'avons donné ici que des pistes de recherche et les critères à la fois d'heuristiques et d'évaluations devraient faire l'objet d'une longue étude par exemple dans le cadre d'une thèse.

8

La double boucle externe : mise en collaboration de plusieurs bases

DANS ce chapitre, nous cherchons à mettre l'accent, non plus sur les doubles boucles uniquement internes à la base, mais sur celles qui sont en partie externes. Nous étudions, à cette fin, la mise en collaboration de plusieurs bases de vecteurs. Dans un premier temps, nous présentons la création d'une base lexicale sémantique monolingue à partir d'une base déjà existante pour une autre langue. Dans l'expérience menée, il s'agit de construire une base pour l'anglais à partir de la base du français dont la constitution a été présentée dans les chapitres précédents et à l'aide de dictionnaires bilingues. Dans une seconde partie, nous montrons comment ces deux bases peuvent collaborer en s'échangeant des informations mutuelles et ainsi améliorer leurs représentations respectives. Nous élargissons enfin cette collaboration à des bases d'architectures éventuellement différentes en présentant comme exemple celle réalisée entre les deux bases de vecteurs conceptuels de l'équipe et nous montrons qu'elle met en jeu une double boucle.

Sommaire

DANS le chapitre précédent, nous avons montré comment nous pouvions avantageusement tirer parti des informations disponibles dans la base lexicale sémantique, implémentée par Blexisma, pour la résolution des problèmes posés lors de l'analyse sémantique d'un énoncé écrit. L'apprentissage des informations contenues dans le système est, en partie, basé sur cette analyse. Son amélioration qualitative peut ainsi permettre celle des données stockées dans la base. De la même manière, l'apprentissage, grâce aux fonctions lexicales, permet d'améliorer synchroniquement les agents spécifiques et le système global comme nous l'avons montré avec la synonymie et l'antonymie (cf. chapitres 3 et 4). Ce phénomène est connu sous le nom de double boucle et nous en avons fait un élément fondateur de notre approche (cf. 5.1.6).

Dans ce chapitre, nous allons mettre l'accent, non plus sur les doubles boucles internes à la base[105], mais sur la double boucle dont la première boucle est interne au système et la seconde externe. Nous allons, à cette fin, étudier la mise en collaboration de plusieurs bases de vecteurs. Dans un premier temps, nous présentons la création d'une base lexicale sémantique monolingue à partir d'une base déjà existante pour une autre langue. Dans l'expérience menée, il s'agit de construire une base pour l'anglais à partir de la base du français dont la constitution a été présentée dans les chapitres précédents et à l'aide de dictionnaires bilingues. Dans une seconde partie, nous montrons comment ces deux bases peuvent collaborer en échangeant des informations mutuelles et ainsi améliorer leurs représentations respectives. Nous élargissons enfin cette collaboration à des bases d'architectures éventuellement différentes en présentant, comme exemple, celle réalisée entre les deux bases de vecteurs conceptuels de l'équipe et nous montrons qu'elle met en jeu une double boucle.

8.1 Création d'une base monolingue à partir d'une base déjà existante dans une autre langue

Dans cette section, nous étudions les méthodes possibles pour permettre de créer une base lexicale sémantique monolingue à partir d'une base déjà existante dans une autre langue. Ces travaux ont été, en grande partie, réalisés au cours du DEA de Frédéric Rodrigo co-encadré par Mathieu Lafourcade et moi-même au début 2004 [Rodrigo, 2004]. Ils n'ont pas pu, faute de moyen humain, être poursuivis à l'heure actuelle mais ils ont fait l'objet d'une expérience et d'une publication [Lafourcade *et al.*, 2004] dont nous allons maintenant parler. L'expérience menée ici, et qui sera le fil conducteur de notre propos, est la création d'une base lexicale sémantique de l'anglais basée sur les vecteurs conceptuels à partir de la base du français utilisée dans les chapitres précédents. Cette base repose sur les hypothèses présentées au chapitre 5, son architecture est ainsi composée de trois niveaux d'objets lexicaux qui regroupent à la fois in-

[105] Il faut comprendre, ici, le terme "base" comme le système global implémenté par Blexisma et non pas l'agent particulier de gestion des données `base`.

acception	français	allemand	danois	italien	anglais
végétal	‹*arbre*›	‹*baum*›	‹*trae*›	‹*albero*›	‹*tree*›
matière	‹*bois*›	‹*holz*›		‹*legno*›	‹*timber*›
petit groupement d'arbres		‹*wald*›	‹*skov*›	‹*bosco*›	‹*wood*›
grand groupement d'arbres	‹*forêt*›			‹*foresta*›	‹*forest*›

FIG. 8.1 – Equivalence des termes entre les langues (issu de ([Éco, 1988], p. 113))

formations thématiques (vecteurs conceptuels) et informations lexicales (morphologie, fonctions lexicales, ...) : les LEXIES qui correspondent à un sens suivant une source, les ACCEPTIONS qui regroupent ces dernières en fonction de leur sens et les ITEMS LEXICAUX qui regroupent toutes les informations des ACCEPTIONS. Le principe de base de cette construction est d'utiliser des dictionnaires bilingues pour la fabrication des LEXIES anglaises.

8.1.1 Problèmes posés par la traduction

On considère généralement que deux principaux problèmes fortement liés sont posés lorsqu'il s'agit d'effectuer une traduction d'une langue source vers une langue cible : le *transfert grammatical* et le *transfert lexical* [Hutchins & Somers, 1992].

8.1.1.1 Transfert grammatical

Deux langues ne répondent pas aux mêmes règles de grammaire. Un exemple connu concerne la place de l'adjectif épithète en anglais et sa place en français. Dans la première langue, il sera généralement placé avant le nom tandis que dans la deuxième, il sera placé après. Le transfert grammatical consiste ainsi à transposer la structure grammaticale de la langue source vers une structure équivalente dans la langue cible.

8.1.1.2 Transfert lexical

Le transfert lexical consiste à chercher le meilleur équivalent de la langue cible pour un terme de la langue source en fonction des informations contextuelles disponibles. Les principaux problèmes posés par ce transfert sont dus à la lexicalisation des sens qui se fait de façon rarement identique entre les langues. La figure 8.1 présente les couvertures sémantiques des items lexicaux français ‹*arbre*›, ‹*bois*› et ‹*forêt*› et de leurs équivalents dans d'autres langues. On peut remarquer que l'item français ‹*bois*› recouvre le même champ sémantique que l'allemand ‹*baum*› et que l'anglais ‹*tree*›. Le français ‹*bois*› couvre deux acceptions : la **matière** qui correspond à l'italien ‹*legno*› et à l'allemand ‹*holz*› et le **petit groupement d'arbres** qui correspond, lui, à l'allemand ‹*wald*› et à l'italien ‹*bosco*›.

Le transfert lexical est ainsi confronté à deux problèmes d'ambiguïté : la polysémie du langage source ainsi que les phénomènes contrastifs du langage cible (ou raffinement des sens cf. 1.3.3.2).

En tant que principes, transferts lexical et grammatical sont présentés séparément mais en pratique, lors de la traduction d'un texte, ces deux opérations sont fortement liées. Le problème qui nous occupe ici, construire une base lexicale sémantique à partir d'une telle base déjà existante dans une autre langue, se rapproche, en revanche, du transfert lexical.

8.1.2 Construction de lexies à l'aide de dictionnaires bilingues

8.1.2.1 Dictionnaire bilingue

L'outil principal du transfert lexical est généralement un dictionnaire bilingue. Un dictionnaire bilingue classique est un dictionnaire bidirectionnel composé de deux volumes : un premier qui donne les traductions possibles d'un terme d'une langue vers une autre et un deuxième qui est son symétrique [Mangeot-Lerebours, 2001]. La structure classique d'une définition issue d'un tel dictionnaire est :

$$item \equiv < morph^*, glose^*, equiv^+ >$$

où *morph* correspond à la morphologie, *glose* à un contexte d'utilisation facultatif et *equiv* est la liste des traductions possibles. La figure 8.2 présente un exemple avec le terme français ‹*souris*› dans ses deux acceptions issues pour l'une du domaine de l'informatique et pour l'autre du domaine de la zoologie comme l'indiquent les deux gloses (ici, entre crochets).

```
souris:
        (Nom) [ZOOLOGIE] mouse
        (Nom) [INFORMATIQUE] mouse
```

FIG. 8.2 – Entrée ‹*souris*› dans le dictionnaire multilingue logos, section anglais-français

8.1.2.2 Principe de la construction des lexies

Les dictionnaires bilingues établissent donc des correspondances entre les termes de deux langues. Notre objectif est de fabriquer des LEXIES anglaises à partir des ACCEPTIONS françaises. Dans le cas général, il y a plusieurs traductions possibles pour un terme. L'opération consiste donc, dans un premier temps, à chercher quelle(s) ACCEPTION(S) française(s) peut (peuvent) correspondre le mieux à telle ou telle traduction en fonction des informations disponibles (filtrage) puis à construire les LEXIES en récupérant quelques-unes des informations de l'acception choisie (vecteur et dans certains cas morphologie cf 8.3).

8.1.3 Réalisation

Le processus se déroule en trois étapes. La première consiste à obtenir les traductions possibles $\acute{equiv}_i$ d'un item lexical I_S de la langue source et la construction d'une représentation lexicale et vectorielle pour chacun d'eux. La deuxième étape consiste à choisir pour chacune des traductions $\acute{equiv}_i$ laquelle des ACCEPTIONSde I_S est la plus proche à l'aide d'un filtre multicritères. Enfin nous construisons une LEXIE à partir de l'ACCEPTION choisie pour chacune des $\acute{equiv}_i$ au cours de la dernière étape.

8.1.3.1 Étape 1 : obtention des traductions possibles et construction de leur représentation

À l'aide d'un dictionnaire bilingue, on récupère toutes les traductions possibles $\acute{equiv}_i$ d'un item lexical I_S de la langue source. Il s'agit comme nous l'avons déjà dit de triplets de la forme $item \equiv < morph^*, glose^*, equiv^+ >$. Dans cette étape nous construisons un objet lexical par équivalent, le vecteur conceptuel correspond à celui qui résulte de l'analyse sémantique de la glose (cf. 2.3.7). L'ensemble des champs d'un objet lexical est rempli en fonction des données disponibles en vue d'une utilisation dans l'étape 2. Nous obtenons ainsi un ensemble d'objets lexicaux $equiv_i$[106].

[106] Dans la suite de notre exposé, nous parlerons indifféremment de *equiv* et de l'objet lexical qui porte le même nom mais qui est la représentation informatique des informations calculées pour ce terme.

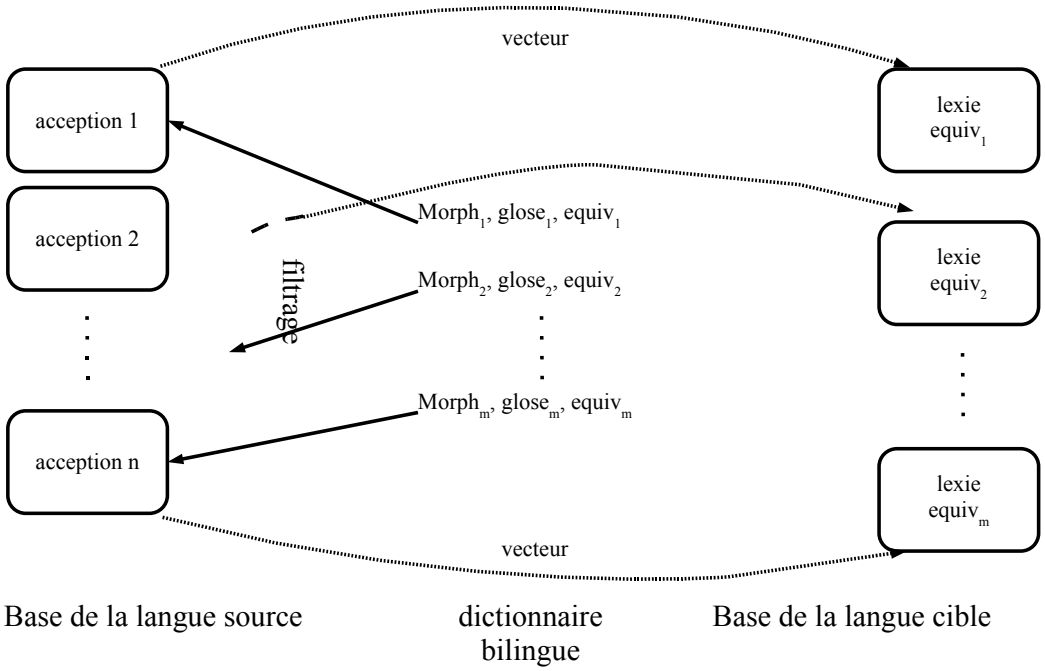


FIG. 8.3 – Le processus de mise en correspondance entre termes et acceptions cible et source

8.1.3.2 Étape 2 : choix par filtres

L'opération de filtrage consiste à trouver la traduction *equiv* de l'item I_S, issue de la base source, la plus proche d'une des ACCEPTIONS A_j de I_S. Elle est basée sur un système de scores donnés pour chaque critère aux traductions possibles. Nous appelons score une valeur entre zéro et un qui est d'autant plus importante que le critère est pertinent. La traduction *equiv* qui a le meilleur score global est choisie.

Score global Le $score_i$ global est calculé à l'aide d'une moyenne des scores de chaque critère.

$$\sigma \times \kappa \rightarrow [0,1] : score_i(equiv_i, A_j) = \frac{1}{n} \sum_{0}^{n} \alpha_k \times score_{crit_k} \tag{8.1}$$

où σ représente l'ensemble des objets lexicaux, κ l'ensemble des ACCEPTIONS et n le nombre de critères applicables pour $equiv_i$. On considère ainsi qu'une LEXIE de $equiv_i$ dans la base cible peut être construite à partir de l'ACCEPTION A_j qui a le meilleur score (cf. étape 3).

Comme nous l'avons déjà remarqué précédemment avec les dictionnaires classiques (cf. 5.1.4), les dictionnaires bilingues peuvent avoir des manques, en particulier l'absence de gloses. Ainsi, certains critères ne sont pas nécessairement applicables. Dans ce cas, le critère n'est pas utilisé dans le calcul du score global.

Calcul des scores de chaque critère

Filtrage morphologique Dans une première approche, nous avons filtré sur la morphologie, et conservé uniquement les traductions pour lesquelles elle est identique à $m(A_j)$. Mais cette stratégie a pour conséquence de limiter le choix des termes à des morphologies identiques. Nous avons donc opté pour une méthode plus flexible, celle du poids morphologique présenté en 2.3.6.4 dont les bornes sont ramenées linéairement à [0,1].

$$\sigma \times \kappa \rightarrow [0,1] : score_{morph}(equiv_i, A_j) = P_{morpho}(m(equiv_i), m(A_j)) \times \frac{2}{\pi} \quad (8.2)$$

Filtrage sur le vecteur glose Il s'agit de calculer la distance entre le vecteur conceptuel de l'ACCEPTION A_j, $V(A_j)$ et le vecteur de la glose $glose_i$, $V(glose_i)$. La mesure utilisée est une relinéarisation de la distance angulaire. Plus les vecteurs sont proches, plus le score avoisine 1 :

$$\sigma \times \kappa \rightarrow [0,1] : score_{glose}(equiv_i, A_j) = D_A(\mathrm{V}(equiv_i),\ \mathrm{V}(A_j)) \times \frac{2}{\pi} \quad (8.3)$$

Le score transforme le domaine image de la distance angulaire de façon linéaire. Pour une transformation vers $[0,1]$, nous aurions pu utiliser le cosinus mais on se garde bien de le faire car on souhaite focaliser le pouvoir de discrimination pour les faibles valeurs de l'angle (cf. 2.1.3.1).

Ces deux critères sont applicables dans le cas général d'un dictionnaire bilingue classique. D'autres critères peuvent être utilisés avec certains dictionnaires. Dans notre expérience en particulier, le dictionnaire utilisé, logos[107], fournit en plus de la glose un thème, ainsi que des informations sur les relations sémantiques de synonymie et d'antonymie entre les termes.

Filtrage sur le sujet Ce score est calculé de la même manière que celui de la glose :

$$\sigma \times \kappa \rightarrow [0,1] : score_{sujet}(equiv_i, A_j) = D_A(V_{sujet}(equiv_i),\ V(A_j)) \times \frac{2}{\pi} \quad (8.4)$$

Filtrage à l'aide des relations sémantiques : synonymie Dans le cas où le dictionnaire considéré présente une liste de synonymes, le score est calculé grâce à la fonction lexicale de construction partielle de synonymie généralisée (cf. 4.1.1.1). Nous calculons ensuite la distance angulaire ramenée à l'intervalle $[0,1]$ entre le vecteur du terme source $vecteur(terme_S)$ et le vecteur résultat de cette fonction.

$$\sigma \times \kappa \rightarrow [0,1] : score_{syn}(equiv_i, A_j) = D_A(Csyn_P(Syn(equiv_i)),\ V(A_j)) \times \frac{2}{\pi} \quad (8.5)$$

8.1.3.3 Étape 3 : construction des lexies

Une fois choisie la meilleure ‹*acception*› issue de la base source, on crée une LEXIE pour la base cible. Les différentes informations linguistiques de cet objet lexical sont remplies de la manière suivante :

- un **identifiant** : par exemple le nom de l'item concaténé à un marqueur permettant de différentier ses diverses ACCEPTIONS ;
- la **morphologie** : celle présentée par le dictionnaire bilingue ;
- la **fréquence en usage** : aucune heuristique ne semble pouvoir estimer cette fréquence avec cette méthode ;
- un **vecteur conceptuel** : celui de l'ACCEPTION issue de la base source ;
- les **fonctions lexicales** associées : : celles qui sont éventuellement disponibles dans le dictionnaire bilingue ;
- des **informations étymologiques** : celles qui sont éventuellement disponibles dans le dictionnaire bilingue ;
- des **gloses** :celles qui sont éventuellement disponibles dans le dictionnaire bilingue.

[107] http://www.logosdictionary.com

8.1.4 Expérience, résultats et évaluation

8.1.4.1 Expérience

L'expérience menée a consisté à utiliser la base du français, dont l'architecture et la construction ont été présentées dans les chapitres précédents, pour construire une base de l'anglais d'architecture identique. Dans un premier temps, on cherche, pour chaque ACCEPTION française, son meilleur équivalent anglais et on construit une lexie pour cet équivalent suivant la méthode proposée dans la section précédente. Les différentes lexies anglaises sont alors catégorisées par un agent spécialiste identique à celui pour le français présenté en 5.4.6.

Au moment de l'expérience, la base française contient environ 96 000 ITEMS LEXICAUX, 130 000 ACCEPTIONS et 160 000 LEXIES créées à partir de sources monolingues. Dans cette première expérience seulement 14 500 ITEMS ont été utilisés dans le processus de traduction (la liste abu[108]). Nous n'avons pas pu mener plus avant l'expérience par manque de temps. Ainsi, la création de ces objets a duré une centaine d'heures dues, en grande partie, aux accès Web pour le dictionnaire bilingue. La présence de ce dictionnaire sur le réseau local accélérerait grandement le rendement et nous avons estimé à environ 12H la création de la base dans ce cas et donc à environ 80H la génération complète de la base anglaise à partir de la base française[109].

À la fin du processus de traduction, la base anglaise contient 21 000 ITEMS, 41 000 ACCEPTIONS et 63500 LEXIES.

8.1.4.2 Résultats

La figure 8.4 présente quelques voisins thématiques dans les bases françaises et anglaises au moment de l'expérience en Mai-Juin 2004.

La figure 8.5 présente les voisins thématiques du vecteur conceptuel construit par la contextualisation faible de l'item anglais ‹*stack*› en fonction de termes contextes.

8.1.4.3 Évaluation

Description du protocole d'évaluation L'évaluation de la base anglaise s'est faite par sondage. Il s'agit pour les sondés de donner une note à une liste de termes en fonction du rapport de sens que ces termes entretiennent avec un item donné. Pour chaque item test, nous donnons deux listes :

- *le contenu de l'entrée du thésaurus Roget* c'est-à-dire les termes les plus proches de l'item considéré selon la version électronique du thésaurus Roget[110].
- *les plus proches voisins dans la base anglaise*

Le sondage a été réalisé sur 19 personnes issues de notre laboratoire (hors équipe) sur une vingtaine de termes choisis aléatoirement dans la base anglaise. L'opération s'est déroulée en deux phases : une première lors du DEA de Frédéric Rodrigo (interrogation de 7 personnes) et une seconde en Juin 2005 (interrogation de 12 nouvelles personnes). Nous avons souhaité l'augmentation du panel de test car le nombre de 7 nous paraissait relativement faible mais il faut reconnaître que mener à bien l'expérience puis une telle évaluation dans les trois mois que durent le stage de DEA est une tâche plus que délicate à entreprendre. Nous verrons dans les résultats que cette augmentation de la population sondée n'a finalement rien changé à l'évaluation obtenue.

Résultats de l'évaluation La figure 8.6 présente quelques exemples représentatifs de listes soumises à la population sondée et les notes moyennes obtenues. Notons que sur les 20 couples

[108] http://abu.cnam.fr/DICO/
[109] Temps obtenus sur un Sun à 8 processeurs 800 Mhz
[110] http://poets.notredame.ac.jp/Roget

beer
pale ale
lambic
cognac
brandy
stout
public house
kir
lambic
alcohol

bière
kriek
cervoise
bière brune
pale-ale
citronnade
picrate
lambic
vinasse
saké

(a) Termes voisins de ‹*beer*›$_{en}$ et ‹*bière*›$_{fr}$

house
building
domicile
institution
lodging
urban planning
urbanism
planology
manhole
palace

maison
chacunière
hutte
carrée
building
grand ensemble
résidence
intérieur
logement
toit

(b) Termes voisins de ‹*house*›$_{en}$ et ‹*maison*›$_{fr}$

car
motorcar
automobile
passenger car
landau
automotive
Ridge runner
limo
jeep
motoring

voiture
auto
automobile
tacot
landau
limousine
bagnole
parking
berline
automobiliste

(c) Termes voisins de ‹*car*›$_{en}$ et ‹*voiture*›$_{fr}$

commerce
transaction
desktop
agiotage
business
holding company
trade
affiliate
acquisition
firm

commerce
offre et demande
prix marchand
succursale
trafic
agence
holding
étal
caisse de dépôts
caisse de crédit

(d) Termes voisins de ‹*commerce*›$_{en}$ et ‹*commerce*›$_{fr}$

FIG. 8.4 – Termes voisins de quelques items de la base anglaise et de quelques items de la base française.

de listes proposés seul celui de ‹*beer*› a obtenu une bien meilleure note pour le voisinage que pour le thésaurus. En revanche, certains cas (4/20) se révèlent défavorables à la base vectorielle, la grande majorité (15/20) ayant des notes relativement proches ($+/-1$).

En moyenne, sur les 20 cas tests, nous avons obtenus une note $\frac{note_{voisins}}{note_{Roget}} = 0.91$ c'est-à-dire que la population test a trouvé que la qualité des données vectorielles était inférieure de seulement 10% à celle des données du thésaurus Roget sur les 20 cas présentés. En ce qui concerne l'évaluation, notons que la note obtenue avec 7 personnes dans l'expérience menée par Frédéric Rodrigo était 0.87 c'est-à-dire une valeur relativement proche.

8.1.5 Comparaison avec des méthodes existantes

La méthode présentée ici permet ainsi la construction rapide d'une base lexicale sémantique conforme à l'architecture introduite au chapitre 5 à partir d'une base déjà existante pour une autre langue. D'autres méthodes de construction permettant d'obtenir deux bases de données vectorielles reposant sur le même espace vectoriel ont été réalisées dans l'équipe. Parmi celles testées avec les vecteurs sémantiques, on peut citer en particulier celle réalisée par Jean-Michel Delorme pour son mémoire d'ingénierie CNAM[111] [Delorme, 2003]. L'idée, ici, n'est pas de fabriquer des objets lexicaux pour la langue cible à partir d'objets lexicaux de la langue source mais plutôt d'agir de façon pratiquement inverse. L'opération se déroule en trois phases :

[111] Conservatoire National des Arts et Métiers

stack	money	wood	car	people	food
	stack	stack	stack	stack	stack
	purse	park	park	couple	supply
	denier	odd piece	store	dynastic	provision
	clutch bag	sheaf	lens hood	keep	park
	park	board	railroad station	running	store
	stock exchange	winder	railway station	corner	load
	store	harbor	station	hydrography	omelet
	cash received	course	freightage	ciborium	omelette

FIG. 8.5 – Les termes les plus proches de ‹*stack*› dans le contexte de ‹*money*›, ‹*wood*›, ‹*car*›, ‹*people*› et ‹*food*›

- Une base de vecteurs sémantiques pour la langue cible[112] est construite à partir d'un thésaurus suivant la méthode présentée en 2.2.
- On fabrique une matrice permettant de transformer les vecteurs de l'espace cible en vecteurs de l'espace source. Cette matrice est construite à partir de la partie 2 du thésaurus dont est issue la base cible (cf. 1.4.5.2). L'idée ici est de favoriser, pour chaque concept du thésaurus cible, l'émergence des concepts forts dans l'espace source. Cette phase est réalisée en faisant la somme vectorielle normée des vecteurs des traductions des termes proches du concept selon le thésaurus. La matrice est la concaténation de chacune des lignes correspondant aux concepts.
- On transforme grâce à cette matrice les vecteurs issus de la langue cible en vecteurs de l'espace de la langue source.

L'expérience présentée par [Delorme, 2003] utilise la version électronique du thésaurus Roget[113] pour fabriquer une base de l'anglais qu'il convertit ensuite en une base du français fondée sur le thésaurus Larousse [Larousse, 1992]. Le dictionnaire bilingue utilisé est une version électronique de [Correard *et al.*, 2001].

8.1.6 Perspectives : Traduction Automatique

Comme nous l'avons vu à la section 8.1.1, la traduction en général et la traduction automatique (TA) en particulier peut se décomposer en deux sous-parties : le transfert lexical et le transfert grammatical. Nous présentons ici une voie de recherche possible avec les outils dont nous disposons dans l'équipe.

8.1.6.1 Transfert grammatical

L'une des méthodes possibles, et par ailleurs, utilisée dans l'équipe TALN pour la génération de textes est celle adoptée et développée par Violaine Prince. À partir de l'arbre d'analyse morphosyntaxique du français fournit par SYGFRAN, l'idée est de générer un arbre de la structure grammaticale de la phrase anglaise. L'arbre initial est ainsi modifié par un ensemble de règles par le transducteur grammatical SYGFtoE actuellement développé avec SYGMART par Violaine

[112] Nous continuons à utiliser les mêmes termes cible et source même si dans [Delorme, 2003] ce ne sont pas les termes qui sont transformés mais plutôt les espaces vectoriels. Il serait plus juste de parler d'espace vectoriel source et d'espace vectoriel cible mais ceux-ci seraient exactement contraires à ceux de notre méthode et pourrait, sans doute, amener énormément de confusion au lecteur. Ainsi, la base de donnée source de notre méthode correspond à l'espace vectoriel cible de [Delorme, 2003] et la base de données cible de notre méthode à l'espace vectoriel source de Jean-Michel Delorme.

[113] http://poets.notredame.ac.jp/Roget

thésaurus Roget *beer*	voisinage thématique *beer*
ale	alcohol
amber brew	alcoholic drink
barley pop	pale ale
barley sandwich	ambrosia
belly wash	aqua vitae
bock	brandy
brew	cider
brown bottle	cognac
cold coffee	gin
head	julep
hops	kir
lager	lambic
malt	lemonade
malt liquor	mark
oil	piccolo
porter	public house
slops	rough brandy
stout	stout
suds	strong liquor
note = 10	note = 13

(a) Les données d'évaluation pour ‹*beer*›.

thésaurus Rodget *dictionary*	voisinage thématique *dictionary*
concordance	alphabet
cyclopedia	article
encyclopedia	designator
glossary	glossary
language	grammar
lexicon	idiom
palaver	lexicography
promptory	lexicon
reference	linguistics
terminology	nomenclature
vocabulary	orthographical
wordbook	vocabulary
note = 15	note = 15.5

(b) Les données d'évaluation pour ‹*dictionary*›

thésaurus Rodget *sky*	voisinage thématique *sky*
air	attic
azure	banquette
celestial sphere	batten
empyrean	bed curtain
envelope	bezel
firmament	cupola
heavens	eden
lid	esplanade
pressure	lambrequin
substratosphere	mirador
the blue	oasis
troposphere	paradise
upper atmosphere	penthouse
welkin	promenade
note = 15	note = 6

(c) Les données d'évaluation pour ‹*sky*›

thésaurus Rodget *wood*	voisinage thématique *wood*
copse	antlers
forest	baguette
grove	bush
lumber	fin
thicket	flower bed
timber	forest
timberland	grove
trees	lumber
weald	shaft
woodland	woodwind instruments
woods	woodwinds
note = 15	note = 13

(d) Les données d'évaluation pour ‹*wood*›

FIG. 8.6 – Exemples de listes présentées aux sondés pour l'évaluation et note moyenne obtenue

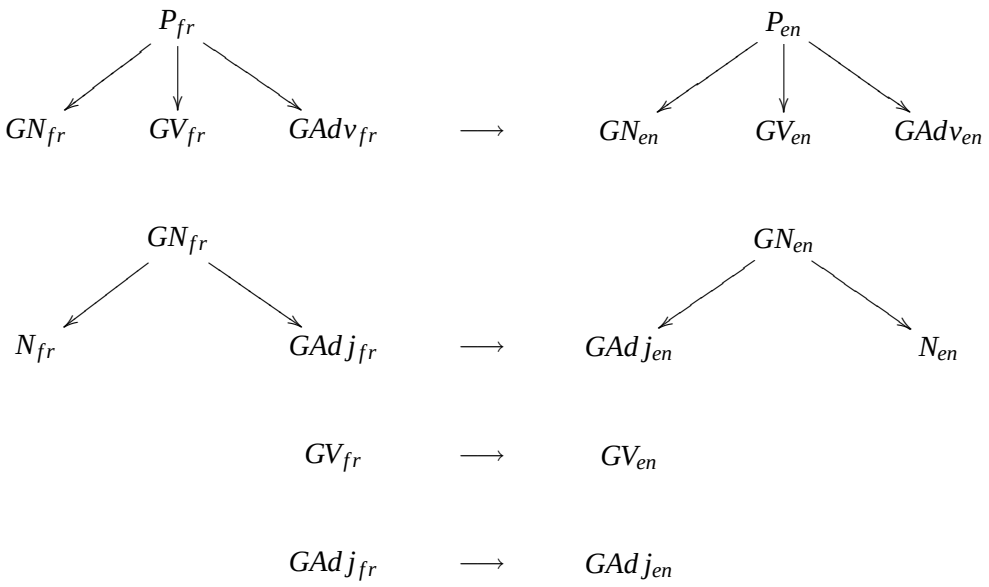


FIG. 8.7 – Exemple de transformations d'arbres grammaticaux

Prince [Delorme, 2003]. La figure 8.7 présente quelques transductions d'arbres pour quelques cas simples de transfert grammatical entre l'anglais et le français.

Une fois l'arbre syntaxique de l'anglais créé, il faut remplacer les feuilles par le résultat de leur transfert lexical (cf 8.1.6.2). Pour finir, il faut appliquer les différentes règles de grammaire et de déclinaisons propres à la langue cible.

8.1.6.2 Transfert lexical

Le transfert lexical consiste à sélectionner un terme du langage cible pour être idéalement substitué à un terme du langage source. Il s'agit donc ici de résoudre un maximum des problèmes posés par l'analyse sémantique (cf. 7.2.1) et en particulier la désambiguïsation lexicale puis de sélectionner le meilleur équivalent de chaque terme dans la langue cible.

La méthode de transfert proprement dite utilise une base pour chaque langue, toutes deux se situant dans le même espace vectoriel. L'équivalent choisi est celui dont la distance thématique avec le terme désambiguïsé est la plus faible.

Cette méthode est actuellement utilisée dans le prototype mis au point par Violaine Prince et qui repose sur la base de vecteurs de l'anglais réalisée par Jean-Michel Delorme (cf 8.1.5) mais rien n'empêcherait d'utiliser la base dont nous avons présenté la création dans la section précédente y compris parallèlement pour croiser les informations.

8.1.6.3 Exemple de traduction

Nous allons étudier la manière dont peut se faire la traduction de la phrase « *L'étudiant fait son devoir.* »

Génération de l'arbre morpho-syntaxique de la langue source SYGFRAN permet d'obtenir l'arbre morpho-syntaxique de la phrase (cf. figure 8.8).

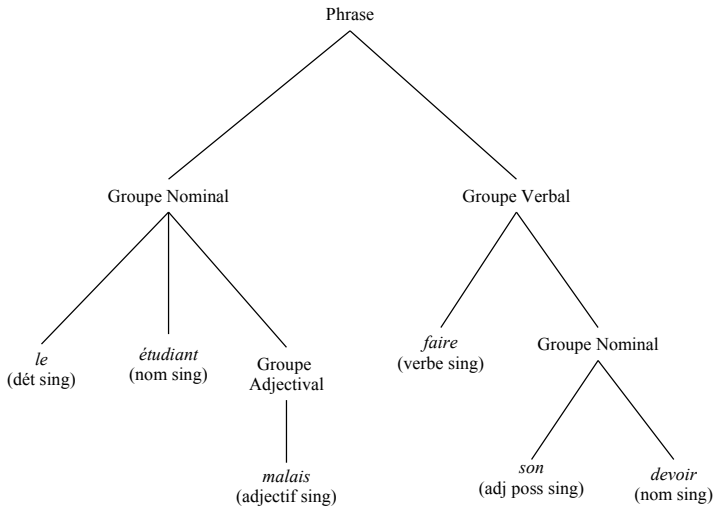


FIG. 8.8 – Analyse morpho-syntaxique de la phrase « *L'étudiant malais fait son devoir.* »

Transduction de l'arbre source vers l'arbre cible La transduction se fait en appliquant des règles sur l'arbre morpho-syntaxique de la langue source. Dans notre exemple, seule la règle de l'inversion du nom et de l'adjectif s'applique (cf. figure 8.9).

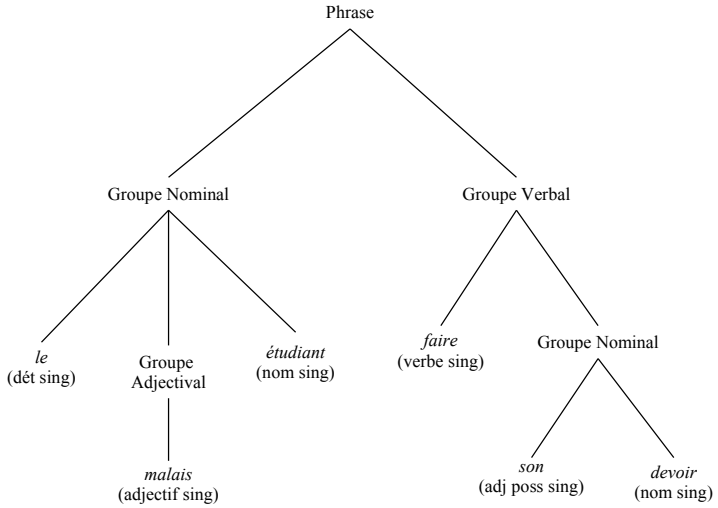


FIG. 8.9 – arbre morpho-syntaxique de la phrase « *L'étudiant malais fait son devoir.* » après transduction vers l'anglais

Traduction des termes : génération de texte Le transfert lexical s'opère en faisant une analyse sémantique de la phrase en français à partir de l'arbre de la figure 8.8. À la stabilisation des vecteurs de l'arbre, on a, pour chacune des feuilles, un vecteur qui correspond aux idées de l'item contextualisées par celles portées par le reste de la phrase.

Le choix du meilleur équivalent se fait en ordonnant les équivalents de l'item d'une feuille donnée en fonction de la distance des vecteurs de chacun de ses équivalents au vecteur de cette

‹*le*› ⟶ ‹*the*›
‹*étudiant*›⟶ ‹*student*›
‹*malais*› ⟶ ‹*malay*›
‹*faire*› ⟶ ‹*to do*›, ‹*to make*›
‹*son*› ⟶ ‹*his*›
‹*devoir*› ⟶ ‹*duty*›, ‹*homework*›

FIG. 8.10 – Les items de la phrase « *L'étudiant malais fait son devoir.* » et ses équivalents en anglais

feuille. Ainsi, dans notre exemple présenté dans la figure 8.10, le vecteur correspondant au *nom* ‹*devoir*› possédera des idées d'*ENSEIGNEMENT*, d'*APPRENTISSAGE* et sera ainsi plus proche du vecteur d'‹*homework*› que de celui de ‹*duty*› qui possède, lui, des idées plutôt proches de *MORALE* et du concept *DEVOIR* qui, dans le thésaurus Larousse, exprime l'idée du devoir moral.

La sélection uniquement basée sur les vecteurs n'est pas forcément possible. Ainsi comment choisir entre ‹*to do*› et ‹*to make*› sans apport d'information concernant les collocations ? Il faut savoir qu'on dit plutôt « *to do his homework* » que « *to make his homework* ». Avec notre méthode, ce type d'informations peut être apporté de deux manières au système :

- *en utilisant les fonctions lexicales de l'anglais* qu'une base lexicale sémantique basée sur les vecteurs conceptuels posséderait.
- *en utilisant des règles SYGMART codant ces fonctions lexicales.* C'est cette méthode qui a été mise en place par Violaine Prince pour la réalisation de son prototype.

Cette seconde méthode apporte une plus grande précision mais un rappel bien moindre. En revanche, celle qui utilise une acquisition automatique de ces relations pourrait permettre une plus rapide adaptativité, donc un rappel plus grand mais une précision moindre. Une solution intermédiaire pourrait être d'acquérir automatiquement les relations et de les mettre en règle manuellement après vérification ce qui est relativement identique au rôle joué par le superviseur dans une BLS.

La figure 8.11 présente l'arbre morpho-syntaxique où les équivalences ont été mises en place.

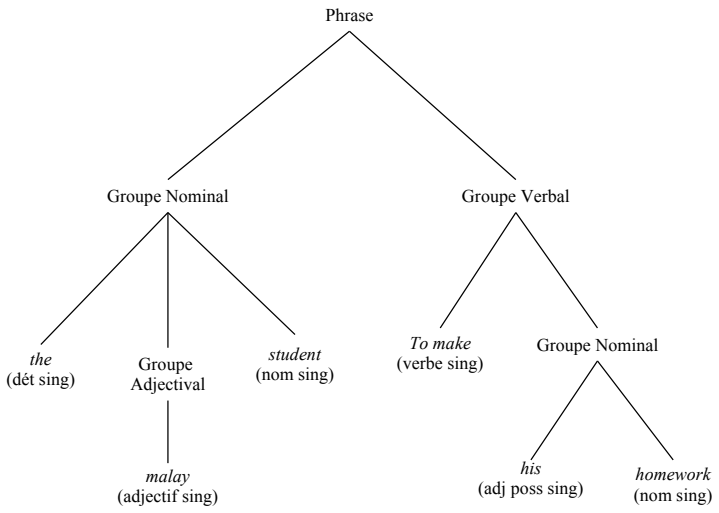


FIG. 8.11 – arbre morpho-syntaxique de la phrase « *L'étudiant malais fait son devoir.* » après transduction vers l'anglais

Applications des règles grammaticales de la langue cible À partir de cet arbre, il faut maintenant appliquer les règles grammaticales de la langue cible afin d'obtenir une phrase correcte. Cette partie peut être réalisée par SYGMART. Pour notre exemple, il s'agira, entre autres, d'accorder le verbe avec le sujet pour obtenir la traduction « *The malay student is doing his homework.* ».

8.2 Perspectives : affinage de la base

L'approche présentée dans la section précédente a permis la construction rapide d'une base de l'anglais à partir d'une base du français déjà existante. Les résultats peuvent être considérés comme encourageants même s'il est bien évident que des méthodes comme celles présentées dans les chapitres précédents sur le français devraient être mises en œuvre pour permettre un affinage de la base. Certains outils (agents) peuvent facilement être adaptés tandis que d'autres ne peuvent absolument pas l'être. On peut ainsi considérer deux types d'agents : *les agents hors langue* et les *agents en langue*. Ces distinctions se font en fonction de la nature des programmes et des données conservées en mémoire.

8.2.1 Agents hors langue

8.2.1.1 Définition

Il s'agit des agents dont les données gardées en mémoire sont soit inexistantes soit indépendantes de la langue. Ces dernières sont de deux natures :

- *numérique* comme les vecteurs ou les fréquences ;
- *symbolique* comme, dans une certaine mesure, les données morphologiques[114].

Ces agents n'ont pas besoin d'être réimplémentés ni réentrainés pour être réutilisés avec une autre langue. On peut même les partager entre des bases évoluant sur des langues différentes.

8.2.1.2 Type d'agents concernés

Parmi les agents de cette classe, on peut citer :

- les `agents experts en synonymie` dont les seules données utilisées sont des vecteurs conceptuels, de la morphologie et des fréquences par l'utilisation de la contextualisation forte (cf. 4.1.2).
- les `agents experts en antonymie naïve`, ceux qui implémentent la partie de la relation d'antonymie qui peut être modélisée uniquement grâce à la contextualisation forte c'est-à-dire ceux qui n'exploitent pas les rapports d'antonymie rencontrés dans les dictionnaires ou extraits de textes mais qui permettent uniquement de gérer certaines tournures négatives (cf. 3.4.2).
- les *agents* `catégoriseurs` qui permettent de fabriquer les ACCEPTIONS à partir des LEXIES et qui ne conservent pas de données en mémoire.

8.2.2 Agents en langue

8.2.2.1 Définition

Il s'agit des agents qui sont tout ou partie dépendants de la langue. Ils se répartissent en deux catégories, ceux qui acquièrent leur données de façon totalement automatique et ceux dont l'intervention d'un être humain est indispensable par l'ajout de règles par exemple. Les agents

[114] Toutes les langues ne semblent pas posséder des verbes, des noms ou des adjectifs. Certaines théories considèrent, par exemple, qu'il n'existe pas d'adjectifs dans la langue thaïe ([Teeraparbseree, 2005], p. 26).

issus de ce premier type peuvent être mis dans une nouvelle base (mais pas partagés) tandis que ceux issus du second ne le peuvent pas.

Parmi les agents de cette classe, on peut citer :
- l'`analyseur morpho-syntaxique` qui sert d'interface à SYGFRAN qui ne permet d'analyser que des textes en français (cf. 5.4.3). Pour permettre une analyse sémantique semblable à celle utilisée pour le français, il faudra alors chercher un analyseur spécifique à la langue cible. Il n'existe pas d'équivalent à SYGFRAN pour l'anglais mais on peut, par exemple, exploiter *Apple Pie Parser*[115] [Sekine & Grishman, 1995] basé sur les statistiques (cf. 1.1.4.2).
- les `extracteurs de définitions` qui sont des agents spécifiques à chaque dictionnaire et donc par conséquent à chaque langue.
- les `extracteurs de relations lexicales` qui, soit recherchent dans des dictionnaires spécialisés pour les relations symétriques et sont donc spécifiques à chaque langue, soit sont basés sur des schémas pour extraire des relations dans des textes et donc nécessitent l'intervention d'un expert.

Dans la classe des agents que l'on peut modifier :
- l'agent `base lexicale` qui conserve les objets lexicaux d'une langue donnée.
- Les *agents experts en relations lexicales* : ceux qui calculent les fonctions d'évaluation correspondant aux fonctions lexicales et utilisent le réseau de l'agent `base lexical`.

À l'aide de ces agents, il est ensuite possible de lancer une collaboration entre les deux bases.

8.3 Collaboration entre plusieurs bases

8.3.1 Motivations

On souhaite faire communiquer une base lexicale avec d'autres bases de même nature qui constituent autant de sources d'information complémentaires. Les informations issues des autres bases peuvent à la fois avoir un rôle de solidification des relations entre termes, mais également de découverte potentielle de nouveaux termes.

Nous appellerons *base cible* la base qui cherche à obtenir des informations d'une autre base et *base source* la base qui lui fournit ces informations. Nous cherchons à faire de la co-évolution, il est donc clair que les deux bases sont, à tout moment, à la fois base source et base cible mais cette terminologie permet de décrire les processus mis en jeu lors d'une requête de manière localiste. Comme dans d'autres sections de cette thèse, il faut comprendre, ici, le terme "base" comme le système global et non pas l'agent particulier de gestion des données `base`.

De façon tout à fait générique, la base cible interroge la base source à l'aide de requêtes et obtient (ou non) une information relative à une fonction lexicale pour un argument donné. Cette information est considérée comme une nouvelle source de données pour la base lexicale sémantique au même titre que des dictionnaires classiques, des dictionnaires de relations sémantiques ou d'informations issues du Web (cf chapitre 5.1).

On remarquera, que faire communiquer au moins deux bases lexicales et utiliser les informations obtenues dans le processus d'apprentissage induit un phénomène de co-évolution. Ce phénomène n'a d'intérêt en terme de qualité des informations lexicales construites que si les bases disposent d'autres sources lexicales et mettent en place des processus de calcul qui ne sont pas communs.

Les agents n'échangent pas leurs états mentaux c'est-à-dire qu'ils n'échangent à aucun moment et de façon directe leurs vecteurs, ils ne font qu'échanger des termes associés pondérés ou

[115] téléchargeable gratuitement à l'adresse http://nlp.cs.nyu.edu/app/

non. L'idée est de faire un système générique qui puisse fonctionner entre les deux bases même si celles-ci ne partagent pas le même espace vectoriel voire même si elles utilisent des méthodes de représentation totalement différentes. Il n'y a ainsi aucune présupposition sur la nature de la base source. Les bases échangent ainsi des symboles que chacune encode à destination de l'autre qui devra alors les décoder.

Dans le cas de bases dont les langues sont différentes, les échanges peuvent se faire via des dictionnaires de traduction et suivant les Fonctions Lexicales concernées. Ainsi, les FLA pour la linguistique ne pourraient être échangées facilement tandis que celles pour les connaissances du monde peuvent l'être. Une collocation ne se retrouve pas forcément de façon simple comme en témoignent par exemple les travaux de [Léon & Millon, 2005] tandis qu'une connaissance du monde comme *Mero*(‹*main*›) = ‹*pouce*› se traduit en *Mero*(‹*hand*›) = ‹*thumb*› ou que ‹*Napoléon*› est une instance d'‹*empereur*› implique que ‹*Napoleon*› est une instance d'‹*emperor*›. Toutefois ce ne sont que des pistes et elles n'ont pas étés testé dans un cadre collaboratif multilingue.

Cette méthode est cognitivement réaliste. En effet, on peut comparer deux bases à deux êtres humains qui échangeraient des informations entre eux. Quelle que soit la manière dont ils conservent les données, ils reçoivent de leur interlocuteur un énoncé qu'ils doivent encoder dans leur propre système de représentation.

8.3.2 Requêtes

8.3.2.1 Généralités

Une requête d'un agent A vers un agent B contient de façon générique les informations suivantes :

– la fonction lexicale d'évaluation concernée f ;
– l'argument de cette fonction x.

La fonction f est soit une fonction lexicale d'évaluation (cf. 6.4.2.2), soit la proximité thématique (cf 2.1.3.2). Pour les valeurs d'argument, il peut s'agir de termes (‹*frégate*›) ou de termes contextualisés et annotés à la manière de [Jalabert & Lafourcade, 2004b] c'est-à-dire grâce à un autre terme du lexique (*frégate*/**navire**). En aucun cas, il ne s'agit de concepts ou d'informations spécifiques à la base lexicale (objets lexicaux).

La requête peut être soit globale soit locale. Une *requête globale* consiste à demander les n premières valeurs possibles pour $f(x)$. Une *requête locale* consiste à demander la vérification d'une valeur $y = f(x)$.

8.3.2.2 Requêtes globales

Une requête globale est une fonction Q_g qui renvoie les n items lexicaux de la base source les plus proches d'un item lexical x selon la fonction lexicale d'évaluation f. En pratique, il s'agit pour la base lexicale cible $\mathcal{B}_c$ de demander à la base lexicale source $\mathcal{B}_s$ le voisinage de x selon la fonction f et les objets lexicaux $u_1, \ldots, u_n$ (cf 6.4.2.3).

$$\mathcal{F} \times \omega \times \mathbb{N} \rightarrow \omega^n : f, x, n \rightarrow Q_g(f, x) = \mathcal{V}_{B_s}(f, x, u_1, \ldots, u_m) \tag{8.6}$$

où $\mathcal{F}$ est l'ensemble des FLE et ω l'ensemble des items lexicaux.

8.3.2.3 Requêtes point à point

La requête point à point $Q_{pp}(f,x,y)$ doit spécifier en plus des arguments f et x, la valeur y ciblée. Le résultat attendu est la valeur de la fonction lexicale d'évaluation f (cf. 6.4.2.2).

Ainsi, la base lexicale cible peut, par exemple, demander à la base source la valeur de la relation d'intensification *Magn* entre ‹*fièvre*› et ‹*forte*› ou la valeur de la fonction de méronymie

entre ‹*corps*› et ‹*bras*›. La valeur 0 indique que la base interrogée ne dispose pas de valeur pour $f(x,y)$. Une valeur de 0 doit être considérée comme étant non-interprétable et ne peut être retenue par la base cible.

8.3.3 Stratégie d'apprentissage et expérimentation

8.3.3.1 Stratégie globale

Les deux types de requêtes sont d'un usage différent. Les requêtes globales servent à faire de la découverte lexicale, les requêtes point à point permettent de valider ou d'invalider certaines informations. On parle, dans ce cas-là, de *consolidation*.

Nous avons expérimenté notre approche par la mise en place de ces modes de communication entre la base lexicale sémantique mise au point au cours de cette thèse et implémentée par le système Blexisma et celle mise au point par Mathieu Lafourcade (cf. 2.1.1). Il s'agit ainsi de deux bases de vecteurs conceptuels dont la manière de représentation des données et de calcul de vecteurs sont en partie différentes.

De façon tout à fait empirique, nous avons établi des périodes alternatives de 5 heures, où la base Blexisma adoptait soit une approche de découverte lexicale soit une approche de consolidation. Ces deux phases se sont faites parallèlement à l'apprentissage sur des données classiques (dictionnaires, relations sémantiques, Web, ...).

Le traitement effectif du résultat de la requête dépend de l'agent demandeur. Dans le cadre de l'expérience menée, les résultats des fonctions lexicales sont intégrés directement dans le réseau lexical. En ce qui concerne le voisinage thématique, il est ajouté à la base vectorielle comme une nouvelle source d'informations permettant d'affiner le calcul du ou des vecteurs conceptuels.

Expérimentalement, nous avons observé que l'influence des informations obtenues depuis la base source ne devait pas dépasser un certain seuil. Nous estimons que l'influence des informations lexicales issues des dictionnaires et issues d'une autre base doit être sensiblement équivalente. Si la communication est trop faible, on ne profite pas assez de l'effet induit par le partage d'informations. Si elle est trop importante, trop peu sont « déduites » de la base et celle-ci se contente de « recopier » les informations des autres bases.

8.3.3.2 Stratégie locale

Pour la base source, la question de savoir si elle doit tenter d'évaluer les résultats d'une requête avant de les intégrer se pose. En pratique, l'agent n'est pas en mesure d'évaluer la pertinence d'un résultat de requête, tout au plus peut-il évaluer la distance entre ce résultat et celui qu'il aurait lui-même produit.

On considèrera que plus le niveau de confiance de la base dans ses propres résultats est élevé moins il accordera de crédit au résultat de la requête. L'influence d'un résultat est déterminée pour une valeur de fonction point à point. Les valeurs de requêtes globales peuvent être considérées comme une liste de valeurs point à point.

Soit v_c la valeur possédée par la base cible pour la fonction et les arguments demandés et v_s la valeur renvoyée par la base source.

La valeur v_c est mise à jour selon la formule :

$$v_{c_{t+1}} = \frac{v_{c_t} + v_s \times \sin(v_{c_t})}{1 + \sin(v_{c_t})} \tag{8.7}$$

où v_{c_t} est la valeur de v_c au temps t.

Si $v_c = 1$ l'agent ne tient pas compte du résultat puisque la valeur 1 correspond en général à une évaluation par un expert et ne saurait ainsi être remise en cause. Si $v_c = 0$, c'est-à-dire si la relation n'existait pas dans $\mathcal{B}_c$, l'agent la rajoute telle quelle. Pour les cas intermédiaires,

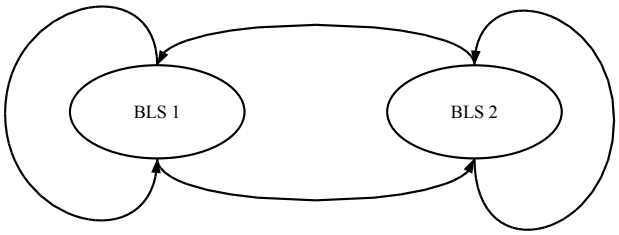


FIG. 8.12 – La double boucle dans la co-évolution de deux bases

l'influence de la nouvelle valeur est d'autant plus grande que la valeur précédente du demandeur est faible.

8.3.4 La double boucle à l'échelle de la base

Nous obtenons un phénomène de double boucle quand deux bases, en plus de leur apprentissage se mettent à partager leurs informations. Cette double boucle est à un niveau supérieur à celles décrites dans les chapitres précédents puisque l'une des deux est externe au système lui-même. La figure 8.12 présente la structure en double boucle de la co-évolution des deux bases. Il y a une double boucle pour chacun des systèmes, chacun ayant son apprentissage propre et chacun bénéficiant des apports de l'autre.

Dans le schéma que nous avons expérimenté, le partage se fait sur la base de la demande, et dans un premier temps l'agent sollicité n'est pas influencé par les demandes qu'il reçoit.

La stratégie que nous avons testée dans un second temps et qui semble prometteuse est de tenir compte des requêtes dont un agent est l'objet dans la stratégie d'apprentissage. Sans rentrer dans le détail, plus une demande est récurrente pour une fonction f et un argument x, plus l'agent va avoir tendance à réviser x. De même, si une demande concerne un mot x inconnu, alors ce mot va être l'objet d'une recherche.

8.4 Conclusions et perspectives

Dans ce chapitre, nous avons montré le niveau le plus haut où nous pouvons trouver une double boucle dans une base lexicale sémantique à savoir lorsqu'une des boucles est interne tandis que l'autre est externe. Un tel phénomène intervient lorsque la base échange des données avec une autre base. Chacune possède son propre apprentissage (boucle interne) et utilise les informations de son interlocutrice pour réviser les siennes. Les échanges réguliers de données sur lesquelles les deux influent forment une boucle externe.

Cette collaboration peut se faire de deux manières différentes, soit avec une base d'une même langue soit même entre bases de langues différentes. Nous avons ainsi montré comment créer une base lexicale sémantique pour une langue à partir d'une base d'une autre langue puis cherché à savoir jusqu'à quel point nous pouvions réutiliser les agents implémentés pour le français dans le cadre d'une autre langue. La mise en collaboration de ces deux bases peut permettre de croiser encore plus d'informations, cette fois y compris multilingue et ainsi améliorer la pertinence globale des bases.

Conclusions de la partie III

Dans la dernière partie de cette thèse, nous sommes revenu sur son point de départ : l'analyse sémantique. Nous avons étudié quels étaient, selon nous, les points fondamentaux à résoudre dans le cadre des applications visées par l'équipe à savoir le résumé automatique, la traduction automatique, la catégorisation de textes ou la recherche d'informations. Parmi ces difficultés, on trouve, outre la désambiguïsation lexicale, la découverte de chemins d'interprétation possibles, les problèmes de références, les rattachements prépositionnels et enfin le dernier, qui nous intéresse plus particulièrement dans cette thèse, celui de l'instanciation des fonctions lexicales.

Dans la partie précédente, nous avions posé les hypothèses de départ que nous considérions importantes pour la création d'une base lexicale sémantique. La première d'entre elles postulait que nous pouvions décrire le sens des termes grâce à deux types d'informations, thématiques d'une part (vecteurs conceptuels) et lexical d'autre part (relations entre objets du lexique) (cf hypothèse I dite de représentation hybride du sens, section 5.1.1). Nous avons montré ici que cette hypothèse était justifiée en mettant en évidence, sur un corpus simple, l'effet bénéfique de l'utilisation conjointe de ces informations dans une analyse sémantique. Nous avons exhibé en particulier l'apport positif de la découverte et l'exploitation des fonctions lexicales pour une telle analyse. Nous avons montré que de telles analyses étaient impossibles avec une remontée-descente et que le paradigme des algorithmes à fourmis pouvait la remplacer avantageusement dans cette tâche. En revanche, elle induit d'autres problèmes comme l'arrêt du système ou la gestion des constructions passives qu'il conviendra d'étudier en détail.

Dans cette dernière partie, nous avons étudié également la mise en collaboration avec une autre base lexicale sémantique.Que cette dernière soit construite pour une autre langue ou bien soit simplement le fruit d'autres sources, d'autres heuristiques donc d'autres expériences, il s'agit ici encore de chercher à croiser le plus d'informations possible afin d'enrichir au maximum la base (cf hypothèse IV dite d'analyse multi-source, section 5.1.4). L'idée est d'échanger des informations non plus statiques comme avec les dictionnaires mais en constant apprentissage au sein de l'autre base. Vu d'une base lexicale sémantique, nous avons alors une boucle interne d'apprentissage ainsi qu'une boucle externe ce qui constitue ainsi le niveau le plus haut où nous pouvons trouver une double boucle (cf hypothèse VI dite de double boucle, section 5.1.6).

Conclusion

Conclusion

Les relations lexicales constituent une importante source d'informations lors d'une analyse sémantique de texte et leurs occurrences doivent être identifiées. Leur détection puis leur exploitation peuvent sensiblement améliorer non seulement la représentation thématique mais aussi la résolution de phénomènes linguistiques comme la sélection correcte des acceptions utilisées ou des rattachements prépositionnels.

Une Base Lexicale Sémantique est une base de données qui contient des informations permettant de résoudre ces tâches. Nous avons énoncé six hypothèses quant à la représentation et l'exploitation du sens sur lesquelles doit reposer sa construction : (I) une *représentation hybride du sens par une approche combinant approche thématique (vectorielle) et approche lexicale (relations lexicales)*, (II) une prise en compte des *relations sémantiques internes* (polysémie), (III) une *génération automatique* des ACCEPTIONS, (IV) la réalisation d'une *analyse multi-source* (à partir de dictionnaires classiques, de listes de synonymes, d'antonymes, de sites Web, ...), (V) un *apprentissage permanent* et (VI) l'hypothèse de *double boucle*. Notre approche se veut ainsi totalement holistique c'est-à-dire que nous pensons que tout élément d'un système plus complexe doit être considéré comme plongé dans un environnement sur lequel il intervient et qui, en retour, intervient sur lui. L'élément doit alors être observé dans cette perspective afin de pouvoir décrire le système dans sa globalité.

Nous avons défini les Fonctions Lexicales d'Analyse (FLA) qui permettent la modélisation de ces relations. Inspirées des fonctions lexicales de Mel'čuk, elles s'en éloignent cependant sur deux points : par l'utilisation dans un but d'analyse pour les premières et dans un but de production, de synthèse pour les secondes ; par leur nature, purement linguistique pour les secondes et à caractère mixte pour les premières (pour la linguistique et pour les connaissances du monde).

Nous avons introduit deux classes de FLA. Les premières, fonctions lexicales de construction, qui n'existent que pour la synonymie et l'antonymie, permettent de fabriquer un vecteur conceptuel à partir des informations lexicales disponibles. Les secondes, fonctions lexicales d'évaluation permettent de mesurer la pertinence d'une relation lexicale entre plusieurs termes. Ces dernières sont modélisables grâce, en partie à des informations thématiques (vecteurs conceptuels) et en partie à des informations lexicales (relations symboliques entre les objets lexicaux) pour la synonymie et l'antonymie, et uniquement à l'aide d'informations lexicales (Relations Lexicales Valuées) pour les autres FLA.

Les informations lexicales sont issues de la base lexicale sémantique dont nous avons présenté

l'architecture à trois niveaux d'objets lexicaux (ITEM LEXICAL, ACCEPTION, LEXIE). Elles sont matérialisées sous la forme de Relations Lexicales Valuées qui traduisent la probabilité d'existence de la relation entre les objets. Le calcul de ces probabilités est un travail de recherche en soi et il n'a pas pu être réellement abordé au cours de cette thèse sauf en ce qui concerne les échanges d'informations entre bases, que celles-ci soient dans la même langue ou une langue différente.

En revanche, l'utilité de ces relations a pu être mise en évidence pour l'analyse sémantique grâce à l'utilisation du paradigme des algorithmes à fourmis. Le modèle introduit dans cette thèse, utilise à la fois les vecteurs conceptuels et les relations du réseau lexical pour résoudre une partie des problèmes posés lors d'une analyse sémantique. Les expériences menées ont montré la faisabilité de cette approche et ouvert des perspectives de recherches fort intéressantes.

Tous nos outils ont été mis en œuvre en Java.Ils reposent sur Blexisma (*Base LEXIcale Sémantique Multi-Agent*) une architecture multi-agent, élaborée au cours de cette thèse, dont l'objectif est d'intégrer tout élément lui permettant de créer, d'améliorer et d'exploiter une ou plusieurs Bases Lexicales Sémantiques. Une quinzaine de type d'agents a ainsi été programmée et testée (base de données, apprentissage, experts en fonctions lexicales, analyse sémantique en remontée-descente, analyse sémantique par fourmis, ...). L'apprentissage et la révision des données de la base du français est permanent : il a occupé dans sa plus grande version jusqu'à 115 agents et tourné simultanément sur cinq machines du laboratoire (PC linux, Sun sous UNIX). Le suivi et le contrôle des agents sont accessibles via le Web[116]. En Septembre 2005, la base contient environ 121 000 ITEMS LEXICAUX pour 276 000 ACCEPTIONS et 842 000 LEXIES. Son exploitation reposant sur les six hypothèses présentées a permis leur validation expérimentale.

Après avoir amélioré la représentation du sens du côté purement lexical, il semble clair que le côté thématique doit être affiné. En effet, les vecteurs sont beaucoup trop tributaires de la hiérarchie, de ses lacunes et des choix effectués par les lexicographes. Ainsi, dans le thésaurus Larousse, certains concepts n'ont, par exemple, pas d'opposé alors qu'il devrait logiquement en exister un (*COMBUSTIBILITÉ*). Ce même, le raffinement semble souvent trop faible (un seul concept pour *INFORMATIQUE* ou *MÉDECINE*). De plus, l'horizon lexical est, comme nous l'avons montré au chapitre 6, fortement influencé par la hiérarchie. Une possibilité serait donc de s'affranchir de cette liste et de ne fonder la construction des vecteurs que sur les sources d'informations disponibles. L'idée ici serait de fixer au début de l'expérience un nombre donné de composantes puis de tirer au hasard les vecteurs conceptuels de tout terme non encore appris et utilisé dans les définitions de l'item lexical en cours d'indexation. Cette technique, compatible avec l'ensemble des voies explorées au cours de cette thèse hormis l'antonymie dans sa partie naïve, permettrait aux vecteurs de s'ajuster les uns par rapport aux autres. Nous n'aurions plus alors la pertinence du noyau alliée à la cohérence de l'analyse qui donne la pertinence de la base augmentée, mais seule la cohérence de la base qui donne sa pertinence.

[116] http://www.lirmm.fr/~schwab

Annexes

A

Espaces Vectoriels

L'approche par vecteurs d'idées repose sur quelques notions mathématiques, principalement liées à la notion d'espace vectoriels que nous rappelons ici brièvement. Avant de présenter les axiomes et les définitions concernant les espaces vectoriels, nous commençons par présenter quelques notions indispensables de structures algébriques.

A.1 Groupes, anneaux, corps

A.1.1 Groupes

Une loi T confère à un ensemble E la structure de groupe si elle vérifie les quatre propriétés suivantes : elle est interne (A.1), associative (A.2), elle possède un élément neutre de E (A.3) et chaque élément de E possède un symétrique par rapport à T (A.4).

$$\begin{aligned}
\text{interne} &: (\forall x,y \in E)\ [xTy \in E] && \text{(A.1)}\\
\text{associativité} &: (\forall x,y,z \in E)\ [(xTy)Tz = xT(yTz)] && \text{(A.2)}\\
\text{neutre} &: (\forall x \in E)(\exists e \in E)\ [xTe = eTx = x] && \text{(A.3)}\\
\text{symétrique} &: (\forall x \in E)(\exists x' \in E)\ [xTx' = x'Tx = e] && \text{(A.4)}
\end{aligned}$$

Nous noterons (E,T) un ensemble E muni d'une opération T.

A.1.2 Groupes abéliens

Un groupe (E,T) est un groupe abélien (ou commutatif), si T vérifie non seulement les propriétés des groupes définies précédemment mais aussi la commutativité.

$$\text{commutativité} : (\forall x,y \in E)\ [xTy = yTx] \qquad \text{(A.5)}$$

A.1.3 Propriétés des groupes

On peut déduire des axiomes précédents les propriétés suivantes :
- L'élément neutre est unique.
- Le symétrique x' d'un élément x est unique.
- $(\forall x,y \in E)\ [(xTy)' = y'Tx']$
- $(\forall x \in E)\ [(x')' = x]$
- $(\forall x,y,z \in E)\ [(xTy = xTz) \Rightarrow (y = z)]$

A.1.4 Anneaux

Un ensemble E possède la structure d'anneau s'il est muni de deux lois internes T et L possédant les propriétés suivantes :

- (E,T) possède la structure de groupe abélien.
- L est associative (A.6) et distributive (A.7) par rapport à T.

$$\text{associativité} \quad : \quad (\forall x,y,z \in E)\ [(xLy)Lz = xL(yLz)] \tag{A.6}$$
$$\text{distributivité} \quad : \quad (\forall x,y,z \in E)\ [(xTy)Lz = (xLz)T(yLz)] \tag{A.7}$$

A.1.5 Corps

Un corps est un anneau K tel que la deuxième loi, L, confère une structure de groupe à l'ensemble $K-\{e\}$ où e est l'élément neutre de la première loi, T. Pour L, il existe donc un élément neutre et chaque élément est symétrisable sauf ce dernier.

A.1.6 Corps $\mathbb{R}$

L'exemple de corps le plus connu est l'ensemble des nombres réels $\mathbb{R}$ qui a une structure de corps pour l'addition et la multiplication. L'addition est interne, associative, commutative, et elle définit un élément neutre, 0. Chaque élément possède un symétrique qui est son opposé (x a pour symétrique $-x$). L'addition confère donc à $\mathbb{R}$ la structure de groupe abélien. La multiplication est interne, associative et distributive par rapport à l'addition. $\mathbb{R}$ possède donc une structure d'anneau. La multiplication admet, de plus, un élément neutre 1, et chaque élément sauf 0 est symétrisable (le symétrique de x étant son inverse $\frac{1}{x}$). La multiplication étant, de plus, commutative, on dit que $\mathbb{R}$ est un corps commutatif.

Les bases des structures algébriques ayant été posées, nous allons maintenant présenter les axiomes des espaces vectoriels.

A.2 Axiomes des espaces vectoriels

Un espace vectoriel sur un corps K est un triplet $(E,+,.)$ où l'ensemble E est muni d'une loi interne (notée +), l'addition (A.8) et d'une loi externe (notée .), la multiplication scalaire (A.9).

$$(\forall \vec{u},\vec{v} \in E) \qquad [\vec{u}+\vec{v} \in E] \tag{A.8}$$
$$(\forall \lambda \in K)(\forall \vec{u} \in E) \qquad [\lambda.\vec{u} \in E] \tag{A.9}$$

Les éléments de E sont appelés *vecteurs*, il sont notés usuellement avec une flèche ($\vec{u}$) ou alors avec l'ensemble auquel ils appartiennent en indice (u_E). Par opposition, les éléments de K sont appelés *scalaires*, ils peuvent êtres notés, eux aussi, avec l'ensemble auquel ils appartiennent en indice (λ_K). Généralement, nous omettrons pour des raisons de clarté cet indice en ce qui concerne les scalaires.

$(E,+)$ est un groupe abélien, il vérifie donc les propriétés définies en A.1 :

$$\text{associativité} \quad : \quad (\forall \vec{u},\vec{v},\vec{w} \in E)\ [(\vec{u}+\vec{v})+\vec{w} = \vec{u}+(\vec{v}+\vec{w})] \tag{A.10}$$
$$\text{neutre} \quad : \quad (\forall \vec{u} \in E)\ [\vec{0}+\vec{u} = \vec{u}+\vec{0} = \vec{u}]\ \text{(le neutre pour } (E,+) \text{ est noté } \vec{0}) \tag{A.11}$$
$$\text{symétrique} \quad : \quad (\forall \vec{u} \in E)[\vec{u}+(-\vec{u}) = (-\vec{u})+\vec{u} = \vec{0}] \tag{A.12}$$
$$\text{commutativité} \quad : \quad (\forall \vec{u},\vec{v} \in E)\ (\vec{u}+\vec{v} = \vec{v}+\vec{u}) \tag{A.13}$$

$(E,.)$ vérifie l'existence d'un élément neutre, 1_K, l'élément unité de K (A.15), les règles de distributivité (A.16, A.17) et la règle d'associativité (A.14).

$$\begin{aligned}
\text{associativité} &: (\forall \vec{u} \in E)(\forall \lambda, \mu \in K)\ [\lambda.(\mu.\vec{u}) = (\lambda.\mu).\vec{u}] && \text{(A.14)}\\
\text{neutre} &: (\forall \vec{u} \in E)\ [1_K.\vec{u} = \vec{u}] && \text{(A.15)}\\
\text{distributivité} &: (\forall \vec{u}, \vec{v} \in E)(\forall \lambda \in K)\ [(\lambda.(\vec{u}+\vec{v})) = (\lambda.\vec{u}) + (\lambda.\vec{v})] && \text{(A.16)}\\
&\quad (\forall \vec{u} \in E)(\forall \lambda, \mu \in K)\ [(\lambda+\mu).\vec{u} = (\lambda.\vec{u}) + (\mu.\vec{u})] && \text{(A.17)}
\end{aligned}$$

Dans toute la suite de cette section, E est un espace vectoriel sur un corps K.

A.3 Propriétés

Des axiomes énoncées dans A.2 nous pouvons déduire les propriétés suivantes :

$$\begin{aligned}
&\forall(\vec{u} \in E) \quad (\forall \lambda \in K) \quad \text{si } \lambda.\vec{u} = \vec{0} \text{ alors } \lambda = 0 \text{ ou } \vec{u} = \vec{0} && \text{(A.18)}\\
&\forall(\vec{u} \in E) \quad (\forall \lambda \in K) \quad [(-\lambda).\vec{u} = \lambda.(-\vec{u}) = -(\lambda.\vec{u})] && \text{(A.19)}
\end{aligned}$$

A.4 Définitions générales

A.4.1 Familles de vecteurs et combinaisons linéaires

Dans l'espace vectoriel E, p éléments $\overrightarrow{u_1}, \overrightarrow{u_2}, \ldots, \overrightarrow{u_p}$ constituent une famille (ou système) de vecteurs de E. On appelle *combinaisons linéaires* sur cette famille les vecteurs construits grâce à des combinaisons entre les opérations linéaires que sont, par définition, l'addition et le scalaire.

$$\overrightarrow{u} = \sum_{i=1}^{i=p} \alpha_i \overrightarrow{u_i} \tag{A.20}$$

Par définition, on dit que les p éléments $\overrightarrow{u_1}, \overrightarrow{u_2}, \ldots, \overrightarrow{u_p}$ constituent une famille :

– *libre (les vecteurs sont indépendants)* si :

$$[\sum_{1}^{p} \alpha_i \overrightarrow{u_i} = \overrightarrow{0}] \Rightarrow [(\forall i \in \{1, 2, \ldots, p\})[\alpha_i = 0]] \tag{A.21}$$

– *liée (les vecteurs sont dépendants)* dans le cas contraire. Dans ce cas, $\exists$ des α_i non tous nuls tels que

$$\sum_{i}^{p} \alpha_i \overrightarrow{u_i} = \overrightarrow{0} \tag{A.22}$$

A.4.2 Sous-espaces vectoriels

Toute partie de l'espace vectoriel E qui possède la structure d'espace vectoriel sur le corps K est appelé *sous-espace vectoriel de* E. Pour montrer qu'une partie F de E est un sous-espace vectoriel de E, il faut montrer que F est stable par combinaison linéaire, c'est-à-dire que toute combinaison de deux vecteurs de F est un vecteur de F.

$$(\forall \vec{u}, \vec{v} \in F)(\forall \alpha, \beta \in K)\ [\alpha\vec{u} + \beta\vec{v} \in F] \tag{A.23}$$

A.4.3 Générateurs

Soit une famille de p vecteurs $\overrightarrow{u_1}, \overrightarrow{u_2}, \ldots, \overrightarrow{u_p}$ de E, l'ensemble des combinaisons linéaires de ces p vecteurs est un sous-espace F de E. On dit que F est *engendré* par cette famille (dite *famille génératrice*).

A.4.4 Bases et composantes

On appelle *base* d'un espace vectoriel E une famille libre de générateurs de E, $\mathcal{B} = \{\overrightarrow{e_1}, \overrightarrow{e_2}, \ldots, \overrightarrow{e_p}\}$. Tout vecteur $\vec{u}$ de E admet une décomposition unique suivant une base

$$\vec{u} = \sum_{1}^{n} x_i e_i \tag{A.24}$$

Les scalaires x_i sont appelés *composantes (ou coordonnées)* de $\vec{u}$ suivant la base $\mathcal{B}$.

A.4.5 Dimensions

Toutes les bases de E ont le même nombre d'éléments. Si E possède une base $\mathcal{B}$ de n vecteurs, alors on dit que n est la *dimension* de E et on note $dim E = n$. La dimension est aussi le nombre maximum d'éléments d'un système libre de E. Conventionnellement, l'espace réduit au vecteur nul $\vec{0}$ est de dimension 0.

A.4.6 Base canonique

Dans un espace E de dimension n, une base peut être favorisée. Dans cette base, dite *base canonique*, un vecteur $\vec{u}$ est représenté par le n-uplet $(x_1, x_2, \ldots, x_n)$. Les vecteurs de la base canonique sont :

$$\mathcal{B} = \{(1,0,0,\ldots,0),(0,1,0,\ldots,0),(0,0,1,\ldots,0)\ldots,(0,0,0\ldots,0,1)\}$$

Lorsque E est un espace vectoriel sur le corps $\mathbb{R}$ on peut assimiler E à $\mathbb{R}^n$ le produit cartésien $\mathbb{R} \times \mathbb{R} \times \ldots \times \mathbb{R}$.

Il est habituel, à cause du caractère souvent abstrait des espaces vectoriels, de ne pas noter de façon particulière les vecteurs et les scalaires. À partir de maintenant, pour une meilleure lisibilité, nous nous affranchirons d'une quelconque notation pour différencier les vecteurs et les scalaires. Par exemple, $\vec{u}$ sera noté u et 1_K, 1. De même, le produit scalaire précédemment noté . pourra être omis. Nous noterons alors λu au lieu de $\lambda.u$.

A.5 Espace vectoriel normé $\mathbb{R}^n$ sur $\mathbb{R}$

Nous allons maintenant nous intéresser plus particulièrement à l'espace vectoriel classique dans lequel évoluent les vecteurs conceptuels, $\mathbb{R}^n$ sur $\mathbb{R}$. Cet espace est muni d'une base orthogonale (*i.e* les vecteurs de cette base forment un angle droit deux à deux), la base canonique définie précédemment (A.4).

A.5.1 Produit scalaire et espace vectoriel euclidien

Un espace vectoriel E sur K possède un *produit scalaire*, si pour chaque paire de vecteurs $(u,v) \in E^2$ on définit un produit scalaire noté $\cdot$ tel que $u \cdot v \in E$ qui satisfait les axiomes suivants :

$$(\forall u, v \in E)\ [u \cdot v = v \cdot u] \tag{A.25}$$
$$(\forall u, v \in E)\ [u \cdot (v + w) = u \cdot v + u \cdot w] \tag{A.26}$$
$$(\forall \lambda \in K]\ (\forall u, v \in E)\ [\lambda u \cdot v = u \cdot \lambda v] \tag{A.27}$$
$$(\forall u \in E)\ [u \cdot u > 0] \text{ si } u \neq \vec{0}] \tag{A.28}$$
$$[u \cdot u = 0] \text{ si } u = \vec{0} \tag{A.29}$$

Le produit scalaire sur $\mathbb{R}^n$ est définie par :

$$\mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R} : u \cdot v = \sum_{i=1}^{n} u_i v_i \tag{A.30}$$

Un espace vectoriel sur $\mathbb{R}$ muni du produit scalaire est appelé *espace euclidien réel*.

A.5.2 Norme

Dans un espace vectoriel euclidien E sur $\mathbb{R}$, une *norme* est une application de E dans $\mathbb{R}^+$ qui, à tout vecteur u, associe le nombre noté habituellement $\| u \|$ et vérifie :

$$\| u \| = 0 \text{ si et seulement si } u = \vec{0} \tag{A.31}$$
$$\forall \lambda \in K, \| \lambda u \| = |\lambda| \ \| u \| \tag{A.32}$$
$$\forall u, v \in E^2, \| u + v \| \leq \| u \| + \| v \| \tag{A.33}$$

Dans l'espace euclidien $\mathbb{R}^n$, le calcul de cette norme est donné par :

$$\mathbb{R}^n \rightarrow \mathbb{R}^+ : \ \| u \| = \sqrt{u \cdot u} = \sqrt{x_1^2 + x_2^2 + \ldots + x_n^2} \tag{A.34}$$

Muni de cette norme, un espace vectoriel est dit *espace vectoriel normé*.

A.5.3 Distance et mesure

On appelle *distance* une application d de E^2 dans $\mathbb{R}^+$ vérifiant $\forall x, y \in E^2$ les axiomes de séparation, de symétrie et d'inégalité triangulaire :

$$\text{séparation} \ : \ d(x,y) = 0 \Leftrightarrow x = y \tag{A.35}$$
$$\text{symétrie} \ : \ d(x,y) = d(y,x) \tag{A.36}$$
$$\text{inégalité triangulaire} \ : \ d(x,y) \leq d(x,z) + d(z,y) \tag{A.37}$$

Dans cette thèse, nous ne parlerons de distance que lorsque les trois axiomes sont vérifiés. Nous parlerons de *mesure* lorsque l'inégalité triangulaire ne le sera pas.

A.5.4 Angle entre deux vecteurs

Dans $\mathbb{R}^n$, la produit scalaire peut être aussi donné par :

$$\mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R} : u \cdot v = \| u \| \times \| v \| \times \cos(\widehat{u,v}) \tag{A.38}$$

De cette formule, il est facile de déduire le cosinus de l'angle entre deux vecteurs dans un espace euclidien réel tel que $\mathbb{R}^n$:

$$\mathbb{R}^n \times \mathbb{R}^n \to [-1,1] : \cos(\widehat{u,v}) = \frac{u \cdot v}{\| u \| \times \| v \|} \tag{A.39}$$

L'angle entre deux vecteurs est donc donné par :

$$\mathbb{R}^n \times \mathbb{R}^n \to [0,\pi] : (\widehat{u,v}) = \arccos\left(\frac{u \cdot v}{\| u \| \times \| v \|}\right) \tag{A.40}$$

où *arccos* désigne la fonction *arccosinus* qui est la fonction réciproque de la fonction *cosinus*.

L'angle entre deux vecteurs est une distance, les trois axiomes des distances sont trivialement démontrables.

L'espace vectoriel E muni d'une distance d est qualifié d'*espace métrique*.

B

La hiérarchie Larousse

CLASSE I. LE MONDE

SECTION I. LES CONCEPTS FONDAMENTAUX

1. Existence
   Existence(1), Inexistence(2), Matérialité(3), Immatérialité(4), Substance(5), Accident(6), État(7), Circonstance(8), Présence(9), Absence(10), Apparition(11), Disparition(12)
2. Identité
   Relation(13), Indépendance(14), Identité(15), Altérité(16), Ambivalence(17), Opposition(18), Substitution(19), Réciprocité(20), Ressemblance(21), Dissemblance(22), Différence(23), Uniformité(24), Diversité(25), Concordance(26), Discordance(27), Conformité(28), Non-conformité(29), Modèle(30), Imitation(31), Innovation(32), Variation(33)
3. Causalité
   Cause(34), Effet(35), Agent(36), Motif(37), But(38), Possibilité(39), Impossibilité(40), Nécessité(41), Éventualité(42), Probabilité(43), Hasard(44)

SECTION II. L'ORDRE ET LA MESURE

1. Ordre
   Ordre(45), Désordre(46), Organisation(47), Désorganisation(48), Classification(49), Méthode(50), Système(51), Règle(52), Norme(53), Normalité(54), Anormalité(55), Commencement(56), Milieu(57), Fin(58), Antériorité(59), Postériorité(60), Continuité(61), Discontinuité(62), Rang(63), Série(64), Gradation(65), Groupement(66), Inclusion(67), Exclusion(68)
2. Quantité
   Quantité(69), Mesure(70), Totalité(71), Partie(72), Unité(73), Pluralité(74), Multitude(75), Répétition(76), Complexité(77), Abondance(78), Paucité(79), Excès(80), Manque(81), Satiété(82), Égalité(83), Inégalité(84), Supériorité(85), Infériorité(86), Intensité(87), Augmentation(88), Diminution(89), Réunion(90), Séparation(91), Intégration(92), Dissociation(93), Proportion(94), Fraction(95), Reste(96), Adjonction(97), Mélange(98), Compensation(99)
3. Nombre
   Nombre(100), Zéro(101), Un(102), Deux(103), Trois(104), Quatre(105), Cinq(106), Six(107), Sept(108), Huit(109), Neuf(110), Dix(111), Douze(112), Cent(113), Mille(114), Infini(115), Calcul(116), Chiffre(117), Addition(118), Soustraction(119), Multiplication(120), Division(121), Mathématique(122)

SECTION III. L'ESPACE

1. Dimensions
   Dimension(123), Longueur(124), Largeur(125), Hauteur(126), Grosseur(127), Petitesse(128), Étroitesse(129)
2. Contours
   Extérieur(130), Intérieur(131), Bord(132), Centre(133), Contenant(134), Contenu(135), Limite(136), Revêtement(137), Barrière(138), Ouverture(139), Fermeture(140)
3. Formes
   Forme(141), Rectitude(142), Angularité(143), Courbure(144), Cercle(145), Géométrie(146)
4. Structures
   Structure(147), Ligne(148), Croix(149), Bande(150), Pointe(151), Bosse(152), Creux(153), Grain(154), Poli(155)
5. Situation

Situation(156), Environnement(157), Intervalle(158), Soutien(159), Suspension(160), Proximité(161), Distance(162), Devant(163), Derrière(164), Dessus(165), Dessous(166), Côté(167), Droite(168), Gauche(169)

SECTION IV. LE TEMPS

1. Temps et durée
   Temps(170), Permanence(171), Durée(172), Éternité(173), Instant(174)
2. Date et chronologie
   Chronologie(175), Calendrier(176), Passé(177), Présent(178), Futur(179), Avance(180), Retard(181), Simultanéité(182), Fréquence(183), Rareté(184), Période(185), Moment(186), Saisons(187), Matinée(188), Soirée(189)
3. Évolution et histoire
   Évolution(190), Histoire(191), Événement(192), Changement(193), Nouveauté(194), Ancienneté(195), Désuétude(196)

SECTION V. LE MOUVEMENT

1. Le mouvement et ses directions
   Mouvement(197), Direction(198), Rapprochement(199), Éloignement(200), Arrivée(201), Départ(202), Entrée(203), Sortie(204), Pénétration(205), Extraction(206), Réception(207), Éjection(208), Expansion(209), Contraction(210), Montée(211), Descente(212), Saut(213), Chute(214), Rotation(215), Oscillation(216), Agitation(217), Déviation(218), Dépassement(219), Inversion(220)
2. Les forces et leurs actions
   Force(221), Traction(222), Attraction(223), Répulsion(224), Impulsion(225), Équilibre(226), Choc(227), Frottement(228), Inertie(229)

SECTION VI. LA MATIÈRE

1. Les sciences de la matière
   Chimie(230), Microphysique(231), Astronomie(232), Mécanique(233), Optique(234), Électricité(235), Magnétisme(236), Géologie(237)
2. Les propriétés de la matière
   Densité(238), Poids(239), Légèreté(240), Chaleur(241), Froid(242), Combustibilité(243), Humidité(244), Sécheresse(245), Solidité(246), Fragilité(247), Rigidité(248), Élasticité(249), Mollesse(250), Pulvérulence(251)
3. Les éléments et les matériaux
   Liquide(252), Gaz(253), Bulle(254), Air(255), Feu(256), Terre(257), Minéraux(258), Minerais(259), Or(260), Argent(261), Fer(262), Bronze(263), Plomb(264), Bois(265), Verre(266), Huile(267)
4. L'environnement terrestre
   Région(268), Plaine(269), Montagne(270), Flots(271), Désert(272), Climats(273), Pluie(274), Vent(275), Nuages(276), Soleil(277), Lune(278)

SECTION VII. LA VIE

1. Le vivant
   Reproduction(279), Hérédité(280), Embryologie(281), Écologie(282), Cellule(283), Micro-organismes(284)
2. Les plantes
   Botanique(285), Arbres(286), Arbustes(287), Fleurs(288), Fruits(289), Herbes et fougères(290), Champignons(291), Mousses et hépatiques(292), Algues(293), Lichens(294)
3. Les animaux
   Zoologie(295), Mammifères(296), Oiseaux(297), Poissons(298), Reptiles(299), Batraciens(300), Insectes et arachnides(301), Crustacés(302), Mollusques et petits animaux marins(303), Vers(304), Cris et bruits d animaux(305)

CLASSE II. LE MONDE

SECTION I. L'ÊTRE HUMAIN

1. Les humains
   Humains(306), Personne(307), Homme(308), Femme(309)
2. L'âge de la vie
   Vie(310), Mort(311), Âge(312), Naissance(313), Enfance(314), Jeunesse(315), Maturité(316), Vieillesse(317)

SECTION II. LE CORPS ET LA VIE

1. Le corps

Judaïsme(486), Christianisme(487), Islam(488), Bouddhisme(489), Hindouisme(490)
4. Les cultes et les pratiques
Culte(491), Religieux et ministres des cultes(492), Lieux de culte(493), Prière(494), Prédication(495), Messe(496), Fêtes religieuses(497), Pape(498), Moines(499)
5. Les croyances
Divinités(500), Textes sacrés(501), Dieu(502), Ange(503), Démon(504), Paradis(505), Enfer(506)

SECTION VIII. LA VOLONTÉ
1. Décision et indécision
Volonté(507), Courage(508), Lâcheté(509), Résolution(510), Irrésolution(511), Persévérance(512), Défection(513), Obstination(514), Renonciation(515)
2. Le libre-arbitre et la nécessité
Liberté(516), Fatalité(517), Obligation(518), Choix(519), Refus(520), Prétexte(521), Caprice(522), Désir(523), Indifférence(524), Persuasion(525), Dissuasion(526)

SECTION IX. L'ACTION
1. L'action et l'inaction
Action(527), Réaction(528), Inaction(529), Effort(530), Repos(531)
2. Le projet et son résultat
Intention(532), Tentative(533), Projet(534), Entreprise(535), Préparation(536), Impréparation(537), Accomplissement(538), Inaccomplissement(539), Succès(540), Échec(541)
3. Les occasions et les circonstances
Opportunité(542), Inopportunité(543), Utilité(544), Inutilité(545), Facilité(546), Difficulté(547), Prospérité(548), Adversité(549), Sécurité(550), Danger(551), Avertissement(552), Alarme(553), Obstacle(554), Détection(555)
4. Les objectifs
Construction(556), Destruction(557), Réparation(558), Préservation(559), Protection(560), Annulation(561)
5. La participation
Participation(562), Aide(563), Stimulation(564), Encouragement(565), Conseil(566)
6. Les manières d'agir
Usage(567), Habitude(568), Abus(569), Adresse(570), Maladresse(571), Prudence(572), Imprudence(573), Soin(574), Négligence(575), Rapidité(576), Lenteur(577), Ponctualité(578), Modération(579), Violence(580)

CLASSE III. LA SOCIÉTÉ
SECTION I. LE RAPPORT A L'AUTRE
1. Les comportements
Sociabilité(581), Insociabilité(582), Compagnie(583), Solitude(584), Bonté(585), Méchanceté(586), Générosité(587), Égoïsme(588), Gratitude(589), Hospitalité(590), Inhospitalité(591), Courtoisie(592), Discourtoisie(593), Loyauté(594), Hypocrisie(595), Promesse(596), Trahison(597), Délicatesse(598), Dureté(599)
2. Les sentiments
Amour(600), Caresse(601), Passion(602), Ressentiment(603), Amitié(604), Inimitié(605), Confiance(606), Défiance(607), Jalousie(608), Pitié(609)
3. L'image de soi
Fierté(610), Honte(611), Modestie(612), Prétention(613), Distinction(614), Affectation(615), Simplicité(616), Ostentation(617), Timidité(618), Décence(619), Indécence(620)

SECTION II. LE RAPPORT HIÉRARCHIQUE
1. Autorité et soumission
Autorité(621), Domination(622), Influence(623), Obéissance(624), Désobéissance(625), Respect(626), Irrespect(627), Soumission(628), Servilité(629), Résistance(630)
2. Commandement et consentement
Commandement(631), Autorisation(632), Interdiction(633), Demande(634), Consentement(635)
3. Louange et repproche
Louange(636), Reproche(637), Pardon(638)
4. Le prestige social
Gloire(639), Ostracisme(640), Honneur(641), Discrédit(642), Promotion(643), Éviction(644), Ridicule(645), Noblesse(646), Roture(647), Titres(648)

SECTION III. GUERRE ET PAIX
1. Le conflit et le compromis

Conflit(649), Guerre(650), Révolution(651), Paix(652), Compromis(653), Pacte(654)
2. Les épisodes du conflit
Attaque(655), Défense(656), Injure(657), Coup(658), Représailles(659), Victoire(660), Défaite(661), Revanche(662)
3. La force armée
Armée(663), Armes(664), Armement ancien(665), Manoeuvres(666), Tir(667)

SECTION IV. LA VIE COLLECTIVE
1. Société et organisation politique
Société(668), Politique(669), Régime(670), Systèmes politiques(671), Élection(672), Représentants(673)
2. Citoyenneté
Citoyen(674), Civisme(675), Habitant(676), Étranger(677)
3. La famille
Famille(678), Père(679), Mère(680), Filiation(681), Mariage(682), Célibat(683), Divorce(684)
4. Les coutumes
Coutume(685), Cérémonies(686), Fête(687), Funérailles(688), Salutations(689)

SECTION V. LA MORALE
1. La loi morale
Morale(690), Devoir(691), Prescription(692), Honnêteté(693), Malhonnêteté(694), Mérite(695), Imperfection(696), Péché(697), Expiation(698)
2. Les vertus et les vices
Vertu(699), Vice(700), Tempérance(701), Ascèse(702), Intempérance(703), Chasteté(704), Luxure(705), Sobriété(706), Gloutonnerie(707), Ivrognerie(708), Avarice(709), Prodigalité(710)

SECTION VI. LE DROIT
1. La justice
Justice(711), Injustice(712), Droit(713), Tribunal(714), Plaidoirie(715), Police(716)
2. Les délits et les peines
Vol(717), Escroquerie(718), Proxénétisme(719), Crime(720), Arrestation(721), Condamnation(722), Détention(723), Libération(724), Supplice(725)

SECTION VII. LA COMMUNICATION ET LE LANGAGE
1. Communication et dissimulation
Communication(726), Secret(727), Tromperie(728), Mensonge(729)
2. Le signe et le sens
Signe(730), Représentation(731), Sens(732), Non-sens(733), Intelligibilité(734), Inintelligibilité(735), Ambiguïté(736), Sous-entendu(737), Interprétation(738)
3. La langue
Langue(739), Grammaire(740), Phrase(741), Mot(742), Nom(743), Lettre(744)
4. La parole
Parole(745), Troubles de la parole(746), Cri(747), Interjections(748), Conversation(749), Plaisanterie(750)
5. Le discours
Discours(751), Figures de discours(752), Rhétorique(753), Récit(754), Description(755), Résumé(756)
6. Le style
Éloquence(757), Platitude(758), Concision(759), Prolixité(760), Grandiloquence(761)

SECTION VIII. LA COMMUNICATION ET L'INFORMATION
1. L'écrit et les médias
Écriture(762), Imprimerie(763), Imprimé(764), Livre(765), Presse(766), Radiotélévision(767), Publicité(768)
2. Circulation et traitement de l'information
Télécommunications(769), Correspondance(770), Enregistrement(771), Informatique(772)

SECTION IX. L'ART
1. Arts plastiques image et décor
Peinture et dessin(773), Iconographie(774), Photographie(775), Sculpture(776), Architecture(777), Ornements(778), Art des jardins(779), Tendances artistiques(780)
2. La musique et la chanson
Musique(781), Musiciens(782), Instruments de musique(783), Chant(784), Chanson(785)
3. Les arts du spectacle

Danse(786), Théâtre(787), Scène(788), Poésie(789), Cinéma(790), Cirque(791)

SECTION X. LES ACTIVITÉS ÉCONOMIQUES

1. Le travail et la production
   Emploi(792), Main-d oeuvre(793), Lieu de travail(794), Salaire(795), Production(796), Improduction(797)
2. L'industrie et l'artisanat
   Énergie(798), Outils(799), Machines(800), Manutention(801), Exploitation minière(802), Pétrole(803), Pétrochimie(804), Sidérurgie(805), Travaux publics(806), Menuiserie(807), Plomberie(808), Serrurerie(809), Textile(810)
3. L'agriculture et la pêche
   Agriculture(811), Arboriculture(812), Élevage(813), Pêche(814)
4. Les transports
   Transports(815), Transports par route(816), Automobile(817), Transports par rail(818), Transports maritimes et fluviaux(819), Transports par air(820), Astronautique(821)
5. Le commerce et les biens
   Possession(822), Cession(823), Restitution(824), Paiement(825), Don(826), Commerce(827), Marchandise(828)
6. L'économie
   Richesse(829), Pauvreté(830), Prix(831), Cherté(832), Modicité(833), Gratuité(834), Dépense(835), Dette(836), Libéralisme(837), Dirigisme(838)
7. La finance
   Monnaie(839), Banque(840), Crédit(841), Bourse(842), Valeurs mobilières(843), Épargne(844), Gestion(845), Fiscalité(846)

SECTION XI. LA VIE QUOTIDIENNE

1. L'habitat
   Habitat(847), Maison(848), Urbanisme(849), Mobilier(850), Vaisselle(851), Éclairage(852), Chauffage(853), Nettoyage(854)
2. L'alimentation
   Repas(855), Gastronomie(856), Pain(857), Sucrerie(858), Boisson(859), Produits laitiers(860), Fromage(861)
3. Le vêtement et la parure
   Vêtement(862), Mode(863), Couture(864), Chaussure(865), Bijou(866), Coiffure(867)
4. Les loisirs
   Passe-temps(868), Voyage(869), Sports(870), Chasse(871), Jeux(872), Jouet(873)

C

La hiérarchie Roget

CLASS I. WORDS EXPRESSING ABSTRACT RELATIONS

SECTION I. EXISTENCE

1. Being, in the Abstract
   Existence(1), Inexistence(2)
2. Being, in the Concrete
   Substantiality(3), Unsubstantiality(4)
3. Formal Existence
   Intrinsicality(5), Extrinsicality(6)
4. Modal Existence
   State(7), Circumstance(8)

SECTION II. RELATION

1. Absolute Relation
   Relation(9), Irrelation(10), Consanguinity(11), Correlation(12), Identity(13), Contrariety(14), Difference(15)
2. Continuous Relation
   Uniformity(16), Nonuniformity(16a)
3. Partial Relation
   Similarity(17), Dissimilarity(18), Imitation(19), Nonimitation(20), Variation(20a), Copy(21), Prototype(22)
4. General Relation
   Agreement(23), Disagreement(24)

SECTION III. QUANTITY

1. Simple Quantity
   Quantity(25), Degree(26)
2. Comparative Quantity
   Equality(27), Inequality(28), Mean(29), Compensation(30)

a. Quantity by comparison with a standard
   Greatness(31), Smallness(32)

b. Quantity by comparison with a simlar object
   Superiority(33), Inferiority(34)

c. Changes in quantity
   Increase(35), Nonincrease, Decrease(36)

3. Conjunctive Quantity
   Addition(37), Nonaddition(38), Adjunct(39), Remainder(40), Decrement(40a), Mixture(41), Simpleness(42), Junction(43), Disjunction(44), Connection(45), Coherence(46), Incoherence(47), Combination(48), Decomposition(49)
4. Concrete Quantity
   Whole(50), Part(51), Completeness(52), Incompleteness(53), Composition(54), Exclusion(55), Component(56), Extraneousness(57)

SECTION IV. ORDER

1. Order
   Order(58), Disorder(59), Complexity(59a), Arrangement(60), Derangement(61)

2. Consecutive Order
Precedence(62), Sequence(63), Precursor(64), Sequel(65), Beginning(66), End(67), Middle(68), Continuity(69), Discontinuity(70), Term(71)
3. Collective Order
Assemblage(72), Nonassemblage(73), Focus(74)
4. Distributive Order
Class(75), Inclusion(76), Exclusion(77), Generality(78), Speciality(79)
5. Order as Regards Categories
Normality(80), Multiformity(81), Conformity(82), Unconformity(83)

SECTION V. NUMBER
1. Number, in the Abstract
Number(84), Numeration(85), List(86)
2. Determinate Number
Unity(87), Accompaniment(88), Duality(89), Duplication(90), Bisection(91), Triality(92), Triplication(93), Trisection(94), Four(95), Quadruplication(96), Quadrisection(97), Five(98), Quinquesection(99), Plurality(100)
3. Indeterminate Number
Fraction(100a), Zero(101), Multitude(102), Fewness(103), Repetition(104), Infinity(105)

SECTION VI. TIME
1. Absolute Time
Time(106), Neverness(107), Period(108), Contingent Duration(108a), Course(109), Diuturnity(110), Transientness(111), Perpetuity(112), Instantaneity(113), Chronometry(114), Anachronism(115)
2. Relative Time
a. Time with reference to Succession
Priority(116), Posteriority(117), Present Time(118), Different time(119), Synchronism(120), Futurity(121), The Past(122), Newness(123), Oldness(124), Morning(125), Evening(126)
b. Time with reference to age ?
Youth(127), Age(128), Infant(129), Veteran(130), Adolescence(131)
c. Time with reference to an Effect or Purpose
Earliness(132), Punctuality(132a), Lateness(133), Occasion(134), Untimeliness(135)
3. Recurrent Time
Frequency(136), Infrequency(137), Regularity of recurrence(138), Irregularity of recurrence(139)

SECTION VII. CHANGE
1. Simple Change
Change(140), Permanence(141), Cessation(142), Continuance in action(143), Conversion(144), Reversion(145), Revolution(146), Substitution(147), Interchange(148)
2. Complex Change
Changeableness(149), Stability(150), Eventuality(151), Destiny(152)

SECTION VIII. CAUSATION
1. Constancy of Sequence in Events
Cause(153), Effect(154), Attribution(155), Chance(156)
2. Connection between Cause and Effect
Power(157), Impotence(158), Strength(159), Weakness(160)
3. Power in Operation
Production(161), Destruction(162), Reproduction(163), Producer(164), Destroyer(165), Paternity(166), Posterity(167), Productiveness(168), Unproductiveness(169), Agency(170), Physical Energy(171), Physical Inertness(172), Violence(173), Moderation(174)
4. Indirect Power
Influence(175), Absence of Influence(175a), Tendency(176), Liability(177)
5. Combinations of Causes
Concurrence(178), Counteraction(179)

CLASS II. WORDS RELATING TO SPACE
SECTION I. SPACE I N GENERAL
1. Abstract Space
Space(180), Inextension(180a), Region(181), Place(182)
2. Relative Space
Situation(183), Location(184), Displacement(185)

1. Vitality
a. Vitality in general
Organization(357), Inorganization(358), Life(359), Death(360), Killing(361), Corpse(362), Interment(363)
b. Special Vitality
Animality(364), Vegetability|(365), Animal(366), Vegetable(367), Zoology(368), Botany(369), Husbandry(370), Agriculture(371), Mankind(372), Man(373), Woman(374), Sexuality(374a)
2. Sensation
a. Sensation in general
Physical Sensibility(375), Physical Insensibility(376), Physical Pleasure(377), Physical Pain(378)
b. Special Sensation
(1) Touch
Touch(379), Sensations of Touch(380), Numbness(381)
(2) Heat
Heat(382), Cold(383), Calefaction(384), Refrigeration(385), Furnace(386), Refrigerator(387), Fuel(388), Insulation, Fire extinction(388a), Thermometer(389)
(3) Taste
Taste(390), Insipidity(391), Pungency(392), Saltiness(392a), Bitterness(392b), Condiment(393), Savoriness(394), Unsavoriness(395), Sweetness(396), Sourness(397)
(4) Odor
Odor(398), Inodorousness(399), Fragrance(400), Fetor(401), Acridity(401a)
(5) Sound
(i) Sound in general
Sound(402), Silence(403), Loudness(404), Faintness(405)
(ii) Specific sounds
Snap(406), Roll(407), Resonance(408), Nonresonance(408a), Hissing sounds(409), Stridor(410), Cry(411), Ululation(412)
(iii) Musical sounds
Melody(413), Discord(414), Music(415), Musician(416), Musical Instruments(417)
(iv) Perception of sound
Hearing(418), Deafness(419)
(6) Light
(i) Light in general
Light(420), Darkness(421), Dimness(422), Luminary(423), Shade(424), Transparency(425), Opacity(426), Turbidity(426a), Semitransparency(427)
(ii) Specific light
Color(428), Achromatism(429), Whiteness(430), Blackness(431), Gray(432), Brown(433)
Primitive Colors
Redness(434), Greenness(435), Yellowness(436), Purple(437), Blueness(438), Orange(439), Variegation(440)
(iii) Perceptions of light
Vision(441), Blindness(442), Dimsightedness(443), Spectator(444), Optical Instruments(445), Visibility(446), Invisibility(447), Appearance(448), Disappearance(449)

CLASS IV. WORDS RELATING TO THE INTELLECTUAL FACULTIES

DIVISION(I). FORMATION OF IDEAS SECTION I. OPERATIONS OF INTELLECT IN GENERAL
Intellect(450), Absence or want of Intellect(450a), Thought(451), Incogitancy(452), Idea(453), Topic(454)

SECTION II. PRECURSORY CONDITIONS AND OPERATIONS
Curiosity(455), Incuriosity(456), Attention(457), Inattention(458), Care(459), Neglect(460), Inquiry(461), Answer(462), Experiment(463), Comparison(464), Incomparability(464a), Discrimination(465), Indiscrimination(465a), Identification(465b), Measurement(466)

SECTION III. MATERIALS FOR REASONING
Evidence(467), Counter Evidence(468), Qualification(469)
Degrees of Evidence
Possibility(470), Impossibility(471), Probability(472), Improbability(473), Certainty(474), Uncertainty(475)

SECTION IV. REASONING PROCESSES
Reasoning(476), Intuition(477), Demonstration(478), Confutation(479)

SECTION V. RESULTS OF REASONING

Judgment(480), Discovery(480a), Misjudgment(481), Overestimation(482), Underestimation(483), Belief(484), Unbelief(485), Credulity(486), Incredulity(487), Assent(488), Dissent(489), Knowledge(490), Ignorance(491), Scholar(492), Ignoramus(493), Truth(494), Error(495), Maxim(496), Absurdity(497), Intelligence(498), Imbecility(499), Sage(500), Fool(501), Sanity(502), Insanity(503), Madman(504)

SECTION VI. EXTENSION OF THOUGHT

1. To the Past
   Memory(505), Oblivion(506)
2. To the Future
   Expectation(507), Inexpectation(508), Disappointment(509), Foresight(510), Prediction(511), Omen(512), Oracle(513)

SECTION VII. CREATIVE THOUGHT

Supposition(514), Analogy(514a), Imagination(515)

DIVISION(II). COMMUNICATION OF IDEAS

SECTION I. NATURE OF IDEAS COMMUNICATED.

Meaning(516), Unmeaningness(517), Intelligibility(518), Unintelligibility(519), Equivocalness(520), Metaphor(521), Interpretation(522), Misinterpretation(523), Interpreter(524)

SECTION II. MODES OF COMMUNICATION

Manifestation(525), Latency(526), Information(527), Correction(527a), Concealment(528), Disclosure(529), Ambush(530), Publication(531), News(532), Secret(533), Messenger(534), Affirmation(535), Negation(536), Teaching(537), Misteaching(538), Learning(539), Teacher(540), Learner(541), School(542), Veracity(543), Falsehood(544), Deception(545), Untruth(546), Dupe(547), Deceiver(548), Exaggeration(549)

SECTION III. MEANS OF COMMUNICATING IDEAS

1. Natural Means
   Indication(550), Record(551), Obliteration(552), Recorder(553), Representation(554), Misrepresentation(555), Painting(556), Sculpture(557), Engraving(558), Artist(559)
2. Conventional Means

a. Language generally
   Language(560), Letter(561), Word(562), Neologism(563), Nomenclature(564), Misnomer(565), Phrase(566), Grammar(567), Solecism(568), Style(569)

Various qualities of style
   Perspicuity(570), Obscurity(571), Conciseness(572), Diffuseness(573), Vigor(574), Feebleness(575), Plainness(576), Ornament(577), Elegance(578), Inelegance(579)

b. Spoken Language
   Voice(580), Aphony(581), Speech(582), Stammering(583), Loquacity(584), Taciturnity(585), Allocution(586), Response(587), Conversation(588), Soliloquy(589)

c. Written Language
   Writing(590), Printing(591), Correspondence(592), Book(593), Description(594), Dissertation(595), Compendium(596), Poetry(597), Prose(598), Drama(599)

CLASS V. WORDS RELATING TO THE VOLUNTARY POWERS(1)

DIVISION(I). INDIVIDUAL VOLITION

SECTION I. VOLITION IN GENERAL

1. Acts of Volition
   Will(600), Necessity(601), Willingness(602), Unwillingness(603), Resolution(604), Perseverance(604a), Irresolution(605), Obstinacy(606), Tergiversation(607), Caprice(608), Choice(609), Absence of Choice(609a), Rejection(610), Predetermination(611), Impulse(612), Habit(613), Desuetude(614)
2. Causes of Volition
   Motive(615), Absence of Motive(615a), Dissuasion(616), Pretext(617)
3. Objects of Volition
   Good(618), Evil(619)

SECTION II. Prospective Volition 1

1. Conceptional Volition
   Intention(620), Chance(621), Pursuit(622), Avoidance(623), Relinquishment(624), Business(625), Plan(626), Method(627), Mid-course(628), Circuit(629), Requirement(630)
2. Subservience to Ends

1. Actual Subservience
   Instrumentality(631), Means(632), Instrument(633), Substitute(634), Materials(635), Store(636), Provision(637), Waste(638), Sufficiency(639), Insufficiency(640), Redundancy(641), Importance(642), Unimportance(643), Utility(644), Inutility(645), Expedience(646), Inexpedience(647), Good qualities(648), Bad qualities(649), Perfection(650), Imperfection(651), Cleanness(652), Uncleanness(653), Health(654), Disease(655), Salubrity(656), Insalubrity(657), Improvement(658), Deterioration(659), Restoration(660), Relapse(661), Remedy(662), Bane(663)
3. Contingent Subservience
   Safety(664), Danger(665), Refuge(666), Pitfall(667), Warning(668), Alarm(669), Preservation(670), Escape(671), Deliverance(672)
4. Precursory Measures
   Preparation(673), Nonpreparation(674), Essay(675), Undertaking(676), Use(677), Disuse(678), Misuse(679)

SECTION III. VOLUNTARY ACTION
1. Simple Voluntary Action
   Action(680), Inaction(681), Activity(682), Inactivity(683), Haste(684), Leisure(685), Exertion(686), Repose(687), Fatigue(688), Refreshment(689), Agent(690), Workshop(691)
2. Complex Voluntary Action
   Conduct(692), Direction(693), Director(694), Advice(695), Council(696), Precept(697), Skill(698), Unskillfulness(699), Proficient(700), Bungler(701), Cunning(702), Artlessness(703)

SECTION IV. ANTAGONISM
1. Conditional Antagonism
   Difficulty(704), Facility(705), Hindrance(706), Aid(707), Opposition(708), Cooperation(709), Opponent(710), Auxiliary(711), Party(712), Discord(713), Concord(714), Defiance(715), Attack(716), Defense(717), Retaliation(718), Resistance(719), Contention(720), Peace(721), Warfare(722), Pacification(723), Mediation(724), Submission(725), Combatant(726), Arms(727), Arena(728)

SECTION V. RESULTS OF VOLUNTARY ACTION
Completion(729), Noncompletion(730), Success(731), Failure(732), Trophy(733), Prosperity(734), Adversity(735), Mediocrity(736)

DIVISION(II). INTERSOCIAL VOLITION
SECTION I. GENERAL INTERSOCIAL VOLITION
Implying the action of the will of one mind over the will of another.
Authority(737), Government(737a), Politics(737b), Laxity(738), Severity(739), Lenity(740), Command(741), Disobedience(742), Obedience(743), Compulsion(744), Master(745), Servant(746), Scepter(747), Freedom(748), Subjection(749), Liberation(750), Restraint(751), Prison(752), Keeper(753), Prisoner(754), Commission(755), Abrogation(756), Resignation(757), Consignee(758), Deputy(759)

SECTION II. SPECIAL INTERSOCIAL VOLITION
Permission(760), Prohibition(761), Consent(762), Offer(763), Refusal(764), Request(765), Deprecation(766), Petitioner(767)

SECTION III. CONDITIONAL INTERSOCIAL VOLITION
Promise(768), Release from engagement(768a), Compact(769), Conditions(770), Security(771), Observance(772), Nonobservance(773), Compromise(774)

SECTION IV. POSSESSIVE RELATIONS
That is, relations which concern property.
1. Property in general
   Acquisition(775), Loss(776), Possession(777), Exemption(777a), Participation(778), Possessor(779), Property(780), Retention(781), Relinquishment(782)
2. Transfer of Property
   Transfer(783), Giving(784), Receiving(785), Apportionment(786), Lending(787), Borrowing(788), Taking(789), Restitution(790), Stealing(791), Thief(792), Booty(793)
3. Interchange of Property
   Barter(794), Purchase(795), Sale(796), Merchant(797), Merchandise(798), Mart(799), Stock Market(799a), Securities(799b)
4. Monetary Relations

Money(800), Treasurer(801), Treasury(802), Wealth(803), Poverty(804), Credit(805), Debt(806), Payment(807), Nonpayment(808), Expenditure(809), Receipt(810), Accounts(811), Price(812), Value(812a), Worthlessness(812b), Discount(813), Dearness(814), Cheapness(815), Liberality(816), Economy(817), Greed(817a), Prodigality(818), Parsimony(819)

CLASS VI. WORDS RELATING TO THE SENTIMENT AND MORAL POWERS

SECTION I. AFFECTIONS IN GENERAL

Affections(820), Feeling(821), Sensibility(822), Insensibility(823), Excitation(824), Excitability(825), Inexcitability(826)

SECTION II. PERSONAL AFFECTIONS

1. Passive Affections
   Pleasure(827), Pain(828), Pleasurableness(829), Painfulness(830), Content(831), Discontent(832), Regret(833), Relief(834), Aggravation(835), Cheerfulness(836), Dejection(837), Rejoicing(838), Lamentation(839), Amusement(840), Weariness(841), Wit(842), Dullness(843), Humorist(844)
2. Discriminative Affections
   Beauty(845), Ugliness(846), Ornament(847), Jewelry(847a), Blemish(848), Simplicity(849), Taste(850), Vulgarity(851), Fashion(852), Ridiculousness(853), Fop(854), Affectation(855), Ridicule(856), Laughingstock(857)
3. Prospective Affections
   Hope(858), Hopelessness(859), Fear(860), Courage(861), Cowardice(862), Rashness(863), Caution(864), Desire(865), Indifference(866), Dislike(867), Fastidiousness(868), Satiety(869)
4. Contemplative Affections
   Wonder(870), Expectance(871), Prodigy(872)
5. Extrinsic Affections
   Repute(873), Disrepute(874), Nobility(875), Commonalty(876), Title(877), Pride(878), Humility(879), Vanity(880), Modesty(881), Ostentation(882), Celebration(883), Boasting(884), Insolence(885), Servility(886), Blusterer(887)

SECTION III. SYMPATHETIC AFFECTIONS

1. Social Affections
   Friendship(888), Enmity(889), Friend(890), Enemy(891), Sociality(892), Seclusion(893), Courtesy(894), Discourtesy(895), Congratulation(896), Love(897), Hate(898), Favorite(899), Resentment(900), Irascibility(901), Sullenness(901a), Endearment(902), Marriage(903), Celibacy(904), Divorce(905)
2. Diffusive Sympathetic Affections
   Benevolence(906), Malevolence(907), Malediction(908), Threat(909), Philanthropy(910), Misanthropy(911), Benefactor(912), Evil doer(913)
3. Special Sympathetic Affections
   Pity(914), Pitilessness(914a), Condolence(915)
4. Retrospective Sympathetic Affections
   Gratitude(916), Ingratitude(917), Forgiveness(918), Revenge(919), Jealousy(920), Envy(921)

SECTION IV. MORAL AFFECTIONS

1. Moral Obligations
   Right(922), Wrong(923), Dueness(924), Undueness(925), Duty(926), Dereliction of Duty(927), Exemption(927a), Respect(928), Disrespect(929), Contempt(930), Approbation(931), Disapprobation(932), Flattery(933), Detraction(934), Flatterer(935), Detractor(936), Vindication(937), Accusation(938)
3. Moral Conditions
   Probity(939), Improbity(940), Knave(941), Disinterestedness(942), Selfishness(943), Virtue(944), Vice(945), Innocence(946), Guilt(947), Good Man(948), Bad Man(949), Penitence(950), Impenitence(951), Atonement(952)
4. Moral Practice
   Temperance(953), Intemperance(954), Sensualist(954a), Asceticism(955), Fasting(956), Gluttony(957), Sobriety(958), Drunkenness(959), Purity(960), Impurity(961), Libertine(962)
5. Institutions
   Legality(963), Illegality(964), Jurisdiction(965), Tribunal(966), Judge(967), Lawyer(968), Lawsuit(969), Acquittal(970), Condemnation(971), Punishment(972), Reward(973), Penalty(974), Scourge(975)

SECTION V. RELIGIOUS AFFECTIONS

1. Superhuman Beings and Regions
   Deity(976), Angel(977), Satan(978), Jupiter(979), Demon(980), Heaven(981), Hell(982)
2. Religious Knowledge

Theology(983), Orthodoxy(983a), Heterodoxy(984), Judeo-Christian Revelation(985), Pseudo-Revelation(986)

3. Religious Sentiments
   Piety(987), Impiety(988), Irreligion(989)
4. Acts of Religion
   Worship(990), Idolatry(991), Sorcery(992), Spell(993), Sorcerer(994)

3. Religious Institutions
   Churchdom(995), Clergy(996), Laity(997), Rite(998), Canonicals(999), Temple(1000)

D

Corpus de phrases

D.1 Ambiguïté lexicale simple

Dans ces phrases, un seul chemin interprétatif est possible.

D.1.1 Ambiguïtés lexicales solubles à l'aide d'informations purement thématiques

1. « *L'avocat plaide à la cour.* »

   L'item ‹*avocat*› peut être le **fruit** ou l'**auxilière de justice** tandis que ‹*cour*› peut être soit l'**espace** entre batiments ou le **tribunal**. Des informations d'ordre thématique suffisent à préférer *avocat*/**avocat** et *cour*/**tribunal**.

2. « *Il connecte sa souris sur son ordinateur.* »

   Les acception de ‹*souris*› font partie du champs sémantique de l'**informatique** ou de celui de la **biologie**. Cependant, ‹*ordinateur*› n'appartient qu'au champs sémantique de l'**informatique** ce qui contraint le choix pour souris.

3. « *L'étudiant malais fait son devoir.* »

   L'item lexical ‹*devoir*› peut être soit **scolaire** soit **moral**. Le thème d'‹*étudiant*› permet de trancher la question.

D.1.2 Ambiguïtés lexicales solubles à l'aide d'informations thématiques et d'informations lexicales

4. « *L'avocat mange une glace.* »

   Avec uniquement des informations d'ordre thématique, les sens axés sur la nourriture dans ‹*avocat*› et ‹*glace*› ne pourraient qu'émerger alors que le sens d'‹*avocat*› est ici celui d'**auxilière de justice**.

5. « *Il faut creuser l'idée.* »

   Le sens métaphorique de ‹*creuser*› est utilisé avec ‹*idée*›.

6. « *Il eut alors l'idée de creuser un trou dans la paroi.* »

7. « *Il ouvre la porte.* »

8. « *La pelle de la rame heurta violemment les feuilles de papier.* »

   L'item ‹*rame*› est ici *rame*/**bateau** plutôt que *rame*/**papier**)

D.2 Ambiguïté lexicale multiple

Dans ces phrases, plusieurs chemins interprétatifs sont possibles.

9. « *La pelle se casse.* »
   Pour ‹*pelle*› *pelle*/**rame** et *pelle*/**outil** sont possibles tandis que pour ‹*casser*› seul *casser*/**morceaux** est possible.
10. « *L'avocat est véreux.* »
    On peut avoir *avocat*/**justice** associé à *véreux*/**crapuleux** ou *avocat*/**fruit** associé à *véreux*/**remplit de vers**. Dans les 2 cas, des informations thématiques ne seront pas suffisantes pour résoudre les ambiguïtés (cf 7.2.1.4).
11. « *Cet homme est très riche.* »
    Le terme ‹*riche*› peut être pris dans son sens propre ou son sens figuré.

D.3 Problème de référence

D.3.1 Résolution anaphorique

12. « *En ce moment, le second attira de nouveau l'attention du capitaine. Celui-ci suspendit sa promenade et dirigea sa lunette vers le point indiqué.* » ([Verne, 1870], p. 93)
    Le pronom ‹*celui-ci*› peut ici faire référence à ‹*second*› ou à ‹*capitaine*›. Ce cas nécessiterait d'autres informations que celles utilisées dans la thèse pour être résolues comme des scénario (savoir que le capitaine était en train de se promener).
    De plus, il est important de noter que dans les expériences réalisées dans cette thèse, nous n'avons pas considéré que « *le second* » pouvait être l'élision anaphorique d'un terme qui pourrait être utilisé avant dans le texte (« *Trois hommes étaient sur le pont (...) le second attira (...)* »).
13. « *L'homme marcha sur la queue du chien, il aboya.* »
    Le pronom personnel ‹*il*› fait-il référence à ‹‹chien›› ou à ‹‹homme›› ?
14. « *L'homme regarda la femme qui s'écroula sur sa chaise.* »
    Le pronom relatif ‹*qui*› fait référence à ‹‹femme›› et non à ‹‹homme››. En revanche les deux interprétations sont possibles pour ‹‹sa››.
15. « *La femme s'assit sur la chaise. Elle se cassa.* »
    Le pronom personnel ‹*elle*› fait-il référence à ‹‹femme›› ou à ‹‹chaise›› ?

D.3.2 Recherche des relations d'identité

Dans chaque phrase, les termes faisant référence à la même entité sont soulignés de la même manière.

16. « *Il cliqua sur la souris pour enregistrer son programme au moment où le chat, trop occupé à poursuivre le morceau de polystyrène poussé par le courant d'air et qu'il avait pris pour une souris, lui heurta le bras.* »
17. « *Le chat est monté sur la chaise. L'animal s'assoupit.* »
18. « *Il creuse avec la pelle, l'outil s'est cassé.* »
19. « *Il monte dans la tractopelle et fait démarrer l'engin.* »

D.4 Rattachement des groupes prépositionnels

Dans chaque phrase, le syntagme souligné d'un trait ondulé peut être rattaché à un des syntagmes soulignés par une droite.

20. « *Il regarde la fille avec un télescope.* »
21. « *Il regarde la fille dans le parc.* »
22. « *Il regarde la fille dans le parc avec un télescope.* »
23. « *Les variations soumettent les particules à des mouvements vibratoires.* »
24. « *L'énergie délivrée au tissu dépend de ces variations de pression qui soumettent les particules du milieu à des mouvements vibratoires.* » [117]
25. « *En -122 avant JC, les Romains soumettent les Allobroges.* » [118]

D.5 Instanciation des Fonctions Lexicales

26. « *Monsieur Smith a remporté une majorité écrasante.* »
    $Magn(\text{‹}majorité\text{›})=\text{‹}écrasante\text{›}$
27. « *Le malade a une forte fièvre.* »
    $Magn(\text{‹}fièvre\text{›})=\text{‹}forte\text{›}$
28. « *Me voilà avec cinquante francs d'appointements par mois, il faut que M. de Rênal ait eu une belle peur. Mais de quoi ?* » ([Stendhal, 1830], p. 76)
    $Magn(\text{‹}peur\text{›})=\text{‹}belle\text{›}$
29. « *Jean a eu une peur bleue.* »
    $Magn(\text{‹}peur\text{›})=\text{‹}bleue\text{›}$
30. « *Le mât du bateau s'est brisé.* »
    $Mero(\text{‹}bateau\text{›})=\text{‹}mât\text{›}$
31. « *La pelle de la rame s'enfonçait dans l'eau.* »
    $Mero(\text{‹}bateau\text{›})=\text{‹}mât\text{›}$ et $Rapport(\text{‹}rame\text{›})=\text{‹}eau\text{›}$
32. « *L'Abraham-Lincoln fut tenu sous petite vapeur, et s'avança prudemment pour ne pas éveiller son adversaire.* » ([Verne, 1870], p. 21)
33. « *La responsabilité incombe à la présidence.* »
34. « *Il est frappé d'interdiction bancaire.* »
35. « *Il a réalisé son rêve.* »
36. « *Il use d'arguments valables.* »
    $Ver(\text{‹}argument\text{›})=\text{‹}valable\text{›}$
37. « *Sa peur est tout à fait justifiée.* »
    $Ver(\text{‹}peur\text{›})=\text{‹}justifiée\text{›}$
38. « *L'homme lui donna un précieux conseil.* »
    $Bon(\text{‹}conseil\text{›})=\text{‹}précieux\text{›}$
39. « *Il se porte comme un charme.* »
    $Magn(\text{‹}se\ porter\text{›})=\text{‹}comme\ un\ charme\text{›}$

[117] http://www.imagemed.org/cerf/cnr/edicerf/BASES/BA003_cv_rb_3.html
[118] http://www.ujf-grenoble.fr/HOUCHES/histoire_et_structures/history-fr.html

40. « *Sa peine le fait souffrir atrocement.* »
    Magn(‹*souffrir*›)=‹*atrocement*›
41. « *D'énormes quartiers de roches nues étaient tombés jadis au milieu de la forêt du côté de la montagne.* »
    L'item lexical ‹*quartier*› peut prendre le sens de **morceau** ou correspondre à une **partie de ville**. L'ensembe des autres termes utilisés sont du domaine de la pierre ce qui contraint le sens de ‹*quartier*› à **morceau**.

E

Les fonctions lexicales standard d'Igor Mel'čuk

Voici la liste des fonctions lexicales d'Igor Mel'čuk [Mel'čuk, 1988], [Mel'čuk *et al.*, 1995], de [Polguère, 2003]. Les FL verbales n'ayant pas été réellement étudiés nous ne les faisons pas figurer ici.

Dans les fonctions, le ième indice indique que la valeur de la fonction est le ième argument. Ainsi, pour la *nominalisation*, on a « *Un* ‹voleur› *(S_1 agent) vole un* ‹butin› *(S_2 patient) à une* ‹victime› *(S_3).* ».

E.1 FL paradigmatiques

1. **Synonyme** [*Syn*].
   $Syn(\text{‹}destin\text{›}) = \text{‹}destinée\text{›}$; $Syn(\text{‹}déshydrater\text{›}) = \text{‹}sécher\text{›}$; $Syn(\text{‹}voiture\text{›}) = \text{‹}automobile\text{›}$
2. **Conversif** [*Conv*].
   $Conv(\text{‹}effrayer\text{›}) = \text{‹}craindre\text{›}$; $Conv(\text{‹}craindre\text{›}) = \text{‹}effrayer\text{›}$; $Conv(\text{‹}acheter\text{›}) = \text{‹}vendre\text{›}$
3. **Antonyme** [*Anti*].
   $Anti(\text{‹}respect\text{›}) = \text{‹}irrespect\text{›}$; $Anti(\text{‹}espoir\text{›}) = \text{‹}désespoir\text{›}$; $Anti(\text{‹}mépris\text{›}) = \text{‹}respect\text{›}$
4. **contrastif** [*Contr*].
   $Contr(\text{‹}d'acier\text{›}) = \text{‹}de\ velour\text{›}$; $Contr(\text{‹}mer\text{›}) = \text{‹}terre\text{›}$; $Contr(\text{‹}glace\text{›}) = \text{‹}feu\text{›}$
5. **Épithète pléonastique** [*Epit*]. Adjectif ou adverbe sans contribution sémantique dans le cadre d'un cliché.
   $Epit(\text{‹}océan\text{›}) = \text{‹}immense\text{›}$; $Epit(\text{‹}gagnant\text{›}) = \text{‹}heureux\text{›}$; $Epit(\text{‹}défier\text{›}) = \text{‹}ouvertement\text{›}$
6. **Générique** [*Gener*].
   $Gener(\text{‹}gaz\text{›}) = \text{‹}substance\ [gazeuse]\text{›}$; $Gener(\text{‹}pistolet\text{›}) = \text{‹}arme\ à\ feu\text{›}$; $Gener(\text{‹}armoire\text{›}) = \text{‹}meuble\text{›}$
7. **figuratif** [*Figur*].
   $Figur(\text{‹}fumée\text{›}) = \text{‹}rideau\ [de\ \tilde{}]\text{›}$; $Figur(\text{‹}haine\text{›}) = \text{‹}feu\ [de\ la\ \tilde{}]\text{›}$; $Figur(\text{‹}jalousie\text{›}) = \text{‹}démon\ [de\ la\ \tilde{}]\text{›}$

Dérivés syntaxiques

8. **Nominalisation** [S_i].
   $S_0(\text{‹}voler\text{›}) = \text{‹}vol\text{›}$; $S_1(\text{‹}voler\text{›}) = \text{‹}voleur\text{›}, \text{‹}coupable\text{›}$; $S_2(\text{‹}voler\text{›}) = \text{‹}butin\text{›}$; $S_3(\text{‹}voler\text{›}) = \text{‹}victime\text{›}$
9. **Verbalisation** [V_0].
   $V_0(\text{‹}arrachage\text{›}) = \text{‹}arracher\text{›}$; $V_0(\text{‹}erreur\text{›}) = \text{‹}se\ tromper\text{›}$; $V_0(\text{‹}serment\text{›}) = \text{‹}jurer\text{›}$

10. **Adjectivisation** [A_0].
A_0(‹*correction*›) = ‹*correct*›; A_0(‹*corriger*›) = ‹*correct*›; A_0(‹*erreur*›) = ‹*erroné*›
11. **Adverbialisation** [Adv_0].
Adv_0(‹*correction*›) = ‹*correctement*›; Adv_0(‹*correct*›) = ‹*correctement*›; Adv_0(‹*corriger*›) = ‹*correctement*›

E.1.1 FL nominales

12. **Dérivés sémantiques nominaux actanciels** [S_i] avec $i \in \{1,2,3\}$.
S_1(‹*parler*›) = ‹*parloir*› ; S_2(‹*parler*›) = ‹*paroles*›, ‹*propos*›, ‹*discours*› ; S_3(‹*parler*›) = ‹*allocutaire*›, ‹*destinataire*›
13. **Dérivés sémantiques nominaux circonstanciels** [S_n] avec $n \in \{instr, loc, med, mod, res\}$.
nom d'instrument [S_{instr}], nom de lieu [S_{loc}], nom de moyen [S_{med}], nom de manière [S_{mod}] et nom de résultat [S_{res}]
S_{instr}(‹*parler*›) = ‹*langue*› ; S_{loc}(‹*parler*›) = ‹*parloir*› ; S_{med}(‹*parler*›) = ‹*façon de˜*› ; S_{res}(‹*laver*›) = ‹*lessive*›
14. **Singulatif** [*Sing*]. Fonction équivalente à « *Unité minimale régulière de...* », fonction inverse de *Mult*.
Sing(‹*flotte*›) = ‹*navire*›; *Sing*(‹*vol*›) = ‹*oiseau*›; *Sing*(‹*riz*›) = ‹*grain*›
15. **Collectif** [*Mult*]. Fonction équivalente à « *Ensemble régulier de...* », fonction inverse de *Sing*.
Mult(‹*navire*›) = ‹*flotte*›; *Mult*(‹*chien*›) = ‹*meute*›; *Mult*(‹*abeille*›) = ‹*essaim*›
16. **Nom du chef** [*Cap*].
Cap(‹*université*›) = ‹*président*›; *Cap*(‹*faculté*›) = ‹*doyen*›; *Cap*(‹*bateau*›) = ‹*capitaine*›
17. **Nom de l'équipe** [*Equip*].
Equip(‹*théâtre*›) = ‹*troupe*›; *Equip*(‹*bateau*›) = ‹*équipage*›; *Equip*(‹*football*›) = ‹*équipe*›
18. **Nom de démarrage** [*Germ*].
Germ(‹*colère*›) = ‹*ferment*›; *Germ*(‹*colère*›) = ‹*raisins*›; *Germ*(‹*match*›) = ‹*coup d'envoi*›
19. **Nom du centre** [*Centr*].
Centr(‹*problème*›) = ‹*cœur*›; *Centr*(‹*Terre*›) = ‹*centre*›; *Centr*(‹*atome*›) = ‹*noyau*›
20. **Nom du point culminant** [*Culm*].
Culm(‹*joie*›) = ‹*comble*›; *Culm*(‹*colère*›) = ‹*paroxisme*›; *Culm*(‹*savoir*›) = ‹*apex*›

E.1.2 FL adjectivales

21. **Dérivé sémantique adjectival actanciel** [A_i] avec $i \in \{1,2,3\}$.
A_1(‹*mépris*›) = ‹*remplit*› ; A_2(‹*mépris*›) = ‹*couvert*› ; A_2(‹*direction*›) = ‹*sous la direction de*›
22. **Dérivé sémantique adjectival potentiel** [$Able_i$] avec $i \in \{1,2,3\}$.
$Able_1$(‹*peur*›) = ‹*peureux*› ; $Able_2$(‹*peur*›) = ‹*effrayant*› ; $Able_2$(‹*lire*›) = ‹*lisible*› ; $Able_2$(‹*brûler*›) = ‹*combustible*›
23. **Dérivé sémantique adjectival virtuel** [$Qual_i$] avec $i \in \{1,2\}$.
$Qual_1$(‹*tromper*›) = ‹*malhonnête*› ; $Qual_2$(‹*tromper*›) = ‹*naïf*› ;

E.2 FL syntagmatiques

E.2.1 FL adjectivales

24. **Intensificateur** [*Magn*].
Magn(‹*amour*›) = ‹*fou*›; *Magn*(‹*peur*›) = ‹*bleu*›; *Magn*(‹*fièvre*›) = ‹*de cheval*›

25. **Comparatifs**[*Plus/Minus*].
Expriment le degrés de comparaison ; ne s'utilise qu'avec d'autres fonctions, produisant le plus souvent un verbe signifiant plus/moins *Magn*.
IncepPredPlus(‹*fièvre*›) = ‹*augmente*› ; *IncepPredPlus*(‹*ouragan*›) = ‹*se déchaîne*›
IncepPredMinus(‹*fièvre*›) = ‹*baisse*›,‹*diminue*› ; *IncepPredMinus*(‹*ouragan*›) = ‹*se calme*›
26. **Confirmateur** [*Ver*].
Ver(‹*argument*›) = ‹*valable*›; *Ver*(‹*succès*›) = ‹*mérité*›; *Ver*(‹*peur*›) = ‹*justifié*›
27. **Laudatif** [*Bon*].
Bon(‹*conseil*›) = ‹*précieux*›; *Bon*(‹*choix*›) = ‹*heureux*›; *Bon*(‹*se porter*›) = ‹*comme un charme*›
28. **Péjoratif**[*Pejor*].
exprime le sens 'pire' et s'utilise surtout dans les FL complexes.
IncepPredPejor(‹*santé*›) = ‹*détériore*› ; *IncepPredPejor*(‹*situation*›) = ‹*s'aggrave*›
29. **Positif** [Pos_2].
Pos_2(‹*opinion*›) = ‹*favorable*›; Pos_2(‹*critique*›) = ‹*élogieuse*›; Pos_2(‹*compte rendu*›) = ‹*favorable*›

E.2.2 FL adverbiales

30. **Dérivés sémantiques adverbiaux actanciels** [Adv_i] avec $i \in \{1,2,3\}$.
Adv_1(‹*mépris*›) = ‹*avec*› ; Adv_1(‹*joie*›) = ‹*avec*› ; $Adv_2(feu/\mathbf{tir})$ = ‹*sous*›
31. **Instrumental** [*Instr*].
Instr(‹*main*›) = ‹*à*›; *Instr*(‹*argument*›) = ‹*à l'aide*›; *Instr*(‹*téléphone*›) = ‹*par*›
32. **Locatif** [Loc^{β}_{α}] avec $\alpha \in \{in, ab, ad\}$ et $\beta \in \{lieu, temp\}$.
$Loc^{lieu}_{in/ad}$(‹*gare*›) = ‹*à la*› ; $Loc^{lieu}_{in/ad}$(‹*ville*›) = ‹*en*›, Loc_{in}(‹*personnel*›) = ‹*au sein du* ›, Loc^{temp}_{in}(‹*antiquité*›) = ‹*dans*›
33. **Consécutif** [*Propt*].
Propt(‹*jalousie*›) = ‹*par*›; *Propt*(‹*maladie*›) = ‹*pour cause de*›; *Propt*(‹*alcool*›) = ‹*sous l'emprise de*›
34. **Locatif** [Loc^{β}_{α}] avec $\alpha \in \{in, ab, ad\}$ et $\beta \in \{lieu, temp\}$.
$Loc^{lieu}_{in/ad}$(‹*gare*›) = ‹*à la*› ; $Loc^{lieu}_{in/ad}$(‹*ville*›) = ‹*en*›, Loc_{in}(‹*personnel*›) = ‹*au sein du* ›, Loc^{temp}_{in}(‹*antiquité*›) = ‹*dans*›

F

Les relations dans UNL

Cette annexe reprend les spécifications d'UNL dont les dernières spécifications sont disponibles à l'adresse http://www.undl.org/unlsys/unl/unl2005/.

UNL 2005 Specifications
7 June 2005

Relations
There are many factors to be considered in choosing an inventory of relations between concepts. Different factors taken into account in choosing the relations lead to different sets of the relations. The UNL relations are selected according to the following principles.

Principles of Relation
PRINCIPLE 1 : NECESSARY CONDITION
When an UW has relations between more than one other UWs, each relation label should be set so as to be able to identify each relation on the premise that there is enough knowledge about the concept of each UW expressed.
PRINCIPLE 2 : SUFFICIENT CONDITION
When there are relations between UWs, each relation label should be set so as to be able to understand the role of each UW only by referring to the relation label.

Definitions of Relations
The following are the relations defined according to the above principles. A relation label is represented as strings of 3 characters or less.

agt	and	aoj	bas	ben	cag	cao	cnt	cob	con	coo	dur	equ	fmt	frm
gol	icl	ins	int	iof	man	met	mod	nam	obj	opl	or	per	plc	plf
plt	pof	pos	ptn	pur	qua	rsn	scn	seq	src	tim	tmf	tmt	to	via

1. **agt** (agent)
   indicates a thing in focus that initiates an action
   agt (do, thing)
   agt (action(icl>event), thing)

   Detailed Definition
   An agent is defined as the relation between :
   UW1 - do, and
   UW2 - a thing
   where :
   · UW2 initiates UW1, or
   · UW2 is thought of as having a direct role in making UW1 happen.

   Examples and Readings
   agt (break(agt>thing,obj>thing), John(iof>person)) John breaks ...
   agt (translate(agt>thing,gol>language,obj>information,src>language), computer(icl>machine))

computer translates ...
agt (run(icl>act(agt>volitional thing)), car(icl>vehicle)) car runs ...
agt (destroy(agt>thing,obj>thing), explosion(icl>event)) explosion destroys ...

Related relations
· An agent is different from **cag** in that an agent initiates the action, whereas a co-agent initiates a different, accompanied action.
· An agent is different from **ptn** in that an agent is the focused initiator of the action, whereas a partner is a non-focused initiator.
· An agent is different from aoj in that an agent initiates an action, whereas aoj indicates a thing that is in a state. A state is expressed by a UW that belongs to 'be'.

2. **and** conjunction
indicates a partner to have conjunctive relation to
and (uw, uw)

Detailed Definition
A conjunction is defined as the relation between :
UW1 - a concept, and
UW2 - another concept,
where :
· The UWs are different, and
· UW1 and UW2 are seen as grouped together, and
· what is said of UW1 is also said of UW2.

Examples and Readings
and (quickly, easily) ... easily and quickly
and (dance(agt>person), sing(agt>person)) ... singing and dancing
and (Mary(iof>person), John(iof>person)) ... John and Mary
Related Relations · A conjunction is different from **or** in that with **and** things are grouped together to say the same thing about both of them, whereas with **or** we separate them to indicate that what is true about one is not true about the other.
· A conjunction is different from **cag** in that when the agents are conjoined, both initiate an explicit event, whereas with **cag** , the co-agent initiates an implicit event.
· A conjunction is different from **ptn** in that when the agents and partners are conjoined, both are in focus, whereas with **ptn** , the partner is not in focus (as compared to the agent).
· A conjunction is different from **coo** and **seq** in meaning, although in many cases the same expressions can be used for both. A conjunction only means that terms are grouped together ; no information about time is implied. **Coo** , on the other hand, means that the terms are in the same time, whether they are considered to be grouped together or not. In turn, **seq** means that the terms are ordered in time, one after the other
· A conjunction is different from **int** and **or** in that as a logical operation and **or** makes differences, **int** makes an intersection, whereasÂ **and** makes a union .

3. **aoj** thing with attribute
indicates a thing that is in s state or has an attribute
aoj (be, thing)
aoj (thing, thing)
aoj (uw(aoj>thing), thing)

Detailed Definition
A thing with an attribute or in a state is defined as the relation between :
UW1 - an attribute or a state or a thing which represents a state, and
UW2 - a thing,
where :
· UW1 is an attribute or state of UW2, or
· UW1 is a state associated with UW2.

Examples and Readings aoj (red(aoj>thing), leaf(pof>plant)) ... leaf is red.
aoj (available(aoj>thing,obj<thing), information) This information is available for ...
aoj (nice, ski(agt>person)) Skiiing is nice.
aoj (teacher(icl>occupation), John(iof>person)) John is a teacher.

aoj (have(aoj>thing,obj>thing), I) I have a pen.
aoj (know(aoj>thing,obj>thing), John(iof>person)) John knows ...

Related Relations · A thing with an attribute is different from **mod** in that **mod** gives some restriction of the concept in focus, whereas **aoj** indicates a thing of a state or characteristic.
· A thing with an attribute is different from **ben** in that a beneficiary is quite independent from a focused event or state. This event or state can be considered as exerting a good or bad influence on the beneficiary, whereas **aoj** indicates a thing that has a direct relation with the event or state, the event or state can be considered as describing a state or characteristic about the thing.
· A thing with an attribute is different from **obj** in that **obj** indicates a thing which is directly affected by an action or phenomenon, whereas, **aoj** indicates a thing in a state.

4. **bas** basis
indicates a thing used as the basis (standard) of comparison
bas (be(aoj>volitional thing,bas>thing,obj>thing), thing)
bas (do(agt>thing,bas>thing,obj>thing), thing)
bas (how(bas>thing), thing)
bas (uw(aoj>thing,bas>thing), thing)

Detailed Definition
A basis is defined as the relation between :
UW1 - a concept expressing comparison, and
UW2 - a thing,
where :
· UW1 is a concept expressing comparison, and
· UW2 is something used as the basis for evaluating the characteristic or quantity of some other (focused) thing.

Examples and readings bas (more(aoj>thing,bas>thing), 7) Ten is three more than seven.
bas (more(icl>how,bas>thing), Jack(iof>person))Betty weighs more than Jack (does).
man (beautiful, more(icl>how,bas>thing)) A tulip is more beautiful than a rose
bas (more(icl>how,bas>thing), rose(icl>flower))
aoj (:01, John(iof>person)) John is more quiet than shy.
man :01 (quiet(aoj>thing), more(icl>how,bas>thing))
bas :01 (more(icl>how,bas>thing), shy(aoj>thing))
bas (prefer(aoj>volitional thing,bas>uw,obj>uw), live(agt>person) :02)
plc (live(agt>person) :02, city(icl>region)) Many people prefer living in the country to living in a city

5. **ben** beneficiary
indicates an indirectly related beneficiary or victim of an event or state
ben (be, thing)
ben (do, thing)
ben (occur, thing)
ben (uw(aoj>thing), thing)

Detailed Definition
A beneficiary is defined as the relation between :
UW1 - an event or state, and
UW2 - a thing,
where :
· UW2 is thought of as being indirectly affected by UW1, as the beneficiary or victim.

Examples and Readings
ben (give(agt>thing,gol>thing,obj>thing), country(icl>region)) To give one's life for one's country.
ben (good(aoj>thing), John(iof>person)) It is good for John to ...

Related Relations · A beneficiary is different from **aoj** in that **aoj** has a direct relation with the focused state or event and the focused state or event can be considered as describing the thing of **aoj**; Whereas a beneficiary is quite independent from a focused event or state, but this event or state can be considered as exerting a good or bad influence on the beneficiary.

6. **cag** co-agent
   indicates a thing not in focus that initiates an implicit event that is done in parallel
   cag (do, thing)
   cag (action(icl>event), thing)

   Detailed Definition
   A co-agent is defined as the relation between :
   UW1 - an action, and
   UW2 - a thing
   where :
   · There is an implicit action that is independent of, but accompanies, UW1, and
   · UW2 is thought of as initiating the implicit action, and
   · UW2 and the implicit action are seen as not being in focus (as compared to the agent's action.

   Examples and Readings
   cag (walk(agt>volitional thing), John(iof>person))To walk with John
   cag (live(agt>volitional thing), aunt(icl>person)) To live with ... aunt

   Related Relations
   · A co-agent is different from **agt** in that differing independent actions occur for an agent and a co-agent. Moreover, an agent and its action are in focus, while a co-agent and its action are not in focus.
   · A co-agent is different from the **ptn** in that the co-agent initiates an action that is independent of an agent's action, whereas a partner initiates the same action together with an agent.
7. **cao** co-thing with attribute
   indicates a thing not in focus that is in a parallel state

   cao (be, thing)
   cao (thing, thing)
   cao (uw(aoj>thing), thing)

   Detailed Definition
   A co-thing with an attribute is defined as the relation between :
   UW1 - a state or a thing which represents a state, and
   UW2 - a thing,
   where :
   · There is an implicit state that is independent of, but accompanies, UW1, and
   · UW2 is associated with the implicit state.

   Examples and readings
   cao (exist(aoj>thing), you) be with you

   Related Relations
   · A co-thing with an attribute is different from **aoj** in that there is a different, independent state for the thing with an attribute and a co-thing with an attribute, respectively.
8. **cnt** content
   indicates the content of a concept

   cao (uw, uw)

   Detailed Definition
   A content is defined as the relation between :
   UW1 - a concept, and
   UW2 - a concept,
   where :
   · UW2 is the content or explanation of UW1.

   Examples and Readings
   cnt (Internet(icl>communication network),amalgamation(icl>harmony)) The Internet : an amalgamation

cnt (language generator, deconverter.@double_quote) a language generator "deconverter"...
cnt (risk(icl>danger), :01)
obj :01 (lose(aoj>thing,obj>thing)@entry, money(icl>mark)) the risk of losing money

9. **cob** affected co-thing
indicates a thing that is directly affected by an implicit event done in parallel or an implicit state in parallel

cob (be, thing)
cob (do, thing)
cob (occur, thing)
cob (event(icl>abstract thing), thing)
cob (uw(aoj>thing,obj>thing), thing)

Detailed Definition
A "co-object" is defined as the relation between :
UW1 - an event or state, and
UW2 - a thing,
where :
· UW2 is thought of as directly affected by an implicit event done in parallel or an implicit state in parallel.

Examples and Readings
cob (die(obj>living thing), Mary(iof>person))
obj (injure(icl>hurt(agt>thing,obj>living thing)), John(iof>person))
cob (injure(icl>hurt(agt>thing,obj>living thing)), friend(icl>comrade).@pl)
pos (friend(icl>comrade).@pl, he)) ... dead with Mary John was injured in the accident with his friends

Related Relations
· A co-object is different from **obj** in that the **obj** is in focus, whereas **cob** is related to a second, non-focused implicit event or state.

10. **con** condition
indicates a non-focused event or state that conditions a focused event or state

con (be, uw)
con (do, uw)
con (occur, uw)
con (uw(aoj>thing), uw)

'*Detailed Definition*
A condition is defined as the relation between :
UW1 - an event or state, and
UW2 - an event or state,
where :
· UW1 is a focused event or state, whereas
· UW2 is a conditioning event or state, and
· UW2 is thought of as having an indirect or external role in making UW1 happen, that is, as some conditioning or inhibiting factor (real or hypothesized) that influences whether or when UW1 can happen.

Examples and Readings
aoj :01 (tired(aoj>thing), you)
con (go(icl>move(agt>thing,gol>place,src>place)), :01) If you are tired, we will go straight home

11. **coo** effected co-thing
indicates a co-occurrent event or state for a focused event or state
coo (be, be)
coo (be, do)
coo (be, occur)
coo (be, thing)
coo (be, uw(aoj>thing))

coo (do, be)
coo (do, do)
coo (do, occur)
coo (do, thing)
coo (do, uw(aoj>thing))
coo (occur, be)
coo (occur, do)
coo (occur, occur)
coo (occur, thing)
coo (occur, uw(aoj>thing))
coo (thing, be)
coo (thing, do)
coo (thing, occur)
coo (thing, thing)
coo (thing, uw(aoj>thing))
coo (uw(aoj>thing), be)
coo (uw(aoj>thing), do)
coo (uw(aoj>thing), occur)
coo (uw(aoj>thing), thing)
coo (uw(aoj>thing), uw(aoj>thing))

Detailed Definition
A co-occurrence is defined as the relation between :
UW1 - an event or state, and
UW2 - an event or state,
where :
· UW1 is a focused event or state, whereas
· UW2 is a co-occurrent event or state, and
· UW1 occurs or is true at the same time as UW2.

Examples and Readings
coo (cry(icl>weep(agt>volitional thing)), run(icl>act(agt>volitional thing))) ... was crying while running
coo (red(aoj>thing), hot(aoj>thing)) ... is red while ... is hot

Related Relations
· A co-occurrence is different from **seq** in that **seq** describes events or states that do not occur at the same time, but one after the other, whereas **coo** describes events that occur simultaneously.
· A co-occurrence is different from **tim** in that **coo** relates the times of events or states with other events or states, whereas **tim** relates events or states directly with points or intervals of time.

12. **dur** duration
indicates a period of time during which an event occurs or a state exists
dur (be, do)
dur (be, event(icl>abstract thing))
dur (be, occur)
dur (be, period(icl>time))
dur (be, state(icl>abstract thing))
dur (be, thing)
dur (be, uw(aoj>thing))
dur (do, do)
dur (do, event(icl>abstract thing))
dur (do, occur)
dur (do, period(icl>time))
dur (do, state(icl>abstract thing))
dur (do, thing)
dur (do, uw(aoj>thing))
dur (occur, do)
dur (occur, event(icl>abstract thing))
dur (occur, occur)
dur (occur, period(icl>time))
dur (occur, state(icl>abstract thing))

dur (occur, thing)
dur (occur, uw(aoj>thing))
dur (thing, do)
dur (thing, event(icl>abstract thing))
dur (thing, occur)
dur (thing, period(icl>time))
dur (thing, state(icl>abstract thing))
dur (thing, thing)
dur (thing, uw(aoj>thing))
dur (uw(aoj>thing), do)
dur (uw(aoj>thing), event(icl>abstract thing))
dur (uw(aoj>thing), occur)
dur (uw(aoj>thing), period(icl>time))
dur (uw(aoj>thing), state(icl>abstract thing))
dur (uw(aoj>thing), thing)
dur (uw(aoj>thing), uw(aoj>thing))

Detailed Definition
A duration is defined as the relation between :
UW1 - an event or a state, and
UW2 - a period during which the event or state continues

Examples and Readings
dur (work(agt>person), hour(icl>period)) ... work nine hours (a day)
qua (hour(icl>period), 9)
dur (talk(icl>express(agt>thing,gol>person,obj>thing), meeting(icl>event)) ... talk ... during meeting
dur (come(icl>move(agt>thing,gol>place,src>place), absence(icl>state)) ... come during (my) absence

13. **equ** effected co-thing
indicates an equivalent concept
equ (uw, uw)

Detailed Definition
An equivalent concept is defined as the relation between :
UW1 - a concept, and
UW2 - a concept,
where :
· The UWs are different, and
· UW2 is an equivalent concept of UW1.

Examples and Readings
equ (deconverter, language generator.@parenthesis) the deconverter (a language generator)

14. **fmt** range/from-to
indicates a range between two things

fmt (thing, thing)

Detailed Definition
A range (from-to) is defined as the relation between :
UW1 - a range-initial thing, and
UW2 - a range-final thing,
where :
· The UWs are different, and
· UW2 describes the beginning of a range and UW1 describes the end.

Examples and Readings

fmt (z(icl>letter), a(icl>letter)) the alphabets from a to z
fmt (New York(iof>city), Osaka(iof>city)) the distance from Osaka to New York

fmt (Friday(icl>day), Monday(icl>day)) the weekdays from Monday to Friday

Related Relations
· A range is different from **src** and **gol** in that for **src** and **gol** the initial and final states of certain **obj** are characterized with respect to some event, whereas **fmt** makes a similar characterization but without linking the endpoints of a range to some event.
· A range is different from **plf** and **plt** or **tmf** and **tmt** in that **fmt** defines endpoints of a range without reference to any sort of event, whereas **plf** , **plt** , **tmf** and **tmt** delimit events.

15. **frm** origin
indicates an initial state of a thing or a thing initially associated with the focused thing

frm (thing, thing)
frm (thing, uw(aoj>thing))

Detailed Definition
An origin is defined as the relation between :
UW1 - a thing, and
UW2 - a state or a thing than can be seen as origin of the thing,
where :
· The UWs are different, and
· UW1 is the focused thing, and
· UW2 is the initial state describing the focused thing UW1, or
· UW2 is a thing that is initially associated with the UW1, origin such as the original position of UW1.

Examples and Readings
frm (visitor(icl>person), Japan(iof>country) a visitor from Japan

Related Relations
· An origin is different from **src** in that **src** is a relation used with an event or a state, whereas **frm** is directly linked to a thing. For instance, "a visitor from Japan" is expressed as "**frm** (visitor(icl>person), Japan(iof>country))", whereas "a visitor came from Japan" is expressed as "**src** (come(agt>thing), Japan(iof>country))" and "agt (come(agt>thing), visitor(icl>person))".

16. **gol** goal/final state
indicates a final state of object or a thing finally associated with the object of an event

gol (be(aoj>thing,gol>thing), thing)
gol (do, thing)
gol (do, uw(aoj>thing))
gol (occur, thing)
gol (occur, uw(aoj>thing))
gol (event(icl>abstract thing), thing)

Detailed Definition
A final state is defined as the relation between :
UW1 - an event, and
UW2 - a state or thing,
where :
· UW2 is the specific state describing the **obj** (of UW1) at the end of UW1, or
· UW2 is a thing that is associated with the **obj** (of UW1) and the end of UW1.

Examples and Readings
gol (change(gol>thing,obj>thing,src>thing), red(aoj>thing)) the lights changed from green to red
gol (deposit(agt>thing,gol>thing,obj>thing), account(icl>record)) millions were deposited in a Swiss bank account

Related Relations
A final state is different from **tmf** and **plf** in that **gol** describes qualitative characteristics and not time nor place related to an event.

17. **icl** included/a kind of
indicates an upper concept or a more general concept

icl (uw, uw)
Detailed Definition

An upper concept or a more general concept is defined as the relation between :
UW1 - a class concept, and
UW2 - a class concept,
where :
· The UWs are different, and
· UW2 is an upper or more general class concept of UW1, i.e.
· UW1 is a subset concept of UW2, and UW1 inherits UW2's property.
Examples and Readings

icl (bird(icl>animal), animal(icl>living thing))
a bird is a (kind of) animal

18. **ins** instrument
indicates an instrument to carry out an event

ins (do, concrete thing)
Detailed Definition

An instrument is defined as the relation between :
UW1 - an event, and
UW2 - a concrete thing,
where :
· UW2 specifies the concrete thing that is used in order to make UW1 happen.
Examples and Readings

ins (look(agt>thing,obj>thing), telescope(icl>optical instrument))
ins (write(agt>thing,obj>thing), pencil(icl>stationery))
ins (cut(agt>thing,obj>thing,opl>thing), scissors(icl>cutley))
look at stars through a telescope
write [draw] with a pencil
He cut the string with a pair of scissors
Related Relations
· An instrument is different from **man** in that **man** describes an event as a whole, whereas **ins** characterizes one of the components of the event : the use of the instrument. And, a manner is an abstract thing whereas an instrument is a concrete thing.
· An instrument is different from **met** in that **met** is used for abstract things (abstract means or methods), whereas **ins** is used for concrete things.

19. **int** intersection
indicates all common instances to have with a partner concept

int (uw, uw)
Detailed Definition

An intersection is taken between :
UW1 - a class concept, and
UW2 - another class concept,
where :
· The UWs are different, and
· UW1 and UW2 have common instances.
Examples and Readings

int (tableware(icl>tool), cookware(icl>tool)))
an intersection of tableware and cookware
Related Relations

· An intersection is different from **and** and **or** in that as a logical operation **and** makes a union and **or** makes differences, whereas **int** makes an intersection.

20. **iof** an instance of
indicates a class concept that an instance belongs to

iof (uw, uw)
Detailed Definition

A class concept is defined as the relation between :
UW1 - an instance, and
UW2 - a class concept,
where :
· The UWs are different, and
· UW2 is a class concept that UW1 belongs to, i.e.
· UW1 is an instance of UW2, and UW1 inherits UW2's property.
Examples and Readings

iof (Tokyo(iof>city), city in Japan)
Tokyo is a city in Japan

21. **man** manner
indicates a way to carry out an event or the characteristics of a state

man (be, how)
man (do, how)
man (occur, how)
man (uw(aoj>thing), how)
Detailed Definition

A "manner" is defined as the relation between :
UW1 - an event or state, and
UW2 - a manner,
where :
· UW1 is done or exists in a way characterized by UW2.
Examples and Readings

man (move(agt<thing,gol>place,src>place), quickly)
man (visit(agt>thing,obj>thing), often)
man (beautiful, very(icl>how))
move quickly
I often visit him.
it is very beautiful.
Related Relations

· A manner is different from **ins** or **met** in that **ins** describes how an event is carried out in terms of the instruments, **met** describes how an event is carried out in terms of the component steps of the event, whereas **man** describes other quantitative or qualitative characteristics of the event as a whole.

22. **met** method/means
indicates a means to carry out an event

met (do, abstract thing)
met (do, do)
Detailed Definition

A "method or means" is defined as the relation between :
UW1 - an action, and
UW2 - an abstract thing or an action,
where :
· UW2 specifies the abstract thing used or the steps carried out in order to make UW1 happen.

Examples and Readings

met (solve(icl>resolve(agt>thing,obj>thing)), dynamics(icl>science))
met (solve(icl> resolve(agt>thing,obj>thing)), algorithm(icl>method))
met (separate(agt>thing,obj>thing,src>thing)), cut(agt>thing,obj>thing,opl>thing))
... solve ... with dynamics
... solve ... using ... algorithm
... separate ... by cutting ...

Related Relations

· A method or means is different from **man** in that **man** describes an event as a whole, whereas **met** characterizes the component steps or procedures of an action.
· A method or means is different from **ins** in that **met** is used for abstract things (abstract means or methods), whereas **ins** is used for concrete things

23. **mod** modification
indicates a thing that restricts a focused thing

mod (thing, thing)
mod (thing, uw(mod<thing))

Detailed Definition

A "modification" is defined as the relation between :
UW1 - a thing, and
UW2 - a restriction or a thing,
where :
· UW1 is the focused thing to be restricted by UW2, and
· UW2 is a restriction or a thing that restricts UW1 in some way.
· When UW2 is a set of UNL expressions of a clause or an phrase, this phrase or clause must be the concrete content of UW1. In this case the whole UNL expression of the phrase or clause must be expressed in a scope and will be treated as a NOMINAL concept.

Examples and Readings

mod (story(icl>tale), whole(mod<thing))
mod (plan(icl>idea), master(mod<thing))
mod (part(pof>thing), main(mod<thing))
the whole story
a master plan
the main part

Related Relations

· A modification is different from **aoj** in that **aoj** indicates a thing that is in a state or has some characteristic, whereas **mod** merely indicates a restriction of the focused thing, which might indirectly suggest some characteristics of the thing described.
· A modification is different from **man** in that **man** describes a way to carry out an event or the characteristics of a state, whereas **mod** restricts a thing

24. **nam** name
indicates a name of a thing

nam (thing, name(icl>mark))

Detailed Definition

A name is defined as the relation between :
UW1 - a thing, and
UW2 - a string used as a name,
where :
· The UWs are different, and
· UW2 is a name of UW1.

Examples and readings

nam (son(icl>relative), Hikari)
his son "Hikari"

25. **obj** affected thing
indicates a thing in focus that is directly affected by an event or state

obj (be, thing)
obj (do, thing)
obj (occur, thing)
obj (event(icl>abstract thing), thing)
obj (uw(aoj>thing,obj>thing), thing)
Detailed Definition

An affected thing is defined as the relation between :
UW1 - an event or state, and
UW2 - a thing,
where :
· UW2 is thought of as directly affected by an event or state.
Examples and Readings

obj (move(gol>place,obj>thing,src>place), table(icl>furniture))
obj (melt(gol>thing,obj>thing), sugar(icl>seasoning))
obj (cure(agt>thing,obj>thing), patient(icl>person))
obj (have(aoj>thing,obj>thing), pen(icl>writing instrument))
the table moved.
the sugar melts into ...
to cure the patient.
I have a pen.
Related Relations
· An affected thing is different from **cob** in that **obj** is in focus, whereas **cob** is related to a second, non-focused implicit event or state.

26. **opl** affected place
indicates a place in focus affected by an event

opl (do(agt>thing,obj>thing,opl>thing), thing)
opl (occur(obj>thing,opl>thing), thing)
Detailed Definition

An affected place is defined as the relation between :
UW1 - an event, and
UW2 - a place or thing defining a place,
where :
· UW2 is a place that is seen as being affected by the event.
Examples and Readings

opl (pat(icl>touch(agt>thing,obj>thing,opl>thing)), shoulder(pof>trunk))
opl (cut(agt>thing,obj>thing,opl>thing), middle(icl>place))
... pat ... on shoulder
... cut ... in middle
Related Relations
· An affected place is different from **obj** and **cob** in that what is affected by the event is a place rather than other kinds of things.
· An affected place is different from **plc** in that an affected place is directly by the event, while the physical and logical place (**plc**) defines the environment in which the event happens.

27. **or** disjunction
indicates a partner to have disjunctive relation to

or (uw, uw)
Detailed Definition

A disjunction is defined as the relation between :
UW1 - a concept, and

UW2 - a concept,
where :
· The UWs are different, and
· Some description is true for either UW1 or UW2 (but not both), or
· Some description is true for either UW1 or UW2 (and perhaps both).
Examples and Readings

or (leave(agt>thing,obj>place), stay(icl>remain(agt>thing)))
or (blue(icl>color), red(icl>color))
or (Jack(iof>person), John(iof>person))
Will you stay or leave ?
Is it red or blue ?
Who is going to do it, John or Jack ?
Related Relations
· A disjunction is different from a conjunction in that the items of disjunction are grouped in order to say that something is true for one or the other, whereas in a conjunction they are grouped to say that the same is true for both.Â A disjunction in formal logic permits three situations for it to be true :Â 1) it is true for UW1, 2) it is true for UW2, and 3) it is true for both. On the other hand, a conjunction only permits the third situation.
· A disjunction is different from **and** and **int** in that as a logical operation **int** makes an intersection and **or** makes differences, whereas **and** makes a union.

28. **per** proportion/rate/distribution
indicates a basis or unit of proportion, rate or distribution

per (thing, thing)
Detailed Definition

A proportion, rate or distribution is defined as the relation between :
UW1 - a quantity, and
UW2 - a quantity, or a thing seen as a quantity,
where :
· The UWs are different, and
· UW1 and UW2 form a proportion, where UW1 is the numerator and UW2 is the denominator, or
· UW2 is the basis or unit for understanding UW1, or
· Each UW expresses a different dimension, of size, for example.
Examples and readings

per (hour(icl>period), day(icl>period))
qua (hour(icl>period), 8)
per (time(icl>frequency), week(icl>period))
qua (time(icl>frequency), 2)
eitgh hours a day

. . . twice a week

29. **plc** place
indicates a place where an event occurs, or a state that is true, or a thing that exists

plc (be, place(icl>thing))
plc (do, place(icl>thing))
plc (occur, place(icl>thing))
plc (thing, place(icl>thing))
plc (uw(aoj>thing), place(icl>thing))
Detailed Definition

A place is defined as the relation between :
UW1 - an event, a state, or a thing, and
UW2 - a place or thing understood as a place.
Examples and Readings

plc (cook(agt>thing), kitchen(pof>building))
plc (sit(agt>thing), beside(icl>place))
plc (cool(icl>cold), here(icl>place))
... cook ... in the kitchen
... sit beside me
It's cool here.

Related Relations

· A place is different from **plf** and **plt** or **src** and **go** l in that **plc** describes a place with respect to an event as a whole, whereas these other relations describe the position with respect to parts of an event.
· A place is different from **opl** in that **plc** is not seen as being affected by an event but merely as a reference point for characterizing it, whereas **op**

30. **l** is seen as being affected
31. **plf** initial place
indicates a place where an event begins or a state that becomes true

plf (be, place(icl>thing))
plf (do, place(icl>thing))
plf (occur, place(icl>thing))
plf (uw(aoj>thing), place(icl>thing))

Detailed Definition

An "initial place" (or "place-from") is defined as the relation between :
UW1 - an event or state, and
UW2 - a place or thing defining a place,
where :
· UW2 is the specific place where UW1 started, or
· UW2 is the specific place from where UW1 is true.

Examples and Readings

plf (travel(agt>volutional thing), Tokyo(icl>city))
plf (deep(aoj>thing), there(icl>place))
traveling from Tokyo
The sea is deep from there to here

Related Relations

· An initial place is different from **plc** in that **plc** describes events or states taken as a whole, whereas **plf** describes only the initial part of an event or state.
· An initial place is different from **plt** in that **plt** describes the final part of an event or state, whereas **plf** describes the initial part of an event or state.
· An initial place is different from **src** in that **plf** describes the place where the event began, whereas **src** describes the initial state of the object.

32. **plt** final place
indicates a place where an event ends or a state that becomes false

plt (be, place(icl>thing))
plt (do, place(icl>thing))
plt (occur, place(icl>thing))
plt (uw(aoj>thing), place(icl>thing))

Detailed Definition

A final place is defined as the relation between :
UW1 - an event or state, and
UW2 - a place or thing defining a place,
where :
· UW2 is the specific place where UW1 ended, or
· UW2 is the specific place where UW2 becomes false.

Examples and readings

plt (travel(agt>volitional thing), Boston(iof>city))
plt (deep(aoj>thing), here(icl>place))
to travel to Boston
The sea is deep from there to here
Related Relations
· A final place is different from **plc** in that **plc** describes events or states taken as a whole, whereas **plt** describes only the final part of an event.
· A final place is different from **plf** in that **plt** describes the final part of an event or state, whereas **plf** describes the initial part of an event.
· A final place is different from **gol** in that **plt** describes the place where an event or state ended, whereas **gol** describes the final state of the object.

33. **pof** part of
indicate a concept of which a focused thing is a part

pof (thing, thing)
Detailed Definition

Part-of is defined as the relation between :
UW1 - a partial thing, and
UW2 - a whole thing,
where :
· The UWs are different, and
· UW1 is a part of UW2
Examples and Readings

pof (preamble(icl>information), document(icl>information))
pof (initial(icl>letter), machine translation)
the preamble of a document
the initials of Machine Translation

34. **pos** possessorÂ
indicates the possessor of a thing

pos (thing, volitional thing)
Detailed Definition

A possessor is defined as the relation between :
UW1 - a thing or a place, and
UW2 - a human or non-human, seen as a volitional thing
where :
· UW2 is a possessor of UW1.
Examples and readings

pos (dog(icl>aminal), John(iof>person))
pos (book(icl>document), I)
John's dog
my book

35. **ptn** partner
indicates an indispensable non-focused initiator of an action

ptn (do, thing)
ptn (action(icl>event), thing)
Detailed Definition

A partner is defined as the relation between :
UW1 - an action, and
UW2 - a human or non-human, seen as a volitional thing
where :
· UW1 is a collaborative event initiated by both the agent and the partner, and
· UW2 is thought of as having a direct role in making an indispensable part of UW1 happen, and

· W2 is seen as not being in focus (as compared to the agent).
Examples and Readings

ptn (compete(agt>thing,ptn>thing), John(iof>person))
ptn (share(icl>divide(agt>thing,obj>thing)), poor(icl>person))
ptn (collaborate(agt>thing,ptn>person), he)
... compete with John
... share ... with the poor
... collaborate with him ...
Related Relations
· A partner is different from **agt** in that an agent and its event are in focus, while a partner and its event are not in focus.
· A partner is different from **cag** in that a co-agent initiates an event that is independent of an agent's event, whereas a partner initiates the same event together with an agent.

36. **pur** purpose
indicates the purpose or objective of an agent of an event or the purpose of a thing that exists

pur (do, do)
pur (do, thing)
pur (thing, uw)
Detailed Definition

A purpose or objective is defined as the relation between :
UW1 - a thing or an action, and
UW2 - a thing or an action,
where :
When UW1 is an action :
· UW2 specifies the agent's purpose or objective, or
· UW2 specifies the thing (object, state, event, etc.) that the agent desires to attain by carrying out UW1, or
When UW1 is a thing :
· UW2 is what UW1 is to be used for.
Examples and Readings

pur (come(icl>move(agt>thing,gol>place,src>place)),Â
see(icl>meet(agt>volitional thing,obj>thing)))
pur (work(agt>person), money(icl>mark))
pur (budget(icl>expense), research(icl>study))
... come to see you

... work for money
our budget for research
Related Relations
· A purpose or objective is different from **gol** in that **pur** describes the desires of an agent, whereas **gol** describes the state of the object at the end of an event.
· A purpose or objective is different from **man** and **met** in that **pur** describes the reason (purpose) why the event is being carried out, while **man** and **met** describe how it is being carried out.

37. **qua** quantity
indicates the quantity of a thing or unit

qua (thing, quantity)
Detailed Definition

A quantity is defined as the relation between :
UW1 - a thing, and
UW2 - quantity,
where :
· UW2 is the number or amount of UW1.
Examples and Readings

qua (cup(icl>tabelware), 2))
qua (coffee(icl>beverage), cup(icl>tableware))
qua (kilogram(icl>unit), many(qua<thing))
qua (dog(icl>animal), 2)
Two cups of coffee

many kilograms
two dogs

Related Relations

· A quantity is different from **per** in that a quantity is an absolute number or amount, whereas **per** is a number or amount relative to some unit of reference (time, distance, etc.).
· A quantity is also used to express iteration, or the number of times an event or state occurs.

38. **rsn** reason
indicates a reason why an event or a state happens

rsn (be, be)
rsn (be, do)
rsn (be, thing)
rsn (be, occur)
rsn (be, uw(aoj>thing))
rsn (do, be)
rsn (do, do)
rsn (do, thing)
rsn (do, occur)
rsn (do, uw(aoj>thing))
rsn (occur, be)
rsn (occur, do)
rsn (occur, occur)
rsn (occur, thing)
rsn (occur, uw(aoj>thing))
rsn (uw(aoj>thing), be)
rsn (uw(aoj>thing), do)
rsn (uw(aoj>thing), occur)
rsn (uw(aoj>thing), thing)
rsn (uw(aoj>thing), uw(aoj>thing))

Detailed Definition

A reason is defined as the relation between :
UW1 - an event or state, and
UW2 - a thing, an event or a state,
where :
· UW2 is a reason why UW1 happens.

Examples and Readings

rsn (go(icl>move(agt>thing,gol>place,src>place)).@not, rain(icl>weather))
agt :01 (arrive(icl>come(agt>thing,gol>place,src>place)), Mary(iof>person))
rsn (start(icl>begin(agt>thing,obj>thing)), :01)
rsn (known(aoj>thing), beauty(icl>abstract thing))
aoj (known(aoj>thing), city(icl>region))
mod (beauty(icl>abstract thing), city(icl>region))
... didn't go because of the rain
They can start because Mary arrived.

a city known for its beauty

39. **scn** scene
indicates a scene where an event occurs, or state is true, or a thing exists

scn (be, thing)
scn (do, thing)
scn (occur, thing)

scn (thing, thing)
scn (uw(aoj>thing), thing)
Detailed Definition

A scene is defined as the relation between :
UW1 - an event or state or thing, and
UW2 - an abstract or metaphorical thing (world) understood as a scene,
where :
· UW2 is the scene that UW1 happens or is true. When UW2 is a concrete thing or place, it is a metaphorical use, and
· UW1 is true or happens in a scene characterized by UW2.
Examples and Readings

scn (win(agt>thing), contest(icl>event))
scn (appear(gol>thing,obj>thing), program(icl>plan))
scn (play(agt>thing,obj>thing), movie(icl>cinema))
... win a prize in a contest
... appear on a TV program
... play in movie
Related Relations
· A scene is different from **plc** in that the reference place for **plc** is in the real place that something happens, whereas for **scn** it is an abstract or metaphorical world.

40. **seq** sequence
indicates a prior event or state of a focused event or state

seq (do, do)
seq (do, occur)
seq (do, uw(aoj>thing))
seq (occur, do)
seq (occur, occur)
seq (occur, uw(aoj>thing))
seq (uw(aoj>thing), do)
seq (uw(aoj>thing), occur)
seq (uw(aoj>thing), uw(aoj>thing))
Detailed Definition

A "sequence" is defined as the relation between :
UW1 - a focused event or state,
UW2 - a prior event or state,
where :
· UW1 occurs or is true after UW2.
Examples and Readings

seq (leap(icl>jump(agt>thing)), look(agt>thing,obj>thing))
seq (red(aoj>thing), green(aoj>thing))
seq (take off(agt>thing,obj>thing), come in(agt>thing))
Look before you leap.
It was green and then red.
She came in and took her coat off
Related Relations
· A sequence is different from **coo** in that **seq** describes events or states that do not occur at the same time, but one after the other, whereas **coo** describes events that occur simultaneously.

41. **src** source/initial state
indicates the initial state of an object or thing initially associated with the object of an event

src (be(aoj>thing,gol>thing), thing)
src (do, thing)
src (do, uw(aoj>thing))
src (occur, thing)

src (occur, uw(aoj>thing))
Detailed Definition

An initial state is defined as the relation between :
UW1 - an event, and
UW2 - a state or thing,
where :
· UW2 is the specific state describing the object of UW1 at the beginning of UW1, or
· UW2 is a thing that is associated with the object of UW1 at the beginning of UW1.
Examples and readings

src (change(obj>thing), red(aoj>thing))
src (withdraw(agt>thing,obj>thing), stove(icl>furniture))
The lights changed from green to red.
I quickly withdrew my hand from the stove
Related Relations
· An initial state is different from **tmf** and **plf** in that src describes qualitative characteristics of the object and not time or place of an event.
· An initial state is different from **gol** in that **gol** describes the characteristics of the object at the final state of the event.

42. **tim** time
indicates the time an event occurs or a state is true

tim (be, time(icl>abstract thing))
tim (do, time(icl>abstract thing))
tim (occur, time(icl>abstract thing))
tim (thing, time(icl>abstract thing))
tim (uw(aoj>thing), time(icl>abstract thing))
Detailed Definition

Time is defined as the relation between :
UW1 - an event or state, and
UW2 - a time, or an event or a state that can be seen as a time,
where :
· UW1, taken as a whole, occurs at the time indicated by UW2.
Examples and Readings

tim (leave(agt>thing,obj>place), Tuesday(icl>day))
tim (do(agt>thing,obj>thing), o'clock(icl>time))
tim (start(icl>>begin(agt>thing,obj>thing)), come(icl>move(agt>thing,gol>place,src>place)))
. . . leave on Tuesday
. . . do . . . at . . . o'clock
Let's start when . . . come
Related Relations
· ime is different from **tmf** and **tmt** in that time characterizes the event or state as a whole, whereas **tmf** and **tmt Â** describe only parts of the event.
· Time is different from **coo** and **seq** in that time does not describe states and events relatively, with respect to each other, but with respect to certain points in time.

43. **tmf** initial time
indicates the time an event starts or a state becomes true

tmf (be, time(icl>abstract thing))
tmf (do, time(icl>abstract thing))
tmf (occur, time(icl>abstract thing))
tmf (thing, time(icl>abstract thing))
tmf (uw(aoj>thing), time(icl>abstract thing))
Detailed Definition

Initial time is defined as the relation between :
UW1 - an event or state, and

UW2 - a time, or an event or a state that can be seen as a time,
where :
· UW2 specifies the time at which UW1 starts, or
· UW2 specifies the time at which UW1 becomes true.
Examples and Readings

tmf (work(agt>person), morning(icl>time))
tmf (change(obj>thing), live(agt>volitional thing))
... work from morning to [till] night
... has changed ... since I have lived here.
Related Relations
· Initial time is different from **tim** in that **tmf** expresses the time at the beginning of the event or state whereas **tim** expresses the time for the event taken as a whole.
· Initial time is different from **src** in that **tmf** expresses the time at the beginning of the event or state whereas **src** expresses characteristics of the object at the beginning of the event.
· Initial time is different from **tmt** in that **tmf** expresses the time at the beginning of the event or state whereas **tmt** expresses the time at its end.

44. **tmt** final time
indicates a time an event ends or a state becomes false

tmt (be, time(icl>abstract thing))
tmt (do, time(icl>abstract thing))
tmt (occur, time(icl>abstract thing))
tmt (thing, time(icl>abstract thing))
tmt (uw(aoj>thing), time(icl>abstract thing))
Detailed Definition

Final time is defined as the relation between :
UW1 - an event or state, and
UW2 - a time, or an event or a state that can be seen as a time,
where :
· UW2 specifies the time at which UW1 ends, or
· UW2 specifies the time at which UW1 becomes false.
Examples and Readings

tmt (work(agt>person), night(icl>time))
tmt (full(aoj>thing), tomorrow(icl>time))
... work from morning to [till] night
... be full till tomorrow
Related Relations
· Final time is different from **tim** in that **tmt** expresses the time at the end of the event or state, whereas **tim** expresses the time for the event taken as a whole.
· Final time is different from **gol** in that **tmt** expresses the time at the end of the event or state, whereas **gol** expresses characteristics of the object at the end of the event.
· Final time is different from **tmf** in that **tmt** expresses the time at the end of the event or state, whereas **tmt** expresses the time at the beginning of the event.

45. **to** destination
indicates a final state of a thing or a final thing (destination) associated with the focused thing

to (thing, thing)
Detailed Definition

A destination is defined as the relation between :
UW1 - a thing, and
UW2 - a state or a thing that can be seen as destination,
where :
· The UWs are different, and
· UW1 is the focused thing, and
· UW2 is the final state describing the focused thing UW1, or

· UW2 is a thing that is finally associated with the UW1, destination such as the final position of UW1.

Examples and Readings

to (train(icl>vehicle), London(iof>city))
to (letter(icl>document), you)
a train for London
a letter to you

Related Relations

· A destination is different with **gol** in that **gol** is a relation used with an event or a state, whereas **to** is directly linked to a thing. For instance, "a letter to you" is expressed as "**to** (letter(icl>document), you)", whereas "a letter sent to you" is expressed as "**gol** (send(agt>thing,gol>thing, obj>thing), you)" and "obj (send(agt>thing,gol>thing,obj>thing), letter(icl>document))".

46. **via** an intermediate place or state
indicates an intermediate place or state of an event

via (do, thing)
via (action(icl>event), thing)
via (occur, thing)
via (phenomenon(icl>event), thing)

Detailed Definition

An intermediate place or state is defined as the relation between :
UW1 - an event, and
UW2 - a place or state,
where :
· UW2 is the specific place or state describing the object of UW1 at some time in the middle of UW1,
· UW2 is a thing that describes a place or state that the object of UW1 passed by or through during UW1..

Examples and Readings

via (go(icl>move(agt>thing,gol>place,src>place)), New York(iof>city))
via (bike(agt>thing), Alps(iof>mountain))
via (drive(agt>thing), tunnel(icl> facilities))
... go ... via New York
... bike ... through the Alps
... drive ... by way of the tunnel

Related Relations

· An intermediate place or state is different from **src** , **plf** and **tmf** in that these all refer to the beginning of an event, whereas **via** describes the middle of an event.
· An intermediate place or state is different from **gol** , **plt** and **tmt** in that these all refer to the end of an event, whereas **via** describes the middle of an event.

G

Les fonctions lexicales pour l'analyse

Voici la liste des fonctions lexicales pour l'analyse étudiées dans cette thèse. Cette liste a été créée à partir de [Mel'čuk, 1988], [Mel'čuk *et al.*, 1995], de [Polguère, 2003] et de la version 2005 des spécifications d'UNL disponible sur la page http://www.undl.org/unlsys/unl/unl2005/.

Dans les fonctions, le ième indice indique que la valeur de la fonction est le ième argument. Ainsi, pour la *nominalisation*, on a « *Un* ‹voleur› *(S_1 agent) vole un* ‹butin› *(S_2 patient) à une* ‹victime› *(S_3).* ».

G.1 Fonctions Lexicales d'Analyse pour les Connaissances Linguistiques (FLACL)

G.1.1 Paradigmatiques : fonctions à caractère à la fois thématique et lexical

1. **Synonymie** [*Syn*].
   Syn(‹*destin*›) = ‹*destinée*›; Syn(‹*déshydrater*›) = ‹*sécher*›; Syn(‹*voiture*›) = ‹*automobile*›
2. **Antonymie** [$Anti_\alpha$].
   $Anti_c$(‹*existence*›) = ‹*inexistence*› ; $Anti_s$(‹*chaleur*›) = ‹*froid*› ; $Anti_d$(‹*père*›) = ‹*fils*› ;
3. **Générique** [*Gener*].
   $Gener$(‹*gaz*›) = ‹*substance [gazeuse]*›; $Gener$(‹*avion*›) = ‹*appareil*›; $Gener$(‹*tractopelle*›) = ‹*engin*›

G.1.2 Syntagmatiques : fonctions à caractère purement lexical

4. **Épithète pléonastique** [*Epit*]. Adjectif ou adverbe sans contribution sémantique dans le cadre d'un cliché.
   $Epit$(‹*océan*›) = ‹*immense*›; $Epit$(‹*gagnant*›) = ‹*heureux*›; $Epit$(‹*défier*›) = ‹*ouvertement*›
5. **Singulatif** [*Sing*]. Fonction équivalente à « *Unité minimale régulière de...* », fonction inverse de *Mult*
   $Sing$(‹*flotte*›) = ‹*navire*›; $Sing$(‹*vol*›) = ‹*oiseau*›; $Sing$(‹*riz*›) = ‹*grain*›
6. **Collectif** [*Mult*]. Fonction équivalente à « *Ensemble régulier de...* », fonction inverse de *Sing*
   $Mult$(‹*navire*›) = ‹*flotte*›; $Mult$(‹*chien*›) = ‹*meute*›; $Mult$(‹*abeille*›) = ‹*essaim*›
7. **Nom du chef** [*Cap*].
   Cap(‹*université*›) = ‹*président*›; Cap(‹*faculté*›) = ‹*doyen*›; Cap(‹*bateau*›) = ‹*capitaine*›
8. **Nom de l'équipe** [*Equip*].
   $Equip$(‹*théâtre*›) = ‹*troupe*›; $Equip$(‹*bateau*›) = ‹*équipage*›; $Equip$(‹*football*›) = ‹*équipe*›

9. **Nom de démarrage** [*Germ*].
   $Germ(\text{‹}colère\text{›}) = \text{‹}ferment\text{›}$; $Germ(\text{‹}match\text{›}) = \text{‹}coup\ d'envoi\text{›}$; $Germ(\text{‹›}) = \text{‹›}$
10. **Nom du centre** [*Centr*].
    $Centr(\text{‹}problème\text{›}) = \text{‹}cœur\text{›}$; $Centr(\text{‹}Terre\text{›}) = \text{‹}centre\text{›}$; $Centr(\text{‹}atome\text{›}) = \text{‹}noyau\text{›}$
11. **Nom du point culminant** [*Culm*].
    $Culm(\text{‹}joie\text{›}) = \text{‹}comble\text{›}$; $Culm(\text{‹}colère\text{›}) = \text{‹}paroxisme\text{›}$; $Culm(\text{‹}savoir\text{›}) = \text{‹}apex\text{›}$
12. **Intensificateur** [*Magn*].
    $Magn(\text{‹}amour\text{›}) = \text{‹}fou\text{›}$; $Magn(\text{‹}peur\text{›}) = \text{‹}bleu\text{›}$; $Magn(\text{‹}fièvre\text{›}) = \text{‹}de\ cheval\text{›}$
13. **Confirmateur** [*Ver*].
    $Ver(\text{‹}argument\text{›}) = \text{‹}valable\text{›}$; $Ver(\text{‹}succès\text{›}) = \text{‹}mérité\text{›}$; $Ver(\text{‹}peur\text{›}) = \text{‹}justifié\text{›}$
14. **Laudatif** [*Bon*].
    $Bon(\text{‹}conseil\text{›}) = \text{‹}précieux\text{›}$; $Bon(\text{‹}choix\text{›}) = \text{‹}heureux\text{›}$; $Bon(\text{‹}se\ porter\text{›}) = \text{‹}comme\ un\ charme\text{›}$
15. **Positif** [Pos_2].
    $Pos_2(\text{‹}opinion\text{›}) = \text{‹}favorable\text{›}$; $Pos_2(\text{‹}critique\text{›}) = \text{‹}élogieuse\text{›}$; $Pos_2(\text{‹}compte\ rendu\text{›}) = \text{‹}favorable\text{›}$
16. **Consécutif** [*Propt*].
    $Propt(\text{‹}jalousie\text{›}) = \text{‹}par\text{›}$; $Propt(\text{‹}maladie\text{›}) = \text{‹}pour\ cause\ de\text{›}$; $Propt(\text{‹}alcool\text{›}) = \text{‹}sous\ l'emprise\ de\text{›}$

G.2 Fonctions Lexicales d'Analyse pour les Connaissances du Monde (FLACM)

G.2.1 Fonctions à caractère à la fois thématique et lexical

17. **Hyperonymie** [*Hyper*].
    $Hyper(\text{‹}cheval\text{›}) = \text{‹}mammifère\text{›}$; $Hyper(\text{‹}siège\text{›}) = \text{‹}meuble\text{›}$; $Hyper(\text{‹}fauteuil\text{›}) = \text{‹}siège\text{›}$
18. **Hyponymie** [*Hypo*].
    $Hypo(\text{‹}mammifère\text{›}) = \text{‹}cheval\text{›}$; $Hypo(\text{‹}meuble\text{›}) = \text{‹}siège\text{›}$; $Hypo(\text{‹}siège\text{›}) = \text{‹}fauteuil\text{›}$
19. **Classe** [*Class*].
    $Class(\text{‹}Jacques\ Chirac\text{›}) = \text{‹}homme\ politique\text{›}$; $Class(\text{‹}Zinédine\ Zidane\text{›}) = \text{‹}footballeur\text{›}$; $Class(\text{‹}Marie\ Curie\text{›}) = \text{‹}scientifique\text{›}$
20. **Instance** [*Inst*].
    $Inst(\text{‹}homme\ politique\text{›}) = \text{‹}Jacques\ Chirac\text{›}$; $Inst(\text{‹}footballeur\text{›}) = \text{‹}Zinédine\ Zidane\text{›}$; $Inst(\text{‹}scientifique\text{›}) = \text{‹}Marie\ Curie\text{›}$

G.2.2 Fonctions à caractère purement lexical

21. **Méronymie** [*Méro*].
    $Méro(\text{‹}bateau\text{›}) = \text{‹}voile\text{›}$; $Méro(\text{‹}livre\text{›}) = \text{‹}page\text{›}$; $Méro(\text{‹}maison\text{›}) = \text{‹}mur\text{›}$
22. **Holonymie** [*Holo*].
    $Holo(\text{‹}voile\text{›}) = \text{‹}bateau\text{›}$; $Holo(\text{‹}page\text{›}) = \text{‹}livre\text{›}$; $Holo(\text{‹}mur\text{›}) = \text{‹}maison\text{›}$
23. **Instrument** [S_{inst}].
    $S_{inst}(\text{‹}creuser\text{›}) = \text{‹}pelle\text{›}$; $S_{inst}(\text{‹}peindre\text{›}) = \text{‹}pinceau\text{›}$; $S_{inst}(\text{‹}écrire\text{›}) = \text{‹}stylo\text{›}$
24. **Lieu** [S_{loc}].
    $S_{loc}(\text{‹}parler\text{›}) = \text{‹}parloir\text{›}$; $S_{loc}(\text{‹}combattre\text{›}) = \text{‹}arène\text{›}$; $S_{loc}(\text{‹}dormir\text{›}) = \text{‹}lit\text{›}$
25. **Moyen** [S_{med}].
    $S_{med}(\text{‹}peindre\text{›}) = \text{‹}peinture\text{›}$; $S_{med}(\text{‹}écrire\text{›}) = \text{‹}encre\text{›}$; $S_{med}(\text{‹}laver\text{›}) = \text{‹}savon\text{›}, \text{‹}lessive\text{›}, \text{‹}détergent\text{›}$

26. **Mode** [S_{mod}].
    $S_{mod}(\text{‹}écrire\text{›}) = \text{‹}écriture\text{›}$; $S_{mod}(\text{‹}parler\text{›}) = \text{‹}façon\text{›}$; $S_{mod}(\text{‹}peindre\text{›}) = \text{‹}peinture\text{›}$
27. **Résultat** [S_{res}].
    $S_{res}(\text{‹}laver\text{›}) = \text{‹}lessive\text{›}$; $S_{res}(\text{‹}copier\text{›}) = \text{‹}copie\text{›}$; $S_{res}(\text{‹}écrire\text{›}) = \text{‹}écrit\text{›}$

H

Blexisma

Blexisma (*Base LEXIcale Sémantique Multi-Agents*) est un Système multi-agents dont l'objectif est d'intégrer tout élément lui permettant de créer, d'améliorer ou/et[119] d'exploiter une ou plusieurs Bases Lexicales Sémantiques dont l'architecture à trois niveaux (LEXIE, ACCEPTION, ITEMS LEXICAL) a été décrite en 5.1. Le développement du moteur du système (caractéristiques des agents, macro-organisation, communications, ...) a été réalisé au cours de cette thèse. J'ai développé ensuite quelques agents qui ont chacun une fonction particulière (base de données, apprentissage, expert fonction lexicale, ...). L'implémentation a été réalisée en Java. Dans cette annexe, nous donnons quelques chiffres sur l'implémentation et l'expérience menée.

Les messages échangés entre agents sont des classes des très simples qui ne comportent que des données et jamais de calculs. Il faut les interpréter comme une requête adressable à l'agent en vue d'une action spécifique.

H.1 Java

Implémenté sous JBuilder[120] puis eclipse 3.0.1.[121] avec le JDK[122] (version 1.3 à 1.5).

H.2 Moteur de Blexisma

- 4 packages
- 37 classes dont les principales classes permettant la construction des structures et les opérations définies sur eux : objets lexicaux, vecteurs conceptuels, ...
- 19 messages

H.3 Agents

total :

- packages principaux : 150 classes
- messages : 59 classes

[119] Avec un objectif fort pour le *et* (cf. hypothèse VI, 5.1.6).
[120] http://www.borland.fr/jbuilder/
[121] http://www.eclipse.org/
[122] http://java.sun.com/j2se/

H.3.1 Base de données vectorielle

H.3.1.1 Base

- package principal : 13 classes
- messages : 31 classes

H.3.1.2 Catégorisateur

Cet agent permet de atégoriser les LEXIES de la BLS et ainsi de fabriquer des ACCEPTIONS (cf. 5.1.3). Sa mise en œuvre est basé sur les travaux de Fabien Jalabert [Jalabert & Lafourcade, 2004a].

- package principal : 4 classes
- messages : 1 classe

H.3.2 Agents pour l'analyse sémantique

H.3.2.1 Contextualiseur

Cet agent met en œuvre la méthode de contextualisation forte présentée en 2.3.6.

- package principal : 4 classes
- messages : 3 classes

H.3.2.2 Distance orthographique

Cet agent permet de corriger les fautes des textes, il est utilisé dans le préformatage des textes (cf. 2.1.6.2). Il donne une distance entre une chaîne de caractères et les mots de la base. La première version a été réalisée par Fabien Jalabert, Jean-Batiste Munier et Mehdi Yousfi-Monod alors étudiants en maîtrise informatique à l'université Montpellier II[123]. Je l'ai reprise pour l'adapter au moteur Blexisma.

- package principal : 27 classes
- messages : 2 classes

H.3.2.3 Lemmatisation

Agent de lemmatisation basé sur la liste ABU[124]. Cet agent indique à l'agent SYGFRAN les informations qui peuvent lui manquer.

- package principal : 6 classes
- messages : 1 classe

H.3.2.4 Analyse sémantique remontée-redescente

Cet agent met en œuvre l'analyse en remontée-redescente présentée en 2.3.7.

- package principal : 4 classes
- messages : 2 classes

H.3.2.5 Analyse sémantique fourmis

Cet agent met en œuvre l'analyse sémantique grâce à un algorithme à fourmis présentée en 7.7.

- package principal : 8 classes

[123] Leur rapport est consultable à l'adresse http://www.lgi2p.ema.fr/~jalabert/publications/terMAITRISE/rapport_ter.pdf

[124] http://abu.cnam.fr/DICO/mots-communs.html

– messages : 2 classes

H.3.2.6 Interfaçage avec SYGFRAN

Cet agent utilise SYGFRAN pour fournir un arbre morpho-syntaxique.
– package principal : 8 classes
– messages : 1 classe

H.3.3 Apprentissage sur dictionnaires

H.3.3.1 Récupérations de données dictionnairiques

Ces agents récupèrent des données dictionnairiques depuis le Web ou à partir de fichiers et les fournissent à l'agent d'apprentissage sur définitions (cf. H.3.3.2).
– packages principaux : 35 classes
– messages : 4 classes

H.3.3.2 Apprentissage sur définitions

Cet agent gère l'apprentissage des définitions de dictionnaires à usage humain. Il formate les définition et en extrait, entre autres, les informations morphologiques (cf. 2.3.5).
– packages principaux : 14 classes
– messages : 5 classes

H.3.4 Agents FLA

H.3.4.1 Agent d'antonymie

Cet agent utilise des listes d'antonymes pour fabriquer des vecteurs conceptuels ou évaluer cette relation entre objets lexicaux (cf. chapitres 3 et 4)
– package principal : 9 classes
– messages : 2 classes

H.3.4.2 Extracteur d'antonymes

Extrait les antonymes grâce à leur morphologie (cf 4.2.2.2) et les mets à disposition des autres agents.
– package principal : 6 classes
– messages : 3 classes

H.3.4.3 Agent de synonymie

Cet agent utilise des dictionnaires de synonymes pour fabriquer des vecteurs conceptuels ou évaluer cette relation entre objets lexicaux (cf. chapitres 3 et 4)
– package principal : 5 classes
– messages : 1 classe

H.3.4.4 Extracteur de synonymes

Extrait les synonymes du dictionnaires des synonymes du CRISCO[125] et les mets à disposition des autres agents.
– package principal : 5 classes

[125] http://elsap1.unicaen.fr/cgi-bin/cherches.cgi

– messages : 1 classe

H.4 Classes Utilitaires

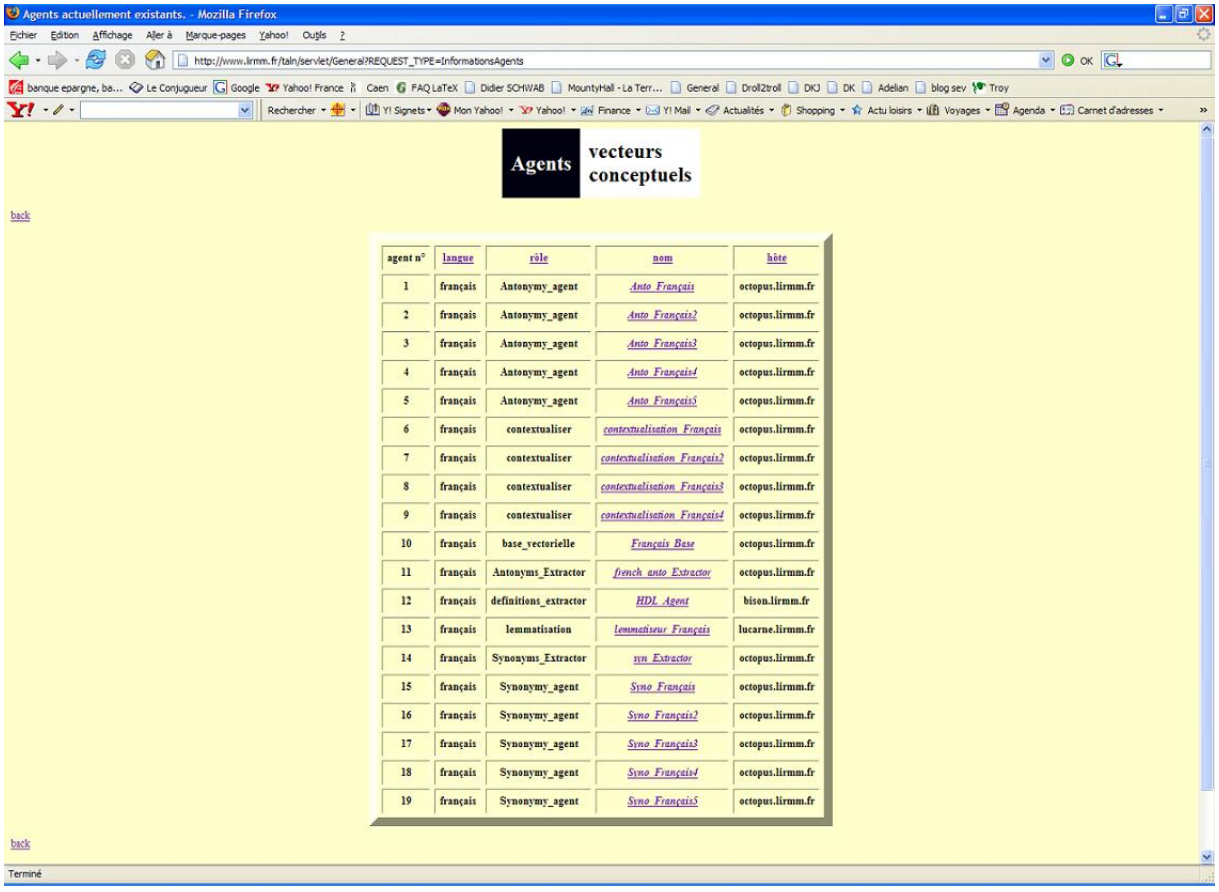
Classe permettant de gérer les entrées sorties disque, réseau, des utilitaires de calculs, de conversions, de transformations XML, ...

– 4 packages
– 21 classes

H.5 Accès Web

Les accès sont gérés par des servlet tournant grâce à Jakarta[126](les serveurs de Java). Via le réseau, il est, entre autre, possible de tuer un agent, de lancer et stopper l'apprentissage, de demander l'indexation de termes, d'obtenir toutes les informations disponibles dans la base sur un terme, de faire des calculs de voisinage, ...

11 services ont été mis en place pour une vingtaine de requêtes possibles.



agent n°	langue	rôle	nom	hôte
1	français	Antonymy_agent	Anto_Français	octopus.lirmm.fr
2	français	Antonymy_agent	Anto_Français2	octopus.lirmm.fr
3	français	Antonymy_agent	Anto_Français3	octopus.lirmm.fr
4	français	Antonymy_agent	Anto_Français4	octopus.lirmm.fr
5	français	Antonymy_agent	Anto_Français5	octopus.lirmm.fr
6	français	contextualiser	contextualisation_Français	octopus.lirmm.fr
7	français	contextualiser	contextualisation_Français2	octopus.lirmm.fr
8	français	contextualiser	contextualisation_Français3	octopus.lirmm.fr
9	français	contextualiser	contextualisation_Français4	octopus.lirmm.fr
10	français	base_vectorielle	Français_Base	octopus.lirmm.fr
11	français	Antonyms_Extractor	french_anto_Extractor	octopus.lirmm.fr
12	français	definitions_extractor	HDL_Agent	bison.lirmm.fr
13	français	lemmatisation	lemmatiseur_Français	lucarne.lirmm.fr
14	français	Synonyms_Extractor	syn_Extractor	octopus.lirmm.fr
15	français	Synonymy_agent	Syno_Français	octopus.lirmm.fr
16	français	Synonymy_agent	Syno_Français2	octopus.lirmm.fr
17	français	Synonymy_agent	Syno_Français3	octopus.lirmm.fr
18	français	Synonymy_agent	Syno_Français4	octopus.lirmm.fr
19	français	Synonymy_agent	Syno_Français5	octopus.lirmm.fr

FIG. H.1 – Vision générale des agents lancés.

[126] http://jakarta.apache.org/

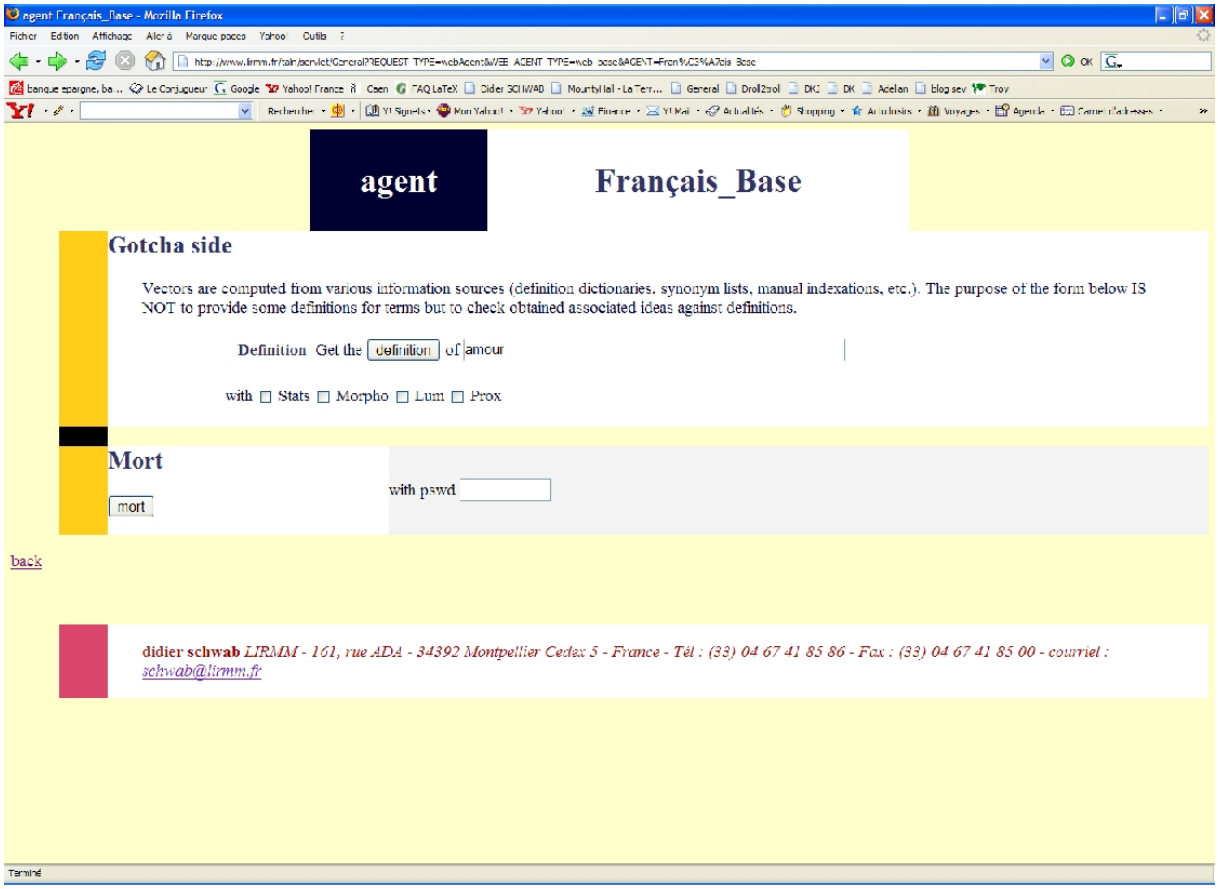


FIG. H.2 – Page d'accès à l'agent Base.

H.6 Expérience

L'apprentissage et la révision des données de la base du français sont permanent : il a occupé dans sa plus grande version jusqu'à 115 agents et tourné simultanément sur cinq machines du laboratoire (PC linux, Sun sous UNIX). Le suivi et le contrôle des agents sont accessibles via le Web[127]. Parmi les dictionnaires utilisés, on trouve les versions électroniques du Larousse [Larousse, 2004], du Robert [Robert, 2000], du thésaurus Larousse [Larousse, 1992] et le dictionnaire des synonymes et des antonymes de Caen[128].

En Septembre 2005, la base contient environ 121 000 ITEMS LEXICAUX pour 276 000 ACCEPTIONS et 842 000 LEXIES. Il faut approximativement 4 jours pour effectuer un cycle complet de révision de la base.

[127] http://www.lirmm.fr/~schwab
[128] http://elsap1.unicaen.fr/cgi-bin/cherches.cgi

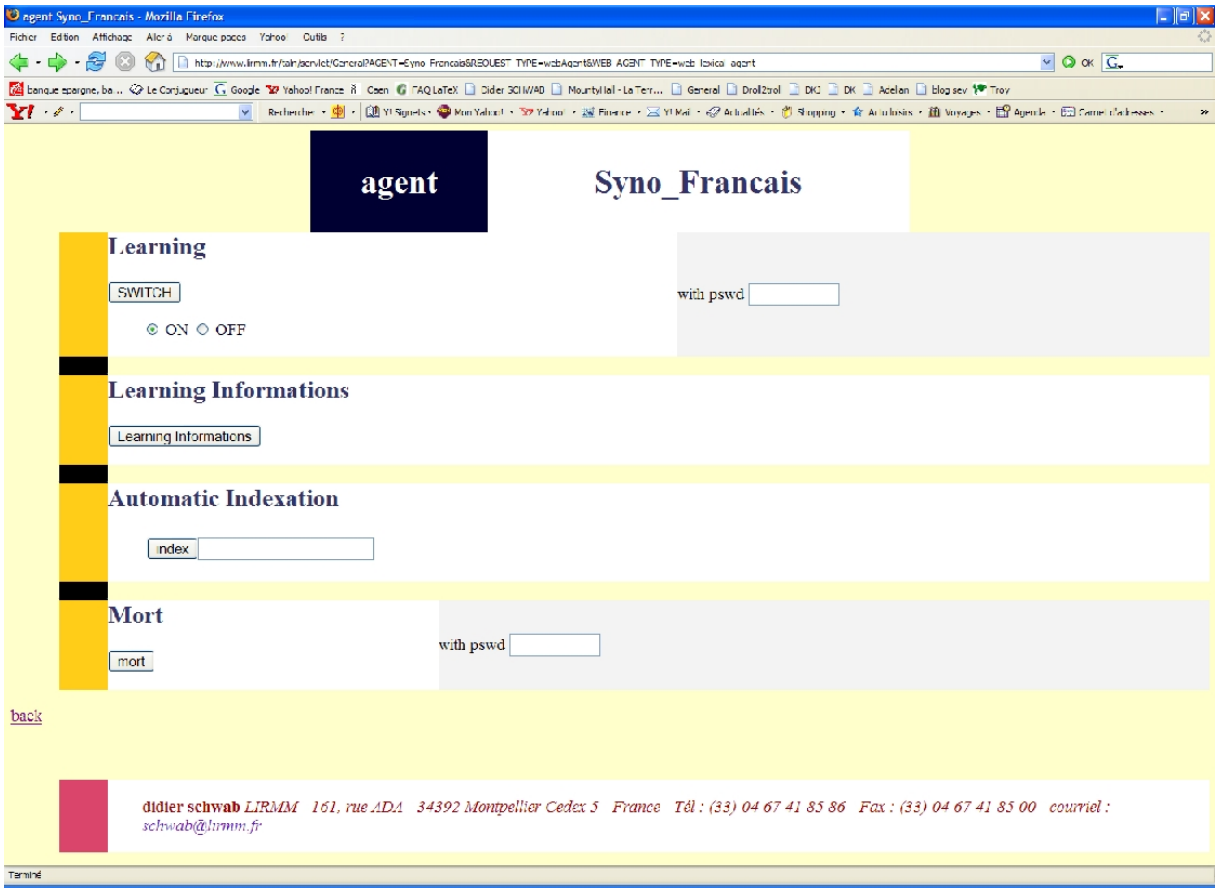


FIG. H.3 – Page d'accès à un des agents gérant les fonctions lexicales, un agent de synonymie.

I

Glossaire

acception : Une acception est un sens particulier d'un mot, admis et reconnu par l'usage. Il s'agit d'une unité sémantique propre à une langue donnée [Sérasset & Mangeot, 2001]. Par exemple, le terme ‹*botte*› possède au moins trois acceptions, la ‹*chaussure*›, l'‹*amas de paille*› et le ‹*coup porté en escrime*›. Les acceptions sont donc monosémiques.

affixe : cf. *morphème*

aire sémantique : L'aire sémantique d'un item lexical est l'ensemble des significations qu'il est susceptible d'avoir.

antidictionnaire : Un antidictionnaire est une liste de mots qui doivent être ignorés car considérés comme non pertinents dans le cadre d'une certaine application. Ainsi, une telle liste, dans le cadre d'une application visant la sémantique, comprendra les mots vides de sens comme les mots outils (pronoms, articles, ...), dans un cadre distributionnel, les mots trop fréquents dans le corpus. (anglais : *stop-list*)

antonymie : Deux items lexicaux sont en relation d'antonymie si on peut exhiber une symétrie de leurs traits sémantiques par rapport à un axe.

anaphore : Un mot à valeur anaphorique ne peut être interprété que lorsqu'il est mis en relation avec un autre élément de l'énoncé. Par exemple, dans « *En ce moment, le second attira de nouveau l'attention du capitaine. Celui-ci suspendit sa promenade et dirigea sa lunette vers le point indiqué.* », ‹‹celui-ci›› est une anaphore de ‹‹capitaine››.

archisémène : Dans une analyse sémique, l'archisémène est l'objet lexicalisé ou non dont les traits sémantiques sont l'intersection mathématique des traits des sémènes étudiés ([Nyckees, 1998], 211). Ainsi dans l'analyse sur les moyens de transport présentée par Pottier [Pottier, 1964] (cf. figure 1.18), ‹*véhicule*› est l'archisémène.

base de la collocations : cf. *collocation*

catégorie grammaticale : cf. *nature*

collocatif : cf. *collocation*

collocation : L'énoncé *AB* (ou *BA*) formé des items lexicaux *A* et *B* est une collocation si, pour produire cette expression, le locuteur sélectionne *A* librement d'après son sens alors qu'il sélectionne *B* pour exprimer un autre sens en fonction de *A* [Polguère, 2003]. On appelle *A* *base de la collocation* et *B* *collocatif*. On peut citer comme exemples de collocations en français : ‹*tir*›$_{[=A]}$ ‹*nourrit*›$_{[=B]}$, ‹*peur*›$_{[=A]}$ ‹*bleue*›$_{[=B]}$, ‹*forte*›$_{[=B]}$ ‹*fièvre*›$_{[=A]}$,‹*dormir*›$_{[=A]}$ ‹*profondément*›$_{[=B]}$.

componentielle (linguistique) : La linguistique componentielle suppose l'existence d'une atomisation de la signification, c'est-à-dire que le sens d'un terme n'est plus considéré comme primitif, mais peut être décomposé en éléments de sens plus petits appelés suivant les diverses écoles : sèmes, traits sémantiques, atomes de sens, primitives, ... Par exemple, ‹*Ferrari*› peut être construit à partir des idées *VOITURE*, *ROUGE*, *RAPIDE*.

compositionnalité sémantique (principe de) : D'après le principe de compositionnalité sémantique, « *le tout est calculable à partir du sens de ses parties* ». Ainsi, un énoncé est directement calculable (dans sa composition lexicale et sa structure syntaxique) à partir de la combinaison du sens de chacun des ses constituants [Polguère, 2003]. Par exemple, le sens d'une phrase comme « *L'enfant voit la mer.* » est calculable à partir :
- des items lexicaux ‹*le*›, ‹*enfant*›, ‹*voir*›, ‹*la*›, ‹*mer*› ;
- des règles syntaxiques et morphologiques du français utilisées dans la phrase.

constituants (d'une phrase) : En syntaxe, les constituants de la phrases sont les unités linguistiques qui composent la phrase : les *mots* et les *syntagmes*.

co-texte : Le co-texte d'un mot est l'ensemble des mots qui constituent son entourage qu'ils apparaissent avant ou après dans l'énoncé. Par exemple, dans la phrase « *Une légère pente aboutissait à un fond accidenté.* », le co-texte de ‹‹pente›› est constitué des mots ‹‹une››, ‹‹légère››, ‹‹aboutissait››, ‹‹à››, ‹‹un››, ‹‹fond››, ‹‹accidenté››. Le co-texte est parfois appelé *contexte linguistique*.

contexte : Au niveau pragmatique, la situation dans laquelle se déroule l'énoncé.

contexte linguistique : cf. *co-texte*

désambiguïsation sémantique : Opération qui consiste à résoudre l'ensemble des ambiguïtés posées par le sens dans un texte : ambiguïté lexicale, résolution d'anaphore, d'ellipse, ...

désambiguïsation lexicale : Opération qui consiste à trouver un sens préférentiel ou une combinaison de sens préférentiel pour les mots d'un énoncé. Ainsi, dans la phrase « *L'avocat est véreux.* » deux combinaisons peuvent être considérées comme préférentielles : d'un côté *avocat*/**personne** et *véreux*/**crapuleux** et de l'autre *avocat*/**fruit** et *véreux*/**ver**.

diachronie : cf. *synchronie*

distributionnelle (linguistique) : La linguistique distributionnelle considère que le sens d'un terme peut être donné par l'ensemble de ses contextes. Par exemple, la sémantique de l'item ‹*lait*› peut être décrite grâce à la liste {‹*vache*› ‹*bouteille*› ‹*fromage*› ‹*yaourt*›, ... }

ellipse : Omission volontaire d'une partie de phrase non nécessaire à la compréhension de l'ensemble. Exemple : « *Elle marche vite, moi (je marche) lentement.* ».

énoncé : Séquence de termes et de phrases en langue naturelle prononcée (appelée alors paroles ou énoncé oral) ou écrite (texte ou énoncé écrit) constituant un tout.

forme canonique : La *forme canonique* d'un mot est la forme de ce mot telle qu'on peut la trouver comme entrée d'un dictionnaire par opposition à la forme fléchie. Par définition, un item lexical est donc toujours dans une forme canonique. Traditionnellement, suivant la nature de l'item, une forme particulière est choisie :
- *verbe* : à l'infinitif
- *nom* : au singulier (s'il existe)
- *adjectif* : au masculin singulier

exemples : ‹*sourire*›, ‹*souris*›, ‹*orgue*›, ‹*orgues*›, ‹*petit*›, ...
Notons que pour les mots invariables, formes fléchie et canonique sont identiques.

forme fléchie : Les mots sous forme *fléchie* comportent un radical et une ou plusieurs désinences. Les désinences sont les morphèmes porteurs des indications de nombre et de genre pour les noms, adjectifs et déterminants, de personnes, de temps et de mode pour les verbes. Ainsi, ‹‹lisions›› est constitué du radical *lis-* issu de l'item ‹*lire*›, de la désinence temporelle *-i-* et de la désinence personnelle *-ons* [Lehmann & Martin-Berthet, 1998] tandis que ‹‹rattes›› est lui formé par *rat* (radical) + *te* (féminin) + s (pluriel). En aucun cas, la flexion ne modifie donc la catégorie syntaxique.

fréquence d'un terme : On appelle fréquence d'un terme (*term frequency*) le nombre de fois où ce terme apparaît, on parle aussi du *nombre d'occurrences* ou de la *fréquence d'occurrence*.

fréquence d'occurrences : cf. *fréquence d'un terme*.

gloses : Informations que l'on trouve dans certains dictionnaires (en particulier de traduction ou de synonymie) pour préciser le sens d'un terme.

holonymie : cf. *méronymie*

homonymie : cf. *polysémie*

hyperonymie : cf. *hyponymie*.

hyponymie : La relation d'hyponymie est la relation hiérarchique qui lie un hyponyme à un item plus général l'hyperonyme. La relation d'hyperonymie est la relation inverse. Exemples : ‹*chat*›\‹*animal*›, ‹*voilier*›\‹*bateau*›, ‹*bateau*›\‹*véhicule*›, ‹*rose*›\‹*fleur*›.

infixe : cf. *morphème*

item lexical : Un item lexical est une suite de caractères formant une unité sémantique et pouvant constituer une entrée de dictionnaire. Par exemple, ‹*voiture*› tout comme ‹*pomme de terre*›, ‹*moulin à vent*› et même des termes techniques comme ‹*pompe bivalve à échappement central*› sont des items lexicaux.

langage naturel (ou langue naturelle) : langage tel qu'il est parlé quotidiennement par les êtres humains et qu'ils ont créé de façon émergente (comme le français, l'anglais, le chinois ou le malais) par opposition aux langages artificiels construits de façon consciente par l'être humain et utilisés en logique, mathématiques ou informatique.

lexie : Objet lexical correspondant à un sens d'un item lexical dans un certain dictionnaire. Ainsi, dans [Larousse, 2004] pour ‹*canard*›, on peut trouver 7 lexies (**oiseau**, **fausse note**, **sucre**, **aviation**, **fausse nouvelle**, **journal**) tandis que dans [Robert, 2000] on en trouve 6 (**oiseau**, **marche**, **sucre**, **fausse note**, **fausse nouvelle**, **terme d'affection**)

locution : Les locutions sont les items lexicaux constitués d'un groupe de mots figé. On distingue :
- les locutions *adjectivales* : ‹*mezza voce*›,‹*quel que*›, ...
- les locutions *adverbiales* : ‹*grosso modo*›, ‹*en partance*›, ...
- les locutions *prépositionnelles* : ‹*en ce qui concerne*›, ‹*au profit de*›, ...
- les locutions *nominales* : ‹*pomme de terre*›, ‹*moulin à vent*›, ...
- les locutions *verbales* : ‹*aller à l'encontre de*›, ‹*se faire marcher sur les pieds*›, ‹*retirer une épine du pied*›, ...

méronymie : La relation de méronymie est la relation hiérarchique qui lie la partie au tout. Un des éléments de la relation est une partie de l'autre élément. Les deux relations sont symétriques c'est-à-dire que le tout est l'holonyme de la partie tandis que la partie est le méronyme du tout.

monosème : cf. *monosémie*

monosémie : Caractéristique des items qui n'ont qu'un seul sens. Ainsi, des termes comme ‹*calame*›, ‹*cajou*›, ‹*neuroleptique*›, ‹*polyamide*› semblent n'avoir qu'une seule signification. On parle alors d'item *monosémique* ou *monosème*

monosémique : cf. *monosémie*

morphologie : Partie de la linguistique et du TALN qui s'intéresse aux morphèmes des mots.

morphologie d'un item lexical : La morphologie d'un item lexical regroupe les informations concernant sa nature et son genre. Ainsi, ‹*courir*› est un *verbe*, ‹*souris*› est un *nom féminin* , ‹*orgues*› est un *nom masculin pluriel*...

morphème : Les morphèmes sont les unités minimales significatives qui constituent les mots. Par exemple, le mot ‹‹fleurs›› est constitué de deux morphèmes : le radical (ou base) correspondant à l'item ‹*fleur*› et du suffixe marquant le pluriel *s*. Il existe deux types de morphèmes :

- les *morphèmes lexicaux* qui correspondent aux items lexicaux ou à une légère variante ;
- les *morphèmes grammaticaux*, autrement appelés *affixes*. Situé avant le radical, un affixe est dit *préfixe*, après le radical, il est dit *suffixe* et dans le radical, *infixe*.

mot : Un mot est la forme fléchie d'un item lexical.

nature : Les termes de même nature se caractérisent par la possibilité de les substituer syntaxiquement. Elle est constituée, entre autres, des *verbes*, des *adjectifs*, des *noms*, des *adverbes*. La nature est aussi appelée parfois *catégorie grammaticale* ou *partie du discours*.

nombre d'occurrences : cf. *fréquence d'un terme*

objet lexical : La classe des *objets lexicaux* regroupe l'ensemble des objets du lexique : *item lexical*, *lexie*, *acception*.

objet linguistique : La classe des *objets linguistiques* regroupe l'ensemble des objets de la langue : les *objets lexicaux* mais aussi les *segments textuels*.

oxymore : On appelle *oxymore* ou *oxymoron* le rapprochement de termes qui semblent contradictoires comme c'est le cas pour ‹*mort-vivant*› ou ‹*clair-obscur*›.

paradigmatique (plan) : Le plan paradigmatique, ou plan du sens, est le plan dans lequel les termes sont unis par leur sens à l'intérieur du lexique. Ces liens sont des relations sémantiques comme la synonymie, l'antonymie ou l'hyperonymie. Paradigmatiquement, ‹*peur*› est, par exemple, relié à ‹*peureux*›, ‹*frayeur*›, ‹*calme*›, ... Le plan paradigmatique est le plan orthogonal au plan syntagmatique.

partie du discours : cf. *nature*. En Anglais, part of speech (POS)

préfixe : cf. *morphème*

polysème : cf. *polysémie*

polysémie : Caractéristique des items qui ont plusieurs sens entre lesquels il existe un lien.

On parle alors d'item *polysémique* ou *polysème*. Habituellement, on distingue de la polysémie, l'*homonymie* qui est la caractéristique des items qui ont plusieurs sens entre lesquels il n'existe pas de lien. Dans cette thèse nous utilisons le qualificatif de polysémique pour un terme sans chercher à distinguer s'il s'agit d'un vrai cas de polysémie ou d'un cas d'homonymie.

polysémique : cf. *polysémie*

phonèmes : Segments phoniques minimaux dont la fonction est de constituer les signifiants et de les distinguer entre eux dans une langue parlée donnée. les sons interchangeables dans une langue sans changer le sens d'un énoncé ne forme qu'un seul phonème. Le français, par exemple, comprend 36 phonèmes (16 voyelles et 20 consonnes). Les phonèmes sont notés habituellement par des lettres placées entre des barres obliques : /a/, /â/, /an/, /b/, /ch/, /d/. Les items ‹*pou*› [pu] et ‹*cou*› [ku] diffèrent par le phonème /p/ et /k/

pragmatique : La pragmatique est l'étude du sens des énoncés en contextes c'est-à-dire l'ensemble des significations que peut lui donner un être humain. Le niveau pragmatique de la compréhension de textes consiste ainsi à découvrir le bon sens d'un énoncé en fonction des conditions situationnelles et contextuelles dans lesquelles il apparaît. La pragmatique s'occupe en particulier des problèmes d'anaphore, de subjectivité.

relations sémantiques externes : Les relations sémantiques externes (ou relations sémantiques lexicales) sont les liens sémantiques qui relient les items lexicaux entre eux. Les principales relations sémantiques externes sont la synonymie, l'antonymie, l'hyperonymie/hyponymie, l'holonymie/méronymie.

relations sémantiques internes : Les relations sémantiques internes sont les liens sémantiques qui relient les différentes acceptions d'un même item lexical.

relations sémantiques lexicales : cf. *relations sémantiques externes*

segment textuel : La classe des *segments textuels* regroupe les portions d'un texte ayant une unité sémantique *mots*, *syntagme*, *phrase*, *paragraphe*, *texte*, ...

sémantique : La sémantique est l'étude du sens des énoncés.

sens : Le sens d'une expression linguistique est la propriété qu'elle partage avec toutes ses paraphrases.

suffixe : cf. *morphème*

synonymie : La synonymie est la relation sémantique qu'il existe entre deux items lexicaux qui diffèrent sur leur forme mais expriment le même sens ou un sens très proche.

synchronie : L'étude synchronique (ou descriptive) de la langue s'intéresse à décrire la langue sans tenir compte des facteurs temps contrairement à l'étude diachronique (ou historique) qui elle porte sur l'évolution de la langue (étymologie).

syntagmatique (plan) : Le plan syntagmatique est le plan dans lequel les termes sont unis à l'intérieur de la phrase. Il s'agit du plan sur lequel s'exercent les phénomènes de collocations. Ainsi, ‹*peur*› est relié à ‹*grande*›, ‹*énorme*›, ‹*avoir la peur de sa vie*›, ... Le plan syntagmatique est le plan orthogonal au plan paradigmatique.

syntagme : cf. *syntaxe*

syntaxe : La syntaxe étudie la manière dont les mots se combinent pour former des syntagmes

et les syntagmes se combinent pour former des phrases.

TALN (Traitement Automatique du Langage naturel (ou des langues naturelles) : domaine d'étude des techniques d'analyse (compréhension) et de génération (production) automatiques d'énoncés oraux ou écrits

taxinomie (ou taxonomie) : Le terme taxinomie signifie littéralement, la « *loi du rangement* ». Il désigne une classification systématique d'un ensemble d'éléments dans un domaine précis (taxinomie des êtres vivants, taxinomie de Flynn sur les architectures informatiques, ...) ou général. Ce terme désigne aussi la science qui vise à établir de telles classifications.

Index

Bibliographie

[Aloulou, 2003] Chafik Aloulou. « Analyse Syntaxique de l'Arabe : Le Système MASPAR ». Dans les actes de *RECITAL'2003*, pp 419–428, Batz-sur-Mer, France, juin 2003. pages 179

[Bailly, 1947] René Bailly. *Dictionnaires des synonymes.* Larousse, Paris, 1947. pages 144

[Bangalore, 1997] Srinivas Bangalore. « *Complexity of lexical descriptions and its relevance to partial parsing.* ». PhD thesis, University of Pennsylvannia., 1997. pages 24

[Bangha, 2003] Kornél Robert Bangha. « *La place des connaissances lexicales face aux connaissances du monde dans le processus d'interprétation des énoncés* ». PhD thesis, Université de Montréal, Montréal, Québec, Canada, 2003. pages 196

[Béchade, 1992] Hervé Béchade. *Phonétique et morphologie du français moderne et contemporain.* Presses Universitaires de France, 1992. pages 149

[Bellot & El-Bèze, 2001] Patrice Bellot et Marc El-Bèze. « Classification et segmentation de textes par arbres de décision ». *Technique et Science Informatiques (TSI)*, pp 397–424, 2001. pages 224

[Berge, 1967] Claude Berge. *Théorie des graphes et ses applications.* Dunod, Paris, France, 1967. pages 145

[Bernard, 2000] Gilles Bernard. « *Intelligence artificielle, linguistique expérimentale, cognition : Principes et mécanismes d'économie des représentations dans la modélisation de systèmes linguistiques* ». Mémoire d'Habilitation à Diriger des Recherches, Université Paris VIII, Paris, France, 2000. pages 35

[Besançon, 2001] Romaric Besançon. « *Intégration de connaissances syntaxiques et sémantiques dans les représentations vectorielles de texte* ». Thèse de doctorat, École Polytechnique Fédérale de Lausanne, Laboratoire d'Intelligence Artificielle, 2001. pages 57, 60, 66

[Bestgen, 2004] Yves Bestgen. « Analyse sémantique latente et segmentation automatique des textes ». Dans les actes de *7èmes Journées internationales d'Analyse statistique des Données Textuelles*, pp 171–181, Louvain-la-Neuve, Mars 2004. pages 31, 224

[Bénac, 1956] Henri Bénac. *Le Dictionnaire Des Synonymes.* Hachette, Paris, 1956. pages 144

[Boitet & Seligman, 1994] Christian Boitet et Marc Seligman. « The "whiteboard" architecture : a way to Integrate heterogenous components of

NLP System ». Dans les actes de *COLING'1994 : 15th International Conference on Computational Linguistics*, volume 1/2, pp 426–430, Japon, August 1994. pages 177

[Boitet, 2000] Christian BOITET. « Handling Texts and Corpuses in Ariane-G5, a complete environment for multilingual MT ». Dans les actes de *ACIDCA'2000, Corpora and Natural Language Processing*, pp 7–11, 2000. pages 25

[Bonabeau & Théraulaz, 2000] Éric BONABEAU et Guy THÉRAULAZ. « L'intelligence en essaim ». *Pour la science*, pp 66–73, 2000. pages 229

[Boser *et al.*, 1992] Bernhard E. BOSER, Isabelle GUYON, et Vladimir VAPNIK. « A Training Algorithm for Optimal Margin Classifiers ». Dans les actes de *Computational Learing Theory*, pp 144–152, Pittsburgh, 1992. pages 72

[Chauché, 1984] Jacques CHAUCHÉ. « Un outil multidimensionnel de l'analyse du discours. ». Dans les actes de *COLING'1984 : 10th International Conference on Computational Linguistics*, pp 11–15, Stanford University, California, 1984. pages 25

[Chauché, 1990] Jacques CHAUCHÉ. « Détermination sémantique en analyse structurelle : une expérience basée sur une définition de distance ». *TAL Information*, pp 17–24, 1990. pages 49, 50

[Chauché *et al.*, 2003] Jacques CHAUCHÉ, Violaine PRINCE, Simon JAILLET, et Maguelonne TEISSEIRE. « Classification automatique de textes à partir de leur analyse syntaxico-sémantique ». Dans les actes de *TALN 2003*, volume 1, pp 55–64, Batz-Sur-Mer, France, Juin 2003. pages 66, 68

[Chazaud, 1979] Henri-Bertaud du CHAZAUD. *Dictionnaire des synonymes.* Éditions Le Robert, Paris, 1979. pages 144, 170

[Church, 1988] Kenneth Ward CHURCH. « A stochastic parts program and noun phrase parser for unrestricted text ». Dans les actes de *2nd Conference on Applied Natural Language Processing*, pp 136–143, 1988. pages 25

[Claveau, 2003] Vincent CLAVEAU. « *Acquisition automatique de lexiques sémantiques pour la recherche d'information* ». Thèse de doctorat, Université de Rennes I, décembre 2003. pages 201

[Collins & Quillian, 1969] Alan COLLINS et Ross QUILLIAN. « Retrivial time from semantic memory ». *Verbal learning and verbal behaviour*, pp 240–247, 1969. pages 1, 35

[Collins & Quillian, 1970] Alan COLLINS et Ross QUILLIAN. « Does category size affect categorization time ? ». *Verbal learning and verbal behaviour*, pp 432–438, 1970. pages 35

[Collins, 1997] Michael COLLINS. « Three generative, lexicalised models for statistical parsing ». Dans les actes de *the Annual Meeting of the Association of Computational Linguistics*, Madrid, 1997. pages 25

[Conrad, 1972] Carol CONRAD. « Cognitive economy in semantic memory ». *Journal of Experimental Psychology*, pp 149–154, 1972. pages 1, 39, 40

[Cori & Léon, 2002] Marcel Cori et Jacqueline Léon. « La constitution du TAL, Étude historique des dénominations et des concepts ». *Traitement Automatiques des Langues (TAL)*, pp 21–55, 2002. pages 17

[Correard *et al.*, 2001] Marie-Helene Correard, Valerie Grundy, et Jean-Benoit Ormal-Grenon. *Oxford-Hachette : French Dictionary*. Oxford University Press, 2001. pages 264

[Damasio & Damasio, 1992] Antonio Damasio et Hanna Damasio. « Le cerveau et le langage ». *Pour la science*, novembre 1992. pages 20, 128, 163, 164

[Damasio, 1995] Antonio Damasio. *L'erreur de Descartes, la raison des émotions*. Odile Jacob, 1995. pages 20

[Deerwester *et al.*, 1990] Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, et Richard A. Harshman. « Indexing by Latent Semantic Analysis ». *Journal of the American Society of Information Science*, pp 391–407, 1990. pages 31

[Delorme, 2003] Jean-Michel Delorme. « *Contribution à la réalisation d'un système de traduction automatique construit autour du moteur d'analyse et de génération transductionnel SYGMART* ». Diplôme d'ingénieur CNAM, Conservatoire National des Arts et Métiers, Montpellier Languedoc-Roussillon, 2003. pages 263, 264, 266

[Deneubourg *et al.*, 1989] Jean-Louis Deneubourg, Gross, Franks, et Pasteels. « The blind leading the blind : Modeling chemically mediated army ant raid patterns ». *Journal of Insect Behavior*, pp 719–725, 1989. pages 229

[Dorigo & Gambardella, 1997] Marco Dorigo et Luca Gambardella. « Ant colony system : A cooperative learning approach to the traveling salesman problem ». *IEEE Transactions on Evolutionary Computation*, pp 53–66, 1997. pages 230

[Dorigo & Stützle, 2004] Dorigo et Stützle. *Ant Colony Optimization*. MIT-Press, 2004. pages 230

[Drogoul, 1993] Alexis Drogoul. « When ants play chess (or can strategies emerge from tactical behaviors) ». Dans les actes de *Maamaw'1993*, 1993. pages 230

[Dutoit & Nugues, 2002] Dominique Dutoit et Pierre Nugues. « A Lexical Network and an Algorithm to Find Words from Definitions ». Dans les actes de Frank van Harmelen, , *ECAI2002, Proceedings of the 15th European Conference on Artificial Intelligence*, pp 450–454, Lyon, July 21-26 2002. IOS Press, Amsterdam. pages 48

[Dutoit, 1992] Dominique Dutoit. « A set-theoric approach to Lexical Semantics ». Dans les actes de *COLING'1992 : 14th International Conference on Computational Linguistics*, pp 539–545, Nantes, France, 1992. pages 49

[Dutoit, 2000] Dominique DUTOIT. « *Quelques opérations Sens -> texte et texte -> Sens utilisant une sémantique linguistique univerliste a priori* ». Thèse de doctorat, Université de Caen, Novembre 2000. pages 47

[Éco, 1988] Umberto ÉCO. *Le signe : histoire et analyse d'un concept.* Livre de poche. Labor, 1988. pages 3, 43, 44, 258

[Erman & Lesser, 1975] L.D. ERMAN et R. LESSER. « A multi-level organization for problem solving using many, diverse, cooperating sources of knowledge ». Dans les actes de *IJCAI'1975*, volume 2, pp 483–390, 1975. pages 179

[Erman *et al.*, 1980] Lee D. ERMAN, Frederick HAYES-ROTH, Victor R. LESSER, et D. Raj REDDY. « The Hearsay-II Speech-Understanding System : Integrating Knowledge to Resolve Uncertainty ». *ACM Comput. Surv.*, pp 213–253, 1980, ACM Press. pages 179

[Ferber, 1995] Jacques FERBER. *Les systèmes multi-agents. Vers une intelligence collective.* InterEditions, 1995. pages 176, 178, 179

[Gala Pavia, 2003a] Núria GALA PAVIA. « *Un modèle d'analyseur syntaxique robuste fondé sur la modularité et la lexicalisation de ses grammaires* ». Thèse de doctorat, Université de Paris-Sud, Mars 2003. pages 225, 227

[Gala Pavia, 2003b] Núria GALA PAVIA. « Une méthode non supervisée d'apprentissage sur le Web pour la résolution d'ambiguïtés structurelles liées au rattachement prépositionnel. ». Dans les actes de *TALN'2003*, pp 353–358, Batz-sur-Mer, France, juin 2003. pages 225

[Genest, 2000] David GENEST. « *Extension du modèle des graphes conceptuels pour la recherche d'informations* ». Thèse de doctorat, Université Montpellier II, 2000. pages 38

[Girault, 1999] François GIRAULT. « Une architecture d'anticipation par réalité augmentée ». Dans les actes de *JFIAD'99 : Journées Francophones sur l'Intelligence Artificielle Distribuée*, pp 253–264, 1999. pages 182

[Glassner, 2001] Jean-Jacques GLASSNER. « L'invention de l'écriture sumérienne : système de notation ou langage ? ». *Les actes de lecture*, pp 94–103, Mars 2001. pages 164

[Goodglass & Baker, 1976] Harold GOODGLASS et Errol BAKER. « Semantic field, naming and auditory comprehension in aphasy ». *Brain and Language*, pp 359–374, 1976. pages 35

[Grabar & Zweigenbaum, 1999] Natalia GRABAR et Pierre ZWEIGENBAUM. « Acquisition automatique de connaissances morphologiques sur le vocabulaire médical. ». Dans les actes de *Actes de la conférence Traitement Automatique du Langage Naturel (TALN'1999)*, Cargese, Juillet 1999. pages 23

[Grefenstette, 1994] Gregory GREFENSTETTE. « Corpus-derived first, second and third-order word affinities ». Dans les actes de *6th EURALEX*, Amsterdam, 1994. pages 31

[Greimas, 1986] Algirdas Julien Greimas. *Sémantique Structurale*. PUF, 1986. pages 45, 102

[Guisot, 1864] François Guisot. *Dictionnaire universel des synonymes de la langue française (7ème édition)*. Didier et Cie, libraires-éditeurs, Paris, 1864. pages 144

[Gutknecht & Ferber, 1999] Olivier Gutknecht et Jacques Ferber. « Vers une méthodologie organisationnelle de conception de systèmes multi-agents ». Dans les actes de *JFIADSMA'99 : Actes des 7èmes Journées Francophones d'Intelligence Artificielle et Systèmes Multi-Agents*, pp 93–104, 1999. pages 186

[Haiman, 1980] John Haiman. « Dictionaries and encyclopedias ». *Lingua*, pp 329–357, 1980. pages 196

[Hao *et al.*, 1999] Jin-Kao Hao, Philippe Galinier, et Michel Habib. « Méthaheuristiques pour l'optimisation combinatoire et l'affectation sous contraintes ». *Revue d'Intelligence Artificielle*, pp 263–324, 1999. pages 230

[Harris *et al.*, 1989] Zellig S. Harris, Michael Gottfried, Thomas Ryckman, Paul Mattick Jr., Anne Daladier, T.N. Harris, et S. Harris. *The form of Information in Science, Analysis of Immunology Sublanguage*, volume 104 de *Boston Studies in the Philosophy of Science*. Kluwer Academic Publisher, Dordrecht, 1989. pages 28, 57

[Hawkins, 1896] N. Hawkins. *New Catechism of Electricity, a pratical treatise*. Theo. Audel & Co., New-York, 1896. pages 165

[Hearst, 1992] Marti Hearst. « Automatic Acquisition of Hyponyms from Large Text Corpora ». Dans les actes de *COLING'1992 : 14th International Conference on Computational Linguistics*, pp 539–545, Nantes, France, 1992. pages 201

[Heylen *et al.*, 1994] Dirk Heylen, Kerry G. Maxwell, et Marc Verhagen. « Lexical functions and machine translation ». Dans les actes de *COLING'1994 : 15th International Conference on Computational Linguistics*, volume 1, pp 1240–1244, Kyoto, Japan, 1994. pages 227

[Hirschman, 1986] Hirschman. « *Analyzing Language in Restricted Domains. Sublanguage Description and Processing* », Chapitre Discovering sublanguage structure, pp 211–234. Grishman and Kittredge, 1986. pages 57

[Hjelmlev, 1968] Louis Hjelmlev. *Prolégolème à une théorie du langage*. éditions de minuit, 1968. pages 43

[Houde *et al.*, 2002] Olivier Houde, Bernard Mazoyer, et Nathalie Tzourio-Mazoyer. *Cerveau et psychologie*. Presses Universitaires de France, 2002. pages 164

[Hutchins & Somers, 1992] W. John Hutchins et Harold L. Somers. *An introduction to machine translation*. Academic Press Limited, 1992. pages 258

[Jaillet, 2005] Simon Jaillet. « *Catégorisation automatique de documents textuels : d'une représentation basée sur les concepts aux motifs séquentiels* ». Thèse de doctorat, Université Montpellier II, 2005. pages 58, 68, 71, 73

[Jalabert & Lafourcade, 2004a] Fabien Jalabert et Mathieu Lafourcade. « Classification automatique de définitions en sens ». Dans les actes de *Traitement Automatique du Langage Naturel (TALN'2005)*, Fèz, Maroc 2004. pages 189, 338

[Jalabert & Lafourcade, 2004b] Fabien Jalabert et Mathieu Lafourcade. « Nommage sens à l'aide de vecteurs conceptuels. ». Dans les actes de *Reconnaissance des Formes et Intelligence Artificielle (RFIA'2004*, volume 2, pp 539–547, Toulouse, Janvier 2004. pages 271

[Jalabert, 2003] Fabien Jalabert. « Catégorisation de définitions et nommage de sens ». Mémoire de dea, Université Montpellier II, LIRMM, Juillet 2003. pages 189

[Juola & Atkinson, 1971] James F. Juola et Richard C. Atkinson. « Memory scanning for word versus categories ». *Journal of verbal learning and verbal behaviour*, pp 449–452, 1971. pages 35

[Kahlmann, 1975] André Kahlmann. « *Traitement automatique d'un dictionnaire de synonymes* ». Thèse de doctorat, Université de Stockholm, Stockholm, 1975. pages 145

[Kerbrat-Orecchioni, 1986] Catherine Kerbrat-Orecchioni. *L'implicite*. Colin, 1986. pages 27

[Kintsch, 2000] Walter Kintsch. « Metaphor comprehension : A computational theory ». *Psychonomic Bulletin and Review*, 2000. pages 31

[Kirkpatrick, 1987] Betty Kirkpatrick, . *Roget's Thesaurus of English Words and Phrases*. Penguin books, London, 1987. pages 52

[Kleiber, 1990] Georges Kleiber. *La sémantique du prototype*. Presses Universitaires de France, 1990. pages 27

[Lafaye, 1841] Pierre-Benjamin Lafaye. *Dictionnaires des synonymes de la langue française*. Librairie Hachette, librairie royale de France, Paris, 1841. pages 144

[Lafourcade & Guinand, 2005] Mathieu Lafourcade et Frédéric Guinand. « Ants for Natural Language Processing ». Rapport interne LIRMM, 2005. pages 230

[Lafourcade & Prince, 2001a] Mathieu Lafourcade et Violaine Prince. « Synonymies et vecteurs conceptuels ». Dans les actes de *TALN'2001*, Tours, France, Juillet 2001. pages 93, 95, 101, 125

[Lafourcade & Prince, 2001b] Mathieu Lafourcade et Violaine Prince. « Synonymy and conceptual vectors ». Dans les actes de *NLPRS'2001*, pp 127–134, Tokyo, Japon, Novembre 2001. pages 95

[Lafourcade & Prince, 2003] Mathieu Lafourcade et Violaine Prince. « Mixing Semantic Networks and Conceptual Vectors : the Case of Hyperonymy ». Dans les actes de *ICCI-2003 (2nd IEEE International Conference on Cognitive Informatics)*, pp 121–128, South Bank University, London, UK, Août 2003. pages 211

[Lafourcade & Prince, 2004] Mathieu LAFOURCADE et Violaine PRINCE. « Modélisation de l'hyperonymie via la combinaison de réseaux sémantiques et de vecteurs conceptuels ». Dans les actes de *JADT 2004 : 7es Journées internationales d'Analyse statistique des Données Textuelles*, volume 2, pp 692–699, Louvain-la-Neuve, Belgique, Mars 2004. pages 3, 203, 206, 211, 215

[Lafourcade, 1994] Mathieu LAFOURCADE. « *Génie logiciel pour le génie linguiciel* ». Thèse de doctorat, Université de Joseph Fourier - Grenoble I, Décembre 1994. pages 177

[Lafourcade, 2003] Mathieu LAFOURCADE. « Algorithmes fourmis et TALN. ». http://www.lirmm.fr/~lafourca/ML-research/directions/TALN-algo-fourmi/TALN-algo-fourmi.html, 2003. pages 185, 230

[Lafourcade *et al.*, 2002] Mathieu LAFOURCADE, Violaine PRINCE, et Didier SCHWAB. « Vecteurs conceptuels et structuration émergente de terminologies ». *Traitement Automatiques des Langues (TAL)*, pp 43–72, 2002. pages 95

[Lafourcade *et al.*, 2004] Mathieu LAFOURCADE, Frederic RODRIGO, et Didier SCHWAB. « Low Cost Automated Conceptual Vector Generation from Mono and Bilingual Resources ». Dans les actes de *PAPILLON-2004*, Grenoble, France, Août 2004. pages 257

[Landauer & Freedman, 1968] Thomas LANDAUER et Jonathan FREEDMAN. « Information Retrivial from long term memory : category size and recognition time ». *Journal of verbal learning and verbal behaviour*, pp 291–331, 1968. pages 35

[Larousse, 1971] LAROUSSE, . *Grand Larousse de la langue française*. Larousse, Paris, 1971. pages 144

[Larousse, 1991] LAROUSSE, . *Grand Larousse Universel*. Larousse, Paris, 1991. pages 100

[Larousse, 1992] LAROUSSE, . *Thésaurus Larousse - des idées aux mots, des mots aux idées*. Larousse, 1992. pages 1, 3, 51, 52, 53, 57, 68, 75, 76, 81, 115, 152, 206, 208, 264, 341

[Larousse, 2001a] LAROUSSE, . *Dictionnaire des synonymes*. Larousse, 2001. pages 93

[Larousse, 2001b] LAROUSSE, . *Le Petit Larousse Illustré 2001*. Larousse, 2001. pages 81

[Larousse, 2004] LAROUSSE, . *Le Petit Larousse Illustré 2004*. Larousse, 2004. pages 2, 21, 77, 92, 99, 120, 146, 147, 152, 154, 167, 169, 205, 341, 345

[Lebarbé, 2001] Thomas LEBARBÉ. « Vers une plate-forme multi-agents pour l'exploration et le traitement linguistique ». Dans les actes de *TALN'2001*, Tours, France, Juillet 2001. pages 181

[Lebarbé, 2003] Thomas LEBARBÉ. « *Hiérarchie inclusive des unités linguistiques en analyse syntaxique coopérative. Le segment, unité intermédiaire entre chunk et phrase dans le traitement linguistique par système multi-agents.* ». Thèse de doctorat, Université de Caen - Basse-Normandie, Caen, France, Mai 2003. pages 179, 181

[Lebarbé, 2004] Thomas Lebarbé. « HACTAR : coopération entre unités, fonctions et domaines linguistiques par une plateforme SMA ». Dans les actes de *Journée d'Etude ATALA Agental "Agents et Langue"*, pp 7–13, Paris, France, 2004. ATALA, ATALA. pages 179, 181

[Lecerf, 1997] Christophe Lecerf. *Une leçon de piano ou la double boucle de l'apprentissage cognitif.*, volume 3-1997. Université Paris 8 - Vincenne-Saint-Denis, Université Paris 8, Vincennes Saint-denis, Mars 1997. revue Travaux et Documents. pages 3, 139, 157, 171, 173, 174, 175, 176

[Lehmann & Martin-Berthet, 1998] Alise Lehmann et Françoise Martin-Berthet. *Introduction à la lexicologie. Sémantique et morphologie.* Dunod, Paris, 1998. pages 23, 345

[Lemaire & Dessus, 2003] Benoît Lemaire et Philippe Dessus. « Modèles cognitifs issus de l'analyse sémantique latente ». *Cahiers Romans de sciences cognitives*, pp 55–74, 2003. pages 31

[Le Ny, 1979] Jean-François Le Ny. *La sémantique psychologique.* PUF, Paris, 1979. pages 20, 43, 44

[Lesser & Erman, 1977] R. Lesser et L.D. Erman. « A retrospective view of the HEARSAY-II architecture ». Dans les actes de *IJCAI'1977*, pp 790–800, 1977. pages 179

[Léon & Millon, 2005] Stéphanie Léon et Chrystel Millon. « Acquisition semi-automatique de relations lexicales bilingues (français-anglais) à partir du Web ». Dans les actes de *RECITAL'2005*, volume 1, page 595, Dourdan, France, Juin 2005. pages 271

[Lyons, 1977] John Lyons. *Semantics.* Cambridge University Press, 1977. pages 101

[Mangeot-Lerebours, 2001] Mathieu Mangeot-Lerebours. « *Environnements centralisés et distribués pour lexicographes et lexicologues en contexte multilingue* ». Thèse de doctorat, Université Joseph Fourier, 2001. pages 259

[Mangeot-Lerebours *et al.*, 2003] Mathieu Mangeot-Lerebours, Gilles Sérasset, et Mathieu Lafourcade. « Construction collaborative d'une base lexicale multilingue : Le projet Papillon ». *TAL (Traitement Automatique des langues) : Les dictionnaires électroniques*, pp 151–176, 2003. pages 41

[Manguin, 2004] Jean-Luc Manguin. « Transitivité partielle de la synonymie : application aux dictionnaires de synonymes ». *CORELA*, 2004. *accessible en ligne à l'adresse* http://edel.univ-poitiers.fr/corela/document80.html. pages 145

[Manguin *et al.*, 2004] Jean-Luc Manguin, Jacques François, Rembert Eufe, Ludwig Fesenmeier, et Corinne Ozouf. « Le dictionnaire électronique des synonymes du CRISCO : un mode d'emploi à trois niveaux. ». *Les Cahiers du CRISCO*, Juillet 2004. pages 144, 145

[Mel'čuk, 1988] Igor Mel'čuk. *Dictionnaire explicatif et combinatoire du français contemporain*, volume 2. Les presses de L'université de Montréal, Montréal, 1988. pages 1, 33, 198, 307, 333

[Mel'čuk, 1994] Igor Mel'čuk. « *TA-TAO, Recherches de pointe et Applications immédiates* », Chapitre Fonctions lexicales dans le traitement du langage naturel, pp 193–219. AUF, Fiches du Monde Arabe, 1994. pages 227

[Mel'čuk *et al.*, 1995] Igor Mel'čuk, André Clas, et Alain Polguère. *Introduction à la lexicologie explicative et combinatoire*. Duculot, 1995. pages 33, 103, 198, 307, 333

[Menézo *et al.*, 1996] J. Menézo, D. Genthial, et J. Courtin. « Reconnaissances pluri-lexicales dans CELINE, un système multi-agents de detection et correction des erreurs ». Dans les actes de *NLP+IA96 : International Conference on Natural Language Processing and Industrial Applications.*, pp 174–180, 1996. pages 179

[Morin, 1999] Emmanuel Morin. « *Extraction de liens sémantiques entre termes à partir de corpus techniques* ». Thèse de doctorat, Université de Nantes, 1999. pages 150, 201

[Muehleisen, 1997] Victoria Lynn Muehleisen. « *Antonymy and semantic range in english* ». PhD thesis, Northwestern university, 1997. pages 101

[Muñoz *et al.*, 2000] Marcia Muñoz, Vasin Punyakanok, Dan Roth, et Dav Zimak. « A learning approach to shallow parsing ». Dans les actes de *EMNLP-WVL-99*, pp 168–178, 2000. pages 25

[Nogier, 1991] Jean-François Nogier. *Génération automatique de langage et graphes conceptuels*. Hermès, 1991. pages 38

[Nyckees, 1998] Vincent Nyckees. *La sémantique*. Belin, 1998. pages 40, 43, 45, 47, 93, 102, 343

[Palmer, 1976] Frank Robert Palmer. *Semantics : a new introduction*. Cambridge University Press, 1976. pages 101

[Ploux & Victorri, 1998] Sabine Ploux et Bernard Victorri. « Construction d'espaces sémantiques à l'aide de dictionnaires informatisés des synonymes ». *Traitement automatique des langues*, 1998. pages 144

[Polguère, 2003] Alain Polguère. *Lexicologie et sémantique lexicale*. Les Presses de l'Université de Montréal, 2003. pages 27, 33, 34, 93, 100, 198, 203, 213, 307, 333, 343, 344

[Pompidor & Vergnaud, 1995] Pierre Pompidor et Jean-François Vergnaud. « Coopération et révision des agents d'un système coopératif gérant des bases de données. Application à la traduction automatique du chinois dans un but pédagogique ». Dans les actes de *Troisièmes Journées IAD et SMA*, Chambéry, St Baldoph, 1995. pages 179, 182

[Pottier, 1964] Bernard Pottier. « Vers une sémantique moderne ». *Travaux de sémantique et de littérature*, pp 107–137, 1964. pages 44, 102, 343

[Prince, 1991] Violaine Prince. « Notes sur l'évaluation de la réponse dans TEDDI : introduction d'une relation d'équivalence pour la synonymie relative. ». Notes et Documents LIMSI-CNRS, 91-20, 1991. pages 93

[Quillian, 1968] Ross Quillian. « *Semantic Informatic processing* », Chapitre Semantic memory, pp 227–270. MIT Press, 1968. pages 20, 36

[Rastier, 1985] François Rastier. « *L'isotopie sémantique, du mot au texte* ». Thèse de doctorat d'État, Université de Paris-Sorbonne, 1985. pages 102

[Rastier, 1989] François Rastier. *Sémantique et Recherche Cognitive*. Presses Universitaires de France, 1989. pages 44

[Rastier, 2004] François Rastier. « Ontologie(s) ». *Revue des sciences et technologies de l'information*, pp 15–40, 2004. pages 36, 40

[Reynar, 1998] Jeffrey Reynar. « *Topic segmentation : algorithms and applications* ». PhD thesis, University of Pennsylvania, 1998. pages 224

[Robert, 1995] Le Robert, . *Le Grand Robert de la langue française, Dictionnaire alphabétique et analogique de la langue française (2ème édition)*. Éditions Le Robert, Paris, 1995. pages 144

[Robert, 2000] Le Robert, . *Le Nouveau Petit Robert, dictionnaire alphabétique et analogique de la langue française*. Éditions Le Robert, 2000. pages 2, 21, 81, 146, 147, 152, 167, 169, 206, 341, 345

[Rodrigo, 2004] Frédéric Rodrigo. « Construction automatique d'une base d'acceptions à l'aide de dictionnaires bilingues et du modèle des vecteurs conceptuels ». Mémoire de dea, Université Montpellier II, Montpellier, 2004. pages 257

[Roget, 1852] Peter Mark Roget. *Roget's Thesaurus of English Words and Phrases*. Longman, London, 1852. pages 52

[Sabah, 1988] Gérard Sabah. *L'intelligence artificielle et le langage*. Hermès, Paris, 1988. pages 35

[Sabah, 1990] Gérard Sabah. « CARAMEL : Un système multi-expert pour le traitement automatique des langues. ». *Modèles linguistiques*, 1990. pages 179, 180

[Sabah, 1996] Gérard Sabah. « le sens dans les traitements automatiques des langues - le point après 50 ans de recherches ». Dans les actes de *journée ATALA (un demi-siècle de traitement automatique des langues : état de l'art)*, 1996. pages 46

[Salton & McGill, 1983] Gerard Salton et Michael McGill. *Introduction to Modern Information Retrieval*. McGrawHill, New York, 1983. pages 29, 57, 60

[Salton, 1968] Gerard Salton. *Automatic Information Organisation and Retrieval*. McGrawHill, New York, 1968. pages 57

[Salton, 1971] Gerard Salton. « The SMART Retrieval System – Experiments in Automatic Document Processing ». 1971. pages 29, 57

[Salton, 1991] Gerard Salton. « The Smart Document Retrieval Project ». Dans les actes de *Proc. of the Fourteenth Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, pp 357–358, Chicago, IL, 1991. pages 29

[Sandford, 1998] Eugène Sandford. « *Augmentation lexicale sémantique et désambiguïsation lexicale sémantique : application à la traduction automatique du français vers le tahitien* ». Thèse de doctorat, Université Montpellier II, 1998. pages 68

[Saussure, 1916] Ferdinand Saussure. *Cours de linguistique générale*. Grande bibliothèque Payot, 1916. pages 166

[Schank, 1972] Roger C. Schank. « Conceptual Dependency : A Theory of Natural Language Understanding ». *Cognitive Psychology*, pp pages 532–631, 1972. pages 46

[Schwab, 2001] Didier Schwab. « Vecteurs conceptuels et fonctions lexicales : application à l'antonymie ». Mémoire de dea, Université Montpellier II, LIRMM, Juillet 2001. pages 99, 100, 104

[Schwab *et al.*, 2002a] Didier Schwab, Mathieu Lafourcade, et Violaine Prince. « Amélioration de la représentation sémantique lexicale par les vecteurs conceptuels, le rôle de l'Antonymie. ». Dans les actes de *JADT 2002*, volume 2, pp 701–712, Saint-Malo, Mars 2002. pages 104

[Schwab *et al.*, 2002b] Didier Schwab, Mathieu Lafourcade, et Violaine Prince. « Antonymy and Conceptual Vectors ». Dans les actes de *COLING'2002 : 19th International Conference on Computational Linguistics*, volume 2/2, pp 904–910, Taipei, Taiwan, Août 2002. pages 185

[Schwab *et al.*, 2002c] Didier Schwab, Mathieu Lafourcade, et Violaine Prince. « Vers l'apprentissage automatique, pour et par les vecteurs conceptuels, de fonctions lexicales. L'exemple de l'antonymie ». Dans les actes de *TALN 2002*, volume 1, pp 125–134, Nancy, Juin 2002. pages 100, 104

[Schwab *et al.*, 2004] Didier Schwab, Mathieu Lafourcade, et Violaine Prince. « Hypothèses pour la construction et l'exploitation conjointe d'une base lexicale sémantique basée sur les vecteurs conceptuels. ». Dans les actes de *JADT 2004*, pp 1008–1018, Louvain-La-Neuve, Belgique, Mars 2004. pages 152

[Sekine & Grishman, 1995] Satoshi Sekine et Ralph Grishman. « A corpus-based probalistic grammar with only two non-terminals ». Dans les actes de *Fourth international workshop on parsing technology*, 1995. pages 270

[Sowa, 1984] John Sowa. *Conceptual Structures : Information Processing in Mind and Machine*. Addison-Wesley, Reading, 1984. pages 38

[Sowa, 2000] John Sowa. *Knowledge Representation : Logical, Philosophical, and Computational Foundations*. Brooks Cole Publishing Co., Pacific Grove, CA, 2000. pages 1, 38

[Sparck Jones, 1986] Karen Sparck Jones. *Synonymy and Semantic Classification (thesis, 1964)*. Edinburgh University Press, Edinburgh, 1986. pages 93

[Sérasset & Mangeot, 2001] Gilles Sérasset et Mathieu Mangeot. « Papillon lexical databases project : monolingual dictionaries and interlingual

links ». Dans les actes de *NLPRS 2001*, pp 119–125, 2001. pages 41, 164, 343

[Stefanini, 2004] Marie-Hélène STEFANINI. « TALISMAN : Bilan et perspectives ». Dans les actes de *Journée d'Etude ATALA Agental "Agents et Langue"*, pp 23–30, Paris, France, 2004. ATALA, ATALA. pages 179, 181

[Stendhal, 1830] STENDHAL. *Le rouge et le noir*. 1830. http://www.diogene.ch/textes%20libres/roman/stendhal_le_rouge_et_le_noir.pdf. pages 305

[Teeraparbseree, 2005] Aree TEERAPARBSEREE. « *Méthodes et outils pour la création automatique et l'évaluation de structures de bases lexicales multilingues (symétriques) à lexies et axies* ». Thèse de doctorat, Université Joseph Fourier, Grenoble, France, Septembre 2005. pages 269

[Universalis, 1968] Encyclopædia UNIVERSALIS, . *Encyclopædia Universalis*, volume 17. Encyclopædia Universalis France, 1968. pages 49

[Vergne & Giguet, 1998] Jacques VERGNE et Emmanuel GIGUET. « Regards théoriques sur le "Tagging" ». Dans les actes de *TALN'1998*, Paris, France, Juin 1998. pages 25

[Verne, 1870] Jules VERNE. *20000 lieues sous les mers*, volume 1. 1870. http://www.jules-verne.co.uk/. pages 304, 305

[Walter, 1988] Henriette WALTER. *Le Français dans tous les sens*. Livre de poche, 1988. pages 149

[Warren, 1998] Karine WARREN. « *Gestion de conflits dans une architecture multi-agents d'analyse automatique de textes* ». Thèse de doctorat, Université Stendhal - Grenoble III, Janvier 1998. pages 176, 181

[Wehrli, 1992] Éric WEHRLI. « The IPS system ». Dans les actes de *COLING'92 : 14th International Conference on Computational Linguistics*, pp 870–875, 1992. pages 25

[Wierzbicka, 1993] Anna WIERZBICKA. « La quête des primitifs sémantiques : 1965-1992 ». *Langue française*, Mai 1993. pages 1, 45

[Wilks, 1977] Yorick WILKS. « Good and Bad Arguments About Semantic Primitives. ». *Communication and Cognition*, pp 181–221, 1977. pages 46

[Winograd, 1978] Terry WINOGRAD. « On primitives, prototypes, and other semantic anomalies ». Dans les actes de *conference on Theoretical Issues in Natural Language Processing*, pp 25–32, University of Illinois, 1978. pages 46

[Yousfi-Monod & Prince, 2005a] Mehdi YOUSFI-MONOD et Violaine PRINCE. « Automatic summarization based on sentence morpho-syntactic structure : narrative sentences compression ». Dans les actes de *2nd International Workshop on Natural Language Understanding and Cognitive Science (NLUCS-2005)*, pp 161–167, Miami, USA, Mai 2005. pages 69

[Yousfi-Monod & Prince, 2005b] Mehdi Yousfi-Monod et Violaine Prince. « Utilisation de la structure morpho-syntaxique des phrases dans le résumé automatique ». Dans les actes de *TALN'2005*, volume 1, pp 193–202, Dourdan, Juin 2005. pages 58, 69, 228

[Zamora, 2005] Thibaud Zamora. « FOETAL : FOurmis et Émergence pour le Traitement Automatique des Langues ». Mémoire de dea, Université Montpellier II, LIRMM, Juin 2005. pages 238

[Zweigenbaum *et al.*, 1989] Pierre Zweigenbaum, Bruno Bachimont, Jacques Bouaud, Marc Cavazza, et Laurent Doré. « *Informatique et Gestion des Unités de Soins* », Chapitre Hélène : Compréhension de comptes-rendus d'hospitalisation, pp 257–268. Springer-Verlag, Paris, 1989. pages 179

résumé de la thèse :

Utilisée à la fois pour l'apprentissage et l'exploitation des vecteurs conceptuels, l'analyse sémantique de texte est centrale à nos recherches. L'amélioration qualitative du processus d'analyse entraîne celle des vecteurs. En retour, cette meilleure pertinence a un effet positif sur l'analyse. Parmi les différentes voies à explorer pour obtenir ce cercle vertueux, l'une des pistes les plus intéressantes semble être la découverte puis l'exploitation des relations lexicales entre les mots du texte. Ces relations, parmi lesquelles la synonymie, l'antonymie, l'hyperonymie, la bonification ou l'intensification, sont modélisables sous la forme de fonctions lexicales. Énoncées essentiellement dans un cadre de production par Igor Mel'čuk, nous cherchons, dans cette thèse, à les adapter à un cadre d'analyse. Nous introduisons ici deux classes de Fonctions Lexicales d'Analyse. Les premières, les FLA de construction permettent de fabriquer un vecteur conceptuel à partir des informations lexicales disponibles. Les secondes, les FLA d'évaluation permettent de mesurer la pertinence d'une relation lexicale entre plusieurs termes. Ces dernières sont modélisables grâce à des informations thématiques (vecteurs conceptuels) et/ou grâce à des informations lexicales (relations symboliques entre les objets lexicaux).

Les informations lexicales sont issues de la base lexicale sémantique dont nous introduisons l'architecture à trois niveaux d'objets lexicaux (ITEM LEXICAL, ACCEPTION, LEXIE). Elles sont matérialisées sous la forme de Relations Lexicales Valuées qui traduisent la probabilité d'existence de la relation entre les objets. L'utilité de ces relations a pu être mis en évidence pour l'analyse sémantique grâce à l'utilisation du paradigme des algorithmes à fourmis. Le modèle introduit dans cette thèse, utilise à la fois les vecteurs conceptuels et les relations du réseau lexical pour résoudre une partie des problèmes posés lors d'une analyse sémantique.

Tous nos outils ont été implémentés en Java. Ils reposent sur Blexisma (*Base LEXIcale Sémantique Multi-Agent*) une architecture multi-agent élaborée au cours de cette thèse dont l'objectif est d'intégrer tout élément lui permettant de créer, d'améliorer et d'exploiter une ou plusieurs Bases Lexicales Sémantiques. Les expériences menées ont montré la faisabilité de cette approche, sa pertinence en termes d'amélioration globale de l'analyse et ouvert des perspectives de recherches fort intéressantes.

abstract :

Used both for learning and exploitation of conceptual vectors, text semantic analysis is a central aspect of our research. The qualitative enhancement of the analysis process improves computed conceptual vectors. In return, this has a positive effect on the analysis. Amoung the paths likely to be explored to obtain this vertuous circle, one of the most promising seems to be the modelisation then the exploitation of lexical relations between words of the text. These relations, amongst them, synonymy, antonymy, hyperonymy, intensification, are modelisable under the form of lexical functions. Enunciated essentially in the framework of text generation (or synthesis) by Igor Mel'čuk, in this work we aim at their adaptation to the analysis context. Here, we introduce two classes of Analysis Lexical Functions (ALF). The first ones, constructive ALF, allow to build a conceptual vector from available lexical information. The second ones, evaluative ALF, allow to measure the relevance of a lexical relation between several terms. They are modelisable thanks to thematic information (conceptual vectors) and/or thanks to lexical information (symbolic relations between lexical objects).

Lexical information are issued from the semantic lexical database, of which we introduce the architecture with three levels of lexical objects (LEXICAL ITEM, ACCEPTION, LEXIE). They are materialized under the form of Valuated Lexical Relation which translate the existing probability of the relation between the objects. The use of these relations has been ascertained for the semantic analysis thanks to the ant algorithm paradigm. The model introduced in this thesis makes usage both of the conceptual vectors and of the relations of the lexical network to solve some of the problems at stake during a semantic analysis.

Our tools and algorithms have been implemented in the Java language. They rely on Blexisma an multi-agent architecture conceived during this thesis which focuses on the integratation of all functionality which allow to create, enhance and exploit one or several Semantic Lexical Database. Experiments undertaken showed the feasibility of our approach, its relevance in terms of global enhancement of the analysis process and also open some quite promising research perspectives.