Université de Montréal

Désambiguïsation de corpus monolingues par des approches de type Lesk

par

Florentina Vasilescu

Département d'informatique et de recherche opérationnelle

Faculté des arts et des sciences

Mémoire présenté à la Faculté des études supérieures
en vue de l'obtention du grade de maîtrise ès sciences (M.Sc.)
en informatique

août, 2003

Université de Montréal
Faculté des études supérieures

Ce mémoire intitulé :

Désambiguïsation de corpus monolingues par des approches de type Lesk

présenté par :
Florentina Vasilescu

a été évalué par un jury composé des personnes suivantes :

Yoshua Bengio
président-rapporteur

Philippe Langlais
directeur de recherche

Guy Lapalme
membre du jury

Mémoire accepté le 14 octobre 2003

Résumé

La désambiguïsation d'un texte consiste à déterminer le sens correct des mots de ce texte. C'est une tâche nécessaire à la bonne réalisation de nombreuses applications du traitement des langues naturelles, telles que : *la traduction automatique, la recherche d'informations, la reconnaissance de la parole* ou *l'analyse grammaticale.* De ce fait, c'est devenu de droit une discipline clé de la linguistique informatique.

L'algorithme de Lesk est une méthode de désambiguïsation bien connue qui consiste à compter le nombre de mots communs entre les définitions d'un mot (généralement trouvées dans un dictionnaire électronique) et les définitions des mots de son contexte. Le sens retenu correspond à la définition pour laquelle on compte le plus de mots communs avec le contexte. Cette idée simple a donné des résultats intéressants et s'est avéré meilleure que bon nombre de techniques plus évoluées.

L'étude que nous avons entreprise s'appuie sur une série d'expériences visant la méthode originelle de Lesk et des variantes que nous avons adaptées aux caractéristiques de *WordNet,* une base de données lexicale organisée comme un ensemble de réseaux sémantiques. Les résultats obtenus nous semblent intéressants par leur capacité de mettre en évidence, d'un côté, certaines modalités d'amélioration par rapport aux études antérieures, d'un autre côté, les limites de la méthode.

Le mémoire porte, en principal, sur une présentation du domaine, y compris le cadre organisé d'évaluation des systèmes de désambiguïsation automatique, *Senseval,* une description de la structure de *WordNet,* du corpus de test et des algorithmes implémentés, ainsi qu'une analyse détaillée des résultats de la recherche.

Mots clés : algorithme de Lesk, dictionnaire électronique, WordNet, désambiguïsation automatique, linguistique informatique

Abstract

Word Sense Disambiguation (WSD) can be defined as the task of detecting the correct sense of an ambiguous word in a given context. It is a domain highly connected with other applications from Natural Language Processing (NLP): *automatic translation, information retrieval, speech recognition, grammatical analysis,* etc., and consequently a key discipline of computational linguistics.

Lesk's algorithm is a well-known disambiguation method based on counting the overlaps between the definitions of sense of a target word (supplied by a machine readable dictionary), and the definitions of words in the context. The selected sense for this word corresponds to the definition of sense containing the maximal number of overlaps with the context.

The present study deals with a series of experiments on the original method and on some variants that we adapted to *WordNet,* a lexical data base structured as a set of semantic networks. The results of our study seem interesting, proving, on one hand, some possible ways of improvement as compared with previous studies, and, on the other hand, the limits of the method.

This paper presents an overview of the domain and some references to *Senseval* evaluation framework, a brief description of *WordNet*, test corpus and implemented algorithms, and also a detailed analysis of the results of our research.

Key words: Lesk's algorithm, machine readable dictionary (MRD), WordNet, word sense disambiguation (WSD), computational linguistics

Table des matières

Liste des tableaux

Liste des figures

Liste des abréviations

APOS	catégorie grammaticale à priori connue
BASE	variante de base, choix du sens le plus fréquent
BASEAPOS	variante de base, choix du sens le plus fréquent, catégorie grammaticale à priori connue
BASEDPOS	variante de base, choix du sens le plus fréquent, catégorie grammaticale détectée à partir des instances de test
BASECG	variante de base, choix du sens le plus fréquent, évaluation *coarse-grained*
$C\overline{E} = B$	décisions par défaut **C**orrectes
$CE \neq \overline{B}$	décisions **E**ffectives **C**orrectes différentes de BASE
$CE=B$	décisions **E**ffectives **C**orrectes communes avec **B**ASE
$\overline{C\overline{E}} = \overline{B}$	décisions par défaut **I**ncorrectes, communes avec **B**ASE
$\overline{C}E = \overline{B}$	décisions **E**ffectives **I**ncorrectes communes avec **B**ASE
$\overline{C}E \neq B$	décisions **E**ffectives **I**ncorrectes, différentes de BASE, **C**orrectes dans **B**ASE
$\overline{C}E \neq \overline{B}$	décisions **E**ffectives **I**ncorrectes, différentes de BASE, **I**ncorrectes dans **B**ASE
CL	désambiguïsation basée sur les **C**haînes **L**exicales
DlogF	score pondéré par la **D**istance au mot cible et le $\mathbf{log_2}$ de la **F**réquence d'usage
F	score pondéré par l'inverse de la **F**réquence d'usage des mots superposés ;
logTD	normalisation logarithmique par la **T**aille de **D**escription de sens ;
NP	variante **N**on **P**ondérée
NPOS	catégorie grammaticale non fournie
NRF	variante sans pondération par la fréquence relative du sens candidat

PA	**P**oids **A**ppris des superpositions
RF	pondération par la **F**réquence **R**elative des sens candidats
TD	normalisation du score par la **T**aille de **D**escription de sens
V	score avec tableau de **V**otes (l'absence de *V*, indique une désambiguïsation séquentielle)

Remerciements

Je tiens à remercier Philippe Langlais, mon directeur de recherche, pour ses conseils toujours pertinents, pour sa manière à la fois rigoureuse et agréable de mener les travaux du projet, pour son appui et ses encouragements dans les moments d'impasse de la recherche. Je remercie aussi Elliott Macklovitch et les membres de RALI pour le climat professionnel et agréable et surtout Guy Lapalme pour les séminaires RALI où j'ai trouvé souvent des idées utiles et intéressantes.

À ma famille, d'ici et d'outre-Atlantique

Avant-propos

En dépit du fait qu'il s'agit d'une tâche aisée pour un humain, l'ambiguïté représente une des grandes difficultés du traitement automatique du langage naturel, étant donnée la propriété de certains énoncés d'avoir plusieurs significations en fonction du contexte. Les facultés de compréhension d'un humain et ses connaissances du monde lui permettent, en général, d'accomplir cette tâche sans grande difficulté, le plus souvent sans même qu'il ne soit conscient d'une ambiguïté. Il est, par exemple, naturel de distinguer le sens *"déplacement actif dans l'air"* du mot *vol,* du sens *"action de dérober"* dans le contexte : *"La vitesse moyenne du pigeon voyageur n'est dépassée que par le* ***vol*** *de l'hirondelle (67 mètres à la seconde) ".* Il est en revanche beaucoup plus difficile à une machine de le faire. En fait, les meilleurs systèmes de désambiguïsation sont loin d'approcher les performances humaines. L'ignorance partielle des connaissances à considérer et des méthodes à mettre en œuvre pour capturer ces connaissances constitue la problématique de base de ce champ d'application.

1.1. Aperçu du domaine

Dresser un état de l'art du domaine de la désambiguïsation est un travail qui dépasse de loin l'objectif de cette section dont le but est d'introduire les idées qui sous-tendent ce travail. Le lecteur intéressé trouvera dans (Ide et Veronis 98) une introduction plus détaillée au domaine.

1.1.1. Premiers pas

Les premières approches de désambiguïsation automatique datent des années 50 et sont reliées au domaine de la traduction automatique. Quelques expériences et hypothèses

théoriques proposées à l'époque ont marqué le développement ultérieur du domaine. Le *Memorandum* de Weawer soulignent très tôt l'importance du contexte et des relations syntaxiques dans la désambiguïsation. Weaver rapporte aussi que le domaine du texte joue également un rôle important dans la désambiguïsation. Ceci a suscité l'intérêt des chercheurs des années 50-60 pour le développement de glossaires spécialisés, ou micro-glossaires. Dans un micro-glossaire chaque terme a un seul sens, le sens spécifique au domaine (par exemple un glossaire spécialisé de mathématique contient seulement le sens géométrique de *triangle* et pas celui d'instrument musical). Le même *Memorandum* lance l'hypothèse de la *"structure logique des langues"* qui constitue l'idée génératrice de la notion d'"*interlingua"*, une représentation sémantique, conceptuelle, fondée sur des principes mathématiques et logiques, qui pourrait capter la signification des mots de toutes les langues. C'est à partir de cette notion qu'à la fin des années 50 dérive le concept de *"réseau sémantique"*, exploité ultérieurement par les méthodes d'intelligence artificielle. Une autre direction tracée par le *Memorandum* vise le traitement statistique d'une langue. En suivant cette approche, l'estimation du degré de polysémie des textes et des dictionnaires ou le calcul de la probabilité d'un sens dans le contexte annoncent, dès les années 60, les méthodes de désambiguïsation de date plus récente, basées sur le corpus.

1.1.2. Les approches d'intelligence artificielle

Les approches d'intelligence artificielle des années 60-80 plaçaient la désambiguïsation sémantique dans un contexte plus large de la compréhension du langage humain. Le fonctionnement des systèmes dédiés à cette tâche était basé sur une modélisation des connaissances de nature sémantique et syntaxique. (Ide et Veronis 1998) particularisent deux types de méthodes d'intelligence artificielle caractérisant cette période: les méthodes *symboliques* et les méthodes *connexionnistes.*

Les méthodes dites *symboliques* s'appuient sur la *représentation symbolique du sens des mots* par l'intermédiaire de réseaux sémantiques. Certains chercheurs ont construit des systèmes permettant de choisir le sens correct d'un mot en calculant le plus court chemin entre les nœuds d'un réseau de concepts. D'autres approches ont tenté de résoudre le même problème en utilisant des réseaux sémantiques enrichis par des informations sur les rôles, les relations et les contraintes gouvernant la combinaison des mots dans une phrase. On

proposait aussi des modules de raisonnement permettant de trouver dans une ontologie les ancêtres communs des mots co-occurant dans le même contexte, idée qui anticipait le concept de *"similarité sémantique"* développé dans les années 90 par exemple par Resnik (1995).

Les méthodes *connexionnistes* regroupaient les approches basées sur le modèle des réseaux d'activation *(spreading activation network),* terme utilisé pour désigner un réseau dont les nœuds, activés par un certain contexte, produisent l'activation des nœuds connexes. On retrouve ici également le schéma des réseaux neuronaux capables d'apprendre à partir d'une collection d'exemples préalablement désambiguïsés.

1.1.3. Méthodes basées sur les connaissances

L'essor des ressources de type *dictionnaires électroniques, thésaurus* et *lexiques,* qui a marqué le début des années 80, a offert une autre direction de développement au domaine de la désambiguïsation automatique. Cette nouvelle perspective s'est matérialisée par les *méthodes basées sur les connaissances* qui essayent d'extraire de manière automatique de ces ressources l'information nécessaire à la désambiguïsation.

De nombreuses expériences ont testé l'adéquation des définitions données par les dictionnaires électroniques de type *Collins English Dictionary (CED), Merriam-Webster New Pocket Dictionary, Dictionary of Contemporary English (LDOCE)* pour le traitement automatique de la désambiguïsation.

D'autres approches ont essayé d'extraire les informations utiles à la désambiguïsation des relations entre les mots, décrites par les thésaurus de type *Roget's International Thesaurus* ou par les lexiques sémantiques de type *WordNet* explicitant les sens et les relations entre les sens.

1.1.4. Les approches basées sur le corpus

A côté du développement des dictionnaires, thésaurus et lexiques, l'évolution des systèmes informatiques des années 80 a encouragé la création et le stockage des corpus de textes de grande taille et le retour de l'étude des mots aux méthodes *empiriques (statistiques) basées sur le corpus,* avancées dès les années '30-40. A partir de ces

prémisses, le domaine de la désambiguïsation sémantique a connu deux orientations principales.

L'une regroupe les *approches supervisées* qui utilisent des corpus d'entraînement annotés, comportant des étiquettes de sens, pour désambiguïser les nouvelles occurrences des mots polysémiques, en faisant appel à des hypothèses de type théorie de l'information, Naïve Bayes ou modèles Markov cachés.

L'autre, regroupant les *approches non-supervisées*, essaye de dériver les informations nécessaires à la désambiguïsation à partir des corpus non-annotés, par des méthodes de classification des sens ou *clustering*.

1.1.5. Méthodes hybrides

Une autre tendance actuelle dans le domaine de la désambiguïsation automatique, signalée par (Steven et Wilks 2001), est la conception des *systèmes hybrides* qui combinent plusieurs sources d'informations (fréquence des mots, informations d'ordre morphologique, sémantique, contextuel) et types de méthodes (extracteur de collocations et de définitions, étiqueteur syntaxique, analyseur des traits sémantiques etc.). Des expériences récentes ont montré la validité de ce type d'approche pour la désambiguïsation automatique.

1.2. Évaluation des systèmes de désambiguïsation automatique

La thématique de la désambiguïsation a rapidement vu le foisonnement de méthodes dont la comparaison directe est ardue (pas de référence acceptée de tous, différents prérequis, différents formats etc.). Néanmoins, des paradigmes d'évaluation ont vu le jour dès les années 90 et ce, principalement grâce à l'émergence de corpus étalons (*gold standard)* c.a.d., de corpus désambiguïsés manuellement. De tels corpus attestés rendent possible la comparaison de certaines méthodes de désambiguïsation (celles qui produisent des étiquettes compatibles avec les étiquettes utilisées dans le corpus de référence). Etablir un corpus de référence est pourtant une activité coûteuse et ardue (validation par plusieurs experts, vérification de cohérence du jeu d'étiquettes, etc.). Il est également possible de

rendre un corpus de texte artificiellement ambigu par la concaténation de deux mots naturels (par exemple, *banana-door)*, la tâche du système consistant à reproduire le texte originel (Gale et al. 1992).

1.2.1. Repères et mesures de performance

Un autre élément important dans l'évaluation des performances d'un système est l'estimation de la *limite inférieure* et *supérieure* de ses performances. D'habitude la limite supérieure est fixée à la performance humaine, quant à la limite inférieure, plusieurs alternatives ont été proposées : choix du sens le plus fréquent indiqué par un dictionnaire, sélection du sens comportant le plus grand nombres d'occurrences dans un corpus donné, choix aléatoire, etc. Les mesures de performance généralement acceptées pour l'évaluation sont la *précision* et le *rappel,* mesures que nous décrivons plus loin.

1.2.2. Le cadre d'évaluation *Senseval*[1]

L'une des procédures d'évaluation les plus connues, qui s'est imposée comme un standard dans le domaine, est l'exercice pilote d'évaluation *Senseval1* décrit par (Kilgarriff 1998). L'exercice prenait la forme d'une compétition entre 17 systèmes qui disposaient des mêmes données d'entraînement et de test pour 3 langues (anglais, italien et français) et 4 catégories grammaticales (noms, verbes, adjectifs et indéterminés[2]). Pour la majorité des mots à désambiguïser la catégorie grammaticale a été fournie par les organisateurs, avec les données d'entrée. La tâche consistait à lever l'ambiguïté sur un ensemble préétabli de mots (*lexical sample task*). Les résultats des systèmes participants ont été comparés avec des corpus annotés à la main, comportant les réponses correctes, et avec les résultats produits par certaines implémentations de référence (*baseline)* parmi lesquelles 3 variantes de l'algorithme de Lesk (1 supervisée et 2 non-supervisées). Les meilleures performances ont été enregistrées par des systèmes supervisés. Pourtant la majorité des systèmes ont été surpassés par les implémentations de type Lesk (selon leur catégorie, supervisé ou non-supervisé). Aucun système n'a enregistré de gains de plus 2% par rapport aux performances

[1] http://www.senseval.org/

[2] Pour un nombre de 5 mots cette information n'a pas été spécifiée, la classe de ces mots étant annotée par l'étiquette *indéterminé.*

de base (Lesk), pour un ensemble de test de 2500 instances à désambiguïser et la sous-tâche dédiée à la désambiguïsation des verbes (Kilgarriff et Rosenzweig, 2000).

Un deuxième exercice *Senseval2* a eu lieu en 2001, réunissant 94 systèmes participants pour 12 langues[3] et 3 types de tâches : désambiguïsation d'un ensemble préétabli de mots (*lexical sample task)*, désambiguïsation de tous les mots, noms, verbes, adjectifs et adverbes (*all words task)* et traduction (*Japanese to English translation retrieval task)*. Les performances obtenues dépendent des ressources utilisées pour chaque langue, de la méthode implémentée et du type de tâche accomplie. Là encore, les résultats, comparés avec différents types de performances de base (sens le plus fréquent, variantes de Lesk, choix aléatoire, etc.) ont montré la supériorité des systèmes supervisés, basés sur l'exploitation des corpus préalablement annotés.

Une autre compétition *Senseval3* est prévue pour mars 2004 et vise à évaluer des systèmes de désambiguïsation dans un cadre plus large, multi-langue, et plus proche des exigences des applications réelles du domaine (traduction automatique, recherche d'information, acquisition automatique, identification des rôles sémantiques, etc.). 90 équipes ont pour le moment manifesté leur intérêt dans cette campagne d'évaluation.

1.3. Notre projet

Les dernières années, les approches de désambiguïsation automatique supervisées, basées sur le corpus, ont acquis une certaine popularité en raison de leur relative simplicité et de la qualité relative de leurs résultats. Elles s'appuient sur la disponibilité d'un grand nombre de textes annotés à la main, où chaque occurrence est étiquetée par le sens approprié au contexte. Une telle donnée est difficile et coûteuse à construire. De plus, ces méthodes requièrent - pour fonctionner correctement – une certaine similarité entre les sujets présentés dans le corpus d'entraînement et celui de test (Banerjee et Pedersen 2002).

C'est pourquoi les approches basées sur les connaissances, qui ne dépendent pas de la nature des corpus d'entraînement, semblent une alternative viable, et ce, d'autant plus si le système doit faire face à de nombreuses situations d'application (diversité grandissante des ressources de type dictionnaires, lexiques, glossaires de synonymes, inventaires de termes spécialisés etc., disponibles sous forme électronique).

---

[3] anglais, italien, chinois, japonais, basque, estonien, danois, coréen, espagnol, tchèque, suédois, hollandais.

Une des méthodes usant de ce genre de données, devenue prototype et base de référence (voir *Senseval* 1 et 2), est la méthode de Lesk, qui compte le nombre de superpositions (ang. *overlaps*) entre les définitions des sens candidats et les définitions des mots entourant le mot à désambiguïser, dans un contexte donné. En plus de son applicabilité, non conditionnée par un ensemble de textes déjà étiquetés, la méthode a pour avantage sa simplicité, tant au niveau de sa mise en place que de son analyse (il est toujours facile de comprendre pourquoi une décision a été prise). Ceci explique les nombreuses études passées et récentes autour de ce type d'approche : (Wilks et Stevenson 1997), (Amorós et al. 2001), (Haynes 2001), (Litkowski 2001), (Sidorov et Gelbukh, 2001), (Banerjee et Pedersen 2002), (Inkpen et Hirst 2003). Pourtant l'algorithme de Lesk, dans sa forme initiale, a le désavantage que les sens sont dictés par les définitions du dictionnaire électronique utilisé, le niveau de granularité de ces sens n'étant pas nécessairement celui de l'application.

Notre étude a visé, principalement, une analyse plus détaillée des facteurs influençant les performances de cet algorithme de désambiguïsation (longueur de contexte, type de description utilisé – définition et/ou relations, nature des mots superposés, ordre des sens candidats selon leur fréquence, etc.) ainsi que son applicabilité à d'autres sortes de tâches, par exemple, la détection des relations discursives (chaînes lexicales) entre les mots d'un texte donné (Hirst et St-Onge 1998) ou la caractérisation du domaine de discours (Yarowsky 1992).

Les algorithmes implémentés concernent la variante originelle ainsi que des variantes adaptées aux spécificités de *WordNet,* une base de données lexicale, sémantiquement hiérarchisée, décrite au chapitre 2. A des fins de comparaison avec d'autres travaux, nous étudions le comportement de ces variantes dans le cadre de l'exercice *Senseval2.* Plus précisément, nous avons considéré la tâche *English all words* où quatre catégories de mots (nom, verbe, adjectif, adverbe) sont à désambiguïser, en leur assignant un tag *WordNet* (les mots appartenant à d'autres classes - prépositions, conjonctions, déterminants et pronoms – ne sont pas considérés ici, n'ayant pas d'étiquette dans *WordNet*).

Les résultats de nos expériences montrent, d'une part qu'il est possible d'améliorer la version de Lesk de base et, d'autre part, qu'il n'est pas facile à dépasser de manière significative un baseline constitué par le sens le plus fréquent.

Le mémoire, structuré en 6 chapitres, comporte une description de *WordNet* et du corpus de test (*Senseval* 2 et *Semcor*) (chapitres 2), une présentation de l'architecture du système et des algorithmes développés (chapitres 3,4) ainsi qu'une discussion sur les résultats, les conclusions et les directions futures de la recherche (chapitres 5, 6).

Ressources

Le but de ce chapitre est de présenter les ressources utilisées par nos implémentations, d'un côté la base de données lexicale *WordNet,* d'un autre, les corpus de test, provenant de *Senseval2* et de *Semcor.*

2.1. Description de *WordNet*

WordNet[1] est une base de données lexicales où l'information est structurée autour de groupes de synonymes nommés *synsets*. Un *synset* comporte la liste des synonymes exprimant un même concept, la définition du concept (*gloss*), éventuellement des exemples d'usage, et les relations de ce concept avec d'autres concepts. Les relations entre les *synsets* sont de deux types :

- ***lexical*** - les relations sont exprimées à partir des formes des mots. On trouve les relations suivantes :
    - *antonymie* – deux mots sont antonymes s'ils comportent des sens opposés l'un à l'autre (par exemple, *skilled/unskilled, animate/inanimate, alignment/nonalignment, live_in, sleep_in/live_out, sleep_out)*. Dans *WordNet* l'antonymie est considérée plutôt comme une relation entre les formes des mots parce que, dans beaucoup de cas, elle suppose l'ajout d'un préfixe (*un-, in-, non-)* ou d'un suffixe (*-less*) ou une préférence pour une certaine forme lexicale (dans l'usage fréquent, *light* est un antonyme de *heavy,* mais pas un antonyme de *ponderous* qui est pourtant un synonyme de *heavy);*
    - *pertainymie* – relation appliquée aux adjectifs relationnels de *WordNet* pour indiquer le nom de provenance (par exemple, *academic* est relié par ce type de relation au synset *academia, academe);*
    - *participe* – un adjectif est en relation de *participe* avec le verbe d'où il dérive (*WordNet* relie, par exemple, l'adjectif *applied* au synset *use, utilize, utilise, apply, employ);*

[1] http://www.cogsci.princeton.edu/~wn/

- *voir aussi* sont des renvois qui apportent des informations supplémentaires à la description d'un synset (par exemple, le synset *drink, imbibe* comporte une relation *voir aussi* vers le synset *drain_the_cup, drink_up = drink to the last drop)* ;
- *dérivé d'un adjectif* – relation qui relie un adverbe et l'adjectif d'où il dérive *(negatively/negative)* ;

- ***sémantique*** – les relations sont établies à partir des sens des mots. On trouve les relations :
  - *hyperonymie/hyponymie*[2] – on entend par *hyperonyme* un terme dont le sens inclut d'autres termes, qui sont ses *hyponymes*. Par exemple, le synset *canine, canid* est l'hyperonyme de *dog, domestic_dog, Canis_familiaris* qui est lui-même l'hyperonyme de *working_dog*, hyperonyme de *Eskimo_dog, husky;*
  - *méronyme/holonyme* – relation de type *partie/tout, membre/groupe* établie entre deux synsets nominaux, par exemple : *hound, hound_dog* est un membre méronyme du holonyme *pack ;*
  - *engendrement (entailment)* – relation qui suppose l'enchaînement logique entre deux synsets verbaux, comme par exemple, *wear, have_on* suppose logiquement *dress, get_dressed ;*
  - *cause* – relation qui exprime l'aspect *causatif / résultatif* entre deux synsets verbaux (*show/see, produce, bring_on, bring_out/appear*) ;
  - *similarité* – relation indiquant le fait que la classe de noms modifiés par un adjectif est incluse dans la classe de noms correspondant à un autre adjectif (*hygroscopic/absorbent, absorptive*) ;
  - *attribut* – relation qui relie les adjectifs descriptifs de *WordNet* avec les noms qu'ils peuvent déterminer (par exemple, l'adjectif *short* est relié par une relation d'attribut au synset *duration, length*).

WordNet ne traite pas de relations de type syntagmatique, i.e. de relations établies entre les mots appartenant à des catégories syntaxiques différentes dans le cadre de la phrase, les 4 catégories fondamentales (nom, verbe, adjectif et adverbe) étant traitées séparément (sauf les relations de *partainymie, participe, dérivé* décrites plus haut).

---

[2] Pour les verbes, ce type de relation est exprimé par le terme de *troponymie.*

2.1.1. Architecture de *WordNet*

Le système *WordNet* comporte quatre parties (Tengi 1998): les fichiers sources écrits par les lexicographes, le logiciel (Grinder) pour la conversion de ces fichiers dans la base lexicale proprement-dite, la base de données lexicale et des logiciels d'interface entre l'utilisateur et la base (voir Fig. 2.1.).

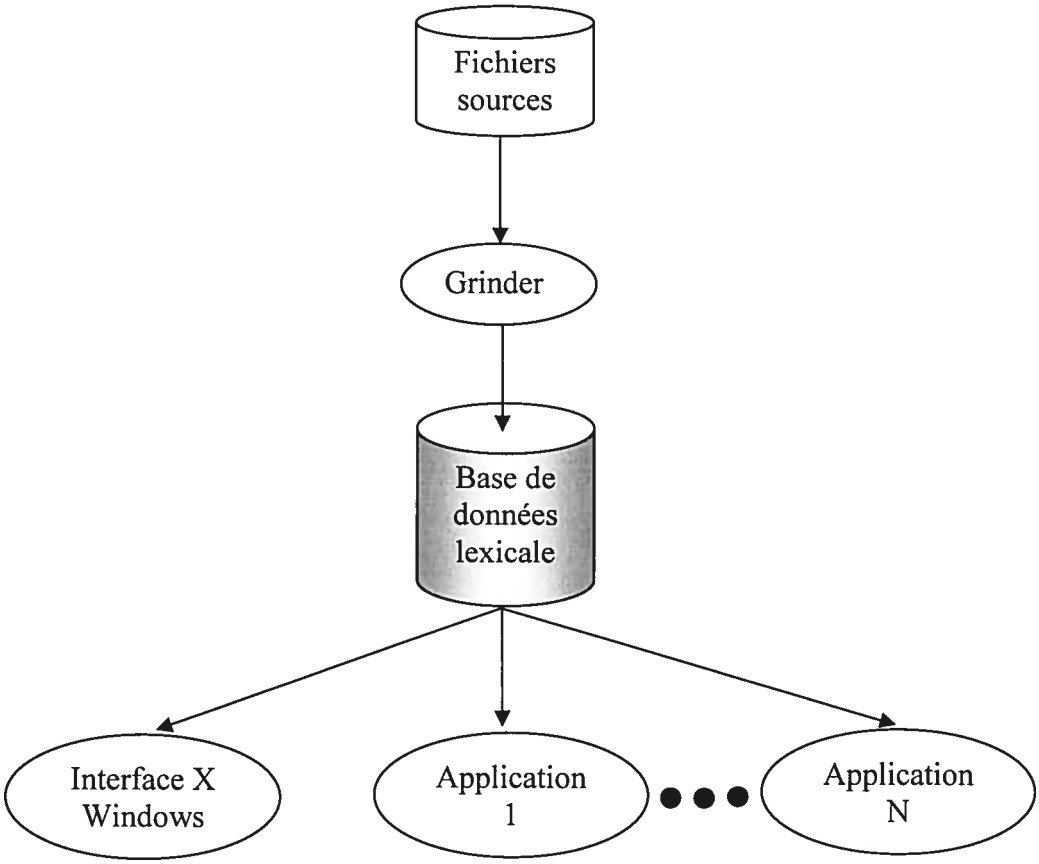


Fig. 2.1. Architecture générale de *WordNet*

Les fichiers sources, créés par les lexicographes, sont le produit d'une analyse minutieuse des relations de type lexical et sémantique entre les mots ainsi que d'une étude sur la fréquence des sens, à partir de corpus sémantiquement annotés.

Le logiciel *Grinder* a pour but de compiler les fichiers des lexicographes dans un format approprié au traitement automatique, facilitant aussi la détection des erreurs

structurales, la construction des pointeurs sémantiques et lexicaux, l'assignation des nombres représentant la fréquence d'usage de chaque sens. La sortie du programme comporte les fichiers de données et d'index, en format ASCII, qui constitue le cœur de *WordNet,* à savoir la base de données. Le format ASCII de la base permet à un utilisateur d'aller chercher facilement l'information dont il a besoin, par l'intermédiaire des ses propres programmes. C'est de cette façon que nous utilisons *WordNet* dans ce travail. Il est également possible d'accéder à l'information via une interface dédiée.

2.1.2. Forme des mots

Les mots dans *WordNet* sont représentés par leur forme canonique (de base) : singulier pour les noms (*book, table)*; infinitif court pour les verbes (*be, read*); degré positif pour les adjectifs (*good, lovely*). Les mots composés, faisant référence à un même concept, sont encodés par une succession de mots individuels, reliés par *underscore* (*fontain_pen, take_for_granted).*

Pour un traitement plus facile des textes en langage naturel, *WordNet* inclut aussi des modules de programmes à fonctions morphologiques et des fichiers d'exceptions, permettant d'obtenir la forme de base à partir de la forme instanciées des mots. Les tableaux ci-dessous indiquent le jeu des suffixes et des terminaisons qui par leur suppression et/ou ajout mènent à la forme de base, ainsi que des exemples extraits des fichiers d'exceptions.

Tab. 2.1. Suffixes et terminaisons par catégorie grammaticale

Noms		**Verbes**		**Adjectifs**	
Suffixe	*Terminaison*	*Suffixe*	*Terminaison*	*Suffixe*	*Terminaison*
s		s		er	
ses	s	ies	y	est	
xes	x	es	e	er	e
zes	z	es		est	e
ches	ch	ed	e		
shes	sh	ed			
		ing	e		
		ing			

Tab. 2.2. Exemples d'exceptions par catégorie grammaticale

Noms		**Verbes**		**Adjectifs**		**Adverbes**	
Forme instanciée	*Forme de base*	*Forme instanciée*	*Forme de base*	*Forme instanciée*	*Forme de base*	*Forme instanciée*	*Forme de base*
activities	activity	accompanied accompanies accompanying	accompany	angrier angriest	angry	best better	well
halves	half	overrunning overruns	overrun	madder maddest	mad	deeper deeper	deeply
men	man	prying	pry	uglier ugliest	ugly	farther further	far
sports_arenas	sports_arena	shook_hands	shake_hands	wetter wettest	wet	harder hardest	hard

2.1.3. Les noms

Les noms dans *WordNet* (G.A. Miller 1998) sont hiérarchisés en plusieurs niveaux de *généralité/spécificité* par l'intermédiaire des relations d'hyperonymie et d'hyponymie entre synsets.

Par exemple, la séquence hyperonymique *{robin, redbreast}* @-> *{bird}* @-> *{animal, animate_being}* @-> *{organism, life_form, living_thing}* décrit une hiérarchie de termes à partir des plus spécifiques vers les plus généraux qui désigne alors une relation de type de type IS-A ou IS-A-KIND-OF. Un parcours inverse peut être aussi tracé, en utilisant les relations d'hyponymie ou de *spécialisation* entre les concepts.

Selon ce principe, les noms de *WordNet* sont divisés en plusieurs hiérarchies, chaque hiérarchie comportant un élément de départ unique ou une *racine* dont les traits sont hérités par tous les hyponymes.

Entre ces hiérarchies existent certaines références croisées mais, en général, elles couvrent des domaines conceptuellement et lexicalement distincts, à partir de 11 mots-racines (voir Fig. 2.2.) :

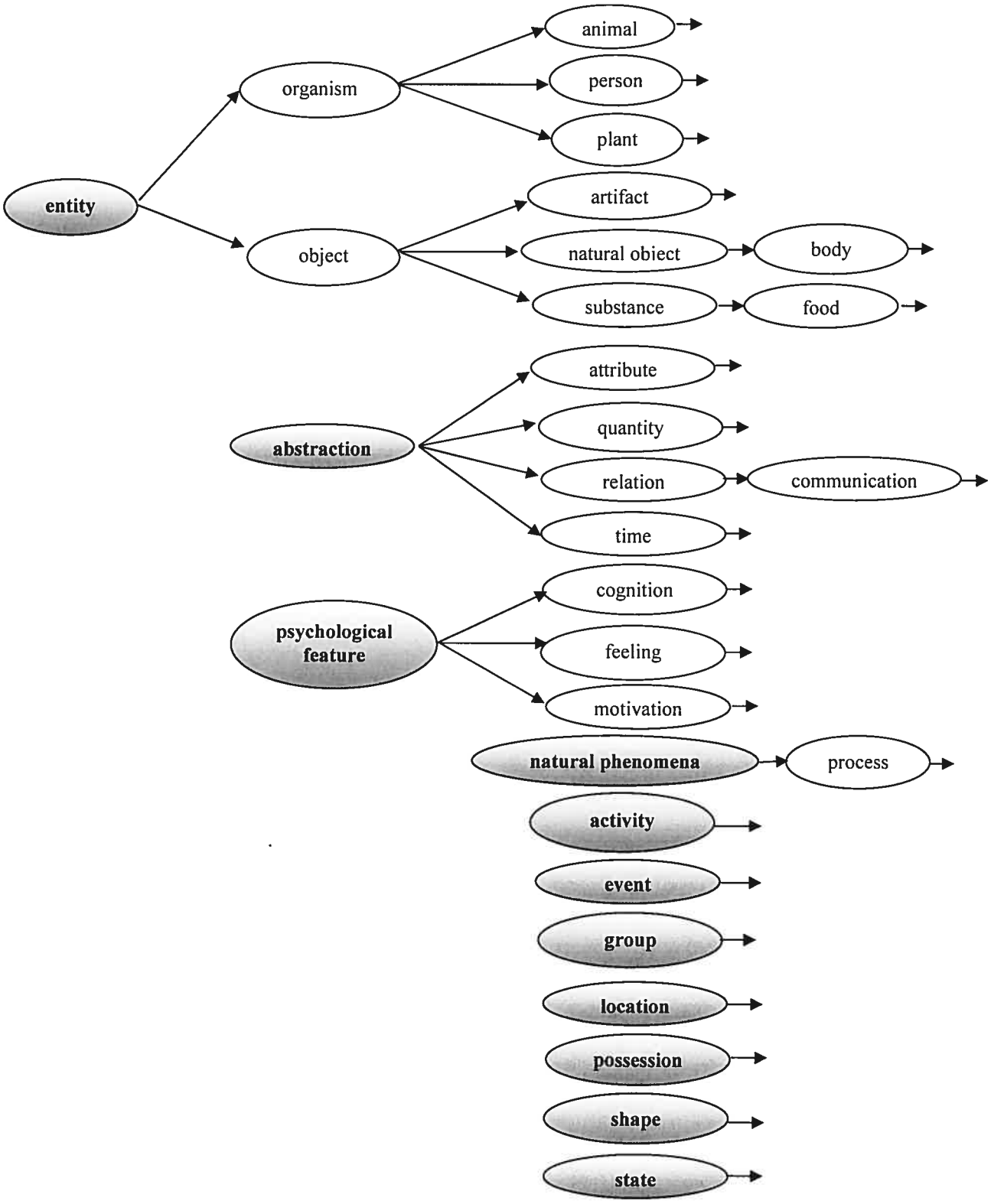


Fig.2.2. Hiérarchies de noms selon *WordNet*

2.1.4. Les verbes

Selon le même principe hiérarchique que pour les noms, les verbes dans *WordNet* (Fellbaum 1998) sont divisés en plusieurs champs sémantiques regroupant des verbes de *mouvement, perception, contact, communication, compétition, changement, cognition, consommation, création, émotion, possession, soin et fonctions du corps,* ou encore *comportement social et interaction.* En plus de ces groupes il y a aussi une catégorie hétérogène englobant les auxiliaires, les verbes de contrôle (*like, want, fail, prevail, succeed),* l'aspectuel *begin* et des concepts élaborés du verbe *be* de type *ressemble, belong* et *suffice.* Certains champs sémantiques sont représentés par plusieurs arbres indépendants, comme par exemple les verbes de mouvement qui comportent deux racines exprimant deux concepts distincts (*move1* – mouvement de translation, *move2* – mouvement sans déplacement) et les verbes de communication dissociés en deux branches indépendantes, verbes de communication verbale et non verbale.

A la différence des noms où la hiérarchisation est réalisée par l'intermédiaire des relations de type IS-A ou IS-A-KIND-OF, pour les verbes une telle sorte de classification semble inadéquate sans une transformation nominale préalable. La relation utilisée dans *WordNet* pour la catégorisation des verbes est la *troponymie*[3] qui peut être exprimée par la phrase :

V1 est un troponyme de V2 si V1 est V2 d'une certaine manière

Par exemple, les troponymes du verbe *fight* dénotent *l'occasion* ou la *forme* de l'action (*battle, war, tourney, duel, feud*), les troponymes d'un verbe de communication encodent *l'intention*, la *motivation* du locuteur (*examine, confess, preach)* ou le *medium* de communication (*fax, e-mail, phone, telex),* etc. La troponymie inclut aussi *l'engendrement (entailment)* et la coexistence temporelle, i.e. V1 est un troponyme d'un verbe plus général V2 si V1 suppose implicitement V2 et les actions de V1 et de V2 se déroulent en même temps. Par exemple, *march* est un troponyme de *walk* parce que *marching* suppose aussi

[3] Les hiérarchies verbales construites par l'intermédiaire de la troponymie n'excèdent pas 4 niveaux hiérarchiques (exemple: *communicate – talk – babble, mumble, slur, murmur, bark).* Par contre, les hiérarchies des noms peuvent atteindre 10 -12 niveaux, cas où la plupart des termes sont des termes techniques, n'appartenant pas au vocabulaire commun (par exemple: *Shetland poney* is a *poney, a horse, an equid, an odd-toed ungulate, a placental mammal, a mammal, a vertebrate, a chordate, an animal, an organisme, an entity.*

walking et ils sont nécessairement coexistants du point de vue temporel. Par contre, la paire *snore / sleep* n'exprime pas une relation troponymique parce que *snore* suppose *sleep* mais leurs actions ne sont pas nécessairement temporellement coexistantes.

Comme les noms et les adjectifs, les verbes dans *WordNet* sont réunis en groupes de synonymes. Pourtant, si on prend en compte la définition exacte de la synonymie qui caractérise des mots interchangeable dans la plupart des contextes, en anglais, le nombre de verbes qui sont de vrais synonymes (*shut / close)* est assez réduit. Par conséquent, dans la majorité des cas, les *synsets* relient des verbes qui expriment le même concept mais qui ne sont pas substituables dans un contexte ou un registre linguistique donné (*begin / commence, end / terminate, rise / ascend, behead / decapitate* etc.). *WordNet* fait ces distinctions d'usage par l'intermédiaire des définitions (*glosses*) et des exemples attachés à chaque *synset.*

2.1.5. Les adjectifs

Les adjectifs dans *WordNet* (K.J. Miller 1998) sont sous-catégorisés en adjectifs *descriptifs* et *relationnels*. Selon le type d'adjectif, il y a une représentation différente dans *WordNet.* A la différence des noms et des verbes, il n'y a pas de hiérarchie dans *WordNet* pour la représentation des adjectifs.

Les adjectifs descriptifs de type *beautiful, interesting, possible, married* sont implicitement reliés à la notion *d'attribut*, i.e. dire que *x est Adj* suppose l'existence d'un attribut A tel que *A(x) = Adj.* Par exemple, la phrase *The package is heavy* implique l'attribut WEIGHT tel que : *WEIGHT(package) = heavy.* Les antonymes *heavy / light* peuvent être considérés ainsi comme des valeurs possibles de l'attribut WEIGHT. Les adjectifs descriptifs sont reliés par des relations de type *attribut* aux noms qu'il peuvent modifier (par exemple, *heavy* est relié au nom *weight).*

Les adjectifs descriptifs sont généralement organisés en *clusters* de synonymes par l'intermédiaire de la relation d'*antonymie* entre des adjectifs dits de *têtes (head synsets)* autour desquels peuvent apparaître des adjectifs *satellites* reliés aux adjectifs-têtes par une relation de *similarité.*

La figure 2.3. montre les relations d'antonymie directe entre les adjectifs-têtes *fast* et *slow* ainsi que les relations de similarité entre les adjectifs satellites et têtes (*rapid* est similaire à *fast, tardy* à *slow)* et d'antonymie indirecte entre les clusters opposés (*laggar* est un antonyme indirecte de *rapid* ou de *fast, quick* de *leisurely* etc.*)*.

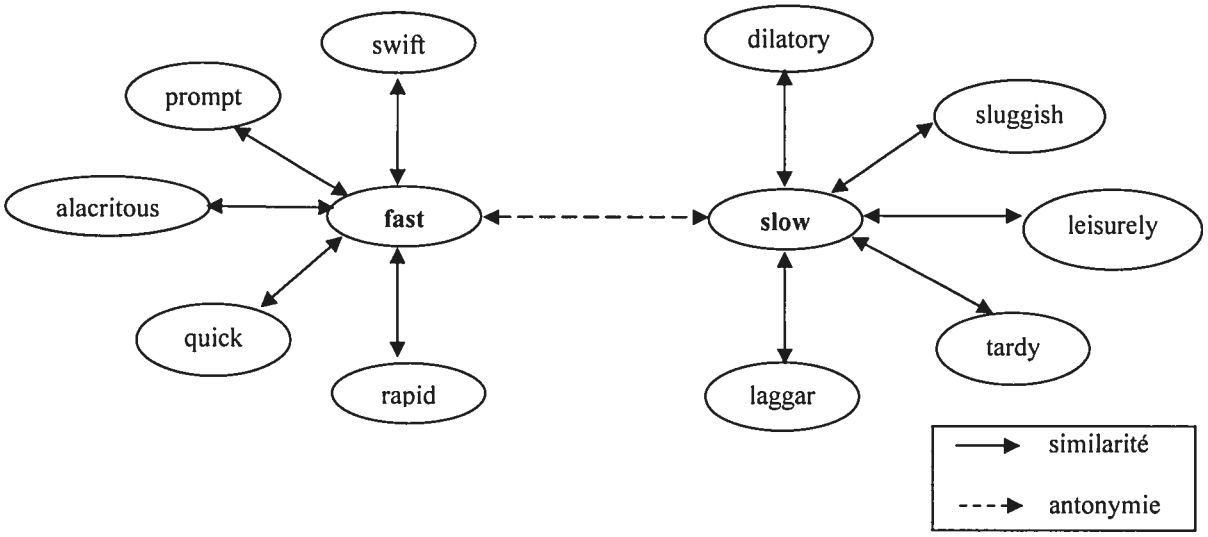


Fig. 2.3. Représentation des adjectifs descriptifs dans *WordNet*

Une sous-classe des adjectifs descriptifs regroupe les adjectifs qui sont des formes participiales des verbes avec lesquels ils sont reliés par des relations de type *participe* (*breaking* est relié au synset *break).* Ce type d'adjectif ne comporte pas d'antonymes.

Les adjectifs relationnels sont des adjectifs dérivés de noms, comme par exemple *electrical* est un dérivé du nom *electricity.* Cette relation implique un lien de type sémantique et morphologique avec le nom d'origine. Pourtant le lien morphologique n'est pas toujours direct, comme dans le cas de l'adjectif *dental* relié au synset *tooth* via le mot latin *dens*.

A la différence des adjectifs descriptifs, les adjectifs relationnels ne suppose pas une relation d'attribut avec le nom déterminé et n'acceptent pas de degrés de comparaison (les expressions telles que : * *the hygiene is dental* ou * *the very electrical field* ne sont pas acceptables).

2.1.6. Les adverbes

L'organisation sémantique des adverbes dans *WordNet* (K.J. Miller 1998) est assez simple, car il n'y a pas de hiérarchies ou de clusters comme pour les autres catégories grammaticales. Chaque *synset* peut comporter un adverbe (éventuellement ses synonymes et/ou antonymes), un pointeur vers l'adjectif à partir duquel l'adverbe est dérivé (s'il y en a un, comme par exemple *quick-quickly, extreme-extremely)* et la partie de définition et d'exemple d'usage (*gloss*) caractérisant le *synset* en question.

2.1.7. Quelques données statistiques

Dans cette section, nous présentons, de manière quantitative, le contenu de *WordNet* 1.7.1 (voir *WordNet Statistics*[4]).

Le tableau 2.3 montre la structure de *WordNet* en nombre de mots, nombre de synsets et nombre de sens, globalement et par catégorie grammaticale. Du nombre total de formes décrites, la plupart sont des noms (74.6%), le reste étant constitué par des adjectifs (14.6%), des verbes (7.6%) et des adverbes (3.2%). La polysémie (nombre de sens par mot) se manifeste dans *Wordnet* par le fait qu'il y a des mots qui peuvent appartenir à plusieurs *synsets* (146350 formes traitées / 111223 synsets).

Tab 2.3. Nombre de mots, *synsets* et sens dans *WordNet*

Partie de discours	Nombre de mots	Nombre de synsets	Nombre de sens
Noms	109195	75804	134716
Verbes	11088	13214	24169
Adjectifs	21460	18576	31184
Adverbes	4607	3629	5748
Total	146350	111223	195817

Le tableau 2.4 donne des informations sur le caractère monosémique ou polysémique des mots selon leur catégorie grammaticale. Les verbes présentent le taux de mots polysémiques le plus élevé, par rapport au nombre total des formes verbales décrites dans *WordNet* (46.6%), tandis que les autres catégories comportent un taux de formes polysémiques plus bas (noms polysémiques – 13.2%, adjectifs polysémiques – 25.5%,

[4] http://www.cogsci.princeton.edu/~wn/man1.7.1/wnstats.7WN.html

adverbes polysémiques - 17%). Ceci semble indiquer, qu'en général, les verbes et les adjectifs présentent une tendance plus accentuée à la polysémie, que les adverbes et les noms.

Tab. 2.4. Répartition des mots dans *WordNet* en monosémiques et polysémiques

Partie de discours	Mots monosémiques	Mots polysémiques
Noms	94685 (86.8%)	14510 (13.2%)
Verbes	5920 (53.4%)	5168 (46.6%)
Adjectifs	15981 (74.5%)	5479 (25.5%)
Adverbes	3820 (83%)	787 (17%)
Total	120406 (82.3%)	25944 (17.7%)

Le tableau 2.5 montre le degré moyen de polysémie des mots dans *WordNet.* Les verbes détiennent le degré de polysémie le plus élevé, suivis par les adjectifs, les noms et les adverbes, le mot le plus polysémique selon WordNet, étant le verbe *give,* avec un nombre de 44 sens.

Tab. 2.5. Polysémie moyenne dans *WordNet*

Partie de discours	Polysémie moyenne (incluant les mots monosémiques)	Polysémie moyenne (excluant les mots monosémiques)
Noms	1.23	2.75
Verbes	2.17	3.52
Adjectifs	1.45	2.76
Adverbes	1.24	2.41
Total	1.52	2.86

La polysémie est souvent considérée comme un *indicateur de familiarité,* directement relié à la fréquence d'usage (Tengi 1998), i.e. en général, plus un mot est fréquent, plus il possède de sens.

Le diagramme de la figure 2.4 présente l'indicateur de familiarité pour chaque catégorie grammaticale, en incluant les mots monosémiques. Selon ce diagramme, le nombre de verbes et d'adjectifs fréquents est plus élevé que celui correspondant aux noms et aux adverbes, ce qui semble indiquer que la plupart des termes spécialisés, moins fréquents, est renfermée dans *WordNet* par ces deux dernières catégories.

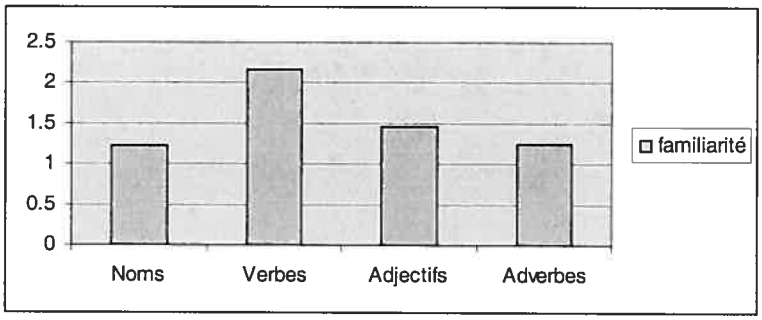


Fig. 2.4. Indicateur de "familiarité" par catégorie grammaticale, selon *WordNet*

2.1.8. Choix de *WordNet*

D'autres études dans le domaine de la désambiguïsation automatique ont utilisé avec succès d'autres types de dictionnaires tels que *Roget's Thesaurus* (Yarowsky 1992), *Longman Dictionary of Contemporary English* (Stevenson et Wilks 2001), *New Oxford Dictionary of English* (Litkowsky 2002). De plus, la granularité trop fine des sens dans *Wordnet* a été plusieurs fois mentionnée comme un inconvénient pour la désambiguïsation automatique (Voorhees 1998), (Véronis 2001), (Palmer et al. 2002), (Preiss et al. 2002). Pourtant, nous avons travaillé avec *Wordnet* par des raisons qui tiennent, d'un côté, de sa complète disponibilité sur Internet (base de données, documentation, fichiers sources etc.) et, d'un autre côté, de sa compatibilité avec l'environnement *Senseval2* que nous avons choisi comme cadre d'évaluation de notre recherche.

2.2. Le corpus de test

Dans ce mémoire, nous avons travaillé sur deux types de corpus : le corpus de test de *Senseval* 2, section *anglais, tous les mots,* et 10 fichiers extraits du corpus *Semcor,* que nous décrivons dans cette section. Pour rendre cette description informative quant à la tâche qui nous préoccupe (la désambiguïsation), il convient d'introduire les métriques avec lesquelles nous allons évaluer nos algorithmes. C'est l'objet de la section 2.2.1.

2.2.1. Métriques d'évaluation

Suivant le modèle d'évaluation de *Senseval2,* nous avons utilisé les même métriques afin de mesurer les performances de notre système. A partir des réponses de celui-ci et des fichiers-clés comportant les réponses correctes nous avons calculé la *précision* et le *rappel* du système pour chaque corpus de test donné.

La *précision* représente une mesure de l'efficacité du système par rapport au nombre de cas traités. Pourtant, elle n'est pas suffisante pour caractériser le comportement global du système parce qu'une précision de 100% n'indique pas toujours un fonctionnement parfait. Par exemple, un système qui ne traite que seulement 2 cas d'un total de 10, même pour une précision de 100% (2 réponses correctes de 2 cas traités), ne représente pas un système satisfaisant.

En revanche, le *rappel* tient compte de cet aspect, en indiquant pour l'exemple considéré une performance de 20% traitements corrects par rapport aux nombre total des cas à traiter.

Les formules que nous avons utilisées pour le calcul des deux métriques sont :

$$prec = 100 \cdot \frac{nb.\ de\ réponses\ correctes}{nb.\ de\ cas\ traités} \qquad (1)$$

$$rapp = 100 \cdot \frac{nb.\ de\ réponses\ correctes}{nb.\ de\ cas\ à\ traiter}. \qquad (2)$$

2.2.2. Le corpus de test de *Senseval* 2

Les données de test de *Senseval2* ainsi que le fichier-clé (de réponses), ont été téléchargés du site officiel de *Senseval*. Le corpus de test consiste en trois articles traitant des sujets différents et un total de 2473 mots à désambiguïser. Ces articles proviennent du *Penn Treebank Text*[5] (un corpus de phrases arborées). L'inventaire des sens a été construit à l'aide des synsets de *WordNet* 1.7.

Le fichier de test, en format XML, contient un système de balises qui mettent en évidence les instances des mots à désambiguïser (noms, verbes, adjectifs ou adverbes) en leur assignant des étiquettes d'identification par document (*d*), phrase (*s*) et mot dans la

[5] http://www.cis.upenn.edu/~treebank/home.html

phrase (t) :

```
<?xml version="1.0"?>
<!DOCTYPE corpus SYSTEM  "all-words.dtd">
<corpus lang="en">
<text id="d00">
The
<head id="d00.s00.t01">art</head>
of
<head id="d00.s00.t03">change-ringing</head>
<head id="d00.s00.t04">is</head>
<head id="d00.s00.t05">peculiar</head>
      …..
They
<head id="d00.s10.t01" sats="d00.s10.t01.s0">belong</head>
<sat id="d00.s10.t01.s0">to</sat>
a
<head id="d00.s10.t04">group</head>
…
and
<head id="d02.s81.t14" >publishes</head>
The
National
Interest
.
</text>
</corpus>
```

Fig. 2.5. Extrait du corpus de test *Senseval* 2, tâche *anglais tous les mots*

Des informations sur les catégories grammaticales, sont aussi disponibles sur le site *Senseval,* dans des fichiers séparés (.MRG), leur exploitation dans le processus de désambiguïsation nécessitant la construction de liens entre ces fichiers et le fichier XML :

```
( (S
    (NP-SBJ
      (NP (DT The) (NN art) )
      (PP (IN of)
        (NP (NN change-ringing) )))
    (VP (VBZ is)
      (ADJP-PRD
        (ADJP (JJ peculiar)
          (PP (TO to)
            (NP (DT the) (NNS English) )))
        (, ,)
        (CC and)
...
```

Fig. 2.6. Extrait d'un fichier *treebank*

Nous avons utilisé ce type d'information pour les expériences qui considèrent à priori connue la catégorie grammaticale des mots à désambiguïser (APOS).

Une analyse plus détaillée du corpus de test de *Senseval* 2 est présentée dans les sous-sections suivantes.

a) Caractérisation globale des données de test *Senseval* 2

Le tableau 2.6 montre que le corpus *Senseval2* contient 1082 mots différents pour un total de 2473 mots à désambiguïser. La distribution par catégories grammaticales indique une majorité nominale (43%), approximativement égale au nombre cumulatif de verbes et d'adjectifs (45%) et une proportion plus restreinte d'adverbes (12%). Les indicatifs *NPOS* et *APOS* ont été choisis pour faire la distinction entre les caractéristiques du corpus si on **n'**utilise pas la catégorie grammaticale (***Part Of Speech***) des mots à désambiguïser *(NPOS),* respectivement, si ce facteur est **à priori** connu *(APOS).* Cette information est fournie dans le *treebank* accompagnant le corpus (les fichiers .MRG décrits plus haut).

Tab.2.6. Structure globale du fichier de test *Senseval* 2

Mots à désambiguïser	*Mots différents*	*Noms*	*Verbes*	*Adjectifs*	*Adverbes*	*No. moyen de sens/mot*		*Performances de base (%)*			
								NPOS		*APOS*	
						NPOS	*APOS*	*Préc.*	*Rapp.*	*Préc.*	*Rapp.*
2473	1082	1067	554	551	301	7.19	4.79	57.9	57.6	61.9	61.3

Les colonnes 7 et 8 de la table 2.6. montrent le nombre moyen de sens par mots dans le corpus de test *Senseval2* selon que l'on utilise *(APOS)* ou pas *(NPOS)* la catégorie grammaticale du mot à désambiguïser. Les colonnes 9-12, quant à elles, montrent les performances (précision et rappel) de l'algorithme du sens majoritaire lorsque l'on fait usage (APOS) ou pas (NPOS) de cette information. Comme on peut s'y attendre, connaître la catégorie grammaticale du mot à désambiguïser diminue le nombre de sens possibles et rend ainsi la tâche plus facile.

b) Structure par catégorie grammaticale du corpus de test *Senseval* 2

Une description du nombre de sens et des performances de base par catégorie grammaticale est présentée dans le tableau 2.7. Comme on peut le constater, les performances de base sont de loin plus faibles pour les verbes que pour les autres catégories (un fait largement reconnu). Le degré de polysémie élevé pour les verbes (7.47) pourrait en partie expliquer ce fait. A l'opposé, on observe que les noms sont plus faciles à désambiguïser que les adjectifs, et ce malgré un taux de polysémie relativement élevé (en moyenne 4.23 sens/mot par rapport à 2.48 pour les adjectifs).

Tab. 2.7. Précision de base et nombre moyen de sens par mot, selon la catégorie grammaticale, corpus *Senseval2*

Noms			***Verbes***			***Adjectifs***			***Adverbes***		
Précision (%)		*No. de*	*Précision (%)*		*No. de*	*Précision (%)*		*No. de*	*Précision (%)*		*No. de*
NPOS	*APOS*	*sens/mot*	*NPOS*	*APOS*	*sens/mot*	*NPOS*	*APOS*	*sens / mot*	*NPOS*	*APOS*	*sens /mot*
67.6	70.3	4.23	37.3	43.6	**7.47**	46.8	50.9	2.48	79.0	80.3	2.55

Cette disparité de performances suggère qu'il est pertinent d'étudier la distribution des sens : un mot avec beaucoup de sens mais dont un des sens est majoritaire est plus facile à désambiguïser qu'un mot qui se réalise dans plusieurs sens équiprobables (Manning et Schütze 1999). Et ceci parce que, habituellement, certains mots sont employés, dans un segment donné (paragraphe ou texte), dans leur sens le plus fréquent.

(Kilgarriff et Rosenzweig 2000b), (Hoste et al. 2002), (Audibert 2003) expriment la difficulté de la tâche par l'entropie, vue comme une mesure de la distribution des fréquences de sens.

Nous avons calculé l'entropie de la distributions des sens s_j, $j=1,n$ d'un mot cible t par la formule:

$$H(t) = -\sum_{j=1,n} p(s_j) \cdot log_2(p(s_j)) \qquad (3)$$

où $p(s_j)$ représente la probabilité du sens s_j de t calculée en tant que fréquence relative[6].

Une valeur élevée de l'entropie indique ainsi une distribution plus uniforme de sens, et par conséquent une tâche plus difficile, tandis qu'une entropie basse suppose une distribution biaisée vers quelques sens et une tâche plus facile. Le tableau 2.8 montre les valeurs de l'entropie, calculées pour le corpus *Senseval2.* Ces valeurs indiquent les degrés de difficulté par catégorie grammaticale : les plus difficiles à désambiguïser sont les verbes et les adjectifs, suivis par les noms et les adverbes.

Tab. 2.8. Entropie de la distribution de sens par catégorie grammaticale, corpus *Senseval2*

Noms	**Verbes**	**Adjectifs**	**Adverbes**
1.16	1.91	1.25	0.71

[6] Le calcul détaillé de la fréquence relative d'un sens candidat à partir de *WordNet* est présenté plus loin, dans les sections 3.1.1. et 5.3.3.

c) Distribution réelle des sens

Cette section réalise l'étude de la distribution de sens réelle dans le corpus de test *Senseval2,* à partir du fichier-clé, comportant les sens corrects. La table 2.9 récapitule les points les plus intéressants de cette étude. Une première remarque qui corrobore l'idée sus-mentionnée est, qu'en moyenne, le nombre de sens par mots est de 1.27, c'est-à-dire sensiblement inférieur aux 7.19 sens possibles. En outre, seulement un cinquième des mots à désambiguïser (19.4%) possèdent plus d'un sens dans le corpus de test de *Senseval2*, et leur sens dominant intervient dans un peu plus de la moitié des cas (55%).

Tab.2.9. Distribution réelle des sens pour le corpus *Senseval* 2

Nombre moyen de sens par mot	Taux moyen des mots à plusieurs sens %	Nombre moyen de sens par mot pour les mots à plusieurs sens	Fréquence relative moyenne du sens dominant pour les mots à plusieurs sens	Intervalle moyen de cooccurrence du même sens dans la même phrase	Taux moyen des phrases à cooccurrence du même sens %	Intervalle moyen de cooccurrence des sens différents du même mot dans la même phrase	Taux moyen des phrases à cooccurrence des sens différents du même mot %	Intervalle moyen de cooccurrence du même sens	Intervalle moyen de cooccurrence des sens différents du même mot
1.27	19.4	2.39	0.55	10.19	19	12.66	1.23	87.22	75.42

Les colonnes 5,6 indiquent que 19% des phrases du corpus de test (46 d'un total de 242 phrases) contiennent plus d'une occurrence du même sens, l'intervalle moyen entre ces occurrences étant de 10 mots (y compris les signes de ponctuation). Par exemple, la phrase "*The **reasons** are complex, but one simple **reason** ought not to be underestimated*" contient le même sens du mot *reason ("an explanation of the cause of some phenomenon"),* à un intervalle de 6 mots.

Le nombre de phrases comportant des sens différents du même mot est bien plus bas (1.23%, c.a.d. 3 phrases d'un total de 242), et la distance moyenne entre ces sens est de 12.6 *tokens* (colonnes 7,8). Un exemple de ce type de phrase est : *"The art of change-ringing is peculiar to the **English**, and, like most **English** peculiarities, unintelligible to the rest of the world.*" qui comporte deux sens différents du même mot, le nom *English ("the people of England")* et l'adjectif *English ("relating to or characteristic of England or its culture").*

Considéré globalement, sans tenir compte des "barrières" imposées par les phrases, l'intervalle moyen de répétition du même sens est de 87 mots et la distance moyenne entre les occurrences des sens différents du même mot est de 75 mots. Par exemple, le mot *English,* qui possède 2 sens et apparaît 6 fois dans le corpus de test, comporte un intervalle

moyen de répétition du même sens de 289 mots et une distance moyenne entre les occurrences de ses sens différents de 140 mots.

Par conséquent, pour le corpus *Senseval2*, les mots tendent à se manifester par un seul sens (Yarovsky 1992). Quant aux occurrences à plusieurs sens, elles présentent généralement un sens dominant qui intervient dans un peu plus que la moitié des cas. Au niveau de la phrase, la répétition du même sens est bien plus fréquente que la répétition d'un mot comportant des sens différents. A l'inverse du comportement dans la phrase, au niveau du corpus, les occurrences d'un même sens sont plus distancées entre elles que les occurrences des sens différentes du même mot.

Ces observations contribuent à expliquer les bonnes performances d'un baseline basé sur le sens le plus fréquent.

2.2.3. Le corpus de test extrait de *Semcor*

Pour compléter notre corpus de test, nous avons également isolé un corpus extrait de *Semcor1.6*[7]. Ce corpus est une collection de texte du *Brown Corpus*[8], où les occurrences des mots polysémiques ont été annotées à la main par des étiquettes de sens provenant de *WordNet1.6. Semcor1.6.* est organisé en 3 répertoires décrits dans le tableau 2.10:

Tab.2.10. Structure et contenu du corpus *Semcor*

Répertoire	Nombre de fichiers	Nombre d'occurrences	Nombre d'occurrences étiquetées	Éléments taggés
brown1	103	198796	106639	noms, verbes, adjectifs et adverbes
brown2	83	160936	86000	noms, verbes, adjectifs et adverbes
brownv	166	316814	41497	verbes

Semcor est présenté dans un format SGML, les données textuelles étant segmentées en fichiers, paragraphes, phrases, mots et signes de ponctuation (voir Fig.2.7). Une étiquette sémantique attachée à un mot indique le sens de *WordNet* approprié au contexte où il apparaît. Seulement les noms, les verbes, les adjectifs et les adverbes sont taggés. Les noms propres sont annotés par quatre types d'étiquettes : *person, location, group* ou *other*.

[7] Maintenant disponible à l'adresse : http://www.cs.unt.edu/~rada/software.html
[8] http://clwww.essex.ac.uk/w3c/corpus_ling/content/corpora/list/private/brown/brown.html

```
<contextfile concordance=brown>
<context filename=br-a01 paras=yes>
<p pnum=1>
<s snum=1>
<wf cmd=ignore pos=DT>The</wf>
<wf cmd=done rdf=group pos=NNP lemma=group wnsn=1 lexsn=1:03:00::
pn=group>Fulton_County_Grand_Jury</wf>
<wf cmd=done pos=VB lemma=say wnsn=1 lexsn=2:32:00::>said</wf>
<wf cmd=done pos=NN lemma=friday wnsn=1 lexsn=1:28:00::>Friday</wf>
<wf cmd=ignore pos=DT>an</wf>
<wf cmd=done pos=NN lemma=investigation wnsn=1 lexsn=1:09:00::>investigation</wf>
…
<punc>.</punc>
</s>
</p>
<p pnum=2>
<s snum=2>
<wf cmd=ignore pos=DT>The</wf>
 …
</context>
</contextfile>
```

Fig. 2.7. Extrait du corpus *Semcor*

Le corpus de test que nous avons utilisé comporte 10 fichiers de test, chacun d'environ 2000 mots à traiter extraits de 2 fichiers du répertoire *brown1*. A l'instar du corpus *Senseval2,* nous présentons dans les sections suivantes les caractéristiques de ce corpus de test.

a) Caractérisation globale des données de test extraites de *Semcor1.6.*

Comme le montre le tableau 2.11, chaque jeu de test comporte environ 2000 mots à désambiguïser (1000 mots différents), dont presque la moitié sont des noms (48.8%), chaque mot possédant en moyenne 7 sens, si la catégorie grammaticale n'est pas à priori connue *(NPOS)* et 4.6 sens en cas contraire *(APOS)*. Ceci est similaire à ce que nous observions sur le corpus *Senseval2.* Les quatre dernières colonnes indiquent les performances de base pour chaque fichier de test (catégorie grammaticale connue ou non), c.a.d. les performances obtenues en choisissant le sens le plus fréquent, sans tenir compte du contexte.

Par rapport au corpus *Senseval2,* les valeurs des performances de base sont plus élevées, ce qui semble indiquer, pour les textes extraits de *Semcor,* une tendance plus marquée des mots de se manifester par leur sens le plus fréquent.

Tab.2.11. Structure globale des fichiers de test extraits de *Semcor*

Fichiers Semcor1.6	*Fichier de test*	*Mots à désambiguïser*	*Noms*	*Verbes*	*Adjectifs*	*Adverbes*	*No. moyen de sens/mot*		*Performances de base (%)*			
							NPOS	*APOS*	*NPOS*		*APOS*	
									Préc.	*Rapp.*	*Préc.*	*Rapp.*
br-a01 + br-a02	test0_1	2037	1184	481	266	106	6.37	4.43	69.61	69.61	76.40	76.29
br-a11 + br-a12	test2_3	2152	1152	467	377	156	8.42	5.12	68.48	68.45	77.31	76.95
br-a13 + br-a14	test4_5	2035	1117	440	340	138	7.54	4.64	68.06	68.06	76.65	76.61
br-a15 + br-b13	test6_7	2113	1040	516	406	151	7.28	4.79	66.64	66.54	73.99	73.78
br-b20 + br-c01	test8_9	2076	970	492	378	236	6.58	4.42	67.15	67.15	72.44	72.30
br-c02 + br-c04	test10_11	2115	976	439	436	264	6.81	4.39	66.05	66.05	72.80	72.62
br-d01 + br-d02	test12_13	1996	803	499	387	307	6.66	4.64	64.78	64.78	69.64	69.54
br-d03 + br-d04	test14_15	2100	971	462	392	275	6.11	4.35	65.90	65.90	69.77	69.57
br-e01 + br-e02	test16_17	2114	888	532	432	262	7.36	4.80	61.62	61.59	69.33	68.87
br-e04 + br-e21	test18_19	2226	1095	432	506	193	7.00	4.57	65.45	65.45	72.95	72.82
	Moyenne (M)	**2096.40**	**1019.60**	**476.00**	**392.00**	**208.80**	**7.01**	**4.61**	**66.37**	**66.36**	**73.13**	**72.94**
	Ecart (σ)	62.15	114.50	32.28	59.79	65.26	0.63	0.22	2.21	2.21	2.91	2.88
	Coef. var (σ*100)/M	2.96	11.22	6.78	15.25	31.25	9.04	4.92	3.35	3.35	3.99	4.00

Les variations des paramètres considérés pour les 10 fichiers de test par rapport aux valeurs moyennes sont exprimées par l'intermédiaire de deux mesures statistiques : l'écart type et le coefficient de variation. Ces mesures indiquent une structure assez homogène du corpus de test, des variations un peu plus élevées étant enregistrées par la catégorie des adverbes.

b) Structure par catégorie grammaticale du corpus de test extrait de *Semcor 1.6.*

L'analyse de la précision de base et du degré de polysémie par catégorie grammaticale est présentée dans le tableau 2.12.

Un comportement assez uniforme pour les 10 fichiers de test, indique, comme pour le corpus *Senseval2*, un degré de difficulté plus élevé dans le cas des verbes. Les meilleures performances de base appartiennent à la catégorie adverbiale, caractérisée par le plus petit degré de polysémie.

A la différence de *Senseval2,* les précisions des noms et des adjectifs sont, cette fois, presque égales, pour un nombre différent de sens / mot.

Tab. 2.12. Précision de base et nombre moyen de sens par mot, selon la catégorie grammaticale, corpus *Semcor*

Fichier de test	*Noms*			*Verbes*			*Adjectifs*			*Adverbes*		
	Précision (%)		*No. de sens/mot*	*Précision (%)*		*No. de sens/mot*	*Précision (%)*		*No. de sens / mot*	*Précision (%)*		*No. de sens / mot*
	NPOS	*APOS*		*NPOS*	*APOS*		*NPOS*	*APOS*		*NPOS*	*APOS*	
test0_1	76	83	3.45	52	57	7.82	66	75	2.01	77	83	2.98
test2_3	73	83	4.60	48	54	8.17	73	83	2.57	79	84	2.79
test4_5	72	81	4.22	54	59	7.30	68	82	2.55	73	76	2.97
test6_7	70	77	4.12	54	57	7.88	65	77	2.62	88	91	2.13
test8_9	70	77	3.84	54	53	7.42	65	77	2.42	88	82	2.28
test10_11	74	74	4.15	48	55	7.42	65	78	2.44	79	86	2.24
test12_13	67	69	4.42	52	55	7.63	66	77	2.35	83	82	2.23
test14_15	62	68	3.96	54	52	7.19	70	84	2.03	78	80	2.21
test16_17	64	72	4.25	49	53	7.51	79	74	2.65	79	77	2.61
test18_19	65	74	4.52	58	59	7.91	67	76	2.31	78	83	2.27
Moyenne (M)	**68.60**	**75.10**	**4.22**	**52.30**	**55.20**	**7.60**	**68.60**	**78.60**	**2.43**	**80.50**	**82.30**	**2.42**
Ecart (σ)	3.80	4.52	0.22	3.03	2.31	0.28	4.22	3.10	0.17	4.41	4.07	0.26
Coef. Var (σ*100)/M	5.53	6.02	5.30	5.80	4.19	3.77	6.15	3.95	7.12	5.47	4.95	11.05

Une étude de la distribution des sens pour les 10 fichiers considérés est présentée dans le tableau 2.13. On observe des entropies presque égales pour les noms et les adjectifs et des valeurs basses pour les adverbes. La plus grande entropie est manifestée par les verbes, ce qui explique leur degré de difficulté plus élevé. En moyenne, les entropies pour *Semcor* comportent des valeurs plus petites que pour *Senseval2* (sauf pour les verbes).

Tab. 2.13. Entropie de la distribution de sens par catégorie grammaticale, corpus *Semcor*

	EntropyNN	**EntropyVB**	**EntropyADJ**	**EntropyADV**
test0_1	0.81	1.91	0.91	0.66
test2_3	0.92	2.23	1.01	0.87
test4_5	0.86	1.96	0.98	0.81
test6_7	0.95	2.02	1.17	0.44
test8_9	0.93	1.98	1.10	0.58
test10_11	1.06	1.93	1.16	0.61
test12_13	1.40	1.90	1.10	0.64
test14_15	1.10	1.79	0.89	0.57
test16_17	1.13	1.95	1.34	0.80
test18_19	1.22	1.95	1.19	0.64
Moyenne	**1.04**	**1.96**	**1.09**	**0.66**
Ecart (σ)	0.1723	0.1059	0.1317	0.1230
Coef var	16.5992	5.3992	12.1288	18.5890

c) Distribution réelle des sens

La distribution réelle des sens, calculée à partir des fichiers clés, indique en général (89% des cas), la même tendance des mots à se manifester dans un seul sens par discours, observée aussi pour le corpus *Senseval.* Les mots à plusieurs sens (11%) comportent une distribution assez équilibrée (en moyenne 2.39 sens / mot, le sens dominant intervenant dans 57% des cas).

Tab.2.14. Distribution réelle des sens, corpus *Semcor*

Fichier	Total mots	Mots différents	Nombre moyen de sens par mot	Taux moyen des mots à plusieurs sens %	Nombre moyen de sens par mot pour les mots à plusieurs sens	Fréquence relative moyenne du sens dominant pour les mots à plusieurs sens	Intervalle moyen de cooccurrence du même sens dans la même phrase	Taux moyen des phrases à cooccurrence du même sens %	Intervalle moyen de cooccurrence des sens différents du même mot dans la même phrase	Taux moyen des phrases à cooccurrence des sens différents du même mot %	Intervalle moyen de cooccurrence du même sens	Intervalle moyen de cooccurrence des sens différents du même mot
key0_1	2037	929	1 15	11 30	2 33	0 56	7 49	19 55	7 60	2 79	62 06	34 47
key2_3	2152	846	1 18	13 00	2 41	0 59	10 32	26 97	9 75	2 81	73 22	54 01
key4_5	2035	913	1 16	11 61	2 43	0 57	8 06	24 22	4 67	1 35	58 54	45 61
key6_7	2113	1055	1 16	11 27	2 42	0 56	6 49	17 32	6 06	3 90	38 43	31 45
key8_9	2076	1230	1 12	9 26	2 38	0 55	6 02	13 88	2 33	1 44	28 03	24 68
key10_11	2115	1128	1 15	12 05	2 25	0 57	7 56	20 62	12 00	2 06	54 01	34 62
key12_13	1996	950	1 17	12 42	2 40	0 58	13 13	25 87	10 17	6 99	54 44	47 11
key14_15	2100	907	1 17	13 12	2.32	0 59	13 80	35 06	14 96	8 44	63 38	47 61
key16_17	2114	995	1 20	13 66	2 50	0 57	9 06	14 50	6 00	2 50	56 20	47 97
key18_19	2226	1042	1 14	10 07	2 42	0 57	8 39	13 71	13 50	2 03	65 30	45 15
Moyenne (M)	**2096.40**	**1000**	**1.16**	**11.78**	**2.39**	**0.57**	**9.03**	**21.17**	**8.70**	**3.43**	**55.36**	**41.27**
Ecart (σ)	62 15	110 25	0 02	1 30	0 06	0 01	2 50	6 55	3 85	2 28	12 54	8 82
Coef Var (σ*100)/M	2 96	11 03	1 78	11 08	2 73	2 34	27 72	30 95	44 29	66 37	22 66	21 37

Pour ce qui est de la répétitivité des sens, on peut constater, malgré les variations d'un fichier à l'autre, une proportion de 21% des phrases avec co-occurrence d'un même sens et de 3.43% des phrases contenant des sens différents du même mot. Ces valeurs, un peu plus élevées que pour *Senseval2*, montrent quand même la même caractéristique : la répétition d'un sens dans une phrase est plus fréquente que l'occurrence des sens différents d'un mot, dans la même phrase. Les intervalles de répétitivité globale et au niveau de la phrase sont, plus petits que dans le cas de *Senseval2,* et ceci principalement parce que les signes de ponctuation n'ont pas été pris en compte pour l'extrait du corpus *Semcor*. On peut observer ainsi, un intervalle de répétitivité des sens différents dans la même phrase plus petit que celui d'un seul sens, si les signes de ponctuation sont éliminés.

2.2.4. Quelques graphiques comparatifs

L'utilisation de plusieurs fichiers de test a pour but l'étude du comportement du système pour des textes différents mais renfermant des propriétés, dans une certaine

mesure, similaires. Les diagrammes suivants résument certaines caractéristiques des données de test, présentées plus en détail dans les sections précédentes.

Le premier diagramme compare le nombre total de mots à désambiguïser et le nombre de mots par catégorie grammaticale, pour les corpus de test *Senseval* 2 et *Semcor*. Les données représentées ci-dessous indiquent des structures comparables du point de vue de leur contenu quantitatif.

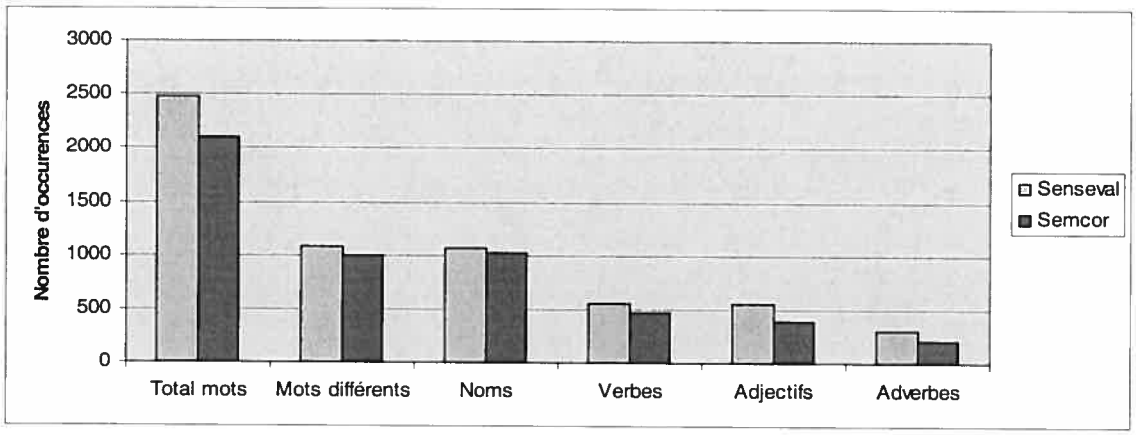


Fig. 2.8. Description quantitative des corpus de test,
en terme de nombre de mots

On peut constater une certaine similarité du nombre moyen de sens par mot (catégorie grammaticale connue APOS et inconnue NPOS) et du degré de polysémie par catégorie grammaticale.

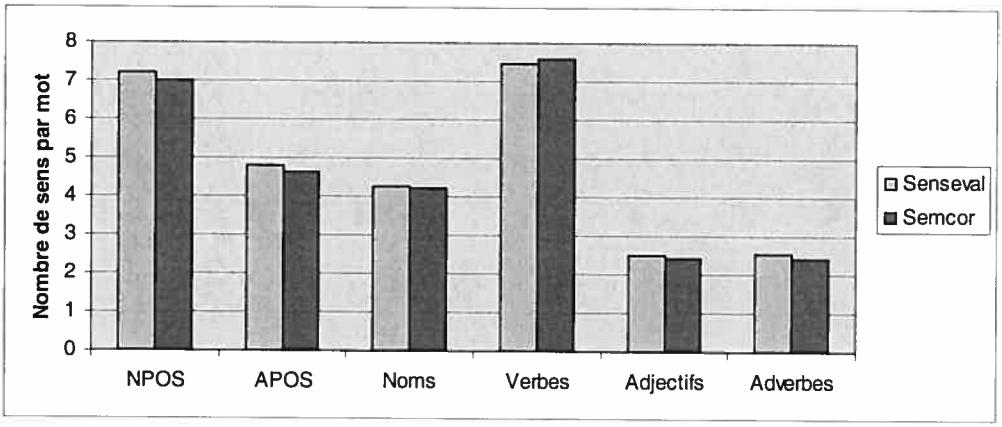


Fig.2.9. Nombre de sens par mot pour les corpus de test,
valeurs globales et par catégorie grammaticale

Le diagramme 2.10 comporte une représentation comparative des précisions de base (catégorie grammaticale à priori connue ou non APOS / NPOS). Les valeurs ci-dessous

indiquent des précisions supérieures pour le corpus *Semcor*, en tant que performances globales et par catégorie, des valeurs assez rapprochées pour les noms et les adverbes, mais des différences entre 10 et 20 % pour les adjectifs et les verbes.

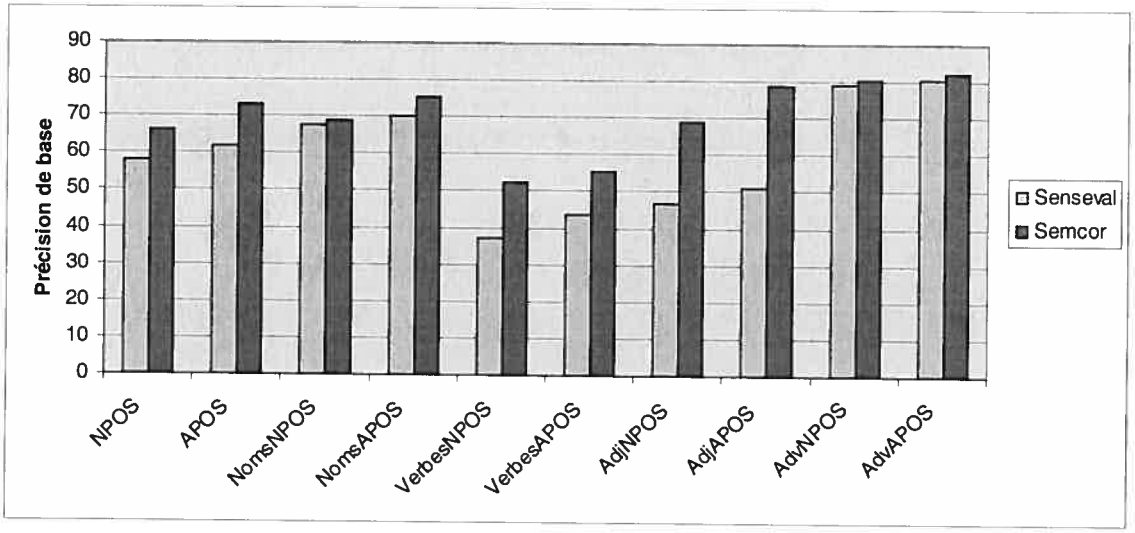


Fig. 2.10. Précision de base pour les corpus de test,
valeurs globales et par catégorie grammaticale

Description de l'architecture

Les implémentations décrites dans ce mémoire comportent, en plus du traitement proprement dit, visant la désambiguïsation, une série de procédures auxiliaires destinées au prétraitement des données ainsi qu'à l'analyse des résultats. Le but de ce chapitre est de présenter l'architecture globale de notre système et d'en décrire les différentes composantes.

Le système est constitué de 3 modules : le premier dédié au prétraitement des données de test et à l'extraction des informations du dictionnaire, le deuxième destiné à la désambiguïsation sémantique basée sur les algorithmes décrits dans le chapitre 4, le troisième s'occupant de l'évaluation des performances et de l'analyse des résultats.

Bien que certains détails de format et de procédure présents dans ce chapitre ne soient d'intérêt qu'au lecteur désirant reproduire nos expériences, nous décrivons dans ce chapitre les éléments nécessaires à la compréhension globale de notre démarche, tels que:

- la détection de la catégorie grammaticale à partir d'une forme instanciée (3.1.1– a1);
- l'ordonnancement des sens candidats selon leur fréquence d'usage et la construction de la variante de *baseline* (3.1.1– a3);
- les exemples de définitions et de relations extraites de *WordNet* (3.1.2);
- la description de la méthode d'évaluation *coarse-grained* que nous avons implémentée (3.3.1 – b).

Les détails techniques qui ne sont pas nécessaires à la compréhension de notre propos, sont regroupés en annexe.

Une représentation schématique de la structure de notre système est illustrée dans la figure 3.1. Les données externes (base lexicale *WordNet*, fichiers de test en forme brute, fichiers clés *Senseval*) sont représentées dans ce diagramme par les cylindres en dégradé, les cylindres simples désignent les fichiers produits par les divers modules du système tandis que les rectangles renferment les procédures.

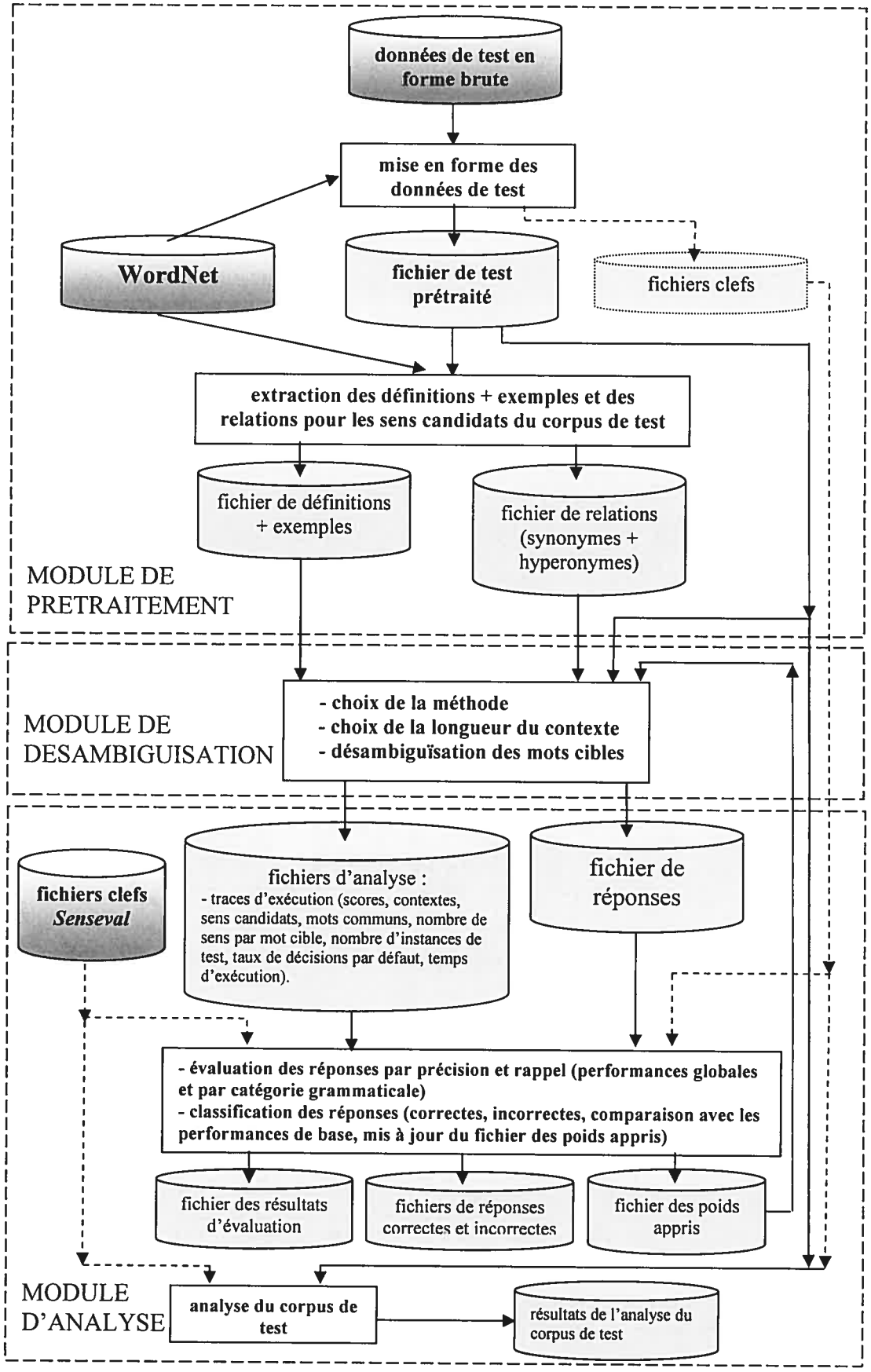


Fig. 3.1. Architecture du système

3.1. Le module de prétraitement

Notre système a été prévu pour fonctionner selon plusieurs modes commandés par des paramètres (méthodes de désambiguïsation différentes, plusieurs longueurs du contexte, différents types de descriptions de sens etc.) et le traitement préalable des données de test s'est imposé comme une exigence d'un fonctionnement en "temps réel", pour chaque variante testée. Le prétraitement comporte deux phases que nous détaillons dans les sections suivantes : la première a pour but la mise en forme des données de test et la création du fichier clé[1], la deuxième effectue la génération des fichiers de définitions et de relations pour tous les sens candidats du corpus de test.

3.1.1. Mise en forme des données de test

Les expériences que nous avons effectuées utilisent deux ensembles de test distincts : un provenant des données de test de *Senseval2* - approche tous les mots -, l'autre étant extrait du corpus annoté *Semcor1.6* (voir chapitre 2 pour un détail descriptif de ces ressources). Pour des raisons d'uniformité, le système convertit vers un format commun (le fichier pré-traité) ces deux types différents de données d'entrée.

a) Prétraitement du fichier de test *Senseval* 2

La procédure de prétraitement transforme chaque instance à désambiguïser dans une ligne du fichier de test prétraité (voir l'annexe 1 pour plus de détails sur le format du corpus *Senseval2* et sur les fonctions spécifiques de traitement du format). Dans ce qui suit on décrit les fonctions plus générales, accomplies par la procédure de prétraitement :

a1) - Transformer en formes de base (lemmes) orthographiées en lettres minuscules toutes les formes instanciées des mots cibles simples et des parties des mots composés (voir Annexe1). La condition pour inclure une instance cible dans le fichier de sortie est de trouver sa forme de base dans *WordNet*. Lorsque cette condition n'est pas vérifiée,

[1] Pour le cas des fichiers de test extraits de *Semcor*.

l'instance n'est pas inclue dans le fichier prétraité[2], fait qui est signalé par un message d'erreur.

Pour certaines instances, la procédure détecte aussi la catégorie grammaticale du mot à désambiguïser, en utilisant les règles morphologiques et les fichiers d'exceptions de *WordNet* (voir le chapitre 2). Si le système trouve, par exemple, l'instance à désambiguïser *is,* il déduit qu'il s'agit du verbe *be* et pas du nom *be*[3], seulement les sens correspondants à la catégorie de verbe étant considérés comme potentiels candidats pour le mot. Dans les cas où il n'est possible de prendre aucune décision sur la catégorie grammaticale (par exemple l'instance *work),* tous les sens candidats du mot cible sont considérés. Pour des raisons de compatibilité avec le cadre *Senseval2,* cette procédure de détection de la catégorie grammaticale à partir d'une forme instanciée n'est pas activée si on génère le fichier de test utilisé comme base de référence (où on choisit toujours le sens le plus fréquent d'un mot, sans s'intéresser à sa catégorie grammaticale).

a2) - Extraire de *WordNet* tous les sens candidats possibles pour les formes de base obtenues au point précédent. Pour les cas où la catégorie grammaticale est connue (soit a priori[4], soit après le traitement décrit plus haut), seulement les sens correspondant à cette catégorie sont ajoutés à la ligne de sortie.

a3) - Ordonner les sens candidats selon leur fréquence d'usage. Dans *WordNet*, les sens d'un mot sont ordonnés selon leur fréquence, par catégorie grammaticale. Par exemple, le mot *work* qui possède 7 sens en tant que nom et 27 en tant que verbe, présente deux ordonnancements différents pour les deux catégories, nom et verbe.

Comme nos implémentations s'appuient sur un ordonnancement global des sens d'un mot, sans tenir compte de son appartenance à une certaine catégorie grammaticale (dans le cas cité, ordonner les sens de *work* de 1 à 34), nous avons construit une fonction qui réalise cette tâche, à partir de l'information de *WordNet.*

Dans *WordNet* les sens des mots pour chaque catégorie grammaticale sont ordonnés[5] par l'intermédiaire d'un nombre décimal (*sense_number)* indiquant le rang selon la fréquence (*sense_number* = 1 dénote le sens le plus fréquent, 2 le deuxième sens etc.). De plus, il y a un autre indicateur (*tag_cnt*) qui compte le nombre de fois le sens apparaît

---

[2] Ce sont des cas qui produisent une valeur plus petite du rappel par rapport à la valeur de la précision parce que le nombre d'instances traitées est plus petit que le nombre de cas à traiter.
[3] Symbole pour *beryllium* dans *WordNet.*
[4] Voir chapitre 2, la description du corpus de test
[5] Voir *WordNet 1.7.1 Reference Manual* http://www.cogsci.princeton.edu/~wn/doc.shtml, sections senseidx(5WN) et wndb(5WN).

dans des textes sémantiquement annotés. La valeur 0 de *tag_cnt* indique le fait que le sens n'a pas été taggé.

Le tableau 3.1 montre quelques exemples de *sense_number* et *tag_cnt* pour le mot *work* dans *WordNet.*

Tab. 3.1. Exemples de *sense_number* et *tag_cnt* pour les sens du mot *work* dans WordNet

Nom			Verbe		
Identificateur	*sense_number*	*tag_cnt*	Identificateur	*sense_number*	*tag_cnt*
work#1	1	435	work#1	1	77
work#2	2	359	work#2	2	62
...	...		...	...	...
work#6	6	18	work#6	6	9
work#7	14	0	...	...	...
			work#9	9	3
			...	...	...
			work#11	11	3
			...	...	...
			work#27	27	0

Pour ordonner les sens d'un mot, de manière globale, indépendante de la catégorie grammaticale, nous avons construit une fonction d'interclassement qui prend en compte les deux indicateurs, *sense_number* et *tag_cnt,* fournis par le fichier d'index de *WordNet.* L'utilité de la combinaison des deux indicateurs devient plus évidente dans les cas d'égalité de *sense_number* ou *tag_cnt,* que les indicateurs utilisés séparément n'arriveraient pas à résoudre. Il n'est pas simple à priori de déterminer le sens le plus fréquent de *work,* i.e. *work#6* comme nom ou *work#6* comme verbe (même *sense_number* selon *WordNet,* voir Tab. 3.1) ou encore *work#9* ou *work#11* (sens verbaux comportant le même *tag_cnt).*

La formule de calcul de la fonction, exprimée comme un indicateur de fréquence, est la suivante:

$$freq\_id(s_j) = 0.6 \cdot tag\_cnt - 0.4 \cdot sense\_number \qquad (4)$$

où s_j, $j = 1,n$ représentent tous les sens du mot à désambiguïser, indifféremment de la catégorie grammaticale.

Afin d'utiliser seulement des indicateurs de fréquence positifs, les valeurs négatives du *freq_id* ont été éliminées par l'ajout de la quantité *(offset)* égale à $|\ min(freq\_id(s_i))| + 1$, $i=1,n$ (la valeur 1 a été ajouté pour éviter les indicateurs de fréquence nulle). Le choix du premier sens classé par l'intermédiaire de cet indicateur détermine les performances de *baseline,* considérées comme référence.

a4) – Enregistrer, dans le fichier de sortie, une ligne pour chaque mot à désambiguïser, contenant : la ou les formes de base et les sens candidats ordonnés selon

leur fréquence et un identificateur de référence ultérieurement utilisée pour identifier, dans le fichier de réponses, le sens choisi par le système pour une instance donnée (voir Annexe 1).

b) Prétraitement du corpus de test extrait de *Semcor 1.6.*

La procédure de prétraitement pour le corpus extrait de *Semcor* est similaire avec celle présentée pour *Senseval2*. Les seules différences proviennent de la manière de générer le fichier de test et du format des données.

Avant le prétraitement proprement dit, pour faciliter la génération des fichiers de test, notre programme produit des combinaisons de deux fichiers *Semcor* placés dans le même répertoire. La méthode de sélection des deux fichiers est la suivante : à partir d'un nombre i entre 0 et $N-1$ (N = le nombre de fichiers *Semcor* dans le répertoire considéré), et d'un nombre $j = i+c, j \neq i, j < N$ (c est une constante à valeurs prédéfinies), on combine les fichiers avec l'index i et j dans la liste des fichiers du répertoire. La combinaison est choisie si les caractéristiques du fichier concaténé (nombre de mots à désambiguïser, nombre de noms, de verbes etc.) sont comparables aux caractéristiques du corpus *Senseval2* (voir Chapitre 2), ce qui confère une structure assez homogène à l'ensemble des fichiers[6] de test utilisé pour la validation des expériences.

Une fois établi le contenu du fichier de test[7], la procédure extrait des fichiers composants toutes les instances annotées par une étiquette de sens en leur assignant un indicateur de référence, semblable à celui utilisé pour *Senseval2* (voir l'annexe 2 sur les détails de mise en forme du corpus extrait de *Semcor*).

Dans certains cas, les formes instanciées peuvent servir à déterminer la catégorie grammaticale, en utilisant les règles morphologiques[8] encodées dans le système (comme déjà mentionné pour *Senseval2*, cette procédure n'est pas active si le fichier de test est destiné au calcul des performances de base).

La procédure de lemmatisation est plus simple que pour *Senseval2,* la forme de base du mot cible étant extraite directement du corpus (voir Annexe 2). Il y a pourtant des cas très rares ou la forme de base extraite de *Semcor1.6*[9] ne correspond à aucune entrée de

---

[6] Au total, 10 fichiers concaténés extraits de *Semcor* et le fichier de test *Senseval2*.

[7] Par exemple *test0_1* englobe les données des fichiers *Semcor1.6. br-a01* et *br-a02.*

[8] Voir chapitre 2, la description de *WordNet.*

[9] *Semcor* 1.6. a été annoté à partir de la version 1.6. de *WordNet.*

WordNet 1.7.1, que nous avons utilisé comme inventaire de sens, ce qui produit une diminution du rappel. Pour le traitement qui considère la catégorie grammaticale a priori connue (APOS), la catégorie grammaticale peut être également extraite du corpus (voir Annexe 2).

L'information de *Semcor* est aussi utilisée dans cette phase de prétraitement pour générer le fichier clé correspondant à l'ensemble de test. A chaque instance à désambiguïser correspond une ligne dans le fichier clé (voir Annexe 2).

Les autres fonctions de la procédure de prétraitement sont identiques à celles décrites au 3.1.1, points a2*), a3), a4).*

3.1.2. Extraction des définitions et des relations de *WordNet*

Une fois le fichier de test mis en forme, l'étape suivante consiste à extraire de *WordNet* les descriptions de tous les sens contenus dans le fichier prétraité (Fig. 3.1.), afin de réduire le temps d'exécution de la phase de désambiguïsation proprement dite. Le type d'entité descriptive (définition + exemples ou relations) recherché dans le dictionnaire représente un paramètre de la procédure d'extraction, à côté du nom du fichier de test mis en forme.

a) Extraction des définitions et des exemples

La procédure d'extraction consiste à parcourir ligne par ligne le fichier de test prétraité et à chercher dans *WordNet* les définitions et les exemples d'usage pour chaque sens candidat. L'information trouvée est soumise à un processus de sélection et de normalisation, puis est enregistrée, sous forme de *sac de mots,* dans le fichier des définitions où chaque ligne comporte un sens et sa description. Les mots retenus font partie des catégories ouvertes (noms, verbes, adjectives et adverbes), les mots fonctionnels (conjonctions, prépositions, déterminants et pronoms) étant exclus de la description. De plus, pour une comparaison facile entre les descriptions de sens dans la phase de désambiguïsation, tous les mots sont réduits à leur forme canonique. Par exemple, le sens 1 de *rejection* est décrit dans *WordNet* par la définition et l'exemple[10] présentés ci-dessous :

[10] Les exemples d'usage sont encadrés entre guillemets dans *WordNet*.

rejection#1 **:** *the act of rejecting something; "his proposals were met with rejection"*

Après la sélection et la normalisation, la ligne de sortie correspondant à *rejection#1* a la forme :

rejection#1 [act, be, meet, proposal, reject, rejection, something]

Les expériences que nous avons effectuées ont fonctionné seulement sur lemmes, les formes instanciées n'ont été utilisées que pour déterminer les lemmes ou, dans certains cas pour détecter la catégorie grammaticale d'un mot cible.

b) Extraction des relations

Afin d'étendre le type des descriptions des sens, nous avons utilisé les relations de synonymie et d'hyperonymie pour les sens présents dans le fichier prétraité. La procédure d'extraction cherche horizontalement dans *WordNet* les synonymes et verticalement[11] les hyperonymes d'un sens, jusqu'à la racine de la hiérarchie.

Ces types de relations, pour le sens *rule#8 (dominance or power through legal authority),* sont présentés dans la figure 3.4:

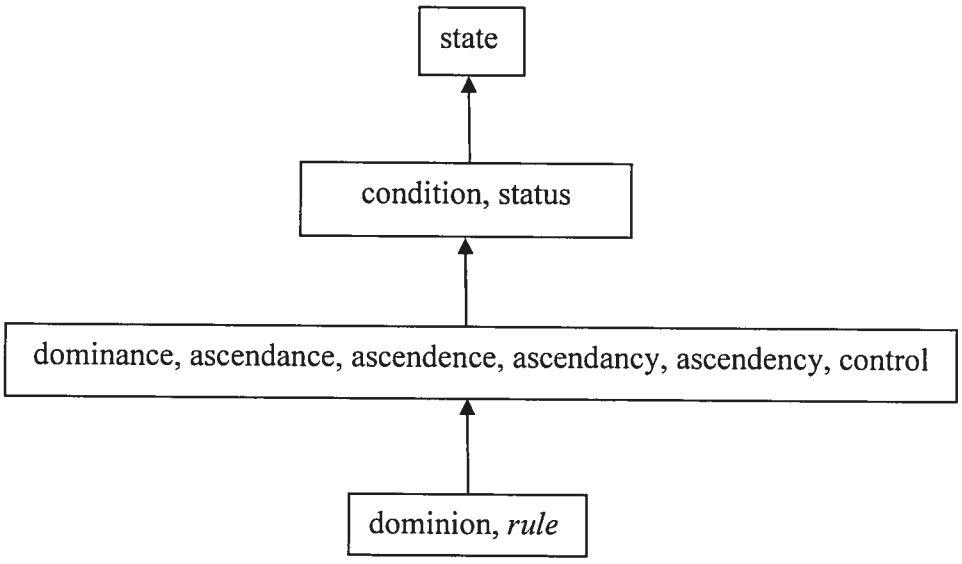


Fig. 3.2. Relations extraites de *WordNet* pour le sens *rule, dominion*

[11] La recherche des hyperonymes est faite seulement pour les noms et les verbes (les adjectifs et les adverbes ne comportent pas de hiérarchie dans *WordNet).*

Dans la figure 3.2, les rectangles représentent des *synsets* (regroupement de concepts synonymes) tandis que les flèches désignent les relations d'hypéronymie entre *synsets*.

La ligne de sortie enregistrée dans le fichier de relations, comporte, pour le sens analysé dans la figure 3.2, l'étiquette de sens et le sac de mots associé (l'ordre n'est pas préservé):

rule#8 [ascendance, ascendancy, ascendence, ascendency, condition, control, dominance, dominion, rule, state, status]

3.2. Le module de désambiguïsation

Comme le montre la figure 3.1, le module de désambiguïsation prend en paramètre d'entrée le fichier de test mis en forme et utilise les descriptions de sens produites dans la deuxième phase de prétraitement. Le type de description (définitions + exemples, relations ou définitions + exemples + relations), la méthode ainsi que la longueur du contexte représentent d'autres paramètres d'entrée dont le programme a besoin pour choisir le meilleur candidat d'un mot cible. Les algorithmes de désambiguïsation que nous avons implémentés sont décrits dans le chapitre 4.

De plus, le module de désambiguïsation contient une version de base qui choisit toujours le sens le plus fréquent, sans tenir compte de contexte et sans détecter la catégorie grammaticale d'un mot à désambiguïser à partir de ses formes instanciées dans le corpus de test. Le principe de fonctionnement de cette version est très simple : comme tous les sens candidats d'un mot cible ont été préalablement ordonnés dans la phase de prétraitement, le programme choisit le premier sens de la liste candidate.

La sortie du module de désambiguïsation consiste dans le fichier de réponses, où chaque ligne comporte un identificateur de référence et l'étiquette du sens candidat choisi par le système (pour plus de détails sur la forme de la ligne de sortie, voir Annexe 3).

Pour des fins d'analyse, le module de désambiguïsation produit divers fichiers renfermant des traces d'exécution : pour chaque instance cible, son identificateur de référence, sa forme de base et le nombre de sens correspondant, le contexte et la description pris en compte pour la désambiguïsation, les superpositions trouvées et le score pour chaque sens candidat, le sens choisi et son score.

Le programme fournit aussi des informations globales, caractérisant le traitement d'un ensemble de test donné : nombre d'instances à traiter, taux de décisions par défaut (choix du sens le plus fréquent), temps d'exécution. Tous ces renseignements sont destinés à l'analyse du système, de manière automatique ou semi-automatique, décrite dans la section suivante.

3.3. Le module d'analyse

Le module d'analyse comporte deux parties : l'une dédiée à l'évaluation du comportement du système pour un ensemble de test donné, l'autre conçue pour l'analyse du corpus de test, dont les caractéristiques ont été présentées dans le chapitre 2. Les sections suivantes décrivent plus en détail ces deux composantes.

3.3.1. Etude des réponses du système

La première composante du module d'analyse est celle qui évalue les réponses du système, produits pendant la phase de désambiguïsation. L'évaluation a comme premier objectif la comparaison du fichier de réponses avec le fichier clé et le calcul des performances, en terme de précision et rappel[12], pour l'ensemble de mots à tester et par catégorie grammaticale. Selon le modèle d'évaluation de *Senseval 2*[13], ces mesures de performances ont été calculées à la fois pour un ordre de granularité fine *(fine-grained)* ainsi que pour une granularité moins fine *(coarse-grained)* du découpage des sens.

a) Evaluation *fine-grained*

La procédure compare le fichier de réponses et le fichier clé en comptant comme réponse correcte seulement celle identique ou inclue[14] dans la réponse clé, pour un mot

[12] Voir chapitre 2.
[13] Voir *Senseval-2 Scoring*, http://www.sle.sharp.co.uk/senseval2/Scoring/
[14] Dans le cas de *Senseval2, English All Words,* il y a des instances de test qui comportent plusieurs réponses correctes, séparées par espace.

cible donné. Puis, la précision et le rappel sont calculés comme le rapport entre le nombre de réponses correctes et le nombre de cas traités, respectivement le nombre de cas à traiter.

b) Evaluation *coarse-grained*

Le but de ce type d'évaluation est de compenser les découpages de sens trop fins, difficiles à cerner même pour les humains, chose souvent reprochée à *WordNet*. L'idée est de regrouper les sens dans des classes moins fines, selon certains critères, d'habitude de nature syntaxique et sémantique, et d'évaluer les réponses du système par rapport aux nouvelles catégories ainsi obtenues. Plusieurs critères de regroupement ont été proposés.

Palmer et al. (2002) discutent la possibilité de regrouper les verbes qui tendent à apparaître dans des structures syntaxiques alternatives reflétant, habituellement, des similarités sémantiques. C'est, par exemple, le cas des sens *eat#1, eat#2* ou *chill#1, chill#2* qui peuvent se manifester dans des structures syntaxiques de type *verbe + objet indéfini / verbe sans objet* (*We **ate#1** fish and chips / We **ate#2** at noon)* ou qui exprime l'aspect *cause / effet (He **chilled#1** the soup / The soup **chilled#2**)*. De plus, les verbes peuvent être regroupés s'ils représentent des versions spécialisées d'un sens plus général comme dans le cas de ***develop#2*** *(create by mental act)* considéré comme une variante spécialisée de ***develop#1*** *(create)*.

Crestan et al. (2001) utilisent le regroupement de sens à partir de l'appartenance à une même classe sémantique[15] de *WordNet.* Par exemple ***sense#1*** *(a general conscious awarness)* et ***sense#4*** *(sound practical judgement)* appartiennent à la même classe sémantique, selon *WordNet,* celle des noms dénotant des processus cognitifs. Une autre alternative serait le regroupement en fonction des relations spécifique/générique entraînant un ou deux niveaux de la hiérarchie *WordNet* (voir le cas du *develop#1* et *develop#2* sus-mentionné).

Les auteurs attirent l'attention sur le fait que le niveau de finesse nécessaire pour une évaluation *coarse-grained* n'est malheureusement pas le même pour tous les mots.

---

[15] Regroupement des mots dans des catégories plus larges, par exemple : noms dénotant des processus cognitifs, noms exprimant des sentiments ou des émotions, noms désignant les parties du corps, verbes de communication, verbes de changement, verbes de la possession et du transfert de la possession, adjectifs relationnels, adjectifs provenant des participes etc.

C'est peut-être la principale raison pour laquelle le regroupement des sens pour l'évaluation *coarse-grained* de *Senseval2* a été fait manuellement par des annotateurs.

La solution que nous avons adoptée est complètement automatique et s'appuie sur les relations de synonymie et d'hyperonymie de *WordNet.* Notre procédure d'évaluation considère comme corrects les cas où le sens choisi par le système et le sens clé sont soit identiques ou synonymes, soit dans une relation hyponyme / hyperonyme de premier ordre, impliquant un seul niveau de la hiérarchie *WordNet*[16].

Par exemple, ***work_out#1, work_up*** *(come up with; "His colleagues worked out his interesting idea"; "We worked up an ad for our client")* et ***work_out#3, elaborate*** *(work out in detail; "elaborate a plan")* ont le même hypéronyme d'ordre supérieur ***develop, make grow*** et sont considérés comme représentant le même concept dans l'évaluation *coarse-grained.*

c) Classification des réponses

Une autre tâche accomplie par le module d'analyse est la classification des réponses du système, pour un ensemble de test donné.

En utilisant le fichier de réponses et les réponses de la version de base (choix du sens le plus fréquent) ainsi que les traces d'exécution générées pendant le processus de désambiguïsation, la procédure de classification renferme les fonctions suivantes :

- calculer la précision et le rappel par catégorie grammaticale ;
- séparer les réponses correctes et incorrectes ;
- déterminer le nombre de réponses correctes communes avec les réponses de la version de base, le nombre de décisions correctes prises par défaut, le nombre de décisions incorrectes à cause de ne pas choisir le sens le plus fréquent, le gain par rapport à la version de base etc[17].

Ces informations, enregistrées dans des fichiers de sortie, nous servent pour interpréter le comportement de notre système.

---

[16] Plus précisément, l'un est l'hyperonyme d'ordre immédiatement supérieur de l'autre ou ils ont le même hyperonyme d'ordre immédiatement supérieur.

[17] Une description plus détaillée de ce type d'analyse est présentée dans le chapitre 5.

d) Mise à jour du fichier des poids appris

La procédure de mise à jour du fichier des poids, décrite dans le chapitre 4, est exécutée pendant la phase d'analyse. Chaque analyse de résultats, pour un corpus de test donné, produit une modification des poids et du nombre d'occurrences des mots superposés, selon l'information fournie par les traces d'exécution et le fichier de réponses.

3.3.2. Analyse du corpus de test

L'étude des performances du système suppose non seulement l'évaluation des résultats mais aussi une analyse de la tâche à résoudre. C'est pour ce motif que nous avons développé un module supplémentaire d'analyse (voir Fig. 3.1), qui, à partir des informations fournies par le fichier de test prétraité et le fichier clé, produit une caractérisation des données de test (voir Chapitre 2).

Description des algorithmes

Ce chapitre présente les algorithmes développés dans le cadre du projet ainsi que les principales études reliées à notre recherche. Cette description s'attache principalement à la méthode de Lesk telle que présentée dans (Lesk 1986) ainsi qu'aux variantes de la méthode que nous avons adaptées à *WordNet*, et à d'autres travaux basés sur l'algorithme de Lesk ou sur l'exploitation de *WordNet* pour la désambiguïsation automatique.

4.1. Algorithme de Lesk

La méthode de désambiguïsation proposée par Lesk (1986) fait partie de la catégorie des méthodes de désambiguïsation basées sur les connaissances. A la différence des méthodes supervisées utilisant des corpus annotés (où chaque occurrence d'un mot polysémique dans le corpus est annotée par une étiquette de sens), les méthodes non supervisées font appel à d'autres types de ressources, d'habitude de nature lexicale et sémantique disponibles par l'intermédiaire d'un dictionnaire électronique. L'avantage de ces méthodes tient principalement au fait qu'elles ne nécessitent pas de corpus annoté, une ressource assez difficile à construire.

L'idée de la désambiguïsation basée sur la superposition (ang. *overlap*) consiste simplement à compter les mots communs entre les définitions des sens d'un mot ambigu (ces définitions proviennent d'un dictionnaire) et les définitions des mots apparaissant dans le contexte du mot à désambiguïser. On choisit alors le sens caractérisé par le plus grand nombre de superpositions.

Par exemple, si on considère la cooccurrence des mots anglais *pine* et *cone* dans le même contexte, un programme de désambiguïsation automatique, basé sur cette idée simple, serait capable de choisir le sens *arbre* du mot *pine* en comptant les intersections entre les différentes définitions de sens des deux mots : "**pine** – *1. kind of* ***evergreen tree***

with needle-shaped leaves ...; 2. waste away through sorrow or illness ...; **cone** – *1. solid body which narrows to a point ... ; 2. something of this shape whether solid or hollow ...; 3. fruit of certain* ***evergreen tree*** *...".* Dans ce cas, le nombre maximal de mots communs (*evergreen, tree*) est donné par l'intersection entre les définitions 1, respectivement 3 de *pine* et *cone,* ce qui détermine le choix du sens correspondant pour le mot *pine* soumis à l'analyse. Lesk mentionne que ses programmes traitent de manière séquentielle les mots à désambiguïser (à un moment donné, on compare les définitions des sens d'un mot cible avec toutes les définitions de chaque mot du contexte). Il suggère cependant que, une fois la décision prise sur le sens d'un mot, seulement la définition de ce sens soit prise en compte pour les désambiguïsations ultérieures, des autres mots. Les sections suivantes portent sur quelques points que Lesk même considère importants pour une meilleure compréhension de son algorithme.

4.1.1. Information syntaxique. Contexte local / global.

Lesk souligne à juste titre la simplicité de son approche et met en avant certains de ses avantages : premièrement, elle ne s'appuie pas sur des connaissances de nature syntaxique; deuxièmement, elle n'est pas dépendante de l'information globale, tenant du domaine de discours (sport, politique, religion etc.).

Il y a de nombreuses situations où la discrimination des sens peut se faire sans information d'ordre syntaxique. Dans l'exemple "1. *I have a* ***mole*** *on my skin;* 2. *There is a* ***mole*** *tunnelling in my lawn;* 3. *They built a* ***mole*** *to stop the waves",* le contexte et les définitions des 3 sens sont suffisamment informatifs pour déterminer le choix du sens correct (1. tache de la peau; 2. mammifère qui creuse des galeries dans le sol; 3. construction à l'entrée d'un port pour briser les vagues).

Cependant, la méthode peut s'avérer mauvaise dans certains cas où seulement l'information syntaxique peut aider à la désambiguïsation. Par exemple, dans la phrase "*I know a* ***hawk*** *from a handsaw",* la distinction automatique entre le sens nominal de *hawk* (oiseau de proie) et celui verbal (offrir de petites marchandises) est dictée par des considérants de nature syntaxique tel que: un verbe ne peut pas apparaître après un article.

Pour ce qui est de l'information globale (le sujet du texte), Lesk semble suggérer la priorité du contexte local sur le contexte global. Pour argumenter, il prend l'exemple des 9

occurrences de *reef* dans un fragment de *Moby Dick*, où les 7 cas reliés à *sail* (voile, en navigation) pourraient conduire au choix erroné (ris, voile d'un navire qui peut être serrée pour diminuer l'action du vent) pour les 2 instances *coral reef* (récif), si l'information locale n'est pas prise en compte.

4.1.2. Qualité du dictionnaire

Un autre aspect important de l'approche est la qualité du dictionnaire utilisé comme source des définitions de sens. Lesk teste à cet effet le comportement de son programme, en fonction de 4 dictionnaires : *Oxford Advanced Learner's Dictionary of Current English* (OALDCE), *Merriam-Webster 7th New Collegiate* (W7), *Collins English Dictionary* (CED) et *Oxford English Dictionnary* (OED) dont les caractéristiques générales sont présentées dans le tableau 4.1 :

Tab. 4.1. Taille des dictionnaires utilisés dans (Lesk 1986)

	OALDCE	W7	CED	OED
Taille (MB)	6,6	15,6	21,3	350
Nombre d'entrées	21000	69000	85000	304000
Nombre de sens	36000	140000	159000	587000
Octets / entrée	290	226	251	1200

Sans surprise, Lesk conclut qu'un dictionnaire comportant une quantité supérieure d'information par entrée serait potentiellement capable de faire plus de distinctions valides entre les sens d'un mot polysémique, qu'un dictionnaire disposant d'un matériau moins riche. Par exemple, il rapporte que pour le mot *galley* dans le contexte de la phrase *"stoke the stove in the galley"*, OALDCE ne fournit aucune intersection de sens entre *galley* et *stove* et produit une désambiguïsation incorrecte. En revanche, OED, incluant dans la définition de sens 2 de *galley* des mots tels que : *stove, cook, cooking-room* et *pot,* mène à un bon résultat.

De manière intéressante, il observe que l'information véhiculée par les exemples d'usage présents dans un dictionnaire n'est que de peu utilité dans le processus de désambiguïsation (ces exemples sont habituellement des extensions des définitions qui sont déjà suffisamment longues).

4.1.3. Calcul des scores et longueur du contexte

Lesk étudie différentes façons de compter les recouvrements des entrées d'un dictionnaire et remarque qu'il n'y a pas de différences significatives entre les variantes de score comptant tout simplement les mots communs entre les définitions de sens, et les variantes pondérées par la taille de l'entrée dans le dictionnaire. De plus, il affirme que les longueurs variables du contexte (4, 6, 8, 10 mots) ne produisent pas en pratique de résultats essentiellement différents. Par contre, d'autres questions sont laissées sans réponse dans son étude, comme par exemple : la limitation de la fenêtre de contexte au cadre de la phrase, l'effet de la pondération du score d'un mot par l'inverse de sa distance au mot à désambiguïser, la prise en compte des sens déjà assignés dans le processus de désambiguïsation. Certains de ces problèmes ont constitué des points de départ de nos recherches.

4.1.4. Performances

Lesk rapporte des performances (ang. *accuracy*) de sa méthode de 50 à 70 % pour de petits fragments de textes extraits de *Pride and Prejudice* et d'*Associated Press,* sans fournir de détails sur les modalités d'évaluation utilisées pour mesurer les performances de son système.

4.2. Travaux connexes

Le but de cette section est de présenter d'autres approches basées sur l'algorithme de Lesk ou sur l'exploitation de *WordNet,* afin de mieux retracer l'encadrement comparatif de notre recherche. Il s'agit principalement de systèmes testés lors de *Senseval1* ou *2*, ainsi que de quelques autres travaux postérieurs à ces campagnes d'évaluation.

4.2.1. Senseval 1

Le premier exercice d'évaluation des systèmes de désambiguïsation automatique, organisé en 1998 (Kilgarriff 1998), a réuni 17 systèmes participants pour trois langues.

Leur tâche (ang. *lexical sample*) consistait à trouver les sens corrects d'un ensemble de mots préalablement sélectionnés pour lesquels la catégorie grammaticale (sauf pour la classe des indéterminés) et des données d'entraînement étaient également fournies. Chaque mot correspondait à une tâche, i.e. un fichier comportant plusieurs instances du mot à désambiguïser et un contexte d'une ou deux phrases pour chaque instance. Le nombre total d'instances à désambiguïser a été de 8448, une caractérisation par catégorie grammaticale de ces données étant présentée dans le tableau 4.2.

Tab. 4.2. Nombres d'instances et de mots à désambiguïser pour *Senseval1, selon* (Kilgarriff et Rosenzweig 2000b)

Noms		**Verbes**		**Adjectifs**		**Indéterminés**		**Total**	
Instances	*Mots*	*Instances*	*Mots*	*Instances*	*Mots*	*Instances*	*Mots*	*Instances*	*Mots*
2756	15	2501	13	1406	8	1785	5	8448	41 (35 mots différents[1])

Le texte suivant représente un échantillon du fichier de test pour le mot *rabbit*, chaque instance à désambiguïser étant identifiée par un nombre de référence :

```
700002
LIFE IN an American orchestra can produce some peculiar contrasts.
A few weeks ago the Pittsburgh Symphony found itself giving a pair of
concerts for the Disney channel with Roger <tag>Rabbit</> (the one who
got framed).

700003
And grand it is to be sure.
The gardens of Ireland have a special dreamlike quality, like gardens
known as a child &dash. where everything was bigger and greener, and
chattering <tag>rabbits</> abounded.
```

Fig. 4.1. Extrait du corpus de test de *Senseval1* pour le nom *rabit*

Les systèmes participants ont été étiquetés *supervisés* (qui ont besoin de données d'entraînement pour chaque mot à désambiguïser) et *non-supervisés*, certains systèmes supervisés étant capables de fonctionner en régime non-supervisé, dans le cas où aucune instance annotée n'est disponible. Les résultats ont été comparés avec différents types de performances de base (choix aléatoire, choix du sens dominant, Lesk). Les meilleures *baselines,* qui ont montré des performances difficiles à surpasser par la majorité des systèmes, étaient des variantes de l'algorithme de Lesk (Kilgarriff et Rosenzweig 2000b), que nous élaborons dans les sections suivantes.

[1] 6 mots ont été utilisés pour plusieurs catégories grammaticale.

a) Lesk simple

Lesk simple, utilisé comme base de comparaison pour la catégorie des systèmes non-supervisés, est une variante de l'algorithme originel de Lesk qui choisit pour un mot à désambiguïser le sens dont la définition du dictionnaire (+ exemples d'usage) comporte le plus grand nombre de superpositions avec les mots du contexte, plutôt qu'avec les définitions des mots du contexte.

Plus formellement, soit t le mot à désambiguïser et $\{s_j\}$ $j=1,n$ ses n sens candidats. Si on considère le contexte formé par la phrase p, comportant les mots w_i, $i=1,k$, alors le score de chaque sens candidat est déterminé par la formule (5), le meilleur candidat comportant le plus grand score :

$$score(s_j) = \sum_{i=1}^{k} poids(w_i) \tag{5}$$

avec

$$poids(w_i) = \begin{cases} -log(p(w_i)), si\ w_i \in D(s_j) \\ 0, sinon \end{cases} \tag{6}$$

où

$D(s_j)$ représente l'ensemble de mots composant la définition du sens s_j (+ exemples d'usage) et $p(w_i)$ est la fraction des définitions et des exemples du dictionnaire qui contient le mot w_i :

$$p(w_i) = \frac{nb.\ de\ définitions\ et\ d'\ exemples\ du\ dictionnaire\ qui\ contiennent\ w_i}{nb.\ total\ de\ définitions\ et\ d'\ exemples\ du\ dictionnaire} \tag{7}$$

b) Lesk definitions

Lesk definitions représente la même variante que la précédente, sauf qu'elle n'utilise pas les exemples du dictionnaire. Comme la variante *Lesk simple*, elle a constitué une base de référence pour les systèmes non-supervisés.

c) Lesk plus corpus

Utilisé comme référence pour les systèmes supervisés, *Lesk plus corpus* est une variante proche de *Lesk simple.* La différence majeure réside dans le fait que le programme teste ici l'occurrence du mot w_i non seulement dans le dictionnaire (définition et exemples

du sens s_j) mais aussi dans un des contextes où s_j apparaît dans le corpus d'entraînement. Dans ce cas $p(w_i)$ est calculée pour la distribution du mot w_i à la fois dans le dictionnaire et dans le corpus.

d) Performances

Les conclusions majeures rapportées par (Kilgarriff et Rosenzweig 2000a, b) sont les suivantes : les meilleures performances ont été enregistrées par des systèmes supervisés qui atteignent des taux de précision et rappel de l'ordre de 77%. Des performances différentes sont observées pour les noms (80%), les verbes (70%), et les adjectifs et les autres catégories (entre 70 et 80%). De manière intéressante, voire surprenante, les variantes de l'algorithme de Lesk, utilisées comme base de référence, ont surpassé la majorité des systèmes participants à la compétition. Aucun système n'a été capable de dépasser avec plus de 2% les performances de base (Lesk) pour la sous-tâche de désambiguïsation des verbes.

4.2.2. Senseval 2

Senseval2 est le deuxième exercice d'évaluation, organisé en 2001, pour 12 langues, 94 systèmes participants et 3 types de tâches (approche lexicale – *lexical sample task*, tous les mots – *all words* et traduction). Dans ce mémoire nous nous intéressons seulement aux approches monolingues et ne détaillons donc que les deux premières tâches.

a) *Senseval 2,* anglais - approche lexicale

La tâche lexicale de *Senseval2* ressemble dans les grandes lignes à celle de *Senseval1*. Les différences tiennent principalement aux données de test et d'entraînement et à l'inventaire de sens utilisé.

L'ensemble de données pour *Senseval1* a été fourni par HECTOR[2], une base de données à double profil : dictionnaire et corpus. A chaque mot de HECTOR correspond une entrée de dictionnaire et des étiquettes de sens pour toutes ses occurrences dans un corpus de 17 MB (extrait de *BNC*). Les mots ont été choisis selon leur nombre d'occurrences dans

[2] HECTOR est un projet de recherche de Oxford University Press et DEC (Atkins 1993).

le corpus, i.e. ceux qui possèdent entre 300 et 1000 occurrences (Kilgarriff et Rosenzweig 2000b).

Le corpus utilisé pour *Senseval2* a été extrait de *BNC-2, Penn Treebank*[3], l'inventaire de sens étant fourni par *WordNet1.7.* Toutes les instances à désambiguïser appartenaient à une des 3 catégories grammaticales : nom, verbe ou adjectif. La tâche de détecter la catégorie grammaticale (le cas des indéterminés dans *Senseval1)* n'a plus été mise en place pour *Senseval2.* Le tableau 4.3 montre la structure du corpus de test, par catégorie grammaticale, nombre d'instances et nombre de mots à désambiguïser.

Tab. 4.3. Nombres d'instances et de mots à désambiguïser pour *Senseval2, lexical sample,* selon (Banerjee et Pedersen 2002)

Noms		**Verbes**		**Adjectifs**		**Total**	
Instances	*Mots*	*Instances*	*Mots*	*Instances*	*Mots*	*Instances*	*Mots*
1754	29	1806	29	768	15	4328	73 (mots distincts)

Les résultats de cet exercice pour l'anglais, tâche lexicale, ont été plus bas que ceux rapportés dans *Senseval1.* Plusieurs explications ont été proposées, que nous reprenons ici. L'inventaire de sens de *Senseval2* est un dictionnaire proprement dit (*WordNet*), pas spécialement conçu pour la désambiguïsation, comme dans le cas de HECTOR utilisé pour *Senseval1.* De plus, l'accord entre les annotateurs comporte un taux de 10% moins élevé que pour *Senseval1* ce qui suppose une tâche plus difficile même pour les humains[4].

Un autre aspect concernant la difficulté de la tâche, vise le degré moyen de polysémie, plus grand pour *Senseval2*, spécialement dans la catégorie des verbes. De plus, les systèmes ont bénéficié d'une quantité de données d'entraînement plus importante dans le cas de *Senseval1* que dans celui de *Senseval2* (Palmer et al. 2002).

Il y a aussi une explication, souvent invoquée, visant la granularité des sens trop fine de *WordNet* (Voorhees 1998), (Véronis 2001), (Preiss et al. 2002), (Edmons 2002).

A l'instar de *Senseval1*, les systèmes participants ont été comparés avec plusieurs systèmes de référence (choix aléatoire, choix du sens le plus fréquent, variantes de l'algorithme de Lesk). Cette fois, les systèmes ont présenté, en général, un comportement comparable aux meilleures performances (précision et rappel) de base pour les systèmes supervisés (Lesk *corpus* – 51.2%) et non-supervisés (*Lesk simple* - 22.6%). Plus de la

[3] Voir aussi http://info.ox.ac.uk/bnc.
[4] Nous nous sommes servi du site Senseval 2, *Review of the Workshop* pour compiler ces explications.

moitié des systèmes ont enregistré des performances supérieures à ces *baselines* (meilleur système supervisé – 64.2%; meilleur système non-supervisé – 40.2/40.1%).[5]

Comme nous nous intéressons aux approches basées sur l'idée de Lesk, nous allons faire référence à deux études qui, à partir des résultats de *Senseval2,* discutent la contribution de l'information de type Lesk (définitions + exemples d'usage) à la désambiguïsation sémantique.

Les systèmes décrits par (Haynes 2001), *IIT1* et *IIT2,* ont obtenu à *Senseval2 – English lexical sample* des performances de 24.3/23.9% et respectivement 24.7/24.4 %, en terme de précision et rappel. Les deux systèmes sont basés sur l'utilisation des exemples d'usages disponibles dans certaines entrées de *WordNet.* Le calcul du score s'appuie sur la comparaison entre les exemples provenant de tous les ancêtres[6] d'un sens candidat et le contexte du mot à désambiguïser. Les cas où aucun exemple n'est disponible sont résolus par la construction de pseudo-exemples à partir des définitions *(glosses)* de *WordNet* (cette alternative semble cependant insuffisante en pratique). Haynes conclut que la méthode pourrait produire de meilleurs résultats en combinaison avec d'autres techniques.

Une conclusion similaire est faite par (Litkowski 2001), qui présente une étude sur les différents facteurs influençant les performances de son système (*CL Research – DIMAP,* 29.3%, précision et rappel à *Senseval2, English lexical sample*). Ses expériences effectuées sur deux inventaires de sens, *WordNet* et *New Oxford Dictionary of English (NODE),* indiquent un meilleur comportement pour *NODE,* qui détermine un nombre plus petit de choix par défaut (sens le plus fréquent) en l'absence de l'information nécessaire à la désambiguïsation. De plus, Litkowski fait l'observation que seulement 30% des instances à désambiguïser tirent profit de l'information de type Lesk (définitions + exemples) et par conséquent, cette valeur reflète la proportion dans laquelle l'information fournie par les dictionnaires pourrait contribuer à la discrimination correcte des sens. C'est une observation assez intéressante qui pourrait expliquer, en partie, les résultats de nos propres expériences.

b) Senseval 2, anglais - tous les mots

La section *tous les mots* de *Senseval2* consistait, quant à elle, à désambiguïser tous les mots appartenant aux catégories grammaticales ouvertes (nom, verbe, adjectif et

[5] Site *Senseval 2, Official Results.*
[6] Déterminés par l'intermédiaire de la relation d'hypéronymie.

adverbe). Comme performances de base ont été considérés les résultats d'un système qui choisit toujours le sens le plus fréquent, sans tenir compte de contexte. Dans le cas de l'anglais, un tel système obtenait 57% de précision et rappel. Des 22 systèmes testés, seulement 4 ont été capables de dépasser cette performance de base. Les meilleurs systèmes ont produit des précisions et des rappels de 69% (supervisés) et de 57.5%,56.9% (non-supervisés)[7]. Les systèmes *IIT1,2,3* (Haynes 2001) et *CL Research - DIMAP* (Litkowski 2001), basés sur l'information de type Lesk extraite d'un dictionnaire, ont enregistré des précisions et des rappels de 29.4%/29.1% à 33.5%/33.2% *(IIT1,2,3)* et de 45.1%/45.1 *(DIMAP).*

4.2.3. D'autres recherches dérivées de l'idée de Lesk

Nous présentons dans cette section des travaux récents inspirés de l'approche Lesk qui nous ont servi de référence dans nos propres expériences.

a) Comparaison floue des mots

(Sidorov et Gelbukh 2001) appliquent la désambiguïsation de sens aux définitions d'un dictionnaire espagnol, considérées comme contexte. Plus précisément, il s'agit de trouver les sens corrects des mots ambigus intervenant dans les définitions d'autres mots. Par exemple, déterminer le sens correct du mot *órgano* (*organ*)[8] dans la définition du mot *glàndula* = *órgano que segrega substancias indispensables para el organismo* (*glandula* = *an organ that segregates indispensable substances for an organism).* La modification que Sidorov et Gelbukh apportent à l'algorithme de Lesk consiste en une comparaison floue des mots, basée sur l'emploi d'un dictionnaire de synonymes et d'un système de morphologie dérivationnelle simple.

D'une manière plus formelle, soit t le mot à désambiguïser qui apparaît dans la définition h d'un autre concept. Soit s_j, $(j = 1,n)$ les sens possibles de t, chacun représenté par leur définition dans le dictionnaire. Le score de chaque sens candidat s'appuie sur le calcul de la *mesure de proximité* $w(s_j,h)$ entre s_j et h (vus comme deux ensembles de mots) :

$$w(s_j,h) = \sum_{x \in s_j, y \in h} w(x,y) \qquad (8)$$

---

[7] Site *Senseval 2, Official Results.*

[8] Les exemples en espagnol et leur traduction en anglais proviennent du texte originel.

où

$$w(x,y) = \begin{cases} 1 \text{ si } x = y \\ 0.5 \text{ si } x \text{ et } y \text{ sont synonymes} \\ 0.5 \text{ si les 5 premières lettres de } x \text{ et } y \text{ coïncident} \\ 0 \text{ dans les autres cas} \end{cases} \quad (9)$$

Sidorov et Gelbukh rapportent un taux d'erreurs de 13% pour 50 entrées testées, à la différence de 17% et respectivement 29% taux d'erreurs dans le cas de l'algorithme de Lesk originel et du choix du sens le plus fréquent. Pourtant, les auteurs attirent l'attention sur le fait que ce type de désambiguïsation des définitions de dictionnaire est plus simple que la désambiguïsation de textes réels, étant donnée l'information sur la catégorie grammaticale contenue dans la définition et la longueur du contexte automatiquement limitée à la taille de la définition. De plus, leur métrique n'est pas normalisée par la longueur de la définition, une limite dont il n'est pas fait mention dans le papier.

b) Combinaison de sens

(Banerjee et Pedersen 2002) proposent une autre variante de la méthode de Lesk, basée sur l'évaluation de toutes les combinaisons de sens des mots co-occurant dans le même contexte. Par exemple, soit N la taille du contexte considéré et w_i $(i=1,N)$ les mots du contexte (y compris le mot à désambiguïser). Par combinaison candidate de sens, on désigne un vecteur à N dimensions comportant à un moment donné un sens pour chacun des mots w_i du contexte. Si $|w_i|$ représente le nombre de sens possibles de w_i, alors, il existe $\prod_{i=1}^{N} |w_i|$ combinaisons candidates possibles. L'idée est d'attribuer un score à chaque combinaison candidate, en choisissant pour le mot à désambiguïser le sens appartenant à la combinaison de sens comportant le score maximal.

Pour chaque combinaison candidate, l'algorithme compare les définitions de sens *(glosses)* de *WordNet* correspondant à toutes les paires de mots dans la fenêtre de contexte. Si N est la taille de la fenêtre de contexte, alors il y a $N(N-1)/2$ paires de mots à comparer. Soit (w_i, w_j), $i,j = 1,N$ une paire de mots du contexte et $s_k(w_i)$ et $s_q(w_j)$ les deux sens *(synsets)* assignés à w_i et w_j dans la combinaison candidate considérée. Les définitions de sens à comparer à un moment donné, sont identifiées par une série de *paires de relations* entre les *synsets* candidats des deux mots. Si la paire de relation à évaluer est *synset-hyperonyme,* alors on compare la définition du *synset* $s_k(w_i)$ avec la définition du *synset*

hypéronyme de $s_q(w_j)$. D'autres types de relations sont également considérées à partir des relations d'hyponymie, d'holonymie, de méronymie, de troponymie et d'attribut des *synsets* $s_k(w_i)$ et $s_q(w_j)$. Une fois toutes les comparaisons[9] possibles étant accomplies pour chaque paire de mot de la fenêtre de contexte, on calcule la somme des scores individuels de chaque comparaison pour obtenir le score d'une combinaison candidate de sens. Le processus est répété jusqu'à ce que toutes les combinaisons candidates soient considérées. La combinaison comportant le score maximal est gagnante et le sens du mot à désambiguïser appartenant à celle-ci est assigné à ce mot. Comme la méthode a été appliquée à l'ensemble de test de *Senseval2, lexical sample* (un seul mot à désambiguïser pour un contexte donné), l'assignation des sens aux autres mots du contexte, considérée comme secondaire, n'a pas été évaluée.

Pour ce qui est de la comparaison des *glosses,* (Banerjee et Pedersen 2002) proposent une acception nouvelle du concept de superposition (*overlap*), défini comme la plus longue séquence commune de mots consécutifs entre deux définitions soumises à l'analyse. Par exemple, les phrases : 1) *"he called for an end to the atrocities"* et 2) *"after bringing an end to the atrocities, he called it a day"* comportent deux superpositions : *"an end to the atrocities"* et *"he called"*. Par contre, les séquences telles que *"of the"*, formées seulement par des mots fonctionnels (pronoms, conjonctions, prépositions et articles), ne sont pas acceptables.

Un autre élément notable de cet algorithme est le score attaché à chaque superposition. Soit n le nombre de mots de la séquence commune, alors l'algorithme y attribue un score quadratique = n^2, qui favorisent les superpositions à plusieurs mots par rapport à celles simples, de mots isolés. Par exemple, le score de la superposition *"he called"* est 4 et il serait 2 si les mots *"he"* et *"called"* apparaissaient séparément.

(Banerjee et Pedersen 2002) rapportent des performances de 31.7% pour l'approche lexicale de *Senseval2*, comparativement aux performances de base (*Lesk simple* - 22.6%, *Lesk definitions* – 16%) et du meilleur système de la catégorie non-supervisée (40%). Ils attirent cependant l'attention que la référence à considérer pour comparer leurs résultats serait plutôt *Lesk definitions* que *Lesk simple,* leur algorithme utilisant seulement les définitions et pas les définitions et les exemples (une variante qu'ils envisagent de tester).

---

[9] Comme il s'agit de 7 types de relations, il y a donc 49 paires possibles de relations à considérer pour une paire de mots donnée.

c) Optimisation de type *simulated annealing*

Une autre approche basée sur la combinaison de tous les sens possibles des mots co-occurant dans le même contexte a été développée par (Stevenson et Wilks 2001). Bien que leur système soit un ensemble hybride de plusieurs modules (filtre de catégorie grammaticale, extracteur des définitions d'un dictionnaire et calcul d'une fonction d'*overlap*, analyseur des traits sémantiques, caractérisation du domaine), ils présentent une analyse sur la contribution de chaque module aux performances globales du système.

L'algorithme implémenté pour le calcul de la fonction d'*overlap* s'appuie sur l'approche de (Cowie, Guthrie et Guthrie 1992) qui tente de déterminer la combinaison optimale des sens des mots dans une phrase, en comptant le nombre de mots communs entre les définitions de tous les sens d'une combinaison, à un moment donné. La méthode d'optimisation qu'ils appliquent est connue sous le nom de *simulated annealing* et elle ne calcule pas toutes les combinaisons de sens possibles, en essayant de trouver une solution approximative. L'algorithme n'est pas garanti de trouver toujours la solution optimale mais, en général, les solutions proposées ne sont pas significativement différentes de celle optimale. Nous décrivons ici l'algorithme utilisé par (Cowie, Guthrie et Guthrie 1992).

Soit N le nombre de mots composant la phrase p. Chaque mot w_i, $(i=1,N)$ de la phrase comporte plusieurs sens, et soit $s_k(w_i)$, un des sens de w_i. Une configuration C du système est un vecteur contenant à un moment donné un sens pour chacun des mots de p. A chaque configuration C possible, l'algorithme attribue une fonction E calculée à partir des définitions des sens qui compose C. La combinaison optimale C^{opt} correspond à la valeur minimale de E, i.e. à la valeur donnée par le plus grand nombre de mots communs entre les définitions des sens candidats.

La procédure est itérative et suppose le calcul d'une nouvelle valeur de E à chaque pas. La configuration initiale C^{init} contient les sens les plus fréquents de chaque mot de la phrase et correspond à une valeur calculée E^{init}. Soit C la configuration de sens à un moment donné. Un pas de l'algorithme consiste à choisir aléatoirement un mot w_j de p et un sens $s_q(w_j)$ et de remplacer dans C le sens précédent de w_j par $s_q(w_j)$. On obtient ainsi une nouvelle configuration C' et une nouvelle valeur correspondante, E'. Soit $\Delta E = E' - E$. Si $\Delta E < 0$, la configuration C est remplacée par C' et le processus continue par un nouveau choix aléatoire de sens dans C'. Si $\Delta E \geq 0$, alors on calcule $P = e^{-\frac{\Delta E}{T}}$ (T est une constante,

initialement $T=1$). L'algorithme génère un nombre aléatoire et remplace C par C' seulement si ce nombre est inférieur à P, sinon C est retenue. Le processus de générer de nouvelles configurations et de faire un choix est répété et à chaque itération $T \leftarrow 0.9T$. Si aucune modification ne survient après un certain nombre d'itérations, alors la procédure s'arrête sur une configuration finale C^{fin} et les sens correspondants sont attribués aux mots de la phrase.

Pour éliminer la "préférence" de l'approche pour les définitions longues, la variante de (Stevenson et Wilks 2001) fait appel à une normalisation du score par la taille en mots de la définition.

Dans leurs expériences, Stevenson et Wilks ont utilisé, comme dictionnaire et inventaire de sens, le *Longman Dictionary of Contemporary English* (LDOCE) et, comme corpus de test, 5 articles de *Wall Street Journal* annotés à la main, contenant 391 mots (de catégorie grammaticales ouverte) à désambiguïser.

Les performances enregistrées pour le module traitant les définitions de dictionnaire sont de 65.24% de désambiguïsations correctes, ce qui représente un gain absolu d'approximativement 35% par rapport aux performances de base (choix du sens le plus fréquent) de 30.9% pour le corpus de test utilisé.

d) Tableau récapitulatif des résultats

Cette section reprend par un tableau récapitulatif (Tab. 4.4) les résultats rapportés pour les travaux dérivés de l'idée de Lesk, décrits dans les paragraphes a)-c):

Tab. 4.4. Tableau récapitulatif des résultats des approches récentes dérivées de Lesk

Approche ou système	Corpus de test	Résultats rapportés
Comparaison floue des mots (Sidorov et Gelbukh 2001)	50 entrées d'un dictionnaire espagnol	13% taux d'erreur par rapport à 17% Lesk originel et 29% choix du sens le plus fréquent
Combinaison de sens (Banerjee et Pedersen 2002)	*Senseval2, lexical sample*	31.7% précision par rapport à 22.6% Lesk simple et 16% Lesk definitions
Optimisation de type *simulated annealing* (Stevenson et Wilks 2001)	5 articles de *Wall Street Journal* (391 mots pleins à désambiguïser)	65.24% précision par rapport à 30.9% choix du sens le plus fréquent

4.3. Algorithmes implémentés dans le cadre du projet

Les algorithmes que nous avons implémentés s'appuient principalement sur la méthode originelle de Lesk (1986) et sur sa variante simplifiée décrite par Kilgarriff et Rosenzweig (2000a,b), adaptée aux caractéristiques constructives de *WordNet.* Chaque variante comporte des versions non-pondérées, comptant tout simplement les mots communs, et pondérées, incluant dans le calcul du score, la taille de l'entité descriptive ou la fréquence d'usage des mots intervenant dans la désambiguïsation. Deux autres variantes, l'une supervisée, basée sur l'apprentissage des poids à partir des désambiguïsations antérieures, et l'autre non-supervisée, basée sur l'idée de chaînes lexicales (Hirst et St-Onge 1998) ont également été implémentées.

Les entités descriptives de sens utilisées dans la désambiguïsation, sont d'un côté, les définitions et les exemples d'usages, et d'un autre, les relations de type synonymie et hyperonymie, ainsi qu'une combinaison des deux (définitions + exemples et relations). Pour des raisons de concision, toutes nos références ultérieures au terme *définition* renfermeront la combinaison *définition + exemples d'usage.*

Chaque entité descriptive est représentée par un *sac de mots,* i.e. une collection de mots dont l'ordre et la dépendance sont ignorés. Les implémentations permettent le choix du type de description, de la fenêtre de contexte et supposent que les sens candidats du mot à désambiguïser ont été préalablement ordonnés, en ordre décroissant de leur fréquence d'usage. L'ordonnancement des sens est effectué pendant la phase de prétraitement[10], à partir de l'information disponible dans *WordNet.*

En ce qui concerne la dépendance entre les sens déjà assignés et ceux en cours d'assignation, les programmes comportent une variante où les sens sont déterminés de manière séquentielle, un, après l'autre, sans relation entre eux. C'est à dire, on choisit le meilleur candidat pour un mot cible donné, sans que ce choix influence ultérieurement la désambiguïsation du mot suivant. Une deuxième alternative que nous avons implémentée tente de capter la "réciprocité" entre les sens des mots du même contexte par l'intermédiaire d'un tableau de votes. Plus précisément, les votes sont attribués mutuellement, par paires *mot cible – mot du contexte*. Par exemple, pour un sens candidat du mot cible, on compte les superpositions avec tous les sens d'un mot pris du contexte. Si le nombre de

[10] Voir chapitre 3.

superpositions n'est pas vide, alors le sens candidat "reçoit" un vote, mais aussi il "accorde" un vote au meilleur sens du mot du contexte, avec lequel il partage le plus grand nombre de superpositions. Les votes sont enregistrés dans un tableau de votes et à la fin du traitement de l'ensemble de test, on assigne à chaque mot à désambiguïser le sens candidat qui a obtenu le nombre maximal de votes.

4.3.1. Algorithme Lesk de base

Avant de présenter l'idée de base de l'algorithme de Lesk, nous introduisons par un schéma les notations utilisées. La figure 4.2 représente le mot à désambiguïser *t*, dans son contexte *C(t)* (centré autour de *t*) qui contient entre autres mots le mot *w*. A chaque sens s_j de *t* correspond une définition $D(s_j)$ dans le dictionnaire. Le mot *w* est représenté dans le dictionnaire par la réunion des définitions de ses sens, *E(w)*.

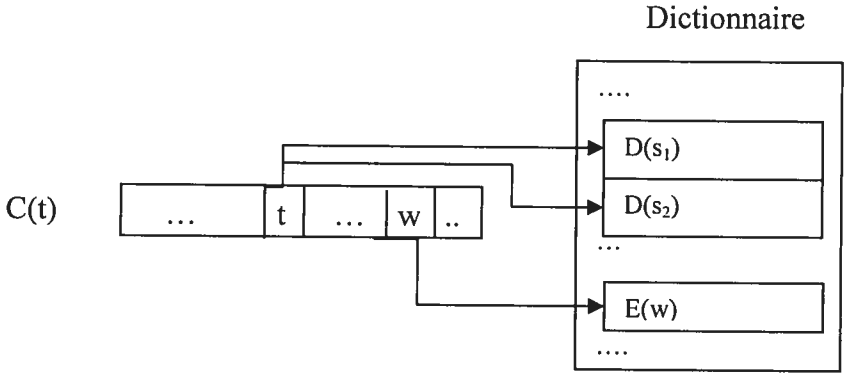


Fig. 4.2. Schéma de l'algorithme de Lesk de base

L'interprétation que nous avons donnée à l'algorithme de Lesk et que nous avons implémentée est décrite par le pseudo-code suivant.

1. pour chaque mot à désambiguïser *t*
2. *best_score* = 0
3. *best_candidate* = s_1 (le sens le plus fréquent)
4. déterminer *C(t)* le contexte de *t*
5. pour chaque sens candidat s_j de *t*
6. extraire du dictionnaire la définition $D(s_j)$
7. *sup* = 0
8. pour chaque mot *w* du contexte *C(t)*
9. extraire du dictionnaire les définitions de ses sens *E(w)*
10. calculer le nombre de superpositions $sup = sup + \left| D(s_j) \cap E(w) \right|$
11. si *best_score* < *sup*
12. *best_score* = *sup*
13. *best_candidate* = s_j
14. attribuer à *t* le sens donné par *best_candidate*

Fig. 4.3. Algorithme de Lesk, variante de base

Les définitions $D(s_j)$ et $E(w)$ sont des sac de mots (lemmes) qui n'appartiennent à aucune des 4 catégories fermées : prépositions, conjonctions, pronoms où déterminants. La réduction à la forme de base des mots (lemme) est une étape réalisée pendant la phase de prétraitement, décrite dans le chapitre précédent. Le nombre de superpositions entre la définition d'un sens candidat et la définition d'un mot du contexte est calculé comme le nombre d'éléments de l'intersection des deux sacs de mots $\left|D(s_j) \cap E(w)\right|$. Pour chaque mot à désambiguïser, l'algorithme assigne initialement, comme meilleur candidat, le sens le plus fréquent (s_1, le premier dans l'ordre des sens). Un autre sens est choisi si et seulement si son score est supérieur à celui du meilleur candidat courant.

Dans notre implémentation, les entités descriptives *D(sj), E(w)* peuvent désigner des définitions + exemples d'usage, des relations ou la combinaison définitions + exemples + relations, éléments extraits de *WordNet*. Le type de relation utilisée pour la désambiguïsation sont la synonymie et l'hyperonymie. Dans ce cas, $D(s_j)$ renferme les mots (en forme de base) provenant des *synsets* synonymes et de tous les *synsets* ancêtres, en relation d'hyperonymie avec le sens s_j. Un ancêtre est tout hypéronyme qui domine le synset s_j (sur n'importe quel niveau, de s_j jusqu'à la racine) dans la hiérarchie de *WordNet*[11]. Pour le cas des relations, $E(w)$ comporte la réunion des relations de tous les sens du mot w. Le même principe du sac de mots s'applique si on considère comme entités descriptives à la fois les définitions et les relations prises ensembles. Alors $D(s_j)$ représente la réunion des mots de la définition et des relations du sens s_j, tandis que $E(w)$ englobe les mots des définitions et des relations de tous les sens de w. Toutes les références ultérieures à $D(s_j)$ et $E(w)$ engloberont ce caractère composite : définitions, relations ou définitions et relations (voir 3.1.2. pour un exemple de définition et de relations extraites de *WordNet).*

4.3.2. Algorithme de Lesk simplifié

La variante simplifiée est similaire à celle de base, la seule différence est que pour le calcul du score on compte les superpositions entre l'entité descriptive du sens candidat $D(s_j)$ et les mots du contexte w (et non plus leur définitions). Soit $C(t)$ la fenêtre de contexte formée par le sac de mots w, en forme de base, alors la représentation en pseudo-code de l'algorithme devient :

[11] Voir chapitre 2 pour une description de l'organisation de *WordNet.*

pour chaque mot à désambiguïser t
    $best\_score = 0$
    $best\_candidate = s_1$
    $sup = 0$
    déterminer $C(t)$ le contexte de t
    pour chaque sens candidat s_j de t
        extraire du dictionnaire la définition $D(s_j)$
        calculer le nombre de superpositions $sup = \left|D(s_j) \cap C(t)\right|$
        si $best\_score < sup$
            $best\_score = sup$
            $best\_candidate = s_j$
    attribuer à t le sens donné par $best\_candidate$

Fig. 4.4. Algorithme de Lesk, variante simplifiée

4.3.3. Normalisation du score par la taille de description de sens

Une des suggestions de Lesk (1986) concernant le calcul du score visait la prise en compte de la taille de l'entrée du dictionnaire pour un sens donné. Nous avons testé deux façons d'intégrer cette information lors du calcul de score d'un sens candidat : la normalisation par l'inverse de la taille de l'entité descriptive $D(s_j)$ et la normalisation par log_2 de la taille de $D(s_j)$. La raison de ces normalisations réside dans le fait que les descriptions trop longues tendent à dominer les plus courtes, la variante logarithmique rendant cette normalisation moins forte.

Soit $|D(s_j)|$ la longueur en mots de $D(s_j)$. Alors, l'algorithme devient :

pour chaque mot à désambiguïser t
    $best\_score = 0$
    $best\_candidate = s_1$
    déterminer $C(t)$ le contexte de t
    pour chaque sens candidat s_j de t
        calculer sup par une des modalités de Fig. 4.2. ou Fig. 4.3.
        $sup1 = \dfrac{sup}{|D(s_j)|}$ ou $sup1 = \dfrac{sup}{log_2 |D(s_j)|}$
        si $best\_score < sup1$
            $best\_score = sup1$
            $best\_candidate = s_j$
    attribuer à t le sens donné par $best\_candidate$

Fig. 4.5. Algorithme de Lesk, variante normalisée par la taille de la description de sens

4.3.4. Contributions des mots pondérés par la distance du mot cible et par la fréquence d'usage

Le facteur de normalisation (taille de la description de sens) utilisé dans la section précédente était relié au sens candidat, en cours d'analyse. Ici, nous considérons un poids caractérisant la contribution d'un mot à la désambiguïsation, apport exprimé par l'inverse de la distance *d(w, t)* entre le mot *w* du contexte et le mot cible (à désambiguïser) *t*, et par l'inverse de la fréquence absolue *f(w)*.

La distance *d(w,t)* représente la distance en mots (noms, verbes, adjectifs et adverbes) entre *w* et *t*. Quant à la fréquence absolue de *w* elle a été calculée à l'aide d'un fichier comptant les occurrences des mots dans le corpus *Hansard*, les textes parlementaires canadiens (N = 31.637.784 instances au total dans la version utilisée ici). Nous avons appliqué le calcul suivant :

$$f(w) = \begin{cases} |w| \; si \; |w| \neq 0 \\ \\ 1 \; si \; |w| = 0 \end{cases} \qquad (10)$$

où |w| représente le nombre d'occurrences de *w* dans le corpus *Hansard*.

Pour les cas des mots inconnus *(|w| = 0)* on considère qu'ils apparaissent une fois dans le corpus *Hansard*.

Nous avons utilisé deux métriques du calcul des poids. La première est exprimée par l'inverse de la fréquence absolue $\frac{1}{f(w)}$. Ce choix est motivé par l'intention de pénaliser les mots trop fréquents (de type *be, have, can* etc.) qui interviennent d'habitude dans les définitions et qui ne sont pas supposés apporter d'information utile à la discrimination des sens (une solution équivalente serait l'utilisation d'une *stop liste* décrite plus loin dans la section 4.3.8). En pratique, nous avons observé que cette métrique fonctionne mieux dans sa forme logarithmique pondérée par l'inverse de la distance *d(w,t)*, mesure qui tente de favoriser les mots du contexte plus proches du mots cible.

L'algorithme de désambiguïsation ressemble à l'algorithme de base, décrit dans la figure 4.3, dont nous reprenons seulement les lignes 8 à 10 :

8. pour chaque mot w_i du contexte $C(t)$
9. extraire du dictionnaire la définition $E(w)$
10. $sup=sup+\frac{1}{f(w)}\cdot\left|D(s_j)\cap E(w)\right|$ ou $sup=sup+\frac{1}{d(w,t)\cdot\log_2(f(w))}\cdot\left|D(s_j)\cap E(w)\right|$

Fig. 4.6. Algorithme de Lesk, variante de base pondérée

La version simplifiée est similaire, sauf que la ligne 9 disparaît et les lignes 8, 10 deviennent dans ce cas:

8. pour chaque mot w du contexte $C(t)$
10. $sup=sup+\frac{1}{f(w)}\cdot\left|D(s_j)\cap\{w\}\right|$ ou $sup=sup+\frac{1}{d(w,t)\cdot\log_2(f(w))}\cdot\left|D(s_j)\cap\{w\}\right|$

Fig. 4.7. Algorithme de Lesk, variante simplifiée, pondérée

où :

$\left|D(s_j)\cap\{w_i\}\right| = 1$ si $w\in D(s_j)$ et 0 en cas contraire.

4.3.5. Poids appris

Le système de désambiguïsation que nous avons développé comporte, en plus des modules de prétraitement et de désambiguïsation proprement dits, un module d'analyse des résultats décrit dans le chapitre 3. Le but de cette analyse est d'évaluer le système pour un ensemble de test donné, ce qui suppose l'existence d'un corpus oracle, contenant les sens corrects des mots à désambiguïser. Dans notre cas, nous avons disposé du fichier de réponses de *Senseval2* et des extraits du corpus annoté *Semcor* pour vérifier les réponses du système. A partir de ce type d'information et en utilisant les traces d'exécution produites par le système [12] on peut étudier de quelle façon certains mots participent à la désambiguïsation d'autres mots, les uns comme facteurs éclaircissants, les autres comme facteurs de bruit. Les sections suivantes présentent, plus en détail, les principes de base de l'approche.

[12] Fichiers où on stocke des informations sur les décisions prises : le mot cible, son contexte, les superpositions trouvées, le score, les décisions correctes et incorrectes etc.

a) Capacité d'un mot d'enlever l'ambiguïté d'autres mots

La méthode consiste à utiliser comme poids d'une superposition dans le calcul du score, la capacité du mot superposé à enlever l'ambiguïté sur d'autres mots. Cet indicateur est calculé en analysant le comportement de ce mot dans la désambiguïsation des corpus antérieurement traités par le système. La formule que nous avons proposée comme mesure de la capacité d'un mot d'enlever l'ambiguïté sur d'autres mots, pour un corpus de test donné (en effet la précision de *w*), est la suivante :

$$CEA(w,T) = \frac{ndc(w,T)}{ntd(w,T)} \qquad (11)$$

où

ndc(w,T) représente le nombre de cas correctement désambiguïsés par *w* dans l'ensemble de test *T*, antérieurement analysé et *ntd(w,T)* est le nombre total de désambiguïsations de *T* où *w* intervient.

Après une analyse des valeurs de cet indicateur pour les données de test traitées par le système, nous avons observé des CEA assez faibles (< 0.2) pour les mots fréquents, à beaucoup de sens ou représentant des concepts à caractère général tels que : *be, have, something, make, use, give, do, move, get, good, work, some, knowledge, action, entity, communication, unit, object* mais, par contre, des taux élevés (= 1) pour les mots peu fréquents, d'habitude à moins de 4 sens, souvent à caractère spécialisé, comme par exemple: *acclaim, pronounce, lapse, bucolic, tetragon, penicillin, guitar, incidence, sabbath, messiah, prophet, victuals, nutriment, nutrition, thyroid_gland.* Ce comportement semble tout à fait normal, puisque les mots à caractère générique sont supposés apporter moins d'informations utiles à la désambiguïsation que les mots à caractère spécifique, chose remarquée aussi par (Amorós et al. 2001).

b) Apprentissage des poids

L'apprentissage consiste à collecter dans un fichier les poids de chaque mot superposé et le nombre de désambiguïsations où il intervient pour un corpus de test donné, et d'ajuster ces valeurs après l'analyse d'un nouveau corpus de test. Premièrement, le fichier des poids appris est vide et les valeurs initiales des poids sont calculées par l'inverse du nombre de sens du mot superposé :

$$CEA_{initial}(w) = \frac{1}{|sens(w)|} \quad (12)$$

où

$|sens(w)|$ représente le nombre de sens de *w,* selon *WordNet.*

Donc au départ, plus le mot a de sens, moins il est informatif. La mise à jour du fichier comporte le calcul de la moyenne entre les valeurs initiales, correspondant aux corpus déjà analysés, et les valeurs courantes, correspondant au corpus de test en cours d'analyse. Il faut noter que les valeurs courantes de *CEA* ne sont pas indépendantes des valeurs précédentes (sauf pour la première apparition d'un mot superposé) parce que la désambiguïsation du corpus en cours d'analyse s'appuie sur les poids appris pendant les expériences antérieures. Nous avons choisi de calculer les moyennes dans l'idée que ces valeurs ne caractérisent pas un corpus de test donné mais plutôt l'ensemble des corpus soumis à l'analyse. Le tableau 4.5 présente les valeurs initiales et les valeurs moyennes extraites de ce fichier après les mises à jours successives pour les 10 fichiers de test de *Semcor.* Chaque ligne représente un mot suivi par le nombre de désambiguïsations où il est intervenu *ntd(w,T)* et son indicateur *CEA(w,T) :*

Tab. 4.5. Extrait du fichier des poids appris

Mot	**Valeurs initiales**		**Valeurs moyennes après le traitement des 10 fichiers *Semcor***	
	ntd	***CEA***	***ntd***	***CEA***
be	0	0.07	1336	0.17
have	0	0.04	150	0.11
must	0	0.25	59	0.04
begin	0	0.09	8	0.03
government	0	0.25	5	0.80
family	0	0.14	3	0.70
total	0	0.25	3	0.0
open-air	0	1.0	2	1.0
singer	0	0.33	1	1.0

Il faut mentionner aussi que pour ne pas obtenir de résultats biaisés l'apprentissage des poids n'a été jamais fait sur le corpus de test courant.

c) Filtrage flou

Un filtrage contrôlé par l'intermédiaire de deux concepts flous (deux prédicats qui prennent des valeurs de 0 à 1), *less_frequent* et *confidence_degree,* permet d'éliminer du

calcul du score les mot superposés trop fréquents ou dont les poids appris ne dépassent pas le seuil de 0.5, condition exprimée par le produit des deux prédicats :

$$poids(w,T) = less\_freq(f(w,T)) \cdot confd\_degree(CEA(w,T)) \quad (13)$$

où

$$f(w,T) = \frac{ntd(w,T)}{ntd(T)} \quad (14)$$

représente la fréquence relative de *w* donnée par le rapport entre *ntd(w,T),* nombre total de désambiguïsations de *T* où *w* intervient, et *ntd(T),* le nombre total de désambiguïsations traitées dans *T*.

Les deux concepts sont décrits, dans la manière de la logique floue, par des fonctions d'appartenance définies sur [0,1] à valeurs dans [0,1]. La fonction d'appartenance de *less_frequent* exprime, par exemple, le degré dans lequel un élément *x* appartient à la classe des phénomènes "moins fréquents". La fonction d'appartenance de *confd_degree* exprime le degré dans lequel un élément *x* peut être considéré "sûr". La forme de ces fonctions (15), (16) a été inspirée par le *principe du moindre effort* énoncé par Zipf (Manning et Schütze 1999) et par les considérations de Tengi (1998) concernant la relation entre la fréquence d'apparition d'un mot et son degré de polysémie (voir aussi 2.1.7). L'idée de base était d'éliminer par l'intermédiaire de deux fonctions simples (linéaires) l'influence des mots trop fréquents, à beaucoup de sens ou à contenu sémantique trop général. Les formules utilisées sont les suivantes :

$$less\_freq(x) = \begin{cases} -100x + 1, & si\ x < 0.01 \\ 0, & si\ x \geq 0.01 \end{cases} \quad (15)$$

$$confd\_degree(x) = \begin{cases} 2x - 1, & si\ x > 0.5 \\ 0, & si\ x \leq 0.5 \end{cases} \quad (16)$$

Le premier concept considère qu'un mot est moins fréquent (*less_frequent)* si son nombre d'occurrences dans un corpus donné ne dépasse pas 1% du nombre total d'occurrences renfermées par le corpus. De la même manière, une superposition présente un degré de confiance suffisant dans sa capacité de désambiguïsation si la valeur apprise de son indicateur CEA est supérieure à 0.5. Ces limites ont été établies après avoir testé plusieurs valeurs de ces seuils (0.1%, 0.5%, 1%, 2%, 3%, 4% respectivement 0.2, 0.4, 0.5, 0.6, 0.7, 0.8) en choisissant finalement les valeurs déterminant les meilleures performances

du système. Ces valeurs semblent en concordance avec les observations directes que nous avons effectuées sur le fichier des poids appris où les mots comportant une fréquence d'apparition *f(w,T)* supérieure à 0.01 sont, en général, des mots à caractère générique ou très familiers[13] de type *be, have, take, go, can, act, activity, entity, mental_object*, *artifact* etc. qui n'apportent pas d'informations utiles à la désambiguïsation (fait exprimé aussi par des valeurs très basses de leur degré de confiance, inférieur à 0.5).

La figure 4.8 présente le graphique des deux fonctions d'appartenance (15), (16):

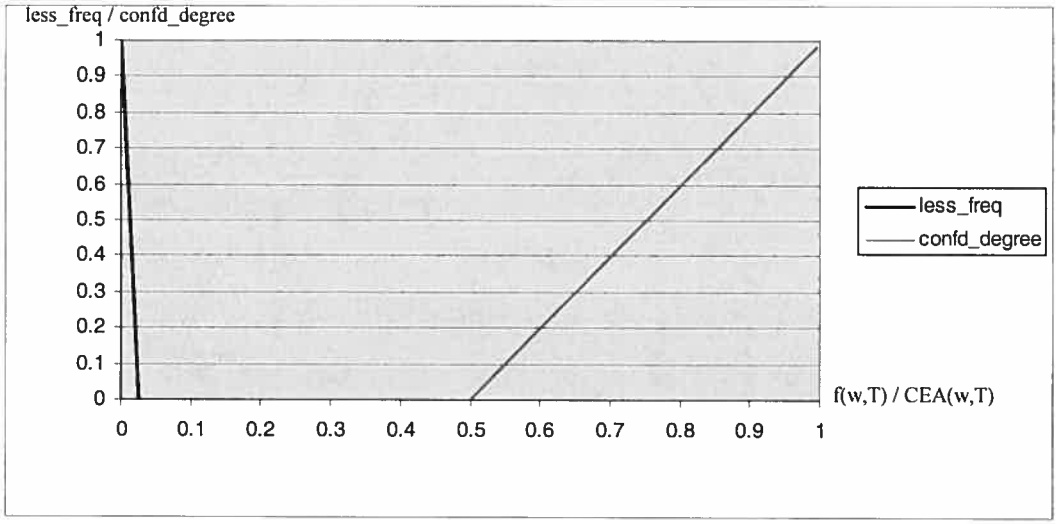


Fig. 4.8. Représentation graphique des concepts flou *less_freq* et *confd_degree* (où x = f(w,T), respectivement x = CEA(w,T))

d) Calcul du score

Les formules de calcul du score présentées dans la figure 4.6 et 4.7 deviennent:

$$sup = sup + poids(w\ ,T) \cdot \left| D(s_j) \cap E(w) \right| \qquad (17)$$

et respectivement :

$$sup = sup + poids(w,T) \cdot \left| D(s_j) \cap \{w\} \right| \qquad (18)$$

où

poids(w,T) représente le poids du mot du contexte *w,* appris par l'analyse des corpus T, antérieurement désambiguïsés.

[13] Voir chapitre 2 pour la définition de la familiarité.

4.3.6. Chaînes lexicales

Une autre variante de l'algorithme de Lesk que nous avons implémentée s'appuie sur la notion de *chaîne lexicale* utilisée par (Hirst et St-Odge 1998) dans la correction du *malapropisme* (confusion de deux mots comportant la même prononciation ou des formes orthographiques très semblables mais des sens différents). Par exemple, le mot *ingenous* (fr. *ingénu*) mis accidentellement à la place de *ingenious* (fr. *ingénieux*) dans "*an ingenous machine for peeling oranges*" est un *malapropisme.* L'idée centrale de cette approche consiste dans le fait que pour rendre un discours cohérent, les mots, co-occurant dans un même contexte, son reliés entre eux par des *relations de cohésion*, en formant des enchaînements logiques nommés *chaînes lexicales.* Selon Hirst et St-Odge, c'est l'utilisation de ces structures qui permettrait de détecter et de corriger automatiquement les cas de *malapropisme*.

Nous avons adapté cette idée à la désambiguïsation sémantique, en considérant que pour enlever l'ambiguïté d'un mot on a besoin seulement des mots appartenant à la même chaîne lexicale et pas de tous les mots apparaissant dans le contexte du mot cible. Pour mieux illustrer notre idée, considérons le texte de la figure 4.9 et supposons que nous devons désambiguïser le mot *committee* qui selon WordNet possède 2 sens. Les mots encadrés forment la chaîne lexicale du mot *committee*. La section suivante décrit en détail notre approche à la construction d'une chaîne lexicale à partir d'un mot tête (le mot à désambiguïser).

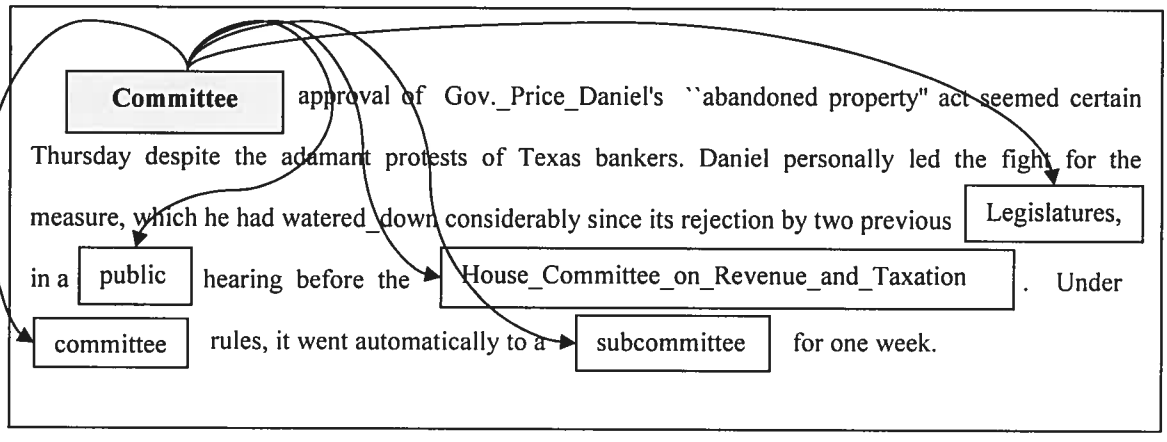


Fig. 4.9. Illustration de la notion de chaîne lexicale pour un fragment de texte et le mot cible *committee*

a) Appartenance à une chaîne lexicale

L'implémentation s'appuie sur l'utilisation des relations de synonymie et d'hyperonymie dans *WordNet* et sur le calcul d'une mesure de similarité entre les mots, calculée par l'intermédiaire de la formule de Jackard, pour tester l'appartenance de deux mots à la même chaîne lexicale. La similarité de deux ensembles est donnée, selon cette formule, par le rapport entre le nombre d'éléments de l'intersection et le nombre d'éléments de l'union des deux ensembles *A* et *B*.

$$S(A,B) = \frac{|A \cap B|}{|A \cup B|} \qquad (19)$$

Prenons par exemple les mots *committee* de notre exemple (voir Fig. 4.9) et le mot *legislature* qui apparaît dans son contexte. D'après *WordNet,* le mot cible a 2 sens possibles et *legislature* en possède 1. Les hiérarchies de synonymes et d'hyperonymes pour les trois sens sont illustrées dans la figure 4.10 où les rectangles représentent les groupes de synonymes (*synsets)* et les flèches, les relations d'hyperonymie entre les *synsets*.

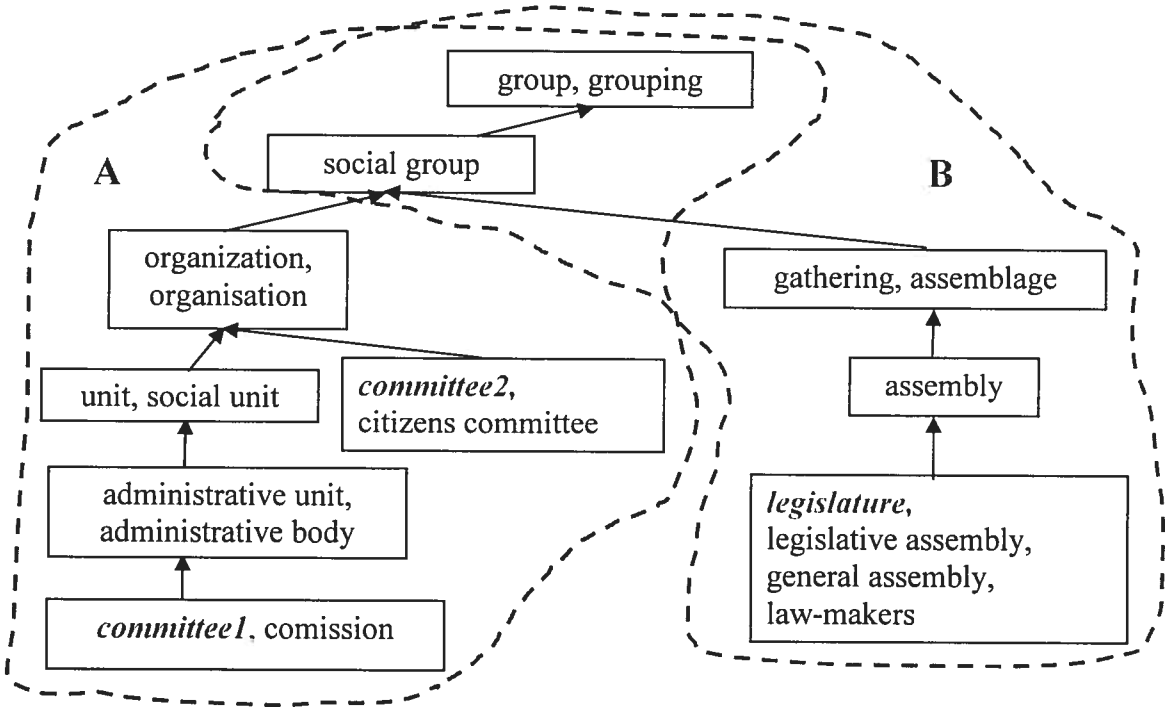


Fig. 4.10. Relations entre les *synsets* dans *WordNet* pour les sens *committee1, committee2* et *legislature*.

Pour déterminer dans quelle mesure les mots w_1 = *committee* et w_2 = *legislature* sont similaires et, donc, appartiennent à la même chaîne lexicale, on regroupe toutes les relations (synonymes + hyperonymes) de chaque mot. Soit $E(w_1)$ l'ensemble de relations caractérisant les deux sens *committee1* et *committee2* de w_1 et $E(w_2)$ l'ensemble de relations pour w_2. Alors, selon la hiérarchie présentée dans Fig. 4.10, le contenu des deux ensembles est : $E(w_1)$ = committee, comission, citizens, committee, administrative unit, administrative body, organization, organisation, social group, group, grouping

$E(w_2)$ = legislature, legislative assembly, general assembly, law-makers, assembly, gathering, assemblage, social group, group, grouping

Nous avons utilisé deux mesures afin de tester l'appartenance d'un mot w_2 à la chaîne lexicale dont la tête est w_1 :

$$I(w_1, w_2) = E(w_1) \cap E(w_2) \tag{20}$$

$$U(w_1, w_2) = E(w_1) \cup E(w_2) \tag{21}$$

Premièrement, on a considéré que w_2 appartient à la chaîne de w_1, si $|I(w_1, w_2)| \neq 0$, i.e. l'intersection des deux ensembles de relations pour w_1 et w_2 est non-vide. En pratique, ce type de métrique produit beaucoup de bruit, incluant, par exemple le nom *one* dans la chaîne lexicale de *committee* à cause d'un seul mot (*unit)* contenu dans *I*.

Une deuxième mesure, basée sur la formule de similarité de Jaccard, donne de meilleurs résultats si on impose que la mesure de similarité $S(w_1,w_2)$, explicitée par la formule (18), dépasse un certain seuil[14] pour que w_2 soit inclus dans la chaîne de w_1.

$$S(w_1, w_2) = \frac{I(w_1, w_2)}{U(w_1, w_2)} \tag{22}$$

Les chaînes suivantes (tête marquée en gras) semblent indiquer que cette métrique est capable de détecter les cas où il y a des niveaux communs dans la hiérarchie et même de saisir des relations de type antonymique[15] (comme par exemple *approval/rejection).*

committee [legislature, subcommittee, public, group, committee],
approval [rejection, witness, hear, act, question, rule],
person [witness, committee_member, banker, person],
seem [appear],
protest [rejection, fight].

Fig. 4.11. Exemples de chaînes lexicales

[14] Plusieurs valeurs ont été testées, entre 0.1 et 0.8. Les meilleures performances ont été obtenues pour 0.3.
[15] Bien que les seuls types de relations prises en compte soient la synonymie et l'hyperonymie.

On rappelle le fait que les mots propres sont encodés dans *WordNet* par les étiquettes *person, location, group* ou *other.* Cet encodage explique l'appartenance du syntagme *House_Committee_on_Revenue_and_Taxation* (*group* dans *WordNet*) à la chaîne lexicale de *committee,* rattaché au *synset* d'ordre supérieur *group* dans la hiérarchie *WordNet.*

L'algorithme que nous avons développé consiste à extraire du contexte, pour chaque mot à désambiguïser, la chaîne lexicale formée par tous les mots en relation avec le mot cible, considéré comme *tête de la chaîne*, et d'appliquer la méthode de Lesk (ou ses variantes) pour calculer le score de chaque sens candidat. Une description plus détaillée de la méthode est présentée dans la section suivante.

b) Algorithme de désambiguïsation basé sur les chaînes lexicales

La variante basée sur les chaînes lexicales que nous avons implémentée est décrite par la séquence suivante, en pseudo-code :

pour chaque mot à désambiguïser *t*
    faire la chaîne *chLex(t) = null*
    extraire de *WordNet* toutes les synonymes et les hyperonymes de *t* regroupés dans *E(t,)*
    déterminer *C(t),* le contexte de *t*
    pour chaque mot *w* du contexte *C(t)*
        extraire ses synonymes et ses hyperonymes de *WordNet* regroupés dans *E(w)*
        calculer la mesure de similarité $S(t,w) = \frac{E(t) \cap E(w)}{E(t) \cup E(w)}$
        si *S(t,w)* > seuil alors
            ajouter *w* à *chLex(t)*
    pour chaque sens candidat s_j de *t*
        pour chaque mot v de *chLex(t)*
            appliquer une des variantes de l'algorithme de Lesk
        attribuer à *t* le sens donné par le meilleur sens candidat

Fig. 4.12. Algorithme de Lesk, variante basée sur les chaînes lexicales

4.3.7. Tableau de votes

Les variantes décrites dans la section 4.3 comportent un traitement séquentiel des mots à désambiguïser, les sens étant assignés de manière indépendante. Une autre alternative, développée dans le cadre du projet, s'appuie sur l'exploitation d'un tableau de

votes où chaque sens candidat du mot cible reçoit et respectivement inscrit un vote pour le meilleur sens candidat d'un mot du contexte. La procédure, en pseudo code, est présentée ici :

1. pour chaque mot à désambiguïser t
2. déterminer $C(t)$, le contexte de t
3. pour chaque sens candidat s_j de t
4. pour chaque mot w du contexte $C(t)$
5. pour chaque sens s'_k de w
6. calculer les superpositions entre les descriptions de s_j et s'_k
7. choisir comme meilleur candidat le sens de w au score maximal, soit s'_{max}
8. ajouter dans le tableau de votes, un vote pour s_j et un vote pour s'_{max} dans les
9. entrées $TabV[t][s_j]$ et $TabV[w][s'_{max}]$
10. pour chaque mot à désambiguïser, t
11. déterminer le nombre maximal de votes v_{max} pour l'entrée TabV[t]
12. si $v_{max} = 0$ alors
13. attribuer à t le sens le plus fréquent
14. sinon
15. s'il y a plusieurs sens de t avec le nombre de votes = v_{max} alors
16. attribuer à t le sens le plus fréquent d'entre eux
17. sinon
18. attribuer à t le sens correspondant à v_{max} (le sens ayant le plus grand nombre de votes)

Fig. 4.13. Algorithme de désambiguïsation avec tableau de votes

Pour le cas avec traitement des chaînes lexicales, on modifie la ligne 4 de la manière suivante :

4. pour chaque mot w de la chaîne lexicale $chLex(t)$

en supposant que la chaîne lexicale de t a été préalablement déterminée par la méthode décrite dans la section précédente.

Le choix effectif du sens se réalise à la fin, par le balayage du tableau de votes, en comptant le nombre de votes pour chaque mot à désambiguïser et en choisissant le sens avec le nombre maximal de votes. Dans le cas où aucun vote n'a été enregistré pour les sens d'un mot cible, on choisit le sens le plus fréquent selon *WordNet.* De même pour les situations d'égalité de vote maximal, on attribue à t le sens le plus fréquent parmi les candidats à *tie break.*

Pour mieux illustrer cet algorithme, prenons, par exemple, le texte présenté dans la figure 4.9 et le mot à désambiguïser *approval* qui possède 4 sens selon *WordNet.* Comme déjà mentionné au début de la section 4.3, le processus de désambiguïsation avec tableau de votes suppose le traitement d'une paire *mot cible – mot du contexte* à un moment donné. Soit cette paire *approval – rejection* appartenant à la même chaîne lexicale, initiée par le

mot tête *approval* (voir Fig. 4.11). Supposons que le sens à traiter est le premier sens du mot *approval* dont la description de *WordNet (définition + exemples)* est la suivante : *the formal act of giving approval; "he gave the project his blessing"; "his decision merited the approval of any sensible person".* L'algorithme compare cette description avec les descriptions des 4 sens de *rejection* présentés dans la figure 4.14 *:*

rejection1 - the act of rejecting something; "his proposals were met with rejection"
rejection2 -- the state of being rejected
rejection3 -- (medicine) immunological response that refuses to accept substances or organisms that are recognized as foreign; "rejection of the transplanted liver"
rejection4 -- the speech act of rejecting

Fig. 4.14. Les descriptions des 4 sens de *rejection* selon *WordNet*

En comptant le nombre de superpositions entre la description du sens *approval1* et les 4 sens de *rejection,* le système trouve un *overlap* (le mot *act*) entre la description de *approval1* et de *rejection1 et 4.* Comme à un score égal, le sens plus fréquent est préféré, le meilleur candidat pour *rejection* est *rejection1.* Alors, les sens *1* des entrées *approval* et *rejection* dans la table de votes reçoivent chacun un vote, comme le montre le tableau 4.6.

Tab. 4.6. Entrées de *approval* et *rejection* dans la table de votes après le traitement de la paire *approval – rejection*

Mot	**Sens1**	**Sens2**	…	…	…	…	…
….						…	
approval	1	0	0	0			
….							
rejection	1	0	0	0			
…						…	

L'algorithme continue avec la comparaison de *approval1* et les sens des autres mots de la chaîne, puis le processus est répété pour *approval2,* etc; chaque traitement d'une paire produisant une modification de la table de votes (si des superpositions sont trouvées). Après le traitement de toutes les instances à désambiguïser, pour chaque entrée du tableau de votes on choisit la case correspondant au sens avec le nombre maximal de votes.

4.3.8. Stop liste

Afin d'éliminer l'influence des mots sans contenu sémantique bien défini (prépositions, conjonctions, pronoms, déterminants), le programme ne prend en compte, dans le processus de désambiguïsation, que les mots appartenant aux 4 catégories ouvertes (noms, verbes, adjectives et adverbes). Cette opération est réalisée pendant la phase d'extraction[16] des définitions et des relations du dictionnaire pour le corpus de test donné. D'autres éléments qui pourraient produire beaucoup de bruit dans le système sont les mots à caractère plus général et, d'habitude, très fréquents (par exemple, les verbes *be,have, can, make* etc.*).* Nous avons testé plusieurs variantes pour écarter ce type d'influences. L'une est basée sur la consultation d'une liste figée de mots "vides" comportant les mots considérés trop fréquents ou ayant un contenu trop vague pour participer à la désambiguïsation. La deuxième variante s'appuie sur les poids appris des mots, méthode décrite dans la section 4.3.5. La dernière s'appuie sur l'utilisation de la fréquence d'usage, calculée à partir du corpus *Hansard* (voir 4.3.4) et modélisée par l'intermédiaire du concept flou de *moins fréquent,* présenté dans 4.3.5.

[16] Voir la phase de prétraitement, chapitre 3.

Résultats expérimentaux

Ce chapitre contient une description des différentes expériences réalisées au cours du projet, visant à mettre en évidence les principaux facteurs qui influencent le plus le comportement du système.

Une convention de nommage des différentes méthodes de désambiguïsation et de leurs paramètres a été adoptée ici, et est décrite dans la section 5.1, afin de synthétiser les résultats des expériences réalisées. Les sections 5.2 et 5.3 sont dédiées aux performances du système pour les deux corpus de test (*Senseval2* et *Semcor*), ainsi qu'à l'interprétation des facteurs qui ont mené aux résultats obtenus. Les paramètres que nous avons étudiés sont : la longueur du contexte, la nature de la description des sens et le nombre de décisions par défaut, la fréquence relative des sens candidats, le rapport correct/incorrect relativement aux choix de BASE, la catégorie grammaticale, l'interdépendance des sens choisis, la granularité du découpage des sens. Une étude comparative avec les résultats officiels de *Senseval2* et avec d'autres approches similaires et décrite en section 5.4.

5.1. Encodage des expériences

La forme générale de l'encodage utilisé pour nos expériences est la suivante:

[L](D | R | DR) v, c (23)

où la présence de la majuscule L signifie la variante de la méthode de Lesk originelle, qui compte les mots communs entre l'entité descriptive des sens candidats (définition, relations ou les deux) et une entité descriptive similaire correspondant aux mots pris du contexte (voir 3.3.1). L'absence de cette lettre indique l'algorithme de Lesk simplifié qui compare les descriptions des sens seulement avec le contexte (voir 3.3.2).

Les majuscules (D, R ou DR) font référence aux entités descriptives mentionnées plus haut (D pour définition, R pour relations, DR pour la combinaison des deux).

La minuscule *v* fait référence à diverses variantes, pondérées ou non, avec ou sans tableau de votes, avec chaînes lexicales, variante de base etc. décrites dans les sections 4.3.3 – 4.3.7. Les codes mnémoniques de ces variantes sont présentés dans la liste suivante:

BASE variante de base, choix du sens le plus fréquent ;

CL désambiguïsation basée sur les **C**haînes **L**exicales ;

DlogF score pondéré par l'inverse de la **D**istance au mot cible et l'inverse de $\log_2$ de la **F**réquence d'usage;

F score pondéré par l'inverse de la **F**réquence d'usage des mots superposés ;

logTD normalisation logarithmique par la **T**aille de **D**escription de sens ;

NP variante **N**on **P**ondérée ;

PA **P**oids **A**ppris des superpositions ;

RF pondération par la **F**réquence **R**elative des sens candidats;

TD normalisation du score par la **T**aille de **D**escription de sens ;

V score avec tableau de **V**otes (l'absence de *V*, indique une désambiguïsation séquentielle)

En plus des paramètres définis ci-dessus, les expériences que nous avons effectuées ont visé aussi une étude du comportement du système pour des longueurs de contexte différentes, afin de déterminer l'apport du contexte local et du contexte global à la désambiguïsation. Toutes les variantes ont été testées pour des fenêtres de contexte de taille 4, 6, 16, 20 et 50.

Chaque code mnémonique de méthode est suivi par la longueur du contexte, *c* dans la formule générale (1), considéré autour du mot à désambiguïser. Par exemple, **LDF,6** codifie une implémentation de la méthode de **Lesk originelle** qui compte les mots communs entre les **définitions** des sens candidats et les définitions des mots du contexte, en pondérant ce nombre par l'inverse de la **fréquence** d'usage des mots superposés, la recherche visant les **3 mots**[1] **avant et après** le mot cible.

Un encodage supplémentaire, pas inclus dans la formule (1), a été attribué à la variante RF qui applique une pondération par la **F**réquence **R**elative des sens candidats aux scores calculés par les méthodes sus-mentionnées. Cette variante est décrite plus en détail dans la section suivante.

[1] Appartenant aux 4 catégories ouvertes.

5.2. Performances

Le but des expériences sur plusieurs fichiers de test a consisté principalement à comparer les résultats obtenus pour des corpus de test provenant de sources différentes (*Senseval2* et *Semcor)* avec l'espoir que cette comparaison nous permettra de tirer des conclusions plus générales, indépendantes de l'ensemble de test choisi. Cette section présente les points principaux de cette analyse comparative, une discussion plus détaillée sur les facteurs menant à ces résultats faisant l'objet de la section 5.3.

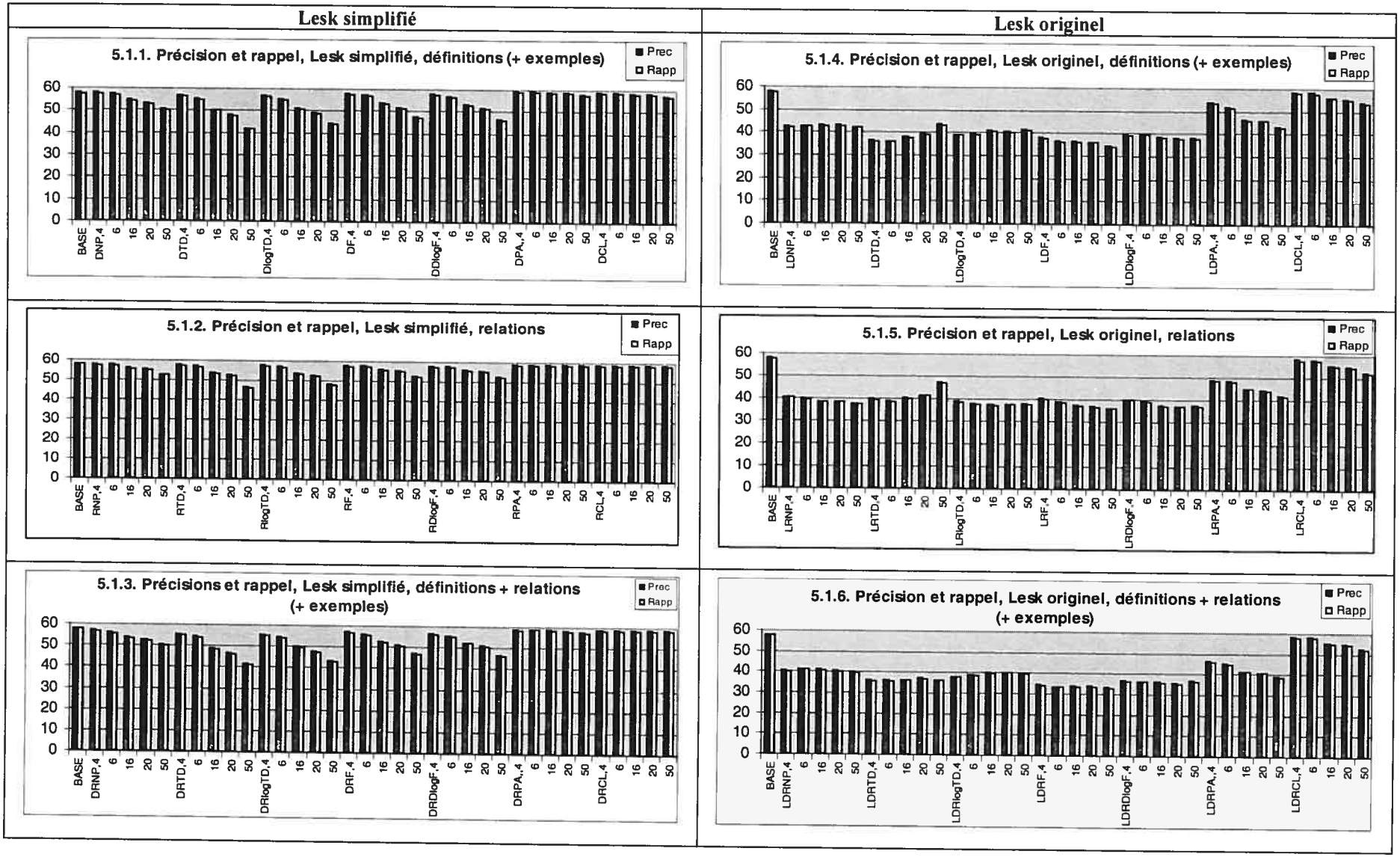
5.2.1. Corpus de test *Senseval2*

Le tableau 5.1 montre les performances des algorithmes décrits dans le chapitre 4, sur les données de test de *Senseval2*, pour cinq longueurs de contexte et la méthode d'évaluation *fine-grained*. Le code mnémonique de chaque variante est spécifié seulement pour le contexte de 4 mots, les autres valeurs le suivant faisant référence à la même méthode. Pour des raisons de clarté, les résultats ont été regroupés en 6 cases selon le type de description utilisée (D – définitions + exemples, R – relations, DR – définitions + relations + exemples) et selon la variante de l'algorithme de Lesk analysée (L - Lesk originel ou simplifié). Chaque case contient aussi les performances de référence (choix du sens le plus fréquent) sous l'étiquette BASE ((précision et rappel 57,9%/57.6).

On observe qu'en général les valeurs de la précision et du rappel sont assez proches, ce qui prouve la capacité du système de traiter à peu près tous les mots à désambiguïser. Les cas non-traités (0.8% des instances à désambiguïser) sont dus aux formes de base non-trouvées dans *WordNet* (voir chapitre 3). On remarque également que les méthodes Lesk originelles sont systématiquement moins bonnes que leur contrepartie simplifiée. Par rapport aux performances de base, les méthodes Lesk simplifiées présentent des valeurs supérieures mais pas de manière très marquée. Par contre, les performances des variantes originelles sont toujours plus basses que les performances de base (sauf pour *CL*).

D'un autre côté, la croissance de la taille du contexte a une influence négative sur les performances des méthodes Lesk simplifiées (tendance moins accentuée dans le cas des descriptions de type *R* que dans le cas de *D* et *DR* et des variantes *PA* et *CL*). Cette tendance n'est pas manifeste de manière systématique pour les méthodes Lesk originelles (il y a même une légère croissance des performances pour les variantes *LDTD* et *LRTD*).

Tab. 5.1. Précision et rappel pour le corpus de test *Senseval* 2, évaluation *fine-grained*



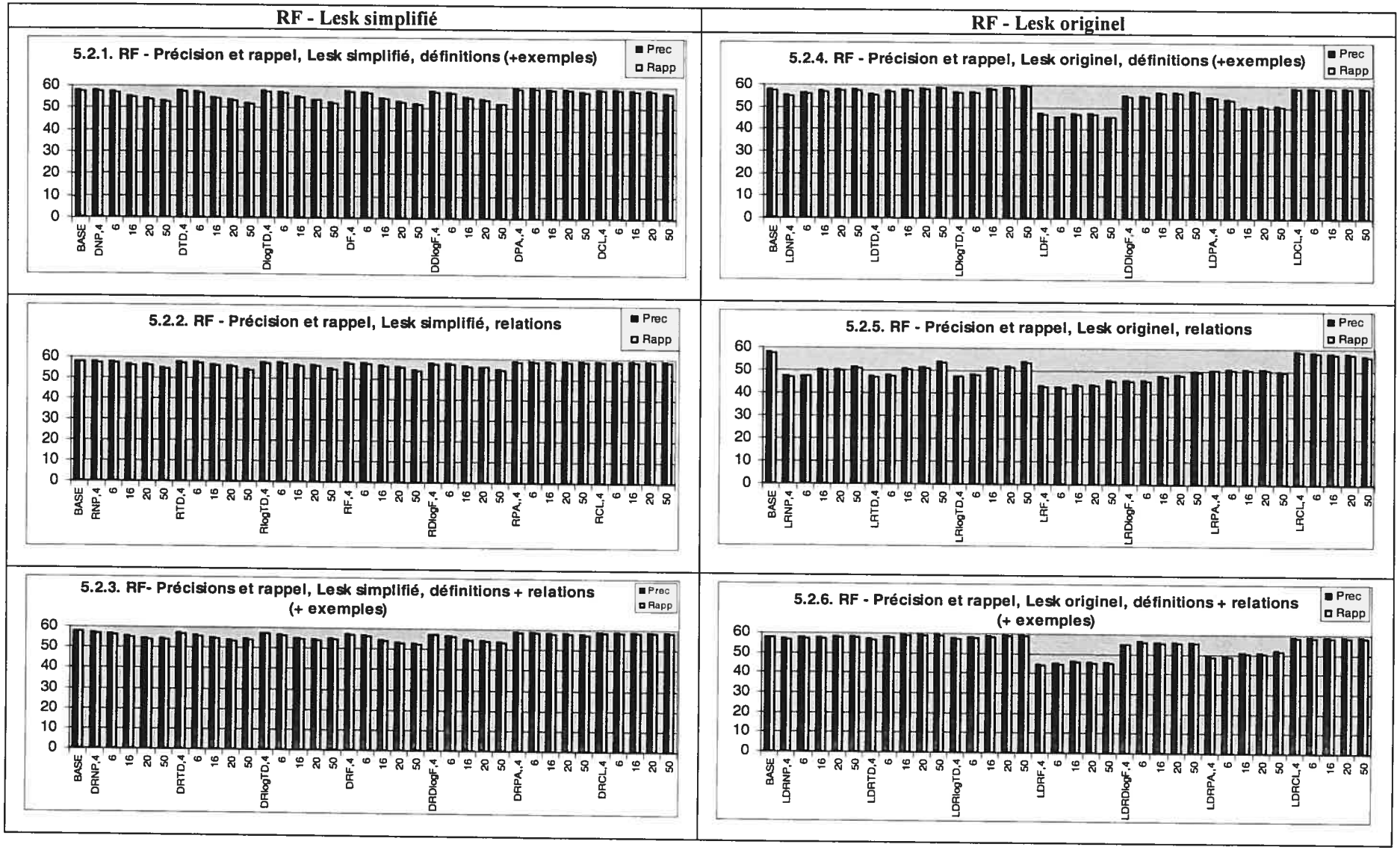
Les méthodes dominantes sont *PA* et *CL,* aussi bien dans leur version simplifiée que dans celle originelle. Les valeurs les plus stables sont enregistrées par la méthode *CL* qui présente les plus petites différences entre les performances de ses deux variantes, simplifiée et originelle. Il existe quand même une tendance de décroissance avec le contexte pour ces méthodes (surtout dans leur version originelle). Les meilleures performances (59.4%/58.8, 59.2%/58.6) appartiennent aux méthodes *DPA* et *DCL* pour un contexte de 6 mots.

Les observations sus-mentionnées semblent suggérer l'hypothèse que les performances du système diminuent lorsque l'information prise en compte pour la désambiguïsation est plus riche (contextes plus larges, descriptions *D, DR* par rapport aux descriptions *R* - pour la variante simplifiée, contexte enrichi par les descriptions des mots composants – pour la variante originelle). C'est-à-dire, un contenu informationnel plus large n'est pas nécessairement pertinent ou est fortement bruité, fait reflété dans la diminution des performances par rapport aux performances de *BASE.* En d'autres mots, une information contextuelle plus riche détermine des choix d'un autre sens que le plus fréquent mais ce choix est dans beaucoup de cas incorrect (chose confirmée, d'ailleurs, par l'analyse sur le nombre de décisions effectives prises par le système, décrite dans la section 5.3.2).

C'est à partir de cette hypothèse que nous avons implémenté une variante supplémentaire qui semble réduire le bruit du système. La méthode utilise les mêmes calculs de score présentés dans le chapitre 4, mais multipliés par un indicateur de fréquence relative du sens candidat. La manière de calculer cet indicateur est décrite dans la section 5.3.3. Le tableau 5.2 reprend les expériences décrites dans 5.1, à la différence que les scores ont été pondérés par l'indicateur de fréquence de chaque sens candidat (variante *RF*).

Pour les variantes Lesk simplifiées on remarque une croissance des performances (surtout pour les descriptions *D* et *DR*) et une tendance moins forte des performances à diminuer avec l'élargissement du contexte. L'amélioration des performances est plus visible dans le cas des méthodes Lesk originelles pondérées et non-pondérées qui produisent, pour ce corpus de test, la meilleure valeur de la précision et du rappel (59.8% / 59.3% – *LDlogTD* pour un contexte de 50 mots). Pour ce qui est de l'influence de la taille du contexte, on observe une tendance de croissance des performances avec le contexte, plus manifeste que dans le cas sans pondération avec la fréquence relative (surtout pour les descriptions *D* et *R*). La méthode la plus stable est toujours CL. La tendance de décroissance des performances avec le contexte, dans sa variante Lesk originelle, est atténuée par la factorisation avec la fréquence relative *(RF).*

Tab. 5.2. Précision et rappel pour le corpus de test *Senseval* 2, évaluation *fine-grained,* implémentation *RF*



5.2.2. Corpus de test extrait de *Semcor*

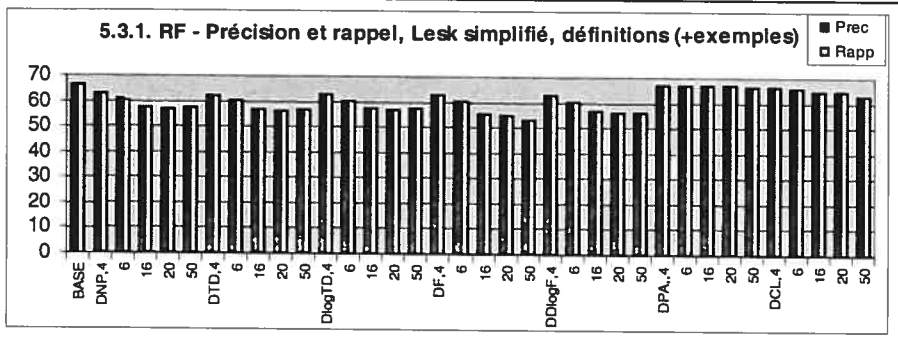
Les expériences effectuées sur les données de test provenant de *Semcor* comportent le traitement de 10 fichiers de test dont les caractéristiques ont été décrites dans le chapitre 2. Les valeurs présentées ci-dessous représentent des valeurs moyennes calculées à partir des résultats individuels obtenus pour chaque fichier de test. Là encore, comme pour le corpus de test *Senseval2,* on a constaté un meilleur comportement du système pour les variantes utilisant la fréquence relative des sens candidats *(RF).* Pour des raisons de concision, nous allons présenter seulement les résultats de ce type d'implémentation.

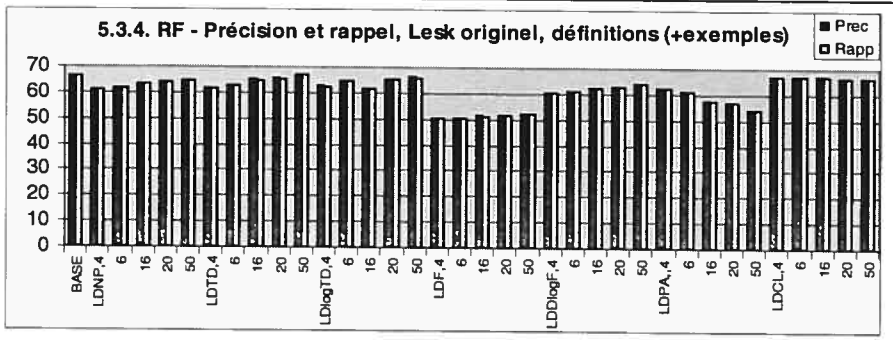
Le tableau 5.3 montre les performances moyennes du système testé sur les 10 fichiers (évaluation *fine-grained)* et la version *RF.* Pour ce corpus les performances moyennes de *BASE* sont de 66.37%/ 66.36. Les valeurs de la précision et du rappel sont plus proches que dans le cas de *Senseval2* et ceci parce que dans le cas de *Semcor* les lemmes sont trouvés directement dans le corpus, les erreurs dues au traitement des formes composées étant ainsi éliminées. Les seules situations qui produisent une diminution du rappel sont les non-concordances entre la version de *Semcor* (annotée avec les étiquettes de *WordNet1.6)* et la version de *WordNet (1.7.1)* que nous avons utilisées. Ces cas de lemmes des mots cibles non trouvés dans le dictionnaire sont cependant très rares (moins de 0.1% du total des mots à désambiguïser).

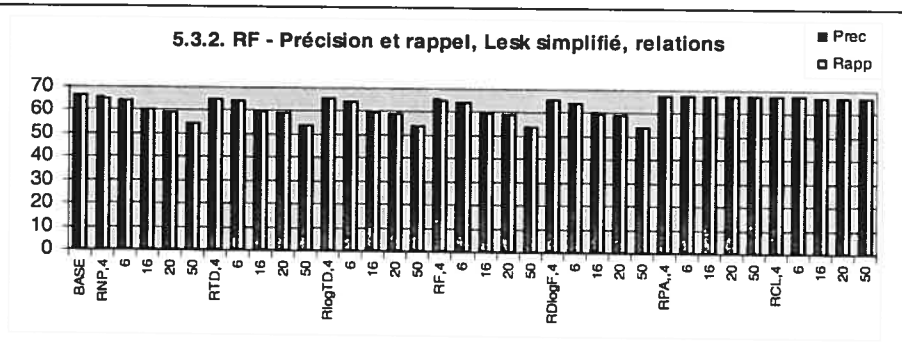
Comme pour *Senseval2,* on peut observer un bon comportement des variantes *PA, CL* dans la version simplifiée (67.30%/67.29% et 67.24%/67.22% meilleures performances de ces méthodes pour *RPA* et *DRCL* et un contexte de 4 mots). Si on compare les deux méthodes, *CL* semble moins sensible aux variations des divers paramètres (description de sens, longueur de contexte, variante simplifiée ou originelle) que *PA,* qui diminue ses performances dans sa version Lesk originelle. Les autres méthodes présentent un comportement assez similaire entre eux, surtout pour la variante Lesk simplifiée. Seule la variante Lesk originelle de la méthode *F* présente une diminution de performances par rapport aux autres (5.3.4-5.3.6 et encore 5.2.4-5.2.6). Pourtant ce "handicap" est récupéré par sa version *L-DlogF,* ce qui semble indiquer que la pondération par l'inverse de la distance du mot cible et l'inverse du log de la fréquence d'usage des mots superposés marche mieux en pratique que la pondération par la fréquence toute seule. La meilleure performance pour la variante Lesk originelle est donnée par la méthode TD, un contexte de 50 mots et une description de sens de type DR (67.23%/67.20%).

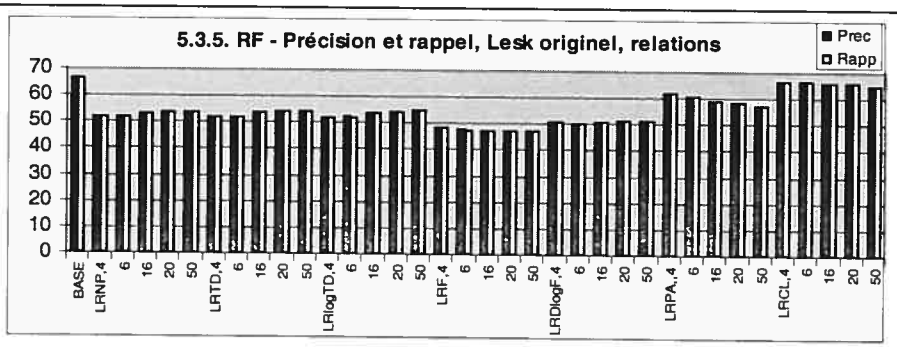
Tab. 5.3. Performances pour le corpus de test extrait de *Semcor,* évaluation *fine-grained,* implémentation *RF*

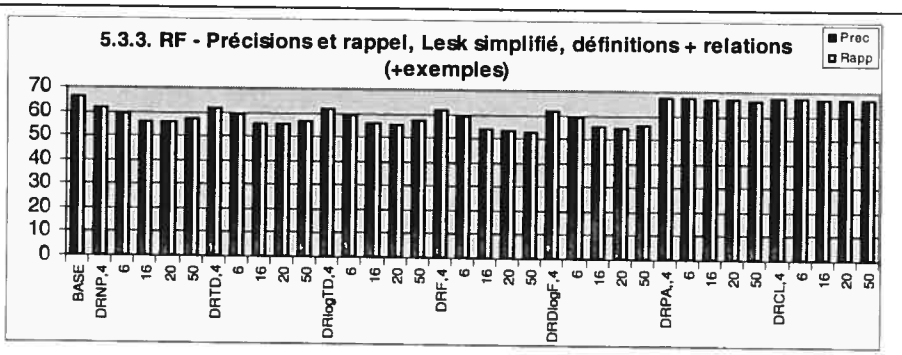
RF - Lesk simplifié	RF - Lesk originel
5.3.1. RF - Précision et rappel, Lesk simplifié, définitions (+exemples)	5.3.4. RF - Précision et rappel, Lesk originel, définitions (+exemples)
5.3.2. RF - Précision et rappel, Lesk simplifié, relations	5.3.5. RF - Précision et rappel, Lesk originel, relations
5.3.3. RF - Précisions et rappel, Lesk simplifié, définitions + relations (+exemples)	5.3.6. RF - Précision et rappel, Lesk originel, définitions + relations (+exemples)

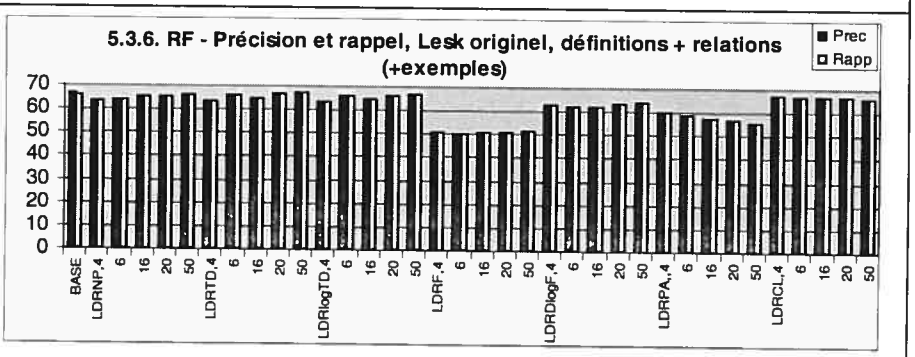












En général, le meilleur gain absolu en précision ne dépasse pas 1.81% pour le corpus *Senseval2* et 0.93% pour *Semcor.* Les variantes capables de surpasser la BASE, de manière systématique, sont seulement les versions Lesk originelles pondérées par la taille de la description et corrigées par la fréquence relative (*L-TD, L-logTD*) pour des contextes larges (20, 50 mots) et les variantes *PA* et *CL* surtout dans leur version Lesk simplifiée, pour des contextes plus petits (4,6,16 mots).

5.3. Influence des paramètres

A partir des observations générales présentées dans 5.2, cette section tente de mettre davantage en lumière l'influence de certains facteurs sur le comportement du système. La discussion porte comparativement sur les résultats obtenus pour les deux corpus de test analysés (*Senseval2 et Semcor)* ainsi que sur des conclusions rapportées par d'autres études.

5.3.1. Longueur du contexte

Nous avons montré dans la section précédente (voir Tab. 5.1, 5.2, 5.3) que la variante Lesk simplifiée montre une décroissance des performances avec l'augmentation du contexte[2], surtout pour les variantes *NP, TD, logTD, F et DlogF* . Ce comportement semble indiquer le fait que les superpositions des descriptions de sens avec le contexte local (4,6 mots pleins [3] autour du mot cible) contribuent plus à la désambiguïsation que les superpositions avec le contexte global (plus de 10 mots). Pourtant les méthodes *PA* et *CL*, usant de certains moyens de sélection des mots superposés (filtrage selon les poids appris ou selon l'appartenance à la même chaîne lexicale), sont moins sensibles à la variation de la taille du contexte et par conséquent, moins influencées par ce type d'enrichissement contextuel. Une explication de ce comportement pour les deux méthodes, consisterait dans le fait que, d'un côté, la méthode *PA* élimine du calcul du score les superpositions "non-sûres" et donc le nombre de superpositions n'a pas beaucoup d'importance. D'un autre côte, la méthode *CL* sélectionne du contexte seulement les mots reliés sémantiquement avec le mot cible et la longueur de la chaîne lexicale ainsi construite est de moindre importance (l'ajout de mots à la chaîne n'influence pas beaucoup le choix du sens pour le mot tête).

---

[2] En moyenne, 10% entre les performances de la même méthode pour un contexte de 4 et de 50 mots.
[3] Noms, verbes, adjectives et adverbes.

Les variantes de l'algorithme de Lesk originel (voir Tab. 5.1) présentent des variations plus petites avec la longueur du contexte, même une légère croissance dans le cas des variantes *TD, logTD.* Pourtant, on peut remarquer une diminution significative des performances par rapport aux performances de base. La seule méthode qui semble mieux gérer le surplus informationnel est la variante *CL*, qui produit des performances supérieures aux performances de *BASE* même dans ces conditions. Pourtant la pondération des scores par un facteur dépendant de la fréquence relative des sens candidats (les tableaux 5.2 et 5.3) produit une augmentation des performances de la variante Lesk originelle ainsi qu'une légère tendance des performances à augmenter avec le contexte (sauf pour les méthodes *PA,CL*).

Il semble donc (du moins dans nos expériences) que les mots les plus proches du mot à désambiguïser sont les plus informatifs. Audibert (2003) arrive à une conclusion similaire. Il teste sa méthode de désambiguïsation basée sur les *coocurrences* pour des contextes entre 1 et 20 mots, en mentionnant de bonnes précisions pour de petites fenêtres, allant de 1 à 4 mots. De plus, dans le cas des verbes, il observe de meilleures performances pour des contextes asymétriques (2 mots avant, 4 mots après le mot cible) ce qui semble indiquer que la désambiguïsation des verbes s'appuie plutôt sur leur objet que sur leur sujet.

D'un autre côté, Lesk (86) rapporte des différences de performances mineures lorsqu'il considère des longueurs de contexte de 4, 6 ou 8 mots. Des études récentes, utilisant d'autres approches, tirent des conclusions différentes concernant l'influence de la taille de la fenêtre de contexte sur la désambiguïsation. Crestan et al. (2003), à partir des résultats obtenus pour des contextes de 3,5 et 7 mots et un algorithme basé sur les *arbres de décision,* concluent qu'il n'y a pas de fenêtre optimale, valable pour tous les mots, mais qu'elle diffère d'un mot à l'autre. Amorós et al. (2001), qui testent un algorithme basé sur la notion de *distance conceptuelle*[4] pour des contextes entre 1 et 500 mots, signalent une augmentation du rappel avec le contexte, même pour des fenêtres de 150 mots et plus.

Dans notre cas, le fonctionnement du système selon la longueur du contexte diffère plutôt d'une méthode à l'autre, ou plus précisément d'un groupe de méthodes à un autre (méthodes *CL, PA* plus stables par rapport aux autres). D'un autre côté, l'information du contexte global peut être quand même utile à la condition d'un filtrage plus strict des données (variantes *RF, CL, PA*).

---

[4] Mesure déterminée à partir de la *densité conceptuelle* proposée par (Agire-Rigau 1996) et à l'aide de la hiérarchie de concepts de *WordNet.*

5.3.2. Description des sens et nombre de décisions par défaut

Les différentes versions que nous avons testées montrent des performances différentes qui sont grandement conditionnées par le nombre d'intersections entre le contexte du mot à désambiguïser et la description du sens candidat. En particulier, dans le cas de la version Lesk originelle, le nombre d'intersections sur lesquelles se base la prise de décisions est à priori plus grande que dans le cas de la variante Lesk simplifiée. Rappelons que dans le premier cas, les descriptions des mots du contexte sont considérées, tandis que dans la version simple, seulement les mots du contexte le sont. Lorsqu'il n'existe pas d'information pour prendre une décision (intersection *contexte-sens* vide), les algorithmes implémentés se rabattent sur le sens le plus fréquent. Il convient donc de caractériser chaque variante en terme de nombre de décisions par défaut, c'est-à-dire de nombre de cas où le système choisit le sens le plus fréquent, par manque d'information (intersection vide).

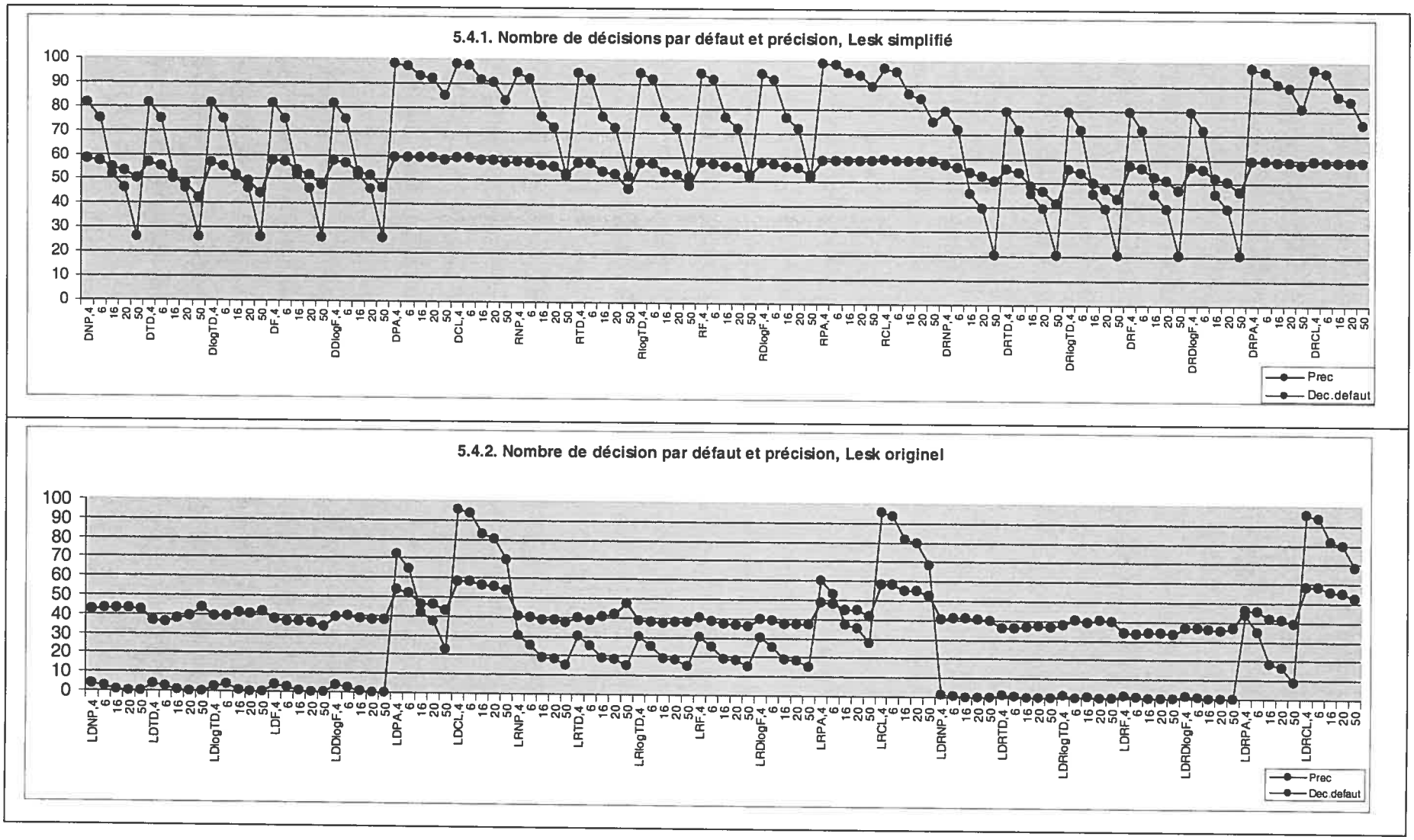
La relation entre la précision du système et le nombre de décisions par défaut, pour le corpus de test *Senseval2,* est présentée dans le tableau 5.4.

Le diagramme 5.4.1 (Lesk simplifié) montre une variation du taux de décisions par défaut entre 20 et 80% pour les variantes *NP, TD, logTD, F, logF* et les description de sens de type D (définitions + exemples) et DR (définitions + exemples + relations). On observe que pour ce genre de méthodes la décroissance du taux de décisions par défaut produit une diminution de la précision.

La même tendance, mais plus atténuée, caractérise les variantes basées sur les relations *(RNP, RTD, RlogTD, RF, RDlogF),* à la seule différence que pour ce type de description le nombre de décisions par défaut est supérieur (entre 50 et 95%), i.e. il y a moins de superpositions entre les relations descriptives des sens R et le contexte, que dans les cas D ou DR.

Par contre, les variantes *PA* et *CL*, qui choisissent le sens le plus fréquent pour 70 - 98% des cas, sont moins influencées par la variation du nombre de décisions par défaut, ce qui semble indiquer que pour ce type de méthodes le choix du sens le plus fréquent est favorisé même dans les conditions où l'information contextuelle devient plus riche. Ce comportement semble tout à fait normal parce que ces méthodes s'appuient sur un filtrage plus dur des données utilisées dans la désambiguïsation.

Tab. 5.4. Nombre de décisions par défaut et précision *fine-grained* pour le corpus *Senseval2*



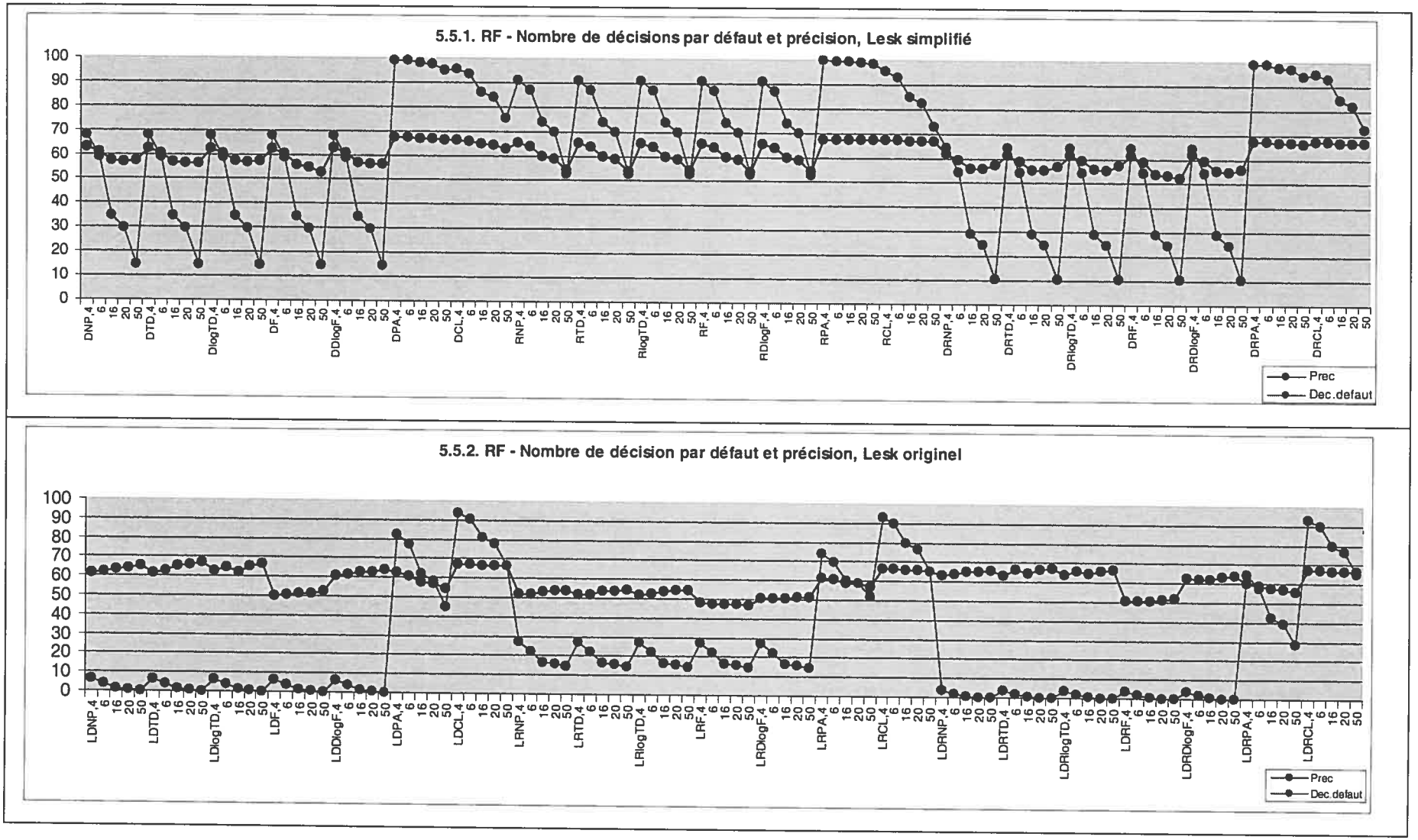
Le diagramme 5.4.2 (Lesk originel) indique une décroissance drastique du nombre de décisions par défaut (moins de 10%) pour les variantes *LNP, LTD, LlogTD, LF, LlogF* et les descriptions de sens *D* et *DR*. Pourtant cette décroissance se traduit par une diminution au niveau des performances, comparativement aux performances de la méthode de Lesk simplifiée. Le contexte enrichi par les descriptions des mots du contexte semble donc augmenter le bruit du système. Les mêmes variantes appliquées aux relations *(LRNP, LRTD, LRlogTD, LRF, LRDlogF)* présentent des précisions comparables avec les valeurs correspondant à *D* et *DR,* mais pour un taux de décisions par défaut plus élevé (entre 15 et 30%). Cette observation pourrait suggérer l'hypothèse que les descriptions *D, DR* favorisent le sens le plus fréquent d'une manière plus significative que les descriptions *R* qui, à leur tour, supposent un nombre plus petit d'intersections non-vides entre les relations des sens candidats et les relations des mots du contexte. D'un autre côté, les relations, utilisées comme descriptions de sens, produisent de meilleurs résultats dans les versions de l'algorithme de Lesk simplifié (précision 45-60%) que dans celles de l'algorithme originel (environ de 40% précision pour toutes les variantes, sauf *PA, CL*) où le contexte est "enrichi" par les relations des mots du contexte. Dans ce cas, la probabilité de trouver des superpositions de type *object, entity, action* etc. augmente. Ce type d'information à caractère générique, véhiculée par les hyperonymes d'ordre supérieur (selon la hiérarchie de *WordNet*), est pourtant supposée apporter peu d'information utile à la désambiguïsation, ce qui pourrait expliquer l'abaissement des performances, fait signalé aussi par (Amorós et al. 2001). De ce point de vue les descriptions *D* ou *DR* semblent fonctionner mieux, pour ce type de méthode, que les relations simples (*R*).

Comme pour la variante Lesk simplifiée, les meilleures performances sont produites par les méthodes avec les taux de décisions par défaut les plus élevés (20-70% *PA,* 75-95% *CL*).

Ces observations semblent expliquer le meilleur comportement du système pour la variante *RF* qui pondère le score d'un sens candidat avec un indicateur de fréquence relative (les tables 5.2 et 5.3) et qui favorise, par conséquent, le choix d'un sens plus fréquent que celui d'un sens moins utilisé.

Le tableau 5.5 montre la même étude menée pour le corpus de test extrait de *Semcor* et la variante *RF.* La tendance générale, manifestée aussi pour le corpus *Senseval2,* est la décroissance de la précision avec la diminution du nombre de décisions par défaut, pour la variante Lesk simplifiée.

Tab. 5.5. Nombre de décision par défaut et précision *fine-grained* pour l'ensemble de test de *Semcor,* implémentation *RF*



La variante Lesk originelle, dans sa variante *RF,* indique par contre une légère croissance de la précision si le nombre de décisions par défaut diminue. Ce comportement est un peu différent pour les versions *PA* et *CL* qui présentent une décroissance faible de la précision, pour les deux cas.

La méthode que Lesk a proposée dépend – comme il le souligne – directement de la qualité des descriptions de sens et de la quantité d'information disponible pour la désambiguïsation. L'auteur rapporte à ce sujet les résultats de sa méthode pour quatre dictionnaires différents et conclut qu'un dictionnaire plus informatif détermine de meilleures performances. De même, Litkowski (2002), qui a testé son système pour deux dictionnaires *WordNet* et *NODE* (New Oxford Dictionary of English), relie la quantité d'information disponible à la désambiguïsation au nombre de décisions prises par défaut. A l'aide de cet indicateur il conclut qu'il y a plus d'information dans *NODE* que dans *WordNet.* Haynes (2001) remarque aussi que l'absence des exemples d'usage pour certains sens candidats dans *WordNet* défavorise le score attribué à ceux-ci par son système.

Dans ce travail nous avons utilisé uniquement *WordNet.* Notre étude sur le nombre de décisions par défaut indique que le manque d'information nécessaire à la désambiguïsation (cas sans superpositions) est plus fréquent pour les relations que pour les définitions et les exemples ou pour les définitions, les relations et les exemples pris ensemble. Les descriptions de type définitions + exemples ou définitions + relations + exemples semblent donc plus informatives que les relations toutes seules dans l'utilisation de *WordNet* pour la désambiguïsation automatique.

En général, compenser l'absence d'information du dictionnaire par l'augmentation du contexte (soit en considérant des contextes plus larges, soit en étendant le contexte par les descriptions des mots) ne se traduit pas nécessairement par une augmentation de performance.

5.3.3. Fréquence relative des sens candidats

Dans la section 3.1.1 nous avons introduit une méthode pour l'ordonnancement des sens candidats d'un mot cible, en calculant un indicateur de fréquence *freq_id* à partir de l'information de *WordNet.* Rappelons que cet indicateur apparaît dans le fichier de test

prétraité, composé par des lignes de la forme ci-dessous, après l'identificateur de chaque sens candidat.

[art, art#1&29.0, art#2&8.2, art#3&3.0, art#4&0.19, d00 d00.s00.t01]

Par exemple, *freq_id(art#1)=29, freq_id(art#2)=8.2*, etc. L'implémentation *RF* susmentionnée fait appel à *freq_id* pour déterminer l'indicateur de fréquence relative d'un sens s_j, $f_r(s_j)$, utilisé pour pondérer le score d'un sens candidat (calculé par une des méthodes décrites dans le chapitre 4). La nouvelle valeur du score devient :

$$score \leftarrow score \cdot f_r(s_j) \tag{24}$$

où :

$$f_r(s_j) = \frac{freq\_id(s_j)}{\sum_{i=1,n} freq\_id(s_i)} \tag{25}$$

et s_j représente un des sens d'un mot à n sens.

Par exemple pour le sens *art#1* du mot *art* considéré plus haut :

$$f_r(art\#1) = \frac{29}{29 + 8.2 + 3 + 0.19} = 0.71$$

Le tableau 5.6 décrit la précision du système pour le calcul des scores, avec (*RF*) et sans fréquence relative (*NRF*), dans le cas de corpus *Senseval2*.

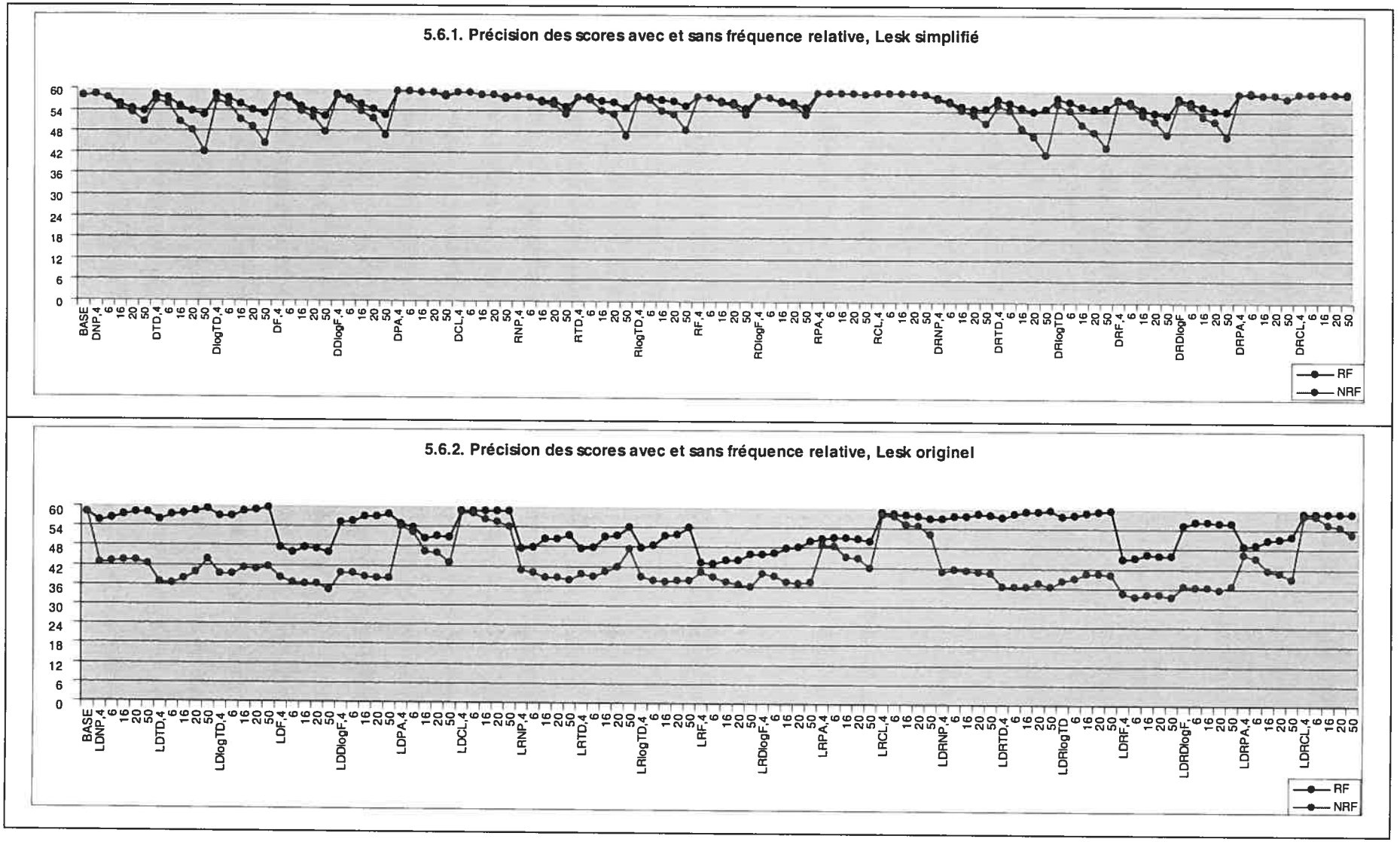
Pour ce qui est de la précision de l'algorithme simplifié (5.6.1), on constate que l'ajout de la fréquence relative produit une atténuation de la tendance de décroissance avec l'augmentation du contexte, pour les variantes *TD, logTD, F, DlogF*. Les versions *NP* et surtout *PA* et *CL* semblent moins influencées par ce facteur.

Par contre, l'amélioration est plus visible pour l'algorithme originel (5.6.2) surtout dans le cas des variantes *NP, TD, logTD, DlogF* et les descriptions *D* et *DR,* qui enregistrent des gains de performances de 15-20% par rapport à la variante *NRF*.

De plus, on observe pour la méthode originelle, une légère croissance avec l'augmentation du contexte, plus marquée pour les variantes *LDlogF, LRNP, LRlogTD, LRF, LRDlogF, LRPA, LDRPA,* qui dans leur version *NRF* présentaient une pente négative de la précision pour les cinq contextes considérés.

Le contexte global influence donc d'une manière positive (pas très forte cependant) les performances du système. Par conséquent, la fréquence relative des sens candidats semble améliorer un peu les performances du système, en réduisant le bruit déterminé par le choix des sens moins fréquents.

Tab. 5.6. Précision *fine-grained* du système pour le calcul des scores avec et sans fréquence relative, corpus *Senseval2*



5.3.4. Topologie des réponses par rapport aux choix de *BASE*

Dans les sections 5.3.1 à 5.3.3, nous avons analysé les performances du système et les éléments déterminant l'évolution de celles-ci. Un autre problème que nous semble intéressant est l'étude du gain par rapport aux performances de *BASE*. Le tableau 5.7. résume les expériences, triées selon la *f-measure*, qui ont produit des gains absolus supérieurs à 1%, par rapport à la précision et au rappel de *BASE*, pour le corpus *Senseval2*. Les expériences ont été regroupées en fonction de la caractéristique *RF / NRF* (avec et sans fréquence relative).

Tab. 5.7. Gains *fine-grained* par rapport aux performances de BASE, (corpus *Senseval2)*

Méthode / contexte		Prec	Rapp	Fmes	Gain absolu (%) *Prec*	*Rapp*	*Fmes*	Gain relatif (%) *Prec*	*Rapp*	*Fmes*
	BASE	*57.99*	*57.62*	*0*	*0*	*0*	*0*	*0*	*0*	*0*
NRF	DPA,6	59.4	58.88	59.14	1.41	1.25	1.33	2.43	2.19	2.31
	DPA,4	59.36	58.84	59.10	1.37	1.21	1.29	2.36	2.12	2.24
	DRPA,6	59.16	58.63	58.89	1.16	1.01	1.09	2.02	1.75	1.88
	DPA,16	59.12	58.59	58.85	1.12	0.97	1.05	1.95	1.68	1.82
	DCL,6	59.12	58.59	58.85	1.12	0.97	1.05	1.95	1.68	1.82
	RCL,4	59.12	58.59	58.85	1.12	0.97	1.05	1.95	1.68	1.82
	DRPA,4	59.12	58.59	58.85	1.12	0.97	1.05	1.95	1.68	1.82
	DPA,20	59.08	58.55	58.81	1.08	0.93	1.01	1.88	1.61	1.75
	DCL,4	59.08	58.55	58.81	1.08	0.93	1.01	1.88	1.61	1.75
	DRCL,4	59.08	58.55	58.81	1.08	0.93	1.01	1.88	1.61	1.75
RF	**LDlogTD,50**	**59.81**	**59.28**	**59.54**	**1.81**	**1.66**	**1.74**	**3.14**	**2.88**	**3.01**
	LDRTD,50	59.57	59.04	59.30	1.57	1.42	1.50	2.72	2.46	2.59
	LDRlogTD,50	59.57	59.04	59.30	1.57	1.42	1.50	2.72	2.46	2.59
	LDRlogTD,20	59.36	58.84	59.10	1.37	1.21	1.29	2.36	2.12	2.24
	DPA,6	59.32	58.79	59.05	1.33	1.17	1.25	2.29	2.03	2.16
	DPA,4	59.28	58.75	59.01	1.28	1.13	1.21	2.22	1.96	2.09
	LDRTD,16	59.16	58.63	58.89	1.16	1.01	1.09	2.02	1.75	1.88
	LDRTD,20	59.16	58.63	58.89	1.16	1.01	1.09	2.02	1.75	1.88
	DCL,6	59.12	58.59	58.85	1.12	0.97	1.05	1.95	1.68	1.82
	RCL,4	59.12	58.59	58.85	1.12	0.97	1.05	1.95	1.68	1.82
	DCL,4	59.08	58.55	58.81	1.08	0.93	1.01	1.88	1.61	1.75
	RCL,6	59.08	58.55	58.81	1.08	0.93	1.01	1.88	1.61	1.75
	DRPA,6	59.08	58.55	58.81	1.08	0.93	1.01	1.88	1.61	1.75
	DRCL,4	59.08	58.55	58.81	1.08	0.93	1.01	1.88	1.61	1.75
	LDTD,50	59.08	58.55	58.81	1.08	0.93	1.01	1.88	1.61	1.75
	DPA,16	59.04	58.51	58.77	1.04	0.89	0.97	1.81	1.54	1.68
	DRPA,4	59.04	58.51	58.77	1.04	0.89	0.97	1.81	1.54	1.68
	DRCL,6	59.04	58.51	58.77	1.04	0.89	0.97	1.81	1.54	1.68
	DRCL,16	59.04	58.51	58.77	1.04	0.89	0.97	1.81	1.54	1.68

Les formules utilisées pour le calcul de la *F-mesure*, des gains absolus, et relatifs sont les suivantes :

$$F\text{-}mes = \frac{2\,Prec \cdot Rapp}{Prec + Rapp} \tag{26}$$

$$Gain_{abs}(exp) = Val(exp) - Val(BASE) \tag{27}$$

$$Gain_{rel}(exp) = \frac{Val(exp) - Val(BASE)}{Val(BASE)} \cdot 100 \tag{28}$$

où : *Val(exp)* – représente la valeur d'une des 3 mesures de performance *(Prec, Rapp ou F-mes)* pour une expérience donnée; *Val(BASE)* – représente la valeur d'une des 3 mesures de performance *(Prec, Rapp ou F-mes)* pour *BASE.*

On observe que le groupe *NRF* comporte seulement des versions *PA* et *CL,* la meilleure performance (1.41% gain absolu et 2.43% gain relatif en précision) étant produite par la variante *DPA,6* (description définitions + exemples et un contexte de 6 mots autour du mot cible). Le groupe *RF* renferme plusieurs variantes *(PA, CL, TD, logTD)*, le meilleur gain absolu (1.81%) et relatif (3.14%) en précision est observé pour la variante *LDlogTD,50* (description définition + exemples et un contexte de 50 mots).

Une autre façon d'appréhender les performances d'un système est de compter le nombre de fois où ce système fait une réponse juste qui n'est pas le sens le plus fréquent ainsi que le nombre de fois où le système n'a pas choisi à tort le sens le plus fréquent. La différence de ces deux comptes permet en quelque sorte d'analyser la prise de risque "payante" / "bénéfique" qu'un système prend.

A partir de l'information fournie par le module d'analyse, nous avons effectué une étude sur les types de réponses générées par le système. Afin de mieux expliciter la dynamique du gain absolu par rapport aux performances de base, nous présentons en table 5.8 une description des catégories[5] de réponses pour les variantes sélectionnées dans la table 5.7, pour le corpus *Senseval2*. Les encodages utilisés ont la signification suivante :

CE=B – taux de réponses correctes, communes avec les réponses de BASE, en cas de nombre de superpositions non zéro[6] (décisions **E**ffectives **C**orrectes **communes** avec **B**ASE);

$CE \neq \overline{B}$ – taux de réponses correctes, différentes des réponses de BASE, prises par le système en cas de nombre de superpositions non zéro (décisions **E**ffectives **C**orrectes **différentes** de **B**ASE);

---

[5] Pour le corpus de test *Senseva2, English All Words Task* il y a une proportion d'approximativement 2.8% occurrences auxquelles aucun sens n'a pas été attribué par les annotateurs (pas d'accord sur le sens à choisir). Ces occurrences, marquées par la lettre *U (Unassigned)* dans le fichier clé, ont été considérées comme des réponses incorrectes par notre module d'évaluation.

[6] Nombre d'overlaps non zéro.

$C\overline{E} = B$ – taux de réponses correctes communes avec les réponses de **B**ASE, dans l'absence des superpositions (décisions par **défaut** Correctes) ;

$\overline{C}E \neq B$ – taux de réponses incorrectes à cause de ne pas choisir le sens le plus fréquent, choisit par BASE (décisions **E**ffectives **I**ncorrectes, **différentes** de BASE, **C**orrectes dans **B**ASE).

$\overline{C}E = \overline{B}$ – taux de réponses incorrectes communes avec les réponses de BASE, sous conditions de nombre de superpositions non zéro (décisions **E**ffectives **I**ncorrectes **communes** avec **B**ASE);

$\overline{C\overline{E}} = \overline{B}$ – taux de réponses incorrectes communes avec les réponses de base, prises en absence des superpositions (décisions par **défaut** **I**ncorrectes, **communes** avec **B**ASE);

$\overline{C}E \neq \overline{B}$ – taux de réponses incorrectes différentes de BASE, incorrectes dans BASE, choisies sous conditions de nombre de superpositions non zéro (décisions **E**ffectives **I**ncorrectes **différentes** de **B**ASE, **I**ncorrectes dans **B**ASE).

Les taux sont calculés par rapport au nombre de cas traités pour chaque expérience. Un schéma de l'encodage est présenté dans la figure 5.1:

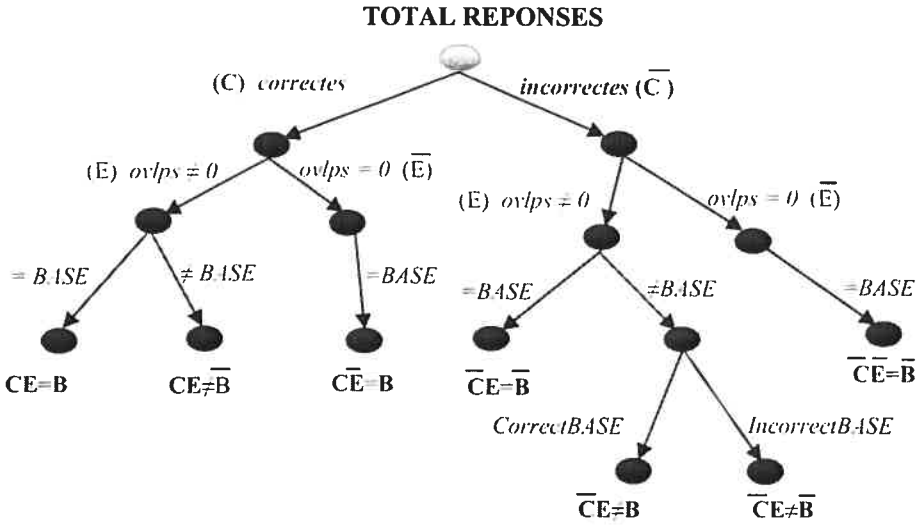
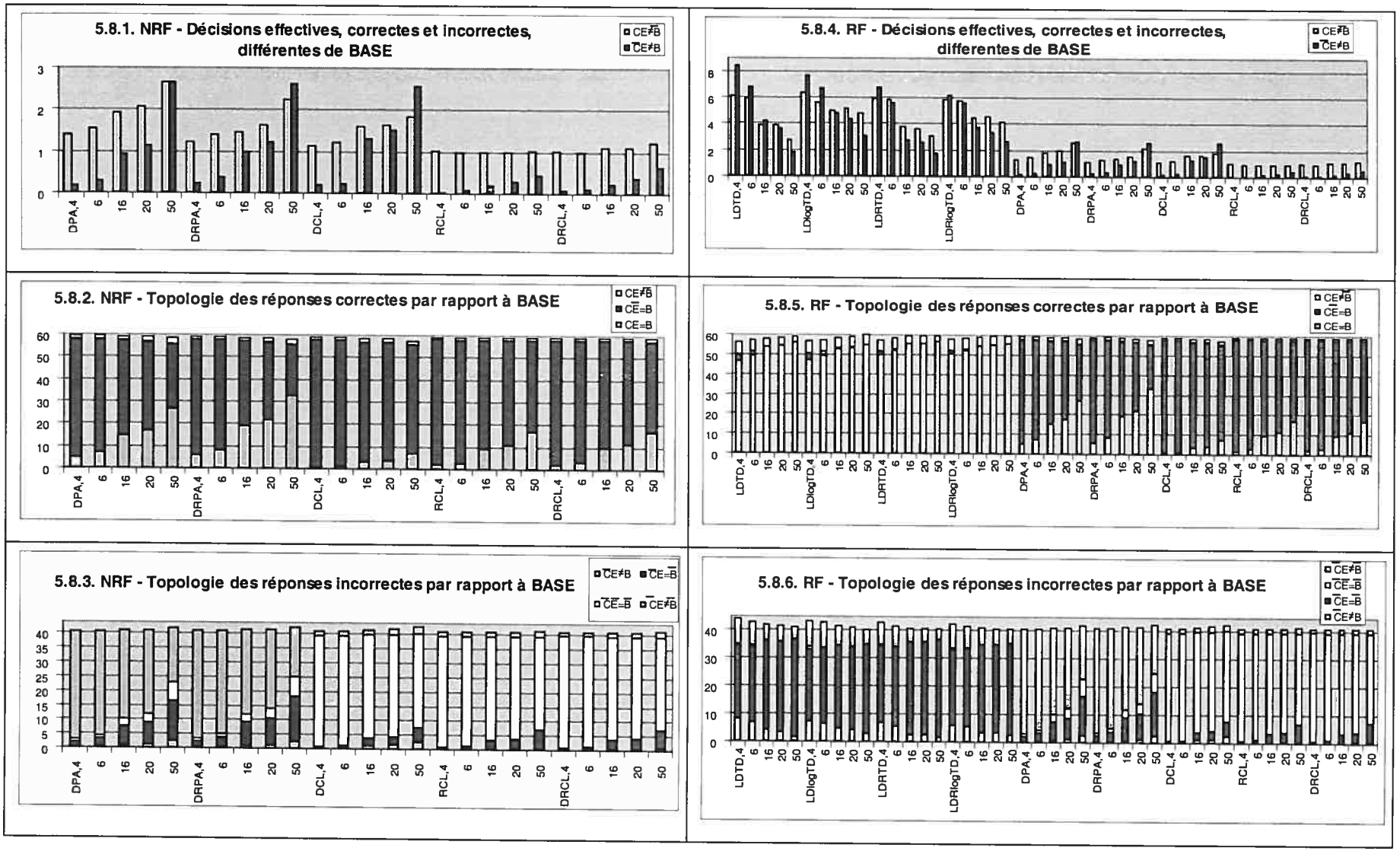


Fig. 5.1. Schéma d'encodage des réponses

Les diagrammes 5.8.1, 5.8.4 montrent le rapport entre le nombre de décisions effectives correctes, différentes des décisions de *BASE* ($CE \neq \overline{B}$), et le nombre de décisions effectives incorrectes, dues au choix d'un autre sens que le sens le plus fréquent ($\overline{C}E \neq B$), pour les deux types d'expériences *NRF* et *RF*. La différence entre les deux mesures ($CE \neq \overline{B}, \overline{C}E \neq B$), donne le gain absolu par rapport aux performances de *BASE*.

Tab. 5.8. Catégories de réponses pour 4 types de méthodes,corpus *Senseval2,* évaluation *fine-grained*



On observe pour les méthodes *PA* et *CL* une tendance de croissance des réponses correctes différentes de BASE avec l'augmentation du contexte[7], contrebalancée par une croissance des réponses incorrectes, ce qui produit graduellement la diminution du gain absolu[8]. En revanche, pour les méthodes *L-TD* et *L-logTD,* pondérées par la fréquence relative *(RF),* l'augmentation du contexte produit une décroissance des réponses correctes différentes de BASE, mais à la fois, une décroissance des réponses incorrectes[9], le gain absolu étant donné par la différence entre les pentes de décroissance des deux mesures. Une première explication sur les valeurs assez petites du gain absolu consisterait donc dans le fait que les deux mesures varient en même temps et de manière presque similaire.

L'analyse de la structure des réponses correctes (5.8.2, 5.8.5) apporte plus d'éclaircissements sur ce problème. On observe que pour toutes les variantes testées, le taux des réponses correctes différentes de *BASE* ($CE \neq \overline{B}$) est très petit (les segments supérieurs des battons, en couleur claire). La majorité des réponses correctes coïncide avec les réponses correctes de BASE; soit qu'elles sont prises par défaut, en absence de superpositions ($C\overline{E} = B$), soit qu'elles sont produites par des décisions effectives *(CE=B).*

On peut remarquer un taux assez élevé des décisions effectives correctes, communes avec *BASE (CE=B)* surtout pour les méthodes *L-TD, L-logTD*[10], ce qui semble indiquer que sous des conditions d'information suffisante (nombre de superpositions non zéro) la plupart des réponses correctes du système pour ce type de méthodes coïncide avec les réponses données par le choix du sens le plus fréquent. Pourtant la normalisation avec la taille de la description de sens ou avec le log de cette mesure semble atténuer un peu cette tendance, les plus grandes valeurs de décisions effectives correctes, différentes de BASE ($CE \neq \overline{B}$), étant produites par ces types de méthodes (entre 3-6% des cas traités, selon 5.8.4).

Quant aux méthodes *PA, CL,* caractérisées par un nombre plus petit de décisions effectives correctes différentes de *BASE* (entre 1-2% des cas traités, voir 5.8.1., 5.8.4.), la plupart de leurs réponses correctes est donnée par les choix par défaut.

Les diagrammes 5.8.3, 5.8.6 présentent la structure des réponses incorrectes du système. Les variantes *PA* doivent la majorité de leurs erreurs aux choix d'un autre sens

---

[7] Moins évidente pour *RCL* et *DRCL.*
[8] Tendance observée pour toutes les expériences de la classe NRF.
[9] Les autres expériences de la classe RF présentent le même comportement.
[10] Observation aussi valable pour les autres variantes de l'algorithme de Lesk originel: *L-NP, L-LF, L-logF* .

que le sens le plus fréquent $(\overline{CE} \neq \overline{B})$. Par contre, les méthodes *CL* et *L-TD, L-logTD* dans la plupart des cas, donnent des réponses incorrectes en choisissant le sens le plus fréquent, les unes par défaut $(\overline{CE} = \overline{B})$, les autres par des décisions effectives $(\overline{CE} = \overline{B})$.

Pour ce qui du corpus extrait de *Semcor,* le tableau 5.9 présente par ordre décroissant de *F-mesure* les variantes, caractérisées par un gain en précision supérieur à 0.5%, pour l'implémentation *RF*.

Tab. 5.9. Gains *fine-grained* par rapport à BASE (corpus *Semcor)*

Méthode / contexte	Prec	Rapp	Fmes	Gain absolu (%)			Gain relatif (%)		
				Prec	*Rapp*	*Fmes*	*Prec*	*Rapp*	*Fmes*
BASE	*66.37*	*66.36*	66.36	*0*	*0*	0	0	0	0
RPA,4	67.30	67.29	67.29	0.93	0.93	0.93	1.40	1.40	1.41
RPA,16	67.30	67.29	67.29	0.93	0.93	0.93	1.40	1.40	1.41
RPA,6	67.29	67.28	67.28	0.92	0.92	0.92	1.39	1.39	1.39
RPA,20	67.29	67.28	67.28	0.92	0.92	0.92	1.39	1.39	1.39
DRPA,4	67.25	67.23	67.24	0.88	0.88	0.87	1.33	1.31	1.33
DRCL,4	67.24	67.22	67.23	0.87	0.86	0.86	1.31	1.30	1.31
LDRTD,50	67.23	67.20	67.21	0.86	0.84	0.86	1.30	1.30	1.30
DPA,4	67.23	67.22	67.22	0.86	0.86	0.85	1.30	1.28	1.30
RCL,4	67.23	67.21	67.22	0.86	0.86	0.85	1.30	1.27	1.29
DRCL,6	67.21	67.20	67.20	0.84	0.84	0.84	1.27	1.27	1.27
RPA,50	67.20	67.19	67.19	0.83	0.83	0.83	1.25	1.25	1.26
RCL,6	67.19	67.18	67.18	0.82	0.82	0.82	1.24	1.24	1.24
DPA,6	67.18	67.17	67.17	0.81	0.81	0.81	1.22	1.22	1.23
DRPA,6	67.18	67.17	67.17	0.81	0.81	0.81	1.22	1.22	1.23
LDCL,4	67.08	67.07	67.07	0.71	0.71	0.71	1.07	1.07	1.08
DRCL,16	67.07	67.06	67.06	0.70	0.70	0.70	1.05	1.05	1.06
LDRCL,4	67.06	67.05	67.05	0.69	0.69	0.69	1.04	1.04	1.05
LDCL,6	67.04	67.02	67.03	0.67	0.67	0.66	1.01	0.99	1.01
RCL,16	67.03	67.02	67.02	0.66	0.66	0.66	0.99	0.99	1.00
DRCL,20	67.03	67.02	67.02	0.66	0.66	0.66	0.99	0.99	1.00
LDRCL,6	67.02	67.01	67.01	0.65	0.65	0.65	0.98	0.98	0.99
DPA,16	67.00	66.98	66.99	0.63	0.63	0.62	0.95	0.93	0.95
RCL,20	67.00	66.98	66.99	0.63	0.63	0.62	0.95	0.93	0.95
DRPA,16	66.98	66.97	66.97	0.61	0.61	0.61	0.92	0.92	0.93
LDTD,50	66.93	66.89	66.91	0.56	0.53	0.55	0.84	0.83	0.84
DRPA,20	66.92	66.91	66.92	0.55	0.55	0.54	0.84	0.80	0.83
DPA,20	66.92	66.90	66.91	0.55	0.54	0.54	0.82	0.82	0.83

On observe, en comparaison avec le corpus *Senseval2,* des valeurs plus petites du gain absolu. Une explication possible consisterait dans le fait que les résultats sont des valeurs moyennes et il y a des fichiers dont la proportion des verbes dépasse celle du corpus de *Senseval2.* Par exemple, dans *test6_7, test12_13, test16_17* les verbes représentent 25% du total des mots traités et 23.6% dans les fichiers *test0_1* et *test8_9,* par rapport à 22% pour *Senseval2* (voir chapitre 2 la description du corpus de test). Les autres fichiers contiennent en moyenne 21% verbes. De plus, les fichiers sus-mentionnés présentent les valeurs les plus grandes de l'entropie (entre 1.90 et 2.02), ce qui indique un degré de difficulté plus élevé que dans le cas de corpus *Senseval2* (1.91). Comme on l'a déjà vu, ce

sont les verbes qui présentent les valeurs les plus basses de la précision et du rappel et, par conséquent, il est probable qu'à cause de ces fichiers on obtient des gains moyens plus petits pour l'ensemble de données de test provenant de *Semcor.*

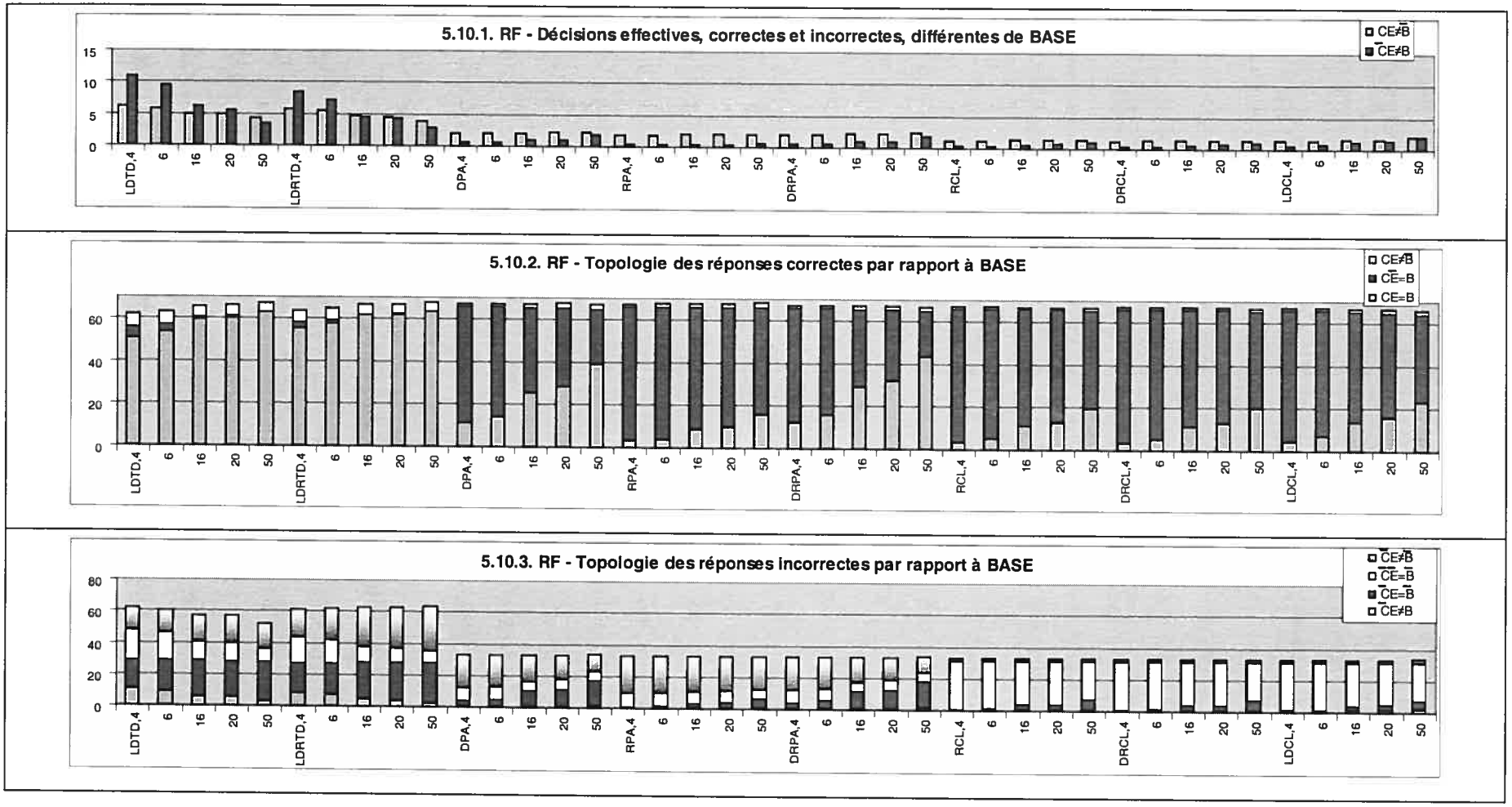
D'un autre côté, on pourrait constater que les variantes qui ont produit les meilleures performances dans le cas du corpus *Senseval2* sont les mêmes pour le corpus *Semcor,* i.e. *PA*, *CL* et *L-TD.* La variante *L-logTD* qui a donné de bons résultats pour *Senseval2* a produit aussi des gains positifs dans le cas du *Semcor,* mais inférieurs à 0.5%.

Le tableau 5.10. montre la structure des réponses pour les expériences qui ont produit les meilleurs gains (voir la table 5.9). Une analyse de cette topologie indique un comportement du système similaire pour les deux types de corpus de test, de *Senseval2* et de *Semcor*. Le gain par rapport à *BASE,* donné par la différence entre le taux des réponses correctes différentes de *BASE* $(CE \neq \overline{B})$ et le taux des réponses incorrectes à cause de ne pas choisir le sens le plus fréquent $(\overline{CE} \neq B)$, comporte des valeurs assez petites parce que les deux mesures $(CE \neq \overline{B}, \overline{CE} \neq B)$ varient de manière similaire (5.10.1).

Si on regarde la structure des réponses correctes (5.10.2), on peut constater aussi que le taux des réponses correctes, différentes de *BASE* $(CE \neq \overline{B})$, est assez bas (moins de 10% des cas traités) et que la plupart des réponses correctes coïncident avec les réponses de BASE, choisis par défaut (méthodes *PA, CL*) ou par décisions effectives (méthodes *L-TD).* Quant aux réponses incorrectes (5.10.3.), elles sont produites en majorité par des choix effectifs communs avec *BASE* (méthodes *L-TD*), par des choix d'un autre sens que le plus fréquent (*PA*) ou par des choix par défaut *(CL).* Ces observations semblent indiquer qu'en général les descriptions de sens de *WordNet* favorisent les sens plus fréquents, c'est-à-dire que pour ceux-ci il y a plus d'information pertinente et utile à désambiguïsation que pour les sens moins fréquents.

Relativement aux descriptions des sens, Véronis (2001) et Palmer et al. (2002) soulignent que les définitions de *WordNet* et des dictionnaires en général, ne fournissent pas l'information nécessaire à la désambiguïsation dans beaucoup de cas. Les définitions, dédiées seulement à la description des sens, sont souvent trop vagues, contradictoires ou incomplètes pour permettre une discrimination correcte entre les sens d'un mot. Une information reliée à l'usage des mots dans des contextes réels (de nature syntaxique, pragmatique, collocations etc.), ajoutée aux entrées d'un dictionnaire, serait plus utile à ce type de tâche.

Tab. 5.10. Topologie des réponses par rapport à *BASE* pour le corpus *Semcor*, évaluation *fine-grained*, implémentation *RF*



5.3.5. Catégorie grammaticale

Une autre influence que nous avons étudiée est celle de la catégorie grammaticale. Cette étude comporte deux aspects : les valeurs de la précision pour chaque catégorie grammaticale et le gain en performance si la catégorie grammaticale est connue, par rapport à la situation où le système ne dispose pas de cette information. On rappelle que dans le cas APOS (catégorie grammaticale à priori connue), nous avons utilisé l'information supplémentaire sur la catégorie grammaticale fournie dans des fichiers *treebank* (pour *Senseval2)* ou extraite directement du corpus *Semcor* (voir le chapitre 2).

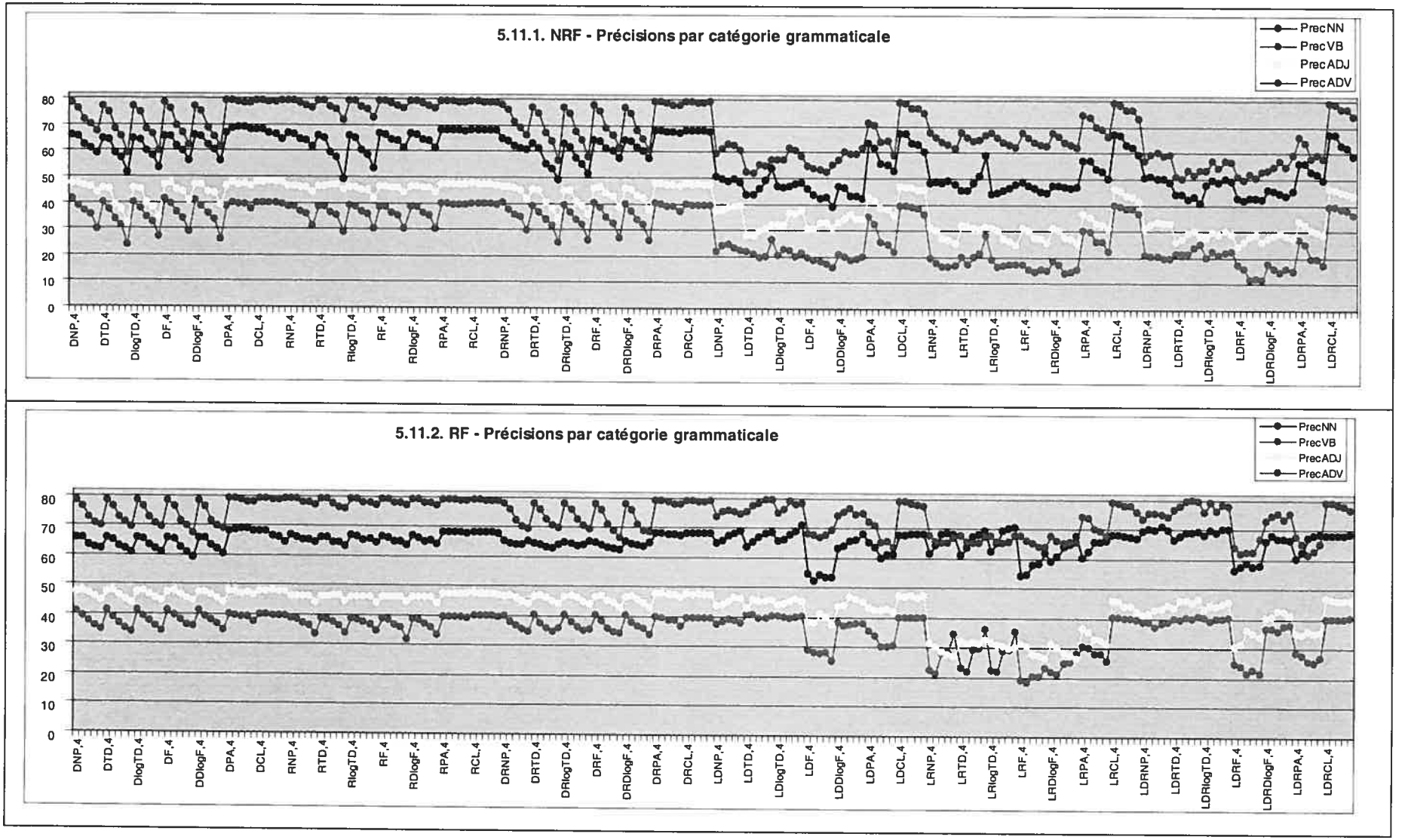
Il y a quand même des cas où le système est capable de détecter la catégorie grammaticale d'un mot à désambiguïser (même si cette information n'est pas fournie), à partir des instances de ce mot dans le corpus de test (par exemple *is* indique la catégorie verbale pour *be)* et des règles morphologiques ou des fichiers d'exceptions de *WordNet.* (voir chapitre 2). C'est, en effet, ce fonctionnement que nous avons évalué dans nos expériences.

Le tableau 5.11 présente les valeurs de la précision pour chaque catégorie grammaticale, les variantes *NRF, RF* et 5 longueurs du contexte. Pour des raisons de lisibilité seulement les codes des méthodes pour le contexte de 4 mots ont été illustrés.

Pour les deux classes de méthodes *NRF* (5.11.1) et *RF* (5.11.2), on observe un comportement similaire, c.a.d. les meilleures performances ont été obtenues pour les adverbes, suivis par les noms, les adjectifs et les verbes. En général, la précision diminue avec l'augmentation du contexte. Pourtant les versions *RF* des méthodes *L-NP, L-TD, L-logTD, L-DlogF* présente une croissance des performances par catégorie grammaticale avec le contexte, ce que nous avons déjà observé pour les performances globales.

Pour ces types de méthodes, cette tendance n'est pas manifeste pour les adjectifs si on utilise des descriptions de type *R,* ce qui semble indiquer que, dans le cas des adjectifs, les relations seules n'apportent pas une information suffisante à la désambiguïsation si on utilise des contextes plus larges. D'un autre côté, la tendance de croissance est plus évidente dans le cas des descriptions *D,* pour les noms et les adverbes, et des descriptions *R,* pour les noms et les verbes. Ce comportement pourrait indiquer le fait que, pour la variante *RF,* les influences à distance pour ces catégories grammaticales sont bien gérées par les descriptions de type *D* et *R.*

Tab. 5.11. Précisions *fine-grained* par catégorie grammaticale, corpus *Senseval2*



Si on compare les deux diagrammes, on peut remarquer aussi que l'amélioration des performances des méthodes *L-TD, L-logTD,* qui donnent les meilleurs résultats pour la variante *RF*, est premièrement due à l'augmentation des performances des noms et des adverbes (croissance de 15-30%).

Pour ce qui est des performances si la catégorie grammaticale des mots cibles est connue, le tableau 5.12 présente les variantes qui ont produit les meilleurs gains par rapport aux performances de *BASE* et par rapport à une nouvelle référence comportant le choix du sens le plus fréquent si la catégorie grammaticale est connue *(BASEAPOS).* On rappelle que *BASE* ne suppose aucun traitement concernant la catégorie grammaticale du mot cible. A des fins comparatives, le tableau 5.12 contient également les performances de *BASEDPOS*, obtenues par le choix du sens le plus fréquent après avoir détecté la catégorie grammaticale de certaines instances à tester (en utilisant les fichiers d'exception et les règles morphologiques de *WordNet,* voir aussi chapitre 3).

Tab. 5.12. Gains *fine-grained* par rapport à différentes performances de base, si la catégorie grammaticale est connue (corpus *Senseval2)*

Méthode / contexte		Prec	Rapp	F-mes	Gain absolu par rapport à BASE (%)			Gain absolu par rapport à BASEDPOS (%)			Gain absolu par rapport à BASEAPOS (%)		
					Prec	*Rapp*	*F-mes*	*Prec*	*Rapp*	*F-mes*	*Prec*	*Rapp*	*F-mes*
BASE		*57.99*	*57.62*	57.80	0	*0*	*0*	-1.12	-0.97	-1.04	-3.91	-3.68	-3.79
BASEDPOS		*59.11*	*58.59*	58.85	1.12	0.97	1.04	0	0	0	-2.79	-2.71	-2.75
BASEAPOS		*61.90*	*61.30*	61.60	3.91	3.68	3.79	2.79	2.71	2.75	0	0	0
NRF	DPA,6	62.23	61.63	61.93	4.24	4.01	4.12	3.12	3.04	3.08	0.33	0.32	0.32
	DPA,16	62.23	61.63	61.93	4.24	4.01	4.12	3.12	3.04	3.08	0.33	0.32	0.32
	DPA,4	62.19	61.59	61.89	4.2	3.97	4.08	3.08	3.00	3.04	0.29	0.28	0.28
	DPA,20	62.15	61.54	61.84	4.16	3.92	4.04	3.04	2.95	2.99	0.24	0.24	0.24
	DRPA,6	62.03	61.42	61.72	4.04	3.8	3.92	2.92	2.83	2.87	0.12	0.12	0.12
	DRPA,4	61.98	61.38	61.68	3.99	3.76	3.87	2.87	2.79	2.83	0.08	0.08	0.08
	DCL,6	61.94	61.34	61.64	3.95	3.72	3.83	2.83	2.75	2.79	0.04	0.04	0.04
	RCL,6	61.94	61.34	61.64	3.95	3.72	3.83	2.83	2.75	2.79	0.04	0.04	0.04
	DRCL,6	61.94	61.34	61.64	3.95	3.72	3.83	2.83	2.75	2.79	0.04	0.04	0.04
RF	LDlogTD,50	62.52	61.91	62.21	4.53	4.29	4.41	3.41	3.32	3.36	0.61	0.61	0.61
	LDRlogTD,50	62.47	61.87	62.17	4.48	4.25	4.36	3.36	3.28	3.32	0.57	0.57	0.57
	LDRTD,50	62.43	61.83	62.13	4.44	4.21	4.32	3.32	3.24	3.28	0.53	0.53	0.53
	LDRTD,16	62.19	61.59	61.89	4.2	3.97	4.08	3.08	3.00	3.04	0.29	0.28	0.28
	LDRlogTD,16	62.19	61.59	61.89	4.2	3.97	4.08	3.08	3.00	3.04	0.29	0.28	0.28
	DPA,6	62.15	61.54	61.84	4.16	3.92	4.04	3.04	2.95	2.99	0.24	0.24	0.24
	DPA,16	62.15	61.54	61.84	4.16	3.92	4.04	3.04	2.95	2.99	0.24	0.24	0.24
	LDRlogTD,20	62.15	61.54	61.84	4.16	3.92	4.04	3.04	2.95	2.99	0.24	0.24	0.24
	DPA,4	62.11	61.5	61.80	4.12	3.88	4.00	3.00	2.91	2.95	0.2	0.2	0.20
	DPA,20	62.11	61.5	61.80	4.12	3.88	4.00	3.00	2.91	2.95	0.2	0.2	0.20
	LDlogTD,20	62.07	61.46	61.76	4.08	3.84	3.96	2.96	2.87	2.91	0.16	0.16	0.16
	RCL,6	61.98	61.38	61.68	3.99	3.76	3.87	2.87	2.79	2.83	0.08	0.08	0.08
	DRCL,6	61.98	61.38	61.68	3.99	3.76	3.87	2.87	2.79	2.83	0.08	0.08	0.08
	LDRTD,20	61.98	61.38	61.68	3.99	3.76	3.87	2.87	2.79	2.83	0.08	0.08	0.08
	DCL,6	61.94	61.34	61.64	3.95	3.72	3.83	2.83	2.75	2.79	0.04	0.04	0.04
	DRPA,6	61.94	61.34	61.64	3.95	3.72	3.83	2.83	2.75	2.79	0.04	0.04	0.04
	LDRCL,4	61.94	61.34	61.64	3.95	3.72	3.83	2.83	2.75	2.79	0.04	0.04	0.04
	LDRCL,6	61.94	61.34	61.64	3.95	3.72	3.83	2.83	2.75	2.79	0.04	0.04	0.04

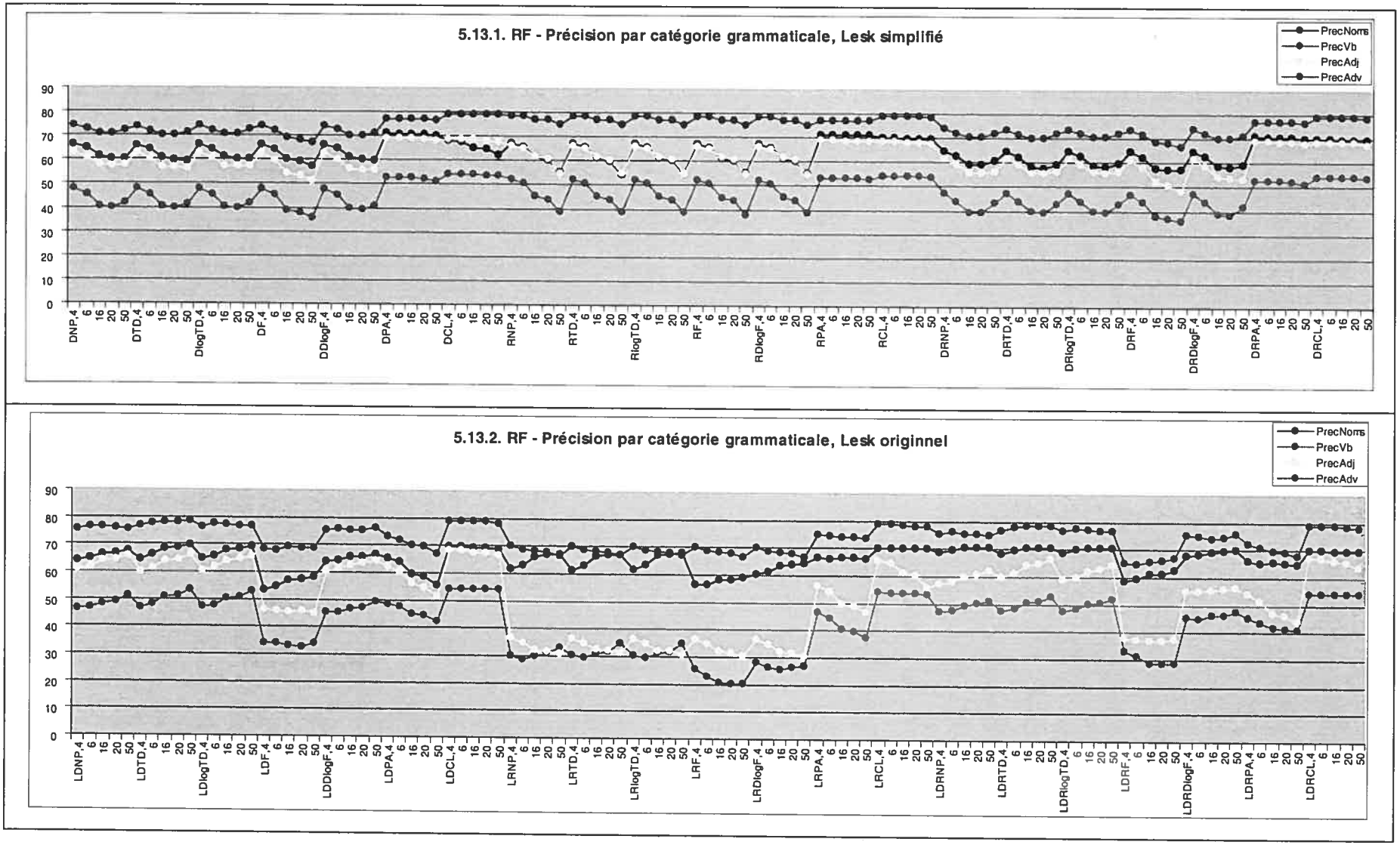
La table 5.12 montre seulement les versions (pour les classes *NRF* et *RF*) comportant des gains positifs comparativement à *BASEAPOS*. Bien que les gains absolus rapportés à *BASE* et à *BASEDPOS* soient supérieurs à 3.9%, respectivement à 2.8%, pour ces variantes, on pourrait remarquer que par rapport à *BASEAPOS* ils restent assez petits (moins de 1%). Cela semble indiquer la même tendance, discutée dans la section 5.3.4, de favoriser le choix du sens le plus fréquent, cette fois représenté par *BASEAPOS*. Les méthodes les plus performantes sont les mêmes que celles signalées dans la section 5.3.4: *PA* et *CL* pour la variante *NRF*, *L-logTD, L-TD, PA, CL* pour la variante *RF*.

Pour le corpus de test extrait de *Semcor,* comme pour le corpus *Senseval2,* les performances selon la catégorie grammaticale (voir la table 5.13) sont globalement comparables : les adverbes et les noms sont plus faciles à désambiguïser que les adjectifs et les adverbes. Pourtant, on observe que dans les variantes de la méthode Lesk simplifiée (5.13.1) la précision obtenue pour les adjectifs est plus proche de la précision des noms, que pour *Senseval2*. Ce comportement pourrait s'expliquer par les précisions de base qui présentent les mêmes valeurs moyennes pour les deux catégories considérées, précision 68.6% dans le cas *Semcor* à la différence d'une précision de base de 67.6% pour les noms et 46.8% pour les adjectifs, dans le cas de *Senseval2* (voir chapitre 2).

L'expansion du contexte par les descriptions des mots le composant semble désavantager les adjectifs dont les performances diminuent par rapport aux performances des noms dans la variante Lesk originelle (5.13.2). Cette tendance est plus accentuée pour les versions *LR-,* et *L-F* usant de ressources (relations de *WordNet*, fréquences du corpus *Hansard*) qui apparemment favorisent les noms par rapport aux adjectifs. Cependant, pour les versions *RF* des méthodes *LD[R]NP, LD[R]TD, LD[R]logTD, LD[R]logF* les adjectifs présentent des pentes de croissance comparables à celles des autres catégories. Pour ce qui est des autres catégories, on constate le même comportement des adverbes (meilleures précisions) et des verbes (les plus basses précisions) comme pour *Senseval2.*

Par conséquent, à partir des résultats de nos expériences, les catégories les plus favorisées dans *WordNet* semblent être les noms et les adverbes. Pourtant, les bonnes performances dans le cas des adverbes sont dues en même temps à une polysémie assez faible (2.55 sens par mots pour *Senseval2,* 2.42 pour *Semcor*), aux valeurs élevées de leur précision de base (79.0% - *Senseval2,* 80.50% - en moyenne pour *Semcor*) et encore aux valeurs basses de l'entropie (0.71 – *Senseval2* et en moyenne 0.66 pour *Semcor*).

Tab. 5.13. Précisions *fine-grained* par catégorie grammaticale, corpus *Semcor*



Ces caractéristiques (voir chapitre 2) présentent pour les noms des valeurs plus petites de la précision de base (67.6% – *Senseval2,* 68.6% – *Semcor)* et plus élevées du degré de polysémie ou de l'entropie (4.23 sens/mot et 1.16 entropie pour *Senseval2*, 4.22 sens par mot et 1.04 entropie pour *Semcor*), chose compensée, en revanche, par de bonnes descriptions de sens des noms dans *WordNet.*

Une étude des gains absolus si la catégorie grammaticale est connue, est résumée dans la table 5.14 qui montre les méthodes, classées par *F-mesure,* dont le gain absolu rapporté à la précision de *BASE* est supérieur à 5% si la catégorie grammaticale est connue. Les valeurs de *BASEDPOS* sont également présentées.

Tab. 5.14. Gains *fine-grained* par rapport à différentes performnaces de base, si la catégorie grammaticale est connue (corpus *Semcor)*

Méthode / contexte	Prec	Rapp	F-mes	Gain absolu par rapport à BASE (%)			Gain absolu par rapport à BASEDPOS (%)			Gain absolu par rapport à BASEAPOS (%)		
				Prec	*Rapp*	*F-mes*	*Prec*	*Rapp*	*F-mes*	*Prec*	*Rapp*	*F-mes*
BASE	*66.37*	*66.35*	66.36	0	*0*	*0*	-0.93	-0.94	-0.93	-6.76	-6.58	-6.67
BASEDPOS	67.30	67.29	67.29	0.93	0.94	0.93	0	0	0	-5.83	-5.64	-5.73
BASEAPOS	73.13	72.93	73.03	6.76	6.58	6.67	5.83	5.64	5.73	0	0	0
LDRCL,4 ; LDRCL,6	73.23	73.17	73.20	6.86	6.82	6.84	5.93	5.88	5.90	0.1	0.24	0.14
LRCL,4 ; RCL,6	73.18	73.11	73.14	6.81	6.76	6.78	5.88	5.82	5.85	0.05	0.18	0.08
RCL,4 ; DRCL,6	73.17	73.11	73.14	6.8	6.76	6.78	5.87	5.82	5.84	0.04	0.18	0.07
LRCL,6 ; DRCL,4	73.16	73.09	73.12	6.79	6.74	6.76	5.86	5.80	5.83	0.03	0.16	0.05
LDCL,4	73.11	73.04	73.07	6.74	6.69	6.71	5.81	5.75	5.78	-0.02	0.11	-0.05
LDCL,6 ; RCL,16 ; RCL,20	73.07	73.01	73.04	6.7	6.66	6.68	5.77	5.72	5.74	-0.06	0.08	-0.48
DRCL,16	73.03	72.96	72.99	6.66	6.61	6.63	5.73	5.67	5.70	-0.1	0.03	0.09
DRCL,20	73.02	72.95	72.98	6.65	6.6	6.62	5.72	5.66	5.69	-0.11	0.02	0.05
LDRCL,16	73.01	72.94	72.97	6.64	6.59	6.61	5.71	5.65	5.68	-0.12	0.01	0.02
LRCL,16	72.98	72.91	72.94	6.61	6.56	6.58	5.68	5.62	5.65	-0.15	-0.02	-0.04
LDRCL,20	72.97	72.9	72.93	6.6	6.55	6.57	5.67	5.61	5.64	-0.16	-0.03	-0.05
LRCL,20	72.93	72.86	72.89	6.56	6.51	6.53	5.63	5.57	5.60	-0.2	-0.07	-0.10
DCL,4	72.91	72.84	72.87	6.54	6.49	6.51	5.61	5.55	5.58	-0.22	-0.09	-0.13
RCL,50	72.86	72.79	72.82	6.49	6.44	6.46	5.56	5.50	5.53	-0.27	-0.14	-0.18
LDCL,16	72.84	72.77	72.80	6.47	6.42	6.44	5.54	5.48	5.51	-0.29	-0.16	-0.21
DRCL,50	72.79	72.72	72.75	6.42	6.37	6.39	5.49	5.43	5.46	-0.34	-0.21	-0.26
DCL,6	72.78	72.71	72.74	6.41	6.36	6.38	5.48	5.42	5.45	-0.35	-0.22	-0.27
LDCL,20	72.77	72.71	72.74	6.4	6.36	6.38	5.47	5.42	5.44	-0.36	-0.22	-0.27
LDRCL,50	72.71	72.64	72.67	6.34	6.29	6.31	5.41	5.35	5.38	-0.42	-0.29	-0.34
LRCL,50	72.63	72.57	72.60	6.26	6.22	6.24	5.33	5.28	5.30	-0.5	-0.36	-0.42
LDRTD,50	72.63	72.44	72.53	6.26	6.08	6.17	5.33	5.15	5.24	-0.5	-0.5	-0.50
LDTD,50	72.41	72.22	72.31	6.04	5.86	5.95	5.11	4.93	5.02	-0.72	-0.72	-0.72
LDCL,50	72.34	72.28	72.31	5.98	5.93	5.95	5.04	4.99	5.01	-0.78	-0.65	-0.71
DCL,16	72.21	72.15	72.18	5.84	5.8	5.82	4.91	4.86	4.88	-0.92	-0.78	-0.84
LDRlogTD,50	72.11	71.92	72.01	5.74	5.56	5.65	4.81	4.63	4.72	-1.02	-1.02	-1.02
DCL,20	72.04	71.98	72.01	5.68	5.63	5.65	4.74	4.69	4.71	-1.08	-0.95	-1.01
LDRTD,20	71.98	71.79	71.88	5.61	5.44	5.52	4.68	4.50	4.59	-1.15	-1.15	-1.15
RPA,4 ; RPA,16; RPA,20 ; RPA,6	71.72	71.58	71.65	5.36	5.22	5.29	4.42	4.29	4.35	-1.4	-1.36	-1.38
DRPA,4	71.69	71.55	71.62	5.33	5.19	5.26	4.39	4.26	4.32	-1.43	-1.39	-1.41
LDlogTD,50	71.71	71.52	71.61	5.34	5.16	5.25	4.41	4.23	4.32	-1.42	-1.42	-1.42
RPA,50 ; DPA,4	71.68	71.53	71.60	5.32	5.17	5.24	4.38	4.24	4.31	-1.44	-1.41	-1.42
DRPA,6	71.65	71.51	71.58	5.29	5.15	5.22	4.35	4.22	4.28	-1.47	-1.43	-1.45
DPA,6	71.64	71.49	71.56	5.27	5.13	5.20	4.34	4.20	4.27	-1.49	-1.45	-1.47
LDRTD,6	71.6	71.41	71.50	5.23	5.05	5.14	4.30	4.12	4.21	-1.53	-1.53	-1.53
LDRlogTD,20	71.54	71.35	71.44	5.17	5	5.08	4.24	4.06	4.15	-1.59	-1.59	-1.59
DRPA,16	71.45	71.3	71.37	5.09	4.95	5.02	4.15	4.01	4.08	-1.67	-1.64	-1.65
DPA,16	71.43	71.28	71.35	5.07	4.93	5.00	4.13	3.99	4.06	-1.69	-1.66	-1.67
DRPA,20	71.42	71.27	71.34	5.05	4.91	4.98	4.12	3.98	4.05	-1.71	-1.67	-1.69
DPA,20	71.39	71.24	71.31	5.02	4.88	4.95	4.09	3.95	4.02	-1.74	-1.7	-1.72

Comparativement au corpus de *Senseval2,* on remarque des valeurs plus grandes du gain absolu par rapport à *BASE* et à *BASEDPOS*, si la catégorie grammaticale est connue (gain maximal : 4.24%, 3.12% – *Senseval2,* 6.86%, 5.93% – *Semcor).* Cette différence correspond approximativement à la différence entre les performances de *BASE, BASEDPOS* et de *BASEAPOS,* pour les deux types de corpus considérés (voir chapitre 2).

On peut observer également que pour le corpus extrait de *Semcor,* les méthodes qui semblent bénéficier le plus de la catégorie grammaticale connue sont *CL, L-TD, L-logTD et PA*, comme pour le corpus de *Senseval2.* Il y a quand même une différence entre l'ordre du classement par méthode et une dominance de la méthode basée sur les chaînes lexicales *(CL).*

Dans leurs études sur la désambiguïsation, de nombreux auteurs s'accordent sur le fait qu'en général, les noms sont plus faciles à désambiguïser que les adjectifs ou les verbes (Kilgarrif et Rosenzweig 2000a), (Banerjee et Pedersen 2002), (Audibert 2003).

L'analyse des performances de notre système (testé sur les noms, les verbes, les adjectifs et les adverbes) indique, quant à elle, de meilleures performances pour les adverbes et pour les noms. Une explication possible de ce comportement pour les adverbes pourrait être le degré de polysémie et d'entropie assez bas selon *WordNet.* Dans le cas des noms, l'information fournie par *WordNet,* semble plus riche que pour les adjectifs et les verbes, fait signalé aussi par (Banerjee et Pedersen 2002).

Une autre observation concerne l'influence du contexte local, plus importante que celle du contexte global, sur la désambiguïsation des quatre catégories. Audibert (2003) présente, de la même manière, des pentes de décroissance avec le contexte pour les noms les adjectifs et les verbes. Cependant, nos expériences ont montré que l'augmentation du contexte pouvait être bénéfique si un filtrage adéquat est appliqué (variantes *RF, CL, PA*) et surtout pour les descriptions de type *D* et *DR.*

En ce qui concerne l'apport de la catégorie grammaticale *(APOS)*, le gain par rapport aux performances d'un système qui ne tient pas compte de cette information *(BASE)* ou le fait, mais pas de manière complète *(BASEDPOS),* est satisfaisant. Cependant, le gain rapporté à une base qui utilise elle aussi ce type d'information *(BASEAPOS)* ne dépasse pas 1%.

5.3.6. Interdépendance des sens choisis

Dans les variantes testées jusqu'à maintenant les prises de décision étaient indépendantes les unes des autres. Nous présentons ici la variante basée sur un tableau de votes qui suppose que les sens choisis sont reliés, c'est-à-dire les sens des mots superposés "accordent" et "reçoivent" des votes des sens candidats du mot cible (voir chapitre 4).

Les variantes qui font appel aux sens des mots du contexte sont les variantes de type *L (*par exemple, *LDNP, LDRPA, LRTD* etc.*)*, est c'est pourquoi le traitement par tableau de votes a été appliqué seulement à ce type d'implémentations.

Une analyse des résultats indiquent dans ces cas un comportement moins performant, inférieur aux performances de *BASE,* pour les méthodes *L-TD,L-logTD* qui ont produit des gains supérieurs à 1% dans le traitement séquentiel des sens. Par contre, la variante *CL* et dans une moindre mesure *PA* semble bénéficier également du traitement par tableau de votes. Une explication possible du bon comportement de la méthode *CL* consisterait dans le fait que pour cette version les votes sont accordés réciproquement seulement entre les sens des mots reliés sémantiquement (appartenant à la même chaîne lexicale), tandis que pour les autres variantes, ce type de contrainte n'existe pas et donc la probabilité des votes "parasites" est plus élevée.

Le tableau 5.15 montre les variantes[11] (dans l'ordre décroissant de *F-mesure)* qui ont produit des gains absolus supérieurs à 1% par rapport aux performances de *BASE,* pour l'implémentation par tableau de votes et le corpus *Senseval2.*

Tab. 5.15. Gains *fined-grained* par rapport à *BASE* implémentations avec tableau de votes (corpus *Senseval2)*

Méthode / contexte	Prec	Rapp	Fmes	Gain absolu (%) *Prec*	*Rapp*	*Fmes*	Gain relatif (%) *Prec*	*Rapp*	*Fmes*
BASE	*57.99*	*57.62*	57.80	*0*	*0*	0	0	0	0
LDCLV,16	59.16	58.63	58.89	1.17	1.01	1.09	2.02	1.75	1.88
LDRCLV,4	59.16	58.63	58.89	1.17	1.01	1.09	2.02	1.75	1.88
LDCLV,20	59.16	58.63	58.89	1.17	1.01	1.09	2.02	1.75	1.88
LDCLV,6	59.12	58.59	58.85	1.13	0.97	1.05	1.95	1.68	1.82
LDRCLV,20	59.12	58.59	58.85	1.13	0.97	1.05	1.95	1.68	1.82
LDRCLV,6	59.08	58.55	58.81	1.09	0.93	1.01	1.88	1.61	1.75
LRCLV,16	59.08	58.55	58.81	1.09	0.93	1.01	1.88	1.61	1.75
LDRCLV,4	59.00	58.47	58.73	1.01	0.85	0.93	1.74	1.48	1.61
LDRCLV,50	59.00	58.47	58.73	1.01	0.85	0.93	1.74	1.48	1.61
LDPAV,6	58.99	58.47	58.73	1.00	0.85	0.92	1.72	1.48	1.60

[11] La lettre V du code des méthodes indique la variante par tableau de Votes.

Le tableau 5.16 présente les variantes (triées par *F-mesure*) qui ont obtenu des gains positifs par rapport aux performances moyennes de *BASE,* pour l'ensemble de test extrait de *Semcor.* Les méthodes qui semblent se comporter mieux pour ce type d'implémentation sont, comme dans le cas de l'ensemble de test *Senseval2,* les méthodes *CL,* surtout pour les descriptions des sens de type *DR.*

Tab. 5.16. Gains *fine-grained* par rapport à *BASE,* implémentations avec tableau de votes (corpus *Semcor*)

Méthode / contexte	Prec	Rapp	Fmes	Gain absolu (%)			Gain relatif (%)		
				Prec	*Rapp*	*Fmes*	*Prec*	*Rapp*	*Fmes*
BASE	*66.37*	*66.36*	66.36	*0*	*0*	0	0	0	0
LDRCLV,4	67.02	67.01	67.01	0.65	0.65	0.65	0.98	0.98	0.99
LDRCLV,6	66.92	66.91	66.91	0.55	0.55	0.55	0.83	0.83	0.84
LDCLV,4	66.89	66.88	66.88	0.52	0.52	0.52	0.78	0.78	0.79
LDCLV,6	66.78	66.77	66.77	0.41	0.41	0.41	0.62	0.62	0.63
LDRCLV,16	66.58	66.57	66.57	0.21	0.21	0.21	0.32	0.32	0.32
LDRCLV,20	66.49	66.48	66.48	0.12	0.12	0.12	0.18	0.18	0.19
LRCLV,4	66.47	66.45	66.46	0.1	0.09	0.09	0.15	0.14	0.15
LDCLV,16	66.38	66.37	66.37	0.01	0.01	0.01	0.02	0.02	0.02

5.3.7. Granularité du découpage de sens

Nous avons jusqu'à maintenant rapporté les résultats selon la méthode de calcul *fine-grained* décrite en section 3.3.1. Ceci permet en effet de comparer nos résultats à ceux d'autres auteurs (ce que nous discutons en section 5.4.) et ce sur une base objective (les réponses sont évaluées strictement, selon l'inventaire de sens de *WordNet).* De nombreux auteurs (Voorhees 1998), (Véronis 2001), (Palmer et al. 2002), (Preiss et al. 2002) ont cependant mentionné que *WordNet* fait des distinctions trop fines. Pour en donner un exemple, prenons le nom *sound* qui possède 8 sens dans *WordNet* (Fig. 5.2) :

1. **sound** -- (the particular auditory effect produced by a given cause; "the sound of rain on the roof"; "the beautiful sound of music")
2. **sound**, auditory sensation -- (the subjective sensation of hearing something; "he strained to hear the faint sounds")
3. **sound** -- (mechanical vibrations transmitted by an elastic medium; "falling trees make a sound in the forest even when no one is there to hear them")
4. **sound** -- (the sudden occurrence of an audible event; "the sound awakened them")
5. audio, **sound** -- (the audible part of a transmitted signal; "they always raise the audio for commercials")
6. phone, speech sound, **sound** -- ((phonetics) an individual sound unit of speech without concern as to whether or not it is a phoneme of some language)
7. strait, **sound** -- (a narrow channel of the sea joining two larger bodies of water)
8. **sound** -- (a large ocean inlet or deep bay; "the main body of the sound ran parallel to the coast")

Fig. 5.2. Les 8 sens du mot *sound* selon *WordNet1.7.1.*

Il est quand même assez difficile de distinguer lequel des trois sens *1, 2,* ou *3* est plus approprié pour *sound* dans la phrase : "*No one speaks, and the snaking of the ropes seems to make as much* ***sound*** *as the bells themselves, muffled by the ceiling."* (les annotateurs du corpus de test de *Senseval2* ont attribué à cette occurrence tous les trois sens*)*.

Dès lors, l'évaluation *coarse-grained,* basée sur les principes décrits dans la section 4.3.1, est une méthode d'évaluation alternative qui tente à palier ce désavantage de *WordNet.* Nous avons ainsi testé le comportement du système pour un découpage moins fin des sens, selon la méthodologie décrite dans le chapitre 3. Les performances du système ont été rapportées à une nouvelle référence, calculée pour ce type de découpage.

Le tableau 5.17 présente les expériences (en ordre décroissant de *F-measure*) qui on produit des gains absolus supérieurs à 1%, par rapport aux performances de base *(BASECG)*, évaluées par la méthode *coarse-grained.*

Tab. 5.17. Gains par rapport aux performances de BASE, évaluation *coarse-grained* (corpus *Senseval2)*

Méthode / contexte		Précision	Rappel	F-mes	Gain absolu par rapport à BASE (%)			Gain absolu par rapport à BASECG (%)		
					Prec	*Rapp*	*F-mes*	*Prec*	*Rapp*	*F-mes*
BASE		*57.99*	*57.62*	57.80	0	*0*	*0*	-4.77	-4.73	-4.75
BASECG		*62.76*	*62.35*	62.55	4.77	4.73	4.75	0	0	0
NRF	DPA,6	64.34	63.77	64.05	6.35	6.15	6.25	1.58	1.42	1.50
	DPA,4	64.3	63.73	64.01	6.31	6.11	6.21	1.54	1.37	1.45
	DPA,16	64.14	63.57	63.85	6.15	5.95	6.05	1.38	1.21	1.29
	DRPA,6	64.1	63.53	63.81	6.11	5.91	6.01	1.34	1.17	1.25
	DPA,20	64.06	63.49	63.77	6.07	5.87	5.97	1.3	1.13	1.21
	RPA,4	64.06	63.49	63.77	6.07	5.87	5.97	1.3	1.13	1.21
	RPA,6	64.06	63.49	63.77	6.07	5.87	5.97	1.3	1.13	1.21
	DRPA,4	64.06	63.49	63.77	6.07	5.87	5.97	1.3	1.13	1.21
	RPA,16	63.93	63.36	63.64	5.94	5.74	5.84	1.17	1.01	1.08
	RPA,20	63.89	63.32	63.60	5.9	5.7	5.80	1.13	0.97	1.04
RF	LDlogTD,50	64.87	64.29	64.58	6.88	6.67	6.77	2.11	1.94	2.02
	LDRlogTD,50	64.59	64.01	64.30	6.6	6.39	6.49	1.83	1.66	1.74
	LDRTD,50	64.5	63.93	64.21	6.51	6.31	6.41	1.74	1.58	1.66
	DPA,6	64.34	63.77	64.05	6.35	6.15	6.25	1.58	1.42	1.50
	LDRlogTD,20	64.34	63.77	64.05	6.35	6.15	6.25	1.58	1.42	1.50
	DPA,4	64.3	63.73	64.01	6.31	6.11	6.21	1.54	1.37	1.45
	LDTD,50	64.18	63.61	63.89	6.19	5.99	6.09	1.42	1.25	1.33
	DPA,16	64.14	63.57	63.85	6.15	5.95	6.05	1.38	1.21	1.29
	DRPA,6	64.1	63.53	63.81	6.11	5.91	6.01	1.34	1.17	1.25
	LDRTD,16	64.1	63.53	63.81	6.11	5.91	6.01	1.34	1.17	1.25
	DPA,20	64.06	63.49	63.77	6.07	5.87	5.97	1.3	1.13	1.21
	RPA,4	64.06	63.49	63.77	6.07	5.87	5.97	1.3	1.13	1.21
	RPA,6	64.06	63.49	63.77	6.07	5.87	5.97	1.3	1.13	1.21
	DRPA,4	64.06	63.49	63.77	6.07	5.87	5.97	1.3	1.13	1.21
	LDlogTD,20	64.01	63.45	63.73	6.02	5.83	5.92	1.26	1.09	1.17
	LDRlogTD,16	64.01	63.45	63.73	6.02	5.83	5.92	1.26	1.09	1.17
	LDRTD,20	63.97	63.4	63.68	5.98	5.78	5.88	1.21	1.05	1.12
	RPA,16	63.93	63.36	63.64	5.94	5.74	5.84	1.17	1.01	1.08
	RPA,20	63.89	63.32	63.60	5.9	5.7	5.80	1.13	0.97	1.04

On observe, en comparaison avec la table 5.7 que les valeurs des gains absolus par rapport à *BASECG* sont plus grandes que celles mesurées par rapport à *BASE* en évaluation

fine-grained. Pour la classe *NRF*, la seule méthode comportant des gains supérieurs à 1% est *PA*. Pour la classe RF les meilleurs résultats ont été enregistrés par les variantes *L-logTD*, *L-TD* et *PA*. En général et sans surprise, les méthodes semblent mieux fonctionner pour des distinctions de sens moins fines.

Cette tendance est observée également pour la partie étudiée de *Semcor*. Le tableau 5.18 présente les variantes comportant des gains supérieurs à 1% par rapport à *BASECG*. Là encore, les variantes qui se comportent mieux sont *PA*, *L-TD* et, en plus, *CL*.

Tab. 5.18. Gains par rapport aux performances de BASE, évaluation *coarse-grained*, implémentation *RF*, (corpus *Semcor)*

Méthode / contexte	Précision	Rappel	F-mes	Gain absolu par rapport à BASE (%)			Gain absolu par rapport à BASECG (%)		
				Prec	*Rapp*	*F-mes*	*Prec*	*Rapp*	*F-mes*
BASE moyenne	*66.37*	*66.35*	66.36	0.00	*0.00*	*0.00*	-2.24	-2.24	-2.24
BASECG moyenne	*68.61*	*68.59*	68.60	*2.24*	*2.24*	2.24	0.00	0.00	0.00
RPA,4	69.94	69.94	69.94	3.57	3.59	3.58	1.33	1.35	1.34
RPA,6	69.94	69.94	69.94	3.57	3.59	3.58	1.33	1.35	1.34
RPA,16	69.92	69.92	69.92	3.55	3.57	3.56	1.31	1.33	1.32
DRPA,4	69.9	69.9	69.90	3.53	3.55	3.54	1.29	1.31	1.30
RPA,20	69.9	69.9	69.90	3.53	3.55	3.54	1.29	1.31	1.30
DPA,4	69.89	69.89	69.89	3.52	3.54	3.53	1.28	1.3	1.29
DRCL,4	69.87	69.87	69.87	3.5	3.52	3.51	1.26	1.28	1.27
RCL,4	69.87	69.87	69.87	3.5	3.52	3.51	1.26	1.28	1.27
DRCL,6	69.86	69.86	69.86	3.49	3.51	3.50	1.25	1.27	1.26
DPA,6	69.86	69.86	69.86	3.49	3.51	3.50	1.25	1.27	1.26
DRPA,6	69.85	69.85	69.85	3.48	3.5	3.49	1.24	1.26	1.25
RCL,6	69.85	69.85	69.85	3.48	3.5	3.49	1.24	1.26	1.25
DRCL,16	69.72	69.72	69.72	3.35	3.37	3.36	1.11	1.13	1.12
LDRTD,50	69.71	69.7	69.70	3.34	3.35	3.34	1.1	1.11	1.10
DPA,16	69.69	69.69	69.69	3.32	3.34	3.33	1.08	1.1	1.09
RCL,16	69.68	69.68	69.68	3.31	3.33	3.32	1.07	1.09	1.08
RPA,50	69.68	69.68	69.68	3.31	3.33	3.32	1.07	1.09	1.08
DRCL,20	69.67	69.67	69.67	3.3	3.32	3.31	1.06	1.08	1.07
DRPA,16	69.67	69.67	69.67	3.3	3.32	3.31	1.06	1.08	1.07
RCL,20	69.65	69.65	69.65	3.28	3.3	3.29	1.04	1.06	1.05
DRPA,20	69.63	69.63	69.63	3.26	3.28	3.27	1.02	1.04	1.03
DPA,20	69.62	69.62	69.62	3.25	3.27	3.26	1.01	1.03	1.02

5.4. Pistes d'investigation

Cette section tente de tracer quelques pistes d'investigation que nous n'avons pas testées à fond mais qui semblent intéressantes dans le cadre des expériences que nous avons décrites dans ce chapitre. Les deux directions d'étude visent la combinaison des meilleurs décideurs et la détection de la catégorie grammaticale par le tagger de RALI. Les résultats de ces tentatives sont résumés dans les sections suivantes.

5.4.1. Combinaison des meilleurs décideurs

Dans une étude sur l'estimation de la *confiance* en une application statistique de traduction automatique, Gandrabur et Foster (2003) présentent le gain maximal obtenu par la combinaison optimale de plusieurs décideurs. Cette combinaison optimale serait obtenue par l'intermédiaire d'un *oracle* qui choisit toujours la réponse correcte.

A partir de cette idée, nous avons calculé le gain maximal donné par la combinaison de 3 des meilleurs décideurs ressortis de nos expériences : *LDLogTD,50, DPA,6, DCL,6* pour le corpus Senseval2 et *RPA,4, DRCL,4, LDRTD,50* pour le corpus *Semcor*. Pour éviter une estimation biaisée, nous avons combiné les meilleurs décideurs pour *Senseval2* afin de calculer le gain maximal pour *Semcor* et les meilleurs décideurs pour *Semcor* afin de déterminer le gain pour *Senseval2*. Les résultats[12] de cette estimation sont présentés dans le tableau 5.19.

Les mesures de performances et les gains ont été calculés par rapport à deux éléments de référence *BASE* (catégorie grammaticale non connue et non détectée) et *BASEAPOS* (catégorie grammaticale connue).

Tab. 5.19. Gain maximal par rapport à *BASE* et à *BASEAPOS*
si on combine les meilleurs décideurs

Variante	Corpus	Prec	Rapp	F-mes	Gain absolu (%)			Gain relatif (%)		
					Prec	*Rapp*	*F-mes*	*Prec*	*Rapp*	*F-mes*
Simple	*Senseval2*	**61.24**	60.69	60.96	3.25	3.07	3.16	5.60	5.32	5.46
	Semcor	**70.51**	70.49	70.50	**4.139**	4.135	4.137	6.236	6.232	6.234
APOS	*Senseval2*	**68.68**	68.01	68.34	6.78	6.71	6.74	10.95	10.94	10.94
	Semcor	**76.03**	75.89	75.96	**2.90**	2.96	2.93	3.97	4.06	4.02

Le tableau montre des gains absolus supérieurs aux gains absolus individuels, pour chaque décideur, ce qui semble indiquer que dans certains cas les 3 décideurs testés se comportent de manière complémentaire dans le choix des réponses, fait déjà mentionné par (Gandrabur et Foster 2003). Une étude approfondie sur la modalité précise de combinaison des décideurs et sur la prise de décision pourrait conduire à des résultats intéressants.

[12] Dans le cas du corpus *Semcor* il s'agit de valeurs moyennes, calculées pour les 10 fichiers de test

5.4.2. Détection de la catégorie grammaticale par le tagger RALI

Nous avons également utilisé le tagger de RALI pour déterminer les catégories grammaticales des mots cibles du corpus de test de *Senseval2*. Le tableau 5.20 présente la valeur des performances de base si le texte a été préalablement taggé par le tagger de RALI *(BASERALI)*. Comme attendu, cette valeur se situe entre les valeurs de *BASEDPOS* et *BASEAPOS* (voir. Tab. 5.12). Le tableau 5.20 indique aussi les performances de trois variantes qui ont produit de bons résultats *LDlogTD,50, DPA,6, RCL,6* dans le cas où la catégorie grammaticale est à priori connue (voir Tab. 5.12, implémentation RF). Les dernières trois colonnes du tableau 5.20 indiquent la perte en précision, rappel et *F-mesure* par rapport aux performances APOS, correspondant à chaque variante étudiée.

Tab. 5.20. Performances et pertes si on utilise le tagger RALI

Méthode/contexte	*Prec*	*Rapp*	*F-mes*	Perte par rapport à APOS (%)		
				Prec	*Rapp*	*F-mes*
BASERALI	60.44	59.88	60.16	-1.46	-1.42	-1.44
LDlogTD,50	61.14	60.57	60.85	-1.38	-1.34	-1.36
DPA,6	60.69	60.13	60.41	-1.46	-1.41	-1.43
RCL,6	60.53	59.96	60.24	-1.45	-1.42	-1.44

On observe, en moyenne, une perte en précision de -1.43% si on utilise le tagger RALI comparativement au cas où la catégorie grammaticale est connue.

5.5. Etude comparative

Le but de cette section est la comparaison des résultats de notre recherche avec les résultats d'autres systèmes testés sur le même corpus de test (le corpus de *Senseval2, English all words)* et décrits dans la littérature.

Le tableau 5.21 présente les performances[13] (triées par rappel) de 9 systèmes, d'un total de 22 participants à cette campagne : les 5 premiers systèmes supervisés, le meilleur

[13] Pour les résultats complets, voir le site officiel de *Senseval*.

système non-supervisé et 4 systèmes utilisant une approche basées sur l'information de type Lesk (définitions et exemples d'usage) extraite de *WordNet.* Nous présentons également les meilleures performance de nos expériences pour les trois cas: l'un *(APOS)* utilisant l'information sur la catégorie grammaticale mise à la disposition des participants sous la forme *treebank* (voir chapitre 2), le deuxième qui utilise le tagger RALI pour détecter la catégorie grammaticale et le troisième qui détecte la catégorie grammaticale à partir des formes instanciées. Le tableau contient les performances évaluées par les deux méthodes décrites auparavant, *fine-grained* et *coarse-grained.* Il faut mentionner que notre évaluation *coarse-grained* diffère de celle utilisée pour *Senseval2* où le regroupement des sens a été réalisé manuellement par les annotateurs.

Tab. 5.21. Performances des systèmes testés sur le corpus de test de *Senseval2, English all words task*

Système	***Fine-grained***		***Coarse-grained***	
	Précision (%)	**Rappel(%)**	**Précision (%)**	**Rappel (%)**
Compétition Senseval2, English All Words *(S – supervisés, U – non-supervisés)*				
Les 5 meilleurs systèmes				
SMUaw (S)	69	69	69.8	69.8
CNTS-Antwerp (S)	63.6	63.6	64.5	64.5
Sinequa-LIA – HMM (S)	61.8	61.8	62.6	62.6
UNED - AW-U2 (U)	57.5	56.9	58.3	57.7
UNED - AW-U (U)	55.6	55	56.5	55.9
4 systèmes utilisant l'information de type Lesk (définitions + exemples d'usage)				
CL Research – DIMAP (U)	45.1	45.1	46	46
IIT 2[14] (U)	32.8	3.8 (32.5)	33.5	3.9 (33.2)
IIT 3 (U)	29.4	3.4 (29.7)	30.1	3.5 (29.1)
IIT 1 (U)	28.7	3.3 (28.3)	29.4	3.4 (29.1)
Notre système				
Variante APOS	62.5	61.9	68.1*	67.5*
Variante tagRALI	61.1	60.5	66.8*	66.2*
Variante simple	59.8	59.2	64.8*	64.3*

[14] Les valeurs faibles du rappel pour les 3 systèmes IIT sont dues à des contraintes de temps, les systèmes n'ayant traité que 12% du corpus de test lors de la campagne. Les résultats entre parenthèses sont ceux rapportés ultérieurement par (Haynes 2001).

Le tableau 5.21 montre que seulement 4 systèmes participant à la compétition ont réussi à surpasser les performances de base[15], 57% précision et rappel *fine-grained,* selon la description officielle (Edmonds 2002).

Comparativement aux résultats officiels, notre système se trouve, par rapport au meilleur système supervisé (précision *fine-grained* 69%), à une distance de 6.5% (variante *APOS)* et de 9.2 % (variante simple). Rapportées aux performances du meilleur système non-supervisé (57.5% précision *fine-grained*), nos expériences indiquent un plus de 5% (APOS) et respectivement de 2.3% (variante simple). En pratique, la détection automatique complète de l'information sur la catégorie grammaticale entraîne une perte mesurée à approximativement 1.4%.

Dans l'interprétation comparative de ces résultats, qui porte seulement sur les valeurs des performances telles quelles, il convient de considérer le fait que les participants à la compétition ont été conditionnés par des contraintes de temps, ce qui n'a pas été notre cas, et que nous avons eu l'occasion de lancer de multiples tests en connaissant la réponse, ce qui n'était bien sûr pas possible aux compétiteurs. Cette comparaison n'est guidée que par un but comparatif.

[15] 57.9 dans nos expériences.

Conclusions

La démarche que nous avons entreprise dans le cadre du projet a visé principalement l'étude détaillée de l'algorithme de désambiguïsation proposé par Lesk. Les expériences effectuées ont eu comme but l'analyse du comportement du système pour différents paramètres et plusieurs variantes de la méthode. Afin de conférer un caractère plus général à notre recherche, le système a été testé pour plusieurs ensembles de test provenant de *Senseval2* et du corpus *Semcor.* Les sections suivantes résument les conclusions et les hypothèses tirées de notre étude.

6.1. Performances des différentes méthodes

Les variantes que nous avons analysées ont eu comme point de départ la variante originelle de l'algorithme de Lesk (1986) et la variante simplifiée du même algorithme, décrite par (Kilgarriff et Rosenzweig 2000a,b). Nous avons adapté ces variantes à la structure spécifique de *WordNet,* ainsi qu'au format des données d'entrée et de sortie imposé par les normes *Senseval* et *Semcor*. Afin de permettre un fonctionnement plus flexible du système, les procédures ont été conçues de telle manière qu'elles soient capables de répondre à un certain nombre de paramètres. Le choix des différents paramètres et caractéristiques constructives de notre système a été guidé par les études déjà mentionnées ou par l'environnement *Senseval* (longueur du contexte, taille de la description, distance par rapport au mot à désambiguïser, mesures de performances, granularité du découpage des sens) ou a été suggéré par l'évolution de nos propres expériences (fréquence des mots superposé, fréquence relative des sens candidats, apprentissage des poids, combinaison des descriptions de sens, nombre de décisions par défaut, structure des réponses, tableau de votes). De plus, à l'aide des résultats de nos expériences nous avons observé que tous les mots du contexte n'apportent pas nécessairement des informations utiles à la désambiguïsation. A partir de cette observation et du concept de *chaîne lexicale* introduit par Hirst et St-Odge (1998) pour la correction du *malapropisme,* nous avons dérivé une

variante de l'algorithme de Lesk qui prend comme contexte d'un mot cible seulement les mots de la chaîne lexicale dont ce mot est la tête.

L'analyse des résultats des différentes expériences indique un meilleur comportement pour trois variantes implémentées :

- la version de l'algorithme originel de Lesk normalisée par la taille de la description et par la fréquence relative des sens candidats *(RF-L-TD, RF-L-logTD)* surtout pour les deux types de description de sens *D* (définitions + exemples) et *DR* (définitions + exemples + relations);

- la variante simplifiée de la méthode de Lesk normalisée par les poids appris des mots superposés, pour les descriptions de types *D* et *DR* et les deux modalités de calcul du score (*RF* – pondéré avec la fréquence relative des sens candidats ou *NRF* - non pondéré);

- la variante simplifiée de l'algorithme de Lesk basée sur les chaînes lexicales, pour tous les types de description *D, R* et *DR* et les deux types de pondération du score (*NRF* et *RF*).

La section suivante présente plus en détail les conclusions concernant l'influence des facteurs saillants sur les performances du système.

6.2. Influence des paramètres

La conception modulaire du système, rendant possible l'analyse systématique d'un grand nombre de variantes, nous a permis d'étudier l'influence des facteurs conditionnant le fonctionnement et les performances du système.

6.2.1. Taille de la fenêtre de contexte

Les résultats de nos expériences ont montré qu'en général les performances diminuent avec l'augmentation du contexte, les meilleures performances étant enregistrées par des fenêtres de 4 à 6 mots pleins autour du mot cible. Pourtant, certaines des méthodes analysées présentent une variation moins significative des performances avec la variation de ce facteur (méthodes basées sur les poids appris – *PA* ou sur les chaînes lexicales - *CL*) ou une tendance de croissance après l'application d'un facteur correcteur (les variantes *RF* de la méthode de Lesk originelle normalisée par la fréquence relative des sens). Ce

comportement indique que l'information du contexte global peut être quand même utile à la désambiguïsation à condition de filtrer de manière adéquate les données (dans notre cas *PA, CL, RF*).

6.2.2. Granularité du découpage des sens

Bien que notre inventaire *coarse-grained* (basée sur le regroupement automatique des sens à partir des relations de *WordNet*) soit différent de celui utilisé à *Senseval2* (créé manuellement par les annotateurs), les résultats de nos expériences ont montré aussi une augmentation des performances pour ce type d'évaluation. De plus, on a pu constater une croissance du gain absolu rapporté aux performances de base (évaluées par les deux méthodes) dans le cas *coarse-grained* comparativement à *fine-grained.* Ceci est cohérent avec le fait, souvent mentionné, que les distinctions de *WordNet* sont trop fines pour la tâche de la désambiguïsation.

6.2.3. Descriptions des sens. Nombre de décisions par défaut

Notre étude sur le nombre de décisions par défaut indique que le manque d'information nécessaire à la désambiguïsation (cas sans superpositions) est plus fréquent pour les relations que pour les définitions et les exemples. En général, la compensation de l'absence d'information par l'augmentation du contexte ne se traduit pas toujours par une croissance en performance (voir la section 6.2.1.).

D'un autre côté, une analyse de la structure des réponses correctes et incorrectes par rapport aux choix du sens le plus fréquent semble indiquer une tendance des descriptions à favoriser le sens le plus fréquent. En d'autres termes, *WordNet* offre plus d'information utile à la discrimination des sens pour les sens plus fréquents que pour les autres.

La normalisation par la taille de la description et par la fréquence relative des sens candidats ou la sélection plus restrictive des éléments participants à la désambiguïsation (poids appris, contexte réduit à la chaîne lexicale du mot cible) sont des solutions possibles pour palier cette lacune, mais les gains par rapport aux performances de base restent cependant assez faibles.

6.2.4. Catégorie grammaticale

L'analyse des performances de notre système indique les meilleures pour les adverbes et pour les noms et des valeurs plus modestes pour les adjectifs et les verbes. Le bon comportement du système dans le cas des adverbes pourrait s'expliquer par le degré de polysémie et d'entropie assez bas de cette catégorie dans *WordNet.* Par contre, les noms semblent favorisés par l'information plus riche trouvée dans *WordNet* (au niveau des définitions et surtout des relations d'hyperonymie) par rapport aux adjectifs et aux verbes.

La catégorie grammaticale agit comme un filtre qui réduit le nombre de sens candidats possibles d'un mot cible. Par conséquent, les performances du système augmentent lorsque la catégorie grammaticale du mot cible est connue ou peut être déduite à partir de la forme instanciée de celui-ci dans le corpus de test. Les gains par rapport aux performances de base sont assez importants si cette information est disponible; ils sont cependant assez faibles si le système de référence en tient compte aussi.

6.2.5. Interdépendance des sens

Comme, dans un texte réel, les sens des mots polysémiques co-occurrant dans le même contexte sont reliés l'un à l'autre, nous avons essayé de modéliser ce type d'interdépendance par l'utilisation d'un tableau de vote. A la différence du traitement séquentiel, unidirectionnel, où chaque sens candidat reçoit un score correspondant au contexte, cette approche prend également en compte le vote du sens candidat accordé à un sens du contexte, avec lequel il partage plus d'éléments en commun. Bien qu'assez intuitive en théorie, en pratique, l'approche s'est prouvée moins performante que la version séquentielle, appliquée aux variantes implémentées. La méthode qui semble plus appropriée à ce type de traitement est la méthode basée sur les chaînes lexicales *(CL)* où l'échange mutuel de votes entre un sens candidat et le contexte est contrôlé par des contraintes d'ordre sémantique, c.a.d. l'appartenance à la même chaîne lexicale.

6.3. Evaluation comparative

Nous avons comparé les résultats de nos expériences avec les résultats officiels de *Senseval2,* tâche *English all words*. Les meilleures performances de notre système sont comparables avec les 4 premiers systèmes classés à cette compétition. Cependant cette interprétation est faite sur les valeurs des performances en tant que telles, sous la réserve que les participants ont été soumis à des contraintes de temps, pas imposées dans le cas de nos expériences, effectuées sur plusieurs variantes et validées sur plusieurs ensembles de test. Pour ce qui est des performances de base, les résultats de *Senseval2* et nos propres tentatives ont prouvé que cette "barrière" n'est pas facile à franchir.

6.4. Travaux futurs

Les expériences que nous avons effectuées ont montré que les variantes d'une méthode de désambiguïsation assez simple, comme l'algorithme de Lesk, pourrait produire des résultats comparables à d'autres techniques, plus compliquées ou nécessitant des ressources coûteuses ou difficiles à construire. Pourtant le type idéal de ressource capable de fournir une information suffisante à la discrimination correcte des sens reste encore questionnable et limite en pratique ce type d'approche. L'ajout des informations tenant de l'usage des mots dans des contextes réels (structures syntaxiques, collocations, information de nature pragmatique) pourrait constituer une réponse possible à cette question ouverte. D'un autre côté, la prise de décision dépendante des choix antérieurs ou basée sur la combinaison optimale des sens dans un contexte donné (HMM, *simulated annealing* etc.), l'exploitation de nombreuses *features* qui se recouvrent, par une méthode de type *maximum entropie,* ou la combinaison de plusieurs décideurs selon des critères probabilistes nous sembleraient des pistes de recherche intéressantes.

Pour conclure, la désambiguïsation sémantique est un problème très complexe, relié à la richesse de la langue et à sa capacité de se manifester par des significations différentes, en fonction de divers contextes. L'accomplissement automatique de cette tâche supposerait par conséquent l'utilisation combinée de plusieurs types de ressources (lexicales, sémantiques, syntaxiques, de type corpus etc.) ainsi que la combinaison de plusieurs méthodes capables de gérer de manière appropriée tout ce contenu informationnel.

Références bibliographiques

Agirre Eneko, Rigau German, *Word Sense Disambiguation using conceptual density,* COLING (International Conference in Computational Linguistics), 1996.

Amorós David Fernández, Gonzalo Julio, Verdejo Felisa, *The Role of Conceptual Relations in Word Sense Disambiguation,* Proceedings of the 6th International Workshop on Applications of natural Language for Information Systems, NLDB-01, 2001.

Atkins Sue, *Tools for computer-aided lexicography: the Hector project,* Papers in Computational Lexicography: COMPLEX '93, Budapest, 1993.

Banerjee Satanjeev, Pedersen Ted, *An Adapted Algorithm for Word Sense Disambiguation Using WordNet,* Proceedings of the Third International Conference on Intelligent Text Processing and Computational Linguistics, February 17-23, Mexico City, 2002.

Cowie Jim, Guthrie Louise, Guthrie Joe, *Lexical disambiguation using simulated annealing,* Proceedings of the 14th International Conference on Computational Linguistics (COLING-92), pp. 359-365, Nantes, France, 1992.

Crestan Eric, El-Bèze Marc, de Loupy Claude, Improving WSD with Multi-Level View of Context Monitored by Similarity Measure, Senseval-2, Toulouse, 2001.

Crestan Eric, El-Bèze Marc, de Loupy Claude, *Peut-on trouver la taille de contexte optimale en désambiguïsation sémantique?,* TALN 2003, Batz-sur-Mer, 11-14 juin 2003.

Edmonds Philip, *SENSEVAL : The Evaluation of Word Sense Disambiguation Systems,* ELRA Newsletter, Vol. 7, No. 3, 2002

Fellbaum Christiane, *A Semantic Network of English Verbs,* WordNet an Electronic Lexical Database, MIT Press, 1998, pp. 69-103.

Gale William A., Church Kenneth W., and Yarowsky David, *Work on Statistical Methods for Word Sense Disambiguation,* Proceedings, AAAI Fall Symposium on Probabilistic Approaches to Natural Language, Cambridge, MA, 1992, pp. 54-60.

Gandrabur Simona and Foster George, *Confidence estimation for translation prediction,* Seventh Conference on Natural Language Learning, CoNLL-2003, Edmonton, Canada, May 31 - June 1, 2003.

Haynes Sherwood, *Semantic Tagging Using WordNet Examples,* Proceedings of Senseval-2 : Second International Workshop on Evaluating Word Sense Disambiguation Systems, Toulouse, France, 2001, pp. 79-82.

Hirst Graeme, St-Onge David, *Lexical Chains as Representations of Context for the etection and Correction of Malapropisms,* WordNet an Electronic Lexical Database, MIT Press, 1998, pp. 305-331.

Hoste Véronique, Daelemans Walter, Hendrickx Iris, Van den Bosch Antal, *Evaluating the results of a memory-based word-expert approach to unrestricted word sense disambiguation.* - Word sense disambiguation: recent successes and future directions / Edmonds Phil [edit.], e.a., New Brunswick, ACL, 2002, p. 95-101.

Ide Nancy, Véronis Jean, *Word Sense Disambiguation: The State of Art,* Computational Linguistics, Vol.24, No.1, March 1998, pp.1-40.

Inkpen Diana Zaiu, Hirst Graeme, *Automatic Sense Disambiguation of the Near-Synonyms in a Dictionary Entry*, Proceedings, 4-th Conference on Intelligent Text Processing and Computational Linguistics (CICLING 2003), Mexico-City, February 2003, pp. 258-267.

Lesk Michael, *Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone,* ACM SIGDOC '86, The Fifth International Conference on Systems Documentation, Proceedings of ACM Press, 1986.

Litkowski Kenneth C., *Sense Information for Disambiguation : Confluence of Supervised and Unsupervised Methods,* Proceedings of the SIGLEX / SENSEVAL Workshop on Word Sense Disambiguation: Recent Successes and Future Directions, Philadelphia, 2002.

Kilgarriff Adam, *SENSEVAL: An Exercise in Evaluating Word Sense Disambiguation Programs* In Proc. LREC, Granada, May 1998, pp. 581-588.

Kilgarriff Adam and Rosenzweig Joseph, *English SENSEVAL: Report and Results.* In Proc. LREC, Athens, May-June 2000a.

Kilgarriff Adam et Rosenzweig Joseph. *Framework and Results for English SENSEVAL.* Computers and the Humanities, 34, 2000b, pp. 15-48.

Kilgarriff Adam, *English lexical sample task description,* Proceedings of Senseval-2 Workshop, Association of Computational Linguistics, 2002.

Manning Christopher D., Hinrich Schütze, *Word Sense Disambiguation,* Foundations of Statistical Natural Language Processing, MIT Press, 1999, pp. 229-264.

Miller George A., *Nouns in WordNet,* WordNet an Electronic Lexical Database, MIT Press, 1998, pp. 23-45.

Miller Katherine J., *Modifiers in WordNet,* WordNet an Electronic Lexical Database, MIT Press, 1998, pp. 47-67.

Martha Palmer, Christiane Fellbaum, Scott Cotton, Lauren Delfs, Hoa Trang Dang. *English Tasks: All-Words and Verb Lexical Sample,* Proceedings of Senseval-2 : Second International Workshop on Evaluating Word Sense Disambiguation Systems, Toulouse FRANCE, July 5-6, 2001.

Palmer Martha, Dang Hoa Trang, Fellbaum Christiane, *Making fine-grained and coarse-grained sense distinctions, both manually and automatically,* Journal of Natural language Engineering, revisions due in march 2003, LREC 2002 Workshop Publications.

Preiss Judita, Korhonen Anna, Briscoe Ted, *Subcategorization Acquisition as an Evaluation Method for WSD,* Proceedings of LREC, pp. 1551-1556, 2002.

Resnik Philip, *Using information content to evaluate semantic similarity in a taxonomy.* Proceedings of the Fourteenth International Joint Conference on Articial Intelligence (IJCAI-95), 1995, pp. 448-453.

Sidorov Grigori, Gelbukh Alexander, *Word Sense Disambiguation in a Spanish Explanatory Dictionary,* Proceedings TALN-2001, pp. 398-402, Tours, France, July 2-5, 2001.

Stevenson Mark, Wilks Yorick, *The Interaction of Knowledge Sources in Word Sense Disambiguation,* Computational Linguistics, Vol. 27, No. 3, September 2001, pp. 321–351.

Tengi Randee I., *Design and Implementation of the WordNet Lexical DataBase and Searching Software,* WordNet an Eectronic Lexical Database, MIT Press, 1998, pp. 105-127.

Véronis Jean, *Sense tagging: does it make sense?,* Proceedings of the Corpus Linguistics 2001 Conference, Vol. 13, Special Issue, 2001.

Vorhees Ellen M., Using WordNet for Text Retrieval, WordNet an Eectronic Lexical Database, MIT Press, 1998, pp. 285-304.

Wilks Yorick, Stevenson Mark, *Sense Tagging: Semantic Tagging with a Lexicon,* Proceedings of the SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What and How?, Washington, D.C., 1997.

Yarowsky David, *Word-sense disambiguation using statistical models of Roget's categories trained on large corpora*, Proceedings of the 14th International conference on Computational Linguistics, Nates, France, August 1992, pp. 454-460.

Yarowsky David, *Unsupervised Word Sense Disambiguation Rivalring Supervised Methods,* Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics, Appendix A A.1 Derivation of Description Length: Two-stage, 1995, pp. 189-196.

Annexe1

Mise en forme du corpus *Senseval2*

La mise en forme des données de test de *Senseval* 2 exige le traitement d'un fichier XML où chaque occurrence à désambiguïser est marquée par une balise <head id = "..." > contenant une étiquette de référence (*id*), qui indique le document, la phrase et l'instance (*token*) du mot cible. Le mot *ringing* présenté dans l'exemple 1 est identifié comme le 37 *token* de la phrase 25 du document 00. Le corpus de test *Senseval2* comporte aussi des balises satellites <sat id = " ... "> permettant le regroupement du mot-tête (*head)* avec le mot-satellite *(sat)*, afin d'obtenir les instances composées à désambiguïser (par exemple *church_of_england*) :

1. Exemples d'instances simples (lignes 1 à 4) et composées (lignes 5-6 et 7-9 du corpus de test *Senseval2*

```
1. <head id="d00.s25.t37">ringing</head>

2. <head id="d00.s00.t04">is</head>
3. <head id="d00.s00.t15">peculiarities</head>
4. <head id="d00.s03.t07">rural</head>

5. <head id="d00.s35.t12" sats="d00.s35.t12">worked</head>
6. <sat id="d00.s35.t12">up</sat>
7. <head id="d00.s37.t04" sats="d00.s37.t04 d00.s37.t04-1">Church</head>
8. <sat id="d00.s37.t04">of</sat>
9. <sat id="d00.s37.t04-1">England</sat>
```

Une des fonctions de la procédure de mise en forme est d'extraire du fichier de test les indicateurs de référence et les instances à désambiguïser, en regroupant sous le même

indicateur les têtes et les satellites des mots composés. Dans le cas des instances composées, le système essaye de produire la forme de base globale en combinant les formes de base individuelles et en les reliant par *underscore* (exemples: *work_up, church_of_england)*. Dans le cas des instances simples, plusieurs formes de base peuvent être proposées[1], comme par exemple, l'instance *ringing* qui pourrait provenir du verbe *ring* ou de la forme nominale *ringing.* Pour accomplir la tâche de lemmatisation des instances simples et composées, la procédure utilise l'information trouvée dans les fichiers d'exceptions et d'index de *WordNet*, ainsi que l'information de nature morphologique, que nous avons encodée dans le système.

D'autres fonctions (décrites dans 3.1.1) ont pour buts la détection de la catégorie grammaticale, l'extraction des sens candidats pour chaque mot cible, l'ordonnancement des sens candidats selon leur fréquence d'usage et la génération du fichier de test prétraité.

Un exemple simplifié d'une ligne de ce fichier, pour une instance du mot *art,* est présenté ci-dessous :

[art, art#1&29.0, art#2&8.2, art#3&3.0, art#4&0.19, d00 d00.s00.t01]

où les sens candidats sont symbolisés par *art#j, j=1,4,* suivi par '&' et la valeur de l'indicateur de fréquence est calculé selon la procédure décrite dans 3.1.1. L'instance est référenciée par la suite *"d00 d00.s00.t01"* (indicateur de référence) qui indique le document (*d00*), la phrase (*s00*) et la position du mot dans le cadre de la phrase (*t01*).

[1] Si la catégorie grammaticale du mot cible n'est pas a priori connue.

Annexe2

Mise en forme du corpus extrait de *Semcor*

Afin de mieux expliquer la mise en forme des données de test, un fragment d'une phrase encodée en format *Semcor* est présenté dans l'exemple 2. La procédure choisit comme instances à désambiguïser seulement les lignes comportant un tag de sens (*lexsn),* les autres occurrences étant ignorées.

A la différence du traitement pour le corpus *Senseval* (où ils étaient déjà établis par les annotateurs), les indicateurs de référence pour l'ensemble de test extrait de *Semcor* sont construits à partir de l'information des fichiers choisis comme fichiers de test. L'encodage que nous avons utilisé pour l'indicateur de référence renferme des renseignements sur le document (*d0* ou *d1)*[1], le paragraphe et la phrase (*p2s2,* voir l'exemple 2) et la position du mot dans le cadre de la phrase (par exemple, *w4* pour l'instance *said).* Pour déterminer la position d'un mot, le programme compte toutes les lignes antérieures qui commencent par "<*wf* ..." et qui appartiennent à la même phrase. Les paragraphes et les phrases sont délimités par les balises de début <*p snum*= ...> , <*s snum*= ...> et de fin </*p*>, </*s*>.

Exemple 2. Extraction des instances à désambiguïser du corpus *Semcor*

```
<p pnum=2>
<s snum=2>
<wf cmd=ignore pos=DT>The</wf>
<wf cmd=done pos=NN lemma=jury wnsn=1 lexsn=1:14:00::>jury</wf>
<wf cmd=done pos=RB lemma=far wnsn=2 lexsn=4:02:00::>further</wf>
<wf cmd=done pos=VB lemma=say wnsn=1 lexsn=2:32:00::>said</wf>
….
<wf cmd=done pos=VB lemma=take_place wnsn=1 lexsn=2:30:00::>took_place</wf>
…
</s>
</p>
```

L'instance de test *said,* de l'exemple considéré, sera référenciée par l'identificateur: *d1p2s2w4.*

---

[1] *d0* s'il s'agit du premier fichier *Semcor* composant le fichier de test, *d1* s'il s'agit du deuxième.

Une autre différence par rapport au prétraitement du corpus de test *Senseval2* consiste dans la manière de déterminer la forme de base à partir des formes instanciées. Comme cette information est déjà présente dans les fichiers *Semcor,* la procédure la reprend tout simplement du champ concerné (le tag *lemma* dans l'exemple 2). Ce procédé est aussi valable pour les cas des instances composées. Les formes de base des instances décrites dans l'exemple 2 sont par conséquent : *jury, far, say* et *take_place.* De plus, le contenu du champ *pos* (voir l'exemple 2) d'une instance annotée peut fournir la catégorie grammaticale du mot cible, utile dans les traitements qui considèrent ce type d'information (APOS).

Le prétraitement du corpus extrait de *Semcor* suppose également la génération du fichier clé. A chaque instance à désambiguïser correspond une ligne dans le fichier clé. Cette ligne regroupe l'identificateur de référence suivi d'un séparateur (*espace)* est de l'étiquette[2] du sens correct, formée par la concaténation du contenu des champs *lemma* et *lexsn.* La ligne du fichier clef, identifiant la réponse correcte pour l'instance *jury* de l'exemple 2 est la suivante :

d1p2s2w2 jury%1:14:00::

Pour ce qui est du fichier de test prétraité, son contenu, est similaire à celui construit pour le corpus *Senseval2,* seulement la forme de l'identificateur de référence diffère. Par exemple, une ligne du fichier de test prétraité pour le mot *committee* a la forme :

[committee, committee#1&10.39, committee#2&1.0, d0p1s1w1]

[2] Selon les règles d'encodage de *WordNet.*

Annexe3

Format du fichier de réponses

La sortie du module de désambiguïsation consiste dans le fichier de réponses du système, où chaque ligne a la forme :

identificateur_de_référence séparateur identificateur_de_sens

et

identificateur_de_référence identifie l'instance à désambiguïser du corpus de test ;

séparateur est le caractère *espace ;*

identificateur_de_sens représente l'étiquette du meilleur sens candidat choisi par le système pour l'instance à désambiguïser.

Les exemples suivants présentent des extraits des fichiers de réponses pour le corpus de test *Senseval2* et *Semcor.*

Exemple 3. Extraits du fichier de réponses pour le corpus de test *Senseval2*

```
d00 d00.s00.t01 art%1:04:00::
d00 d00.s00.t03 change_ringing%1:04:00::
d00 d00.s00.t04 be%2:42:03::
d00 d00.s00.t05 peculiar%5:00:00:unusual:00
d00 d00.s00.t08 english%3:01:00::
d00 d00.s00.t13 most%4:02:00::
d00 d00.s00.t14 english%3:01:00::
```

Exemple 4. Extraits du fichier de réponses pour le corpus de test *Semcor*

```
d0p1s1w2 texas%1:15:00::
d0p1s1w3 halfback%1:18:00::
d0p1s1w6 n't%4:02:00::
d0p1s1w7 even%4:02:00::
d0p1s1w8 know%2:31:01::
d0p1s1w10 team%1:14:00::
d0p1s1w12 play%2:33:00::
d0p1s1w13 person%1:03:00::
```