instance à traiter Senseval 2 instances définies exceptions

14 juillet 2012

Dédicaces

On dédie ce modeste travail à nos familles et nos amis pour leur soutien indéfectible.

Remerciement

Nous exprimons notre reconnaissance et nos amples remerciements à notre encadreur Mr Bentaallah Mohammed Amine qui nous a encouragés, guidés et soutenus d'une main de maître tout au long de la préparation de notre mémoire, nous remercions également l'honorable jury qui a consenti à juger notre travail.

Enfin, Nous remercions tous ceux qui de près ou de loin ont bien voulu nous encourager à accomplir ce travail.

Les modestes contributions apportées tout au long de ce travail en firent une expérience très intéressante et formidablement enrichissante.

Résumé

La désambiguïsation sémantique est une tâche intermédiaire, fondamentale pour la bonne réalisation des applications du traitement automatique du langage naturel nécessitant la compréhension de la langue. Plusieurs approches ont tenté de résoudre le problème de l'ambiguïté de manière automatique, parmi elles celle basée sur l'algorithme de LESK.

Le travail porte sur l'implémentation de l'algorithme de LESK ainsi que sa variante simplifiée, de plus, la proposition de nouvelles variantes qui se basent sur l'utilisation des distances sémantiques ainsi que sur les relations de synonymie. Le « WordNet 2.1 »a été utilisé comme lexique informatique organisé comme un réseau de concepts. Toutes nos expérimentations ont été effectuées sur les corpus fournis par par la campagne d'évaluation « Senseval ».

Les résultats obtenus par les variantes de Lesk proposées et implémentées durant ce travail étaient les meilleurs. Tandis que ceux obtenus par les approches calculant les mesures de similarité étaient sans intérêt.

Mots clés : Désambiguïsation sémantique, algorithme de Lesk, WordNet, Dictionnaire électronique, mesure de similarité.

Table des matières

Liste des tableaux

Table des figures

Introduction générale

Les langues naturelles sont caractérisées par l'omniprésence de l'ambiguïté sémantique qui constitue l'une des sources de richesse et de souplesse des langues. Dans la communication interhumaine, l'ambiguïté ne présente pas un réel problème puisque ces derniers ont accès à autant d'informations (connaissances extralinguistiques). Néanmoins, dans un cadre de traitement automatique de la langue, l'ambigüité lexicale constitue toujours un défi.

L'intérêt porté pour la tâche de désambiguïsation est en plein essor compte tenu du grand nombre de méthodes et de ressources utilisées, comme par exemple les dictionnaires, les thésaurus ou les lexiques sémantiques électroniques, les corpus annotés comportant des étiquettes de sens, les corpus non annotés ou une combinaison de ces ressources.

Parmi les méthodes proposées, l'approche de LESK basée sur l'utilisation de ressources externes telles qu'un dictionnaire électronique (WordNet) est la plus connue et la plus utilisée pour sa simplicité. Notre contribution dans ce mémoire consiste en la proposition de nouvelles approches se basant sur l'algorithme de LESK et une étude des critères pour la tâche de désambiguïsation pour cet algorithme.

Le manuscrit est présenté en trois chapitre, le premier chapitre aborde des notions sur la désambiguïsation sémantique accompagnés en grande partie d'un etat de la recherche pour les différentes approches existantes, Le deuxième chapitre détaille l'algorithme de LESK et sa variante simplifiée et enfin un chapitre qui présente les différentes implémentations avec des résultats et des discussions

Chapitre 1

Notions sur la désambiguïsation sémantique

Introduction

Retrouver le sens d'un mot polysémique dans un énoncé fait partie des principaux verrous du traitement automatique de la langue. La désambiguïsation sémantique étant un thème de recherche important, les publications dans ce domaine sont nombreuses.

Dans ce chapitre, nous allons en préambule définir la désambiguïsation sémantique et les domaines d'applications de la WSD (Word Sense Disambiguation en anglais), puis donner un aperçu général sur le domaine et les types d'approches qui existent à ce jour et enfin, présenter les critères d'évaluation des systèmes de désambiguïsation automatique, notamment, la campagne « SENSEVAL ».

1.1 Définition

La désambiguïsation sémantique (Word Sense Disambiguation, WSD) consiste à identifier le sens d'un mot dans un texte, elle constitue *« une tâche intermédiaire »* Wilks et Stevenson [34] essentielle dans le cadre de nombreux processus de traitement automatique de la langue, et principalement dans les applications visant la compréhension de texte en langage naturel. [8]

En effet, les applications qui ont besoin de gérer le sens d'un texte dans un corpus de documents à travers une analyse linguistique des énoncés tel que la traduction automatique ne peuvent se passer d'une procédure d'identification du sens des mots des textes de ce corpus.

1.2 Types d'ambiguïté

L'ambiguïté inhérente aux langues naturelles est un problème récurrent dans le domaine du Traitement Automatique du Langage. En fonction du niveau d'analyse linguistique où l'on se situe, on distingue différents types d'ambiguïté, parmi elles :

1.2.1 Ambiguïté syntaxique

Lors d'une analyse syntaxique d'un énoncé donné, on peut trouver différents possibilités de points de rattachements. Un type d'ambiguïté particulier serait l'ambiguïté syntaxique. Par exemple, dans : *« Rita a acheté des nappes à pois rouges »*, il nous est impossible de savoir si ce sont les nappes qui sont rouges ou plutôt les pois qui le sont, étant donné que le syntagme adjectival *« rouges »* pourrait s'attacher au syntagme nominal *« des napres à pois »* autant qui'l pourrait faire partie du syntagme prépositionnel *« à pois rouges »*.

1.2.2 Ambiguïté sémantique

Une forme est sémantiquement ambiguë si on peut lui faire correspondre au moins deux sens distincts. Il existe deux types d'ambiguïtés sémantiques (lexicales) :

1.2.2.1 Ambiguïté lexicale polysémique

Les polysémies correspondent à des mots morphologiquement identiques appartenant à une même classe grammaticale mais qui ont des sens différents avec des relations entre ces sens. Par exemple, dans la phrase *« Pierre sent la rose »*, le mot *« sent »* admet deux

sens, le premier peut se comprendre : *« Pierre hume la rose »*, le second comme : *« Pierre a le parfum d'une rose »*.

1.2.2.2 Ambiguïté lexicale homonymique

L'homonymie se définit comme la polysémie avec l'absence de relations entre les sens possibles du mot homonymique. Par exemple, dans la phrase *« Cet ours a mangé un avocat »*, il ya ambiguïté lexicale homonymique. Le mot *« avocat »* correspond a deux mots distincts, à savoir un *« fruit »* ou un *« plaideur »*.

1.3 Applications de la désambiguïsation sémantique

La désambiguïsation sémantique des mots est une tâche fondamentale pour la plupart des applications de traitement automatique du langage, parmi ces applications on trouves :

1.3.1 Traduction automatique (MT)

La traduction automatique est le domaine où la tâche de désambiguïsation sémantique est fondamentale pour aboutir à des traductions correctes des mots ambigus quelque soit leur contexte d'apparition. Dans l'exemple que Audibert [1] a cité : « *la traduction en anglais du mot français « mèche » est « lock, wick, fuse » ou bien « drill » suivant qu'il s'agit d'une « mèche de cheveux, de bougie, de pétard » ou de « perceuse »*. »

1.3.2 Recherche d'information (IR)

Déterminer le sens exacte des mots ambigus d'une requête peut permettre de filtrer la recherche de manière que les documents retournés soient des documents sémantiquement pertinents.

1.4 Etat de l'art

Le domaine de désambiguïsation a connu un foisonnement de travaux cherchant à

résoudre le problème de l'ambiguïté. Dresser un panorama exhaustif d'un état de l'art reste une tâche difficile, on présentera dans cette section un aperçu général sur le domaine de la désambiguïsation sémantique automatique en se référant de l'état de l'art présenté dans [8] qui introduit le domaine en détail et qui a été mis à jour par L.Audibert [1] en 2003 et par Kolhatkar [12] en 2009.

1.4.1 Premiers pas

Les premières recherches pour la tâche de désambiguïsation sémantique ont commençé en 1949 dans le domaine de la traduction autaumatique. Dans son Mémorandum, Weaver [32] introduit le besoin de désambiguïsation lexicale dans la traduction par ordinateur. Il expose le problème de la manière suivante : *« en présentant un mot dénué de tout contexte, il est impossible au lecteur de déterminer quel est son sens, en revanche, en fournissant au lecteur un certain nombre de mots qui se trouvent dans le voisinage (aussi bien à gauche qu'à droite) de ce terme central, il est alors possible pour le lecteur de décider de son sens. La question est de déterminer quel est le nombre minimum de mots voisins requis dont le lecteur a besoin pour obtenir le sens correct de ce mot. »*

Dans [1], Audibert ressort de ces différents états de l'art une distinction implicite ou explicite entre deux grands types d'approches en fonction des ressources utilisées :

1.4.1.1 Approches exogènes

Les approches exogènes utilisent des ressources externes au texte à traiter pour la tâche de désambiguïsation. Certains utilisent des dictionnaires de langue classiques qui ont été mis sous forme électronique, comme par exemple *« le Robert »*, ou l'*« OED (Oxford English Dictionary) »* pour l'anglais. D'autres utilisent des thésaurus tels que le *« Roget's Thesaurus »*, qui sont sollicités pour leur structure hiérarchisée comme par exemple la description des liens **hyperonymiques**[1] entre mots. D'autres encore utilisent des dictionnaires particulièrement adaptés au traitement automatique, comme par exemple, *« WordNet »* qui combine définitions, relations hyperonymiques, relations de

1. Se dit des termes génériques dont le sens comprend celui d'autres termes plus spécifiques. Animal est l'hyperonyme de mammifère, ce dernier terme étant lui-même hyperonyme de chien.

synonymie, d'antonymie, etc.

1.4.1.2 Approches endogènes

Les approches endogènes regroupent les travaux qui ne font appel à aucune ressource externe. Quelques travaux reposent sur des calculs statistiques des occurences du mot ambigu et de ces mots cooccurrents dans différents contextes, d'autres utilisent différents types de relations telles que les relations de cooccurrences ou relations syntaxiques.

On appelle mixtes (hybrides), les approches combinant ressources endogènes et exogènes.

1.4.2 Les approches d'inteligence artificielle

Les approches d'intelligence artificielle reposent sur une modéliation des connaissances de nature sémantique et syntaxique. Dans [8] Ide et Véronis particularisent deux types de méthodes d'intelligence artificielle caractérisant cette période : les méthodes symboliques et les méthodes connexionnistes.

1.4.2.1 Approches connexionnistes

Les approches connexionnistes sont les approches qui ressemble le plus du fonctionnement effectif du cerveau humain. Elles modélisent des ressources telles que les mémoires sémantiques avec la nation de la diffusion de l'activation des concepts, où chaque concept concerné sera activé ainsi que tous ses voisins. L'activation diminue au fur et à mesure de sa diffusion, certains noeuds se renforcent progressivement en recevant plusieurs signaux d'activation.

1.4.2.2 Approches symboliques

Les approches symboliques reposent sur la représentation symbolique des sens, elles modélisent des ressources telles que les réseau sémantiques ou des structures conditioennelles. Certains chercheurs ont construit un système qui permet de téterminer le sens correct en calculant le nombre minimum de noeuds ou d'arcs entre les concepts.

La limitation des approches d'inteligence artificielle est qu'elles ne couvrent que des parties restreintes du langage, donc elles ne sont applicables que sur des petits textes ne couvrant qu'une partie de la langue

1.4.3 Les approches basées sur les connaissances

L'emergence d'un grand nombre de ressources telles que les dictionnaires électroniques, les thésaurus et les léxiques informatiques qui a marqué les années 80 a donné une autre orientation pour la résolution du problème de l'ambiguïté. Les approches basées sur les connaissances tentes d'extraire de manière automatique les informations dont on a besoin pour la tâche de désambiguïsation.

1.4.3.1 Dictionnaires informatisés

La méthode de Lesk [15] est la plus utilisée dans les travaux basés sur les dictionnaires. Lesk a créé une base de connaissances contenant les mots de la définition de chaque sens du mot ambigu, la désambiguïsation est réalisée en choisissant le sens ayant la définition qui compte le plus de mots communs avec les mots qui entourent le mot ambigu.

Plusieurs auteurs ont tenté d'étendre la méthode de Lesk. Dans [35], Wilks et al. cherchent à enrichir la base de connaissances associées à chaque sens en calculant la fréquence d'apparition des mots cooccurrents dans les définitions. Ils en dérivent plusieurs mesures du degré de corrélation entre les mots.

Les dictionnaires ne souffrent pas seulement de l'incovénient d'être inconsistant. Si les dictionnaires contiennent des informations détaillées au niveau lexical, ils manquent cruellement d'**informations pragmatiques** [2], alors que celles-ci constituent une bonne source pour la désambiguïsation. Dans l'exemple [1], le mot *« cendre »* et *« tabac »*, *« cendrillon »* ou *« cigarette »* dans un réseau tel que celui de Quillian, sont très indirectes alors que ces trois mots sont fréquemment en cooccurrence avec le mot *« cendre »* dans les

2. Les informations pragmatiques regroupent, en lexicographie, toutes les informations dont a besoin le locuteur pour savoir utiliser correctement une unité lexicale (substantif, adjectif,verbe, etc.) ou un groupe d'unités lexicales dans un contexte donné en tenant compte des variables de la communication.

corpus.

1.4.3.2 Thésaurus

Les thésaurus comportent des informations sur les relations entre les mots et en particulier sur les relations de synonymie. Le thésaurus international (Roget thésaurus) informmatisé en 1950 a été utilisé à plusieurs reprises dans différentes applications du traitement automatique du langage telles que la traduction automatique (Masterman [5]), la recherche d'information (Sparck Jones [28], [29]) et l'analyse du contenu (Sedelow et Sedelow [24], [25], [26]).

1.4.3.3 Lexiques informatiques

Les lexiques informatiques sont des bases de connaissances informatiques construites manuellement à partir des milieux des années 80. Le plus connue entre ces lexiques est WordNet (Miller, Beckwith, Fellbaum, [17], [6]), CYC (Lenat et Guha [14], Acquilex (Briscoe [4]).

WordNet combine entre les informations que contiennent les dictionnaires informatiques (définitions des sens) et les informations fournies par les thésaurus avec des relations entre les sens telles que la synonymie représentés dans une hiérarchie de concepts (noms et verbes), mais également des liens entre les mots selon divers types de relations sémantiques telles que l'hyperonymie, l'hyponymie, l'antonymie et **la méronymie** [3], etc. un autre avantage que fournit WordNet, est qu'il s'agit d'une ressource lexicale de grande couverture librement et gratuitement utilisable.

Dans [32], Voorhees est l'un des premiers chercheurs avoir utiliser WordNet dans le domaine de la recherche d'informations, il conçoit une structure appelée *« hood »* dans le but de représenter les catégories sémantiques, dans le même esprit que les catégories du thésaurus ROGET, en utilisant les relations d'hyponymie entre les mots de WordNet. Un hood, pour un mot donné, est définit comme étant le plus grand sous-graphe connecté

3. La méronymie est une relation partitive hiérarchisée : une relation de partie à tout. Un méronyme A d'un mot B est un mot dont le signifié désigne une sous-partie du signifié de B. La relation inverse est l'holonymie.

contenant ce mot. Les résultats que Voorhees obtient indiquent que sa technique n'est pas utilisable pour distinguer les sens fins de WordNet. [1]

Le système *« SensLearner »* [16], présenté par Mihalcea et Faruque en 2004, implémente une nouvelle méthode supervisée pour désambiguïser tous les mots d'un texte en utilisant WordNet. Le but de ce système est d'utiliser le plus possible un petit ensemble de données et au même temps réussir à généraliser une fonction d'apprentissage qui peut manipuler tous les mots contenus dans le texte.

En 2007, Navigli et Lapata [18], ont proposé un algorithme non supervisé basé sur un graphe pour la désambiguïsation automatique qui utilise des connaissances depuis un lexique informatique de référence. Les mesures de connectivité du graphe sont bien étudiées et ont été pris en considération pour l'étude des structures des environnements hyperlink et dans l'analyse des réseaux sociaux. Ils utilisent WordNet et ses versions enrichies, ils réclament que la méthode est indépendante du lexique utilisé.

Le critère du sens le plus fréquent (MFS, Most Frequent Sense) est très difficile à battre par les systèmes de désambiguïsation automatique. Preiss et al. (2009) [9] se sont intéressés à raffiner le sens le plus fréquent en améliorant chaque stage de désambiguïsation comme la lemmatisation et l'étiquetage des parties du discours (POS, Part-Of-Speech). Le système supervisé proposé utilise un algorithme de classement et une mesure de similarité. Le système choisit une réponse alternative quand un grand niveau de confiance est observé. Le système de raffinement du sens le plus fréquent bénéficie d'un très faible rappel mais d'une haute précision.

Guo et Diab, 2009 [6] suggèrent une modification de l'algorithme supervisé basé sur un graphe proposé dans [18], en incluant l'utilisation de la mesure de similarité JCN (Jiang And Conrath) pour trouver la similarité entre pairs de verbes. Ils rapportent le meilleur résultat de l'état de l'art dans SensEval2 en utilisant WordNet 1.7.1.

Dans [13], Kolte et Bhirud ont présenté une méthodologie pour la tâche de désambiguïsation sémantique basée sur des informations sur le domaine extraites de WordNet et de sa hiérarchie avec l'idée que les mots d'une phrase contribuent à déterminer le domaine de la phrase. La disponibilité de *« WordNet domains »* rend la tâche possible. Le domaine du mot à désambiguïsé peut être déterminé en se basant sur le domaine des mots appa-

raissant dans son contexte. Le niveau de précision a atteint une efficacité de 63.92% pour les noms.

1.4.4 Les approches basées sur corpus

Les corpus sont des grands ensembles d'exemples d'usages de chaque occurence des mots dans différents contextes. Le principe de base de ces approchess consiste à l'acquisiation de connaissances à partir de ces corpus en étudiant les différents exemples et on utilisant des méthodes statistiques. Deux types de corpus sont utilisés, les corpus étiquetés et les corpus non étiquetés.

Dans le cas des corpus étiquetés, la meilleure approche qui peut être utilisée est probablement d'utiliser des techniques d'apprentissage supervisés. Néanmoins, ces approches souffrent du manque des corpus léxicalement étiquetés ainsi, que la difficulté d'étiquetage de corpus (en terme de temps et d'argent). Ces corpus étiquetés ont étaient utilisés dans plusieurs travaux tels que le travaux cités dans [3], [19].

Les approches basées sur les corpus non étiquetés (schütze[22], [23])ont utilisés des méthodes statistiques afin de détecter le sens directement à partir du corpus traité. celà permet de résoudre le problème de la non disponibilité des corpus étiquetés.

1.4.5 Approches hybrides

Un autre type d'approches utilisé dans le domaine de la désambiguïsation automatique, signalé en 2001 par Stevenson et Wilks [30], est la conception de systèmes hybrides qui combinent plusieurs sources d'informations.

L'approche de Yorowsky cité dans [36] constitue une approche mixte qui utilise le thésaurus ROGET et l'information extraite à partir de corpus pour générer des classes de mots dérivées des catégories de concepts du thésaurus ROGET.

Dans [19], Ng et Lee exploitent également plusieurs sources d'informations comme l'étiquetage morphosyntaxique, la forme morphologique, les cooccurrences, les **collocations**[4], et les relations syntaxiques verbe-objet. Toutes ces informations sont directement

4. En linguistique, une collocation est une cooccurrence privilégiée, une association habituelle d'un mot à un autre au sein d'une phrase

extraites du corpus.

D'autres chercheurs tels que Herley et Glennon [7], qui ont utilisé l'étiqueteur sémantique commercialisé par la « Cambridge Language Survey (CLS) »qui comporte quatre sous-étiqueteurs fonctionnant en parllèle basés sur différents types d'informations telles que les collocations (extraites à partir du dictionnaire CIDE) les catégories du sujets, etc. Les résultats obtenus ont atteint les 78% de précision.

1.5 Evaluation des systèmes de désambiguïsation automatique

Le domaine de la désambiguïsation sémantique a connu un grand nombre de méthodes dont leur évaluation est difficile. L'apparition de corpus manuellement désambiguïsés comme le *« gold standard »* permettent d'évaluer les méthodes en comparant les résultats obtenus pas les approches(sens attribués pour les mots désambiguïsés) avec les sens attribué manuellement dans le corpus traité. La réalisation de ces corpus est trés couteuse en terme de temps et d'argent puisqu'il faut avoir recours à des experts de la linguistique pour la validation des sens attribués aux instances des mots polysémiques.

1.5.1 Mesures de performance

Afin de pouvoir évaluer les performances d'un système de désambiguïsation plusieurs mesures peuvent être utilisées. Le précision et le rappel sont généralement les plus utilisés. La précision représente le pourcentage d'efficacité du système par rapport au mots désambiguïsés, tandis que le rappel représente le pourcentage des réponses correctes par rapport aux mots à traiter dans le texte.

Le calcul de la précision et du rappel se fait selon les deux formules suivantes :

$$\text{précision} = 100 \cdot \frac{\text{nb. de réponses correctes}}{\text{nb. de cas traités}} \tag{1.1}$$

$$\text{rappel} = 100 \cdot \frac{\text{nb. de réponses correctes}}{\text{nb. de cas à traiter}} \tag{1.2}$$

1.5.2 Le cadre d'évaluation Senseval

La compagne d'évaluation Senseval ets la plus utilisée pour la comparaison entre les différents systèmes de désambiguïsation on leur fournissant les mêmes données d'entrainement et de test, la catégorie grammaticale des mots à désambiguïser ainsi que les sens adéquats déterminés par des experts de la linguistique. Ce cadre d'évaluation est détaillé dans le chapitre 2 avec ses trois versions.

Conclusion

Dans ce chapitre, nous avons définit la désambiguïsation automatique en présentant les différents types d'approches proposées pour la levé de l'ambiguïté. Nous avons aussi cité les différents métriques utilisées pour l'évaluation des systèmes de désambiguïsation Dans 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 de leurs résultats. Elles requièrent la disponibilité d'un grand nombre de textes annotés à la main, où chaque occurrence est étiquetée par le sens approprié au contexte. Un tel corpus est difficile et coûteux à réaliser en termes de temps et d'argent. Ces méthodes requièrent aussi une certaine similarité entre les sujets dans le corpus d'entrainement et celui de test.

Parmi les approches proposées ceux basées sur les connaissances qui semblent une alternative viable qui permettent de résoudre le problème du manque de corpus léxicalement étiquetés.

Une des méthodes basée sur les connaissences est celle de LESK (chapitre 2) qui compte le nombre de superpositions entre les définitions des sens candidats et les définitions des mots du contexte du mot à désambiguïser.

Chapitre 2

Algorithme de LESK

Introduction

L'intérêt porté pour la tâche de désambiguïsation sémantique à fait naissances de plusieurs types d'approches. La plus célèbre est celle de Lesk. Dans ce chapitre on s'attache principalement à la méthode de Lesk telle qu'elle est présentée dans [15], sa variante simplifiée ainsi qu'à des travaux connexes.

2.1 Algorithme de LESK

La méthode de désambiguïsation proposée par Lesk est l'une des approches basées sur les connaissances, ces approches cherchent à extraire de manière automatique les informations nécessaire pour la tâche de désambiguïsation. Son principe consite à compter le nombre de mots communs entre la défintion des sens du mot polysémique et les définition des sens des mots se trouvant dans son contexte

La désambiguïsation basée sur la superposition (en anglais, *overlaps*) consiste simplement à compter le nombre de mots communs entre les définitions des sens d'un mot ambigu (les définitions sont extraites d'un dictionnaire tel que WordNet) et les définitions des mots apparaissant dans le contexte du mot à désambiguïser. On choisit alors le sens de la définition qui compte le plus grand nombre de mots communs.

2.2 Critères de l'algorithme de LESK

Dans cette section les critères de désambiguïsation basée sur l'algorithme de LESK sont présentés en deux parties, la première concerne les informations extraites du texte traité et la deuxième, les informations extraites du dictionnaire utilisé.

2.2.1 Informations extraites du texte à traiter

2.2.1.1 Informations syntaxiques

L'approche de lesk est une approche simple, elle a comme avantages :

– elle ne s'appuie pas sur des connaissances de nature syntaxique.

– 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.

Par contre , la méthode peut etre mauvaise dans quelque 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. »[31]

2.2.1.2 Calcul des scores et longueur du contexte

Lesk étudie plusieurs façons pour calculer les recouvrements des entrées d'un dictionnaire et remarque qu'il n'y a pas de différences 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 obligatoirement des résultats qui diffèrent.

Par contre il y'a des questions qui sont laissées sans réponse, par exemple : l'utilisation

de la phrase comme fenêtre de contexte, et la prise en compte des sens déjà assignés aus mots traités, etc.

2.2.2 Informations extraites du dictionnaire

2.2.2.1 Qualité du dictionnaire

le dictionnaire utilisé est une ressource trés importante. Dans [15], Lesk teste à cet effet l'efficacité 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 2.1.

	OALDCE	**W7**	**CED**	**OED**
Taille(MB)	6.6	15.6	21.3	350
Nombre d'entrées	21 000	69 000	85 000	304 000
Nombre de sens	36 000	140 000	159 000	587 000
Octets / entrée	290	226	251	1 200

TABLE 2.1 – Caractérsitiques des dictionnaires utilisés par LESK

Il est tres clair que l'utilisation d'un dictionnaire qui comporte une quantité supérieure d'informations par entrée est capable de faire plus de distinctions valides entre les sens d'un mot polysémique qu'un dictionnaire disposant d'un matériau moins riche.

2.3 Choix de WordNet

Notre choix s'est porté pour le WordNet bien que d'une part, plusieurs études dans le domaine de la désambiguïsation automatique ont utilisé avec succès d'autres types de ressources tels que Roget's Thesaurus (Yarowsky 1992) [36], Longman Dictionary of Contemporary English [30], New Oxford Dictionary of English (Litkowsky 2002 [5]). Et d'autre part le WordNet a pour inconvénient la granularité trop fine des sens dans la désambiguïsation automatique (Voorhees [32]), (Véronis [33]), (Palmer et al. [20]), (Preiss et al. [21]). Les raisons de ce choix sont d'un coté, nous avons travaillé avec WordNet pour

des raisons qui tiennent, d'une part, sa complète disponibilité sur Internet (base de données, documentation, fichiers sources etc.) et, d'une autre , et c'est de loin la raison qui prime, de sa compatibilité avec l'environnement Senseval que nous avons choisi comme cadre d'évaluation de nos implémentations.

2.4 Travaux connexes

Nous présentons d'autres approches basées sur l'algorithme de Lesk et/ou sur l'exploitation de WordNet. Il s'agit principalement de systèmes déjà testés lors de Senseval2, 3.

2.4.1 Senseval 1

La première version de Senseval a eu lieu en 1998 d'évaluation (Kilgarriff [10]) avec la participation de 17 systèmes pour trois langues différentes. Leurs tâche consitait à déteminer le sens correct d'un ensemble d'occurences du mot ambigu dans différents contextes (un ou deux phrases). Le nombre total d'instances à désambiguïser a été de 8448, une caractérisation par catégorie grammaticale de ces données étant présentée dans le tableau 2.2 selon (Kilgarriff et Rosenzweig [11]).

Noms		**Verbes**		**Adjectifs**		**Indeterminés**		**Total**	
Instances	*Mots*	*Instances*	*Mots*	*Instances*	*Mots*	*Instances*	*Mots*	*Instances*	*Mots*
2 756	15	2 501	13	1 406	8	1 785	5	8 448	41

TABLE 2.2 – Nombres d'instances et de mots à désambiguïser pour Senseval 1 [31]

Le texte suivant (figure 2.1) 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.
```

FIGURE 2.1 – Extrait d'un corpus contenant deux instances du mot *rabbit* [31]

La variante simplifiée de l'algorithme de LESK

« Lesk simple » s'agit d'une variante de l'algorithme original de Lesk qui détermine pour un mot ambigu le sens dont la définition comporte le plus grand nombre de superpositions avec les mots du contexte (non leurs définitions).

2.4.2 Senseval 2

L'exercice d'évaluation Senseval2 qui est organisé en 2001, pour 12 langues, 94 systèmes participants et 3 types de tâches (approche lexicale lexical sample, tous les mots all words et traduction). Dans ce mémoire nous nous intéressons seulement aux approches monolingues (anglais).

2.4.2.1 anglais - approche lexicale

Les données pour Senseval2 ont été fournies par *« HECTOR »*, une base de données à double profil : dictionnaire et corpus. A chaque mot d'HECTOR correspond une entrée de dictionnaire et des étiquettes de sens pour toutes ses occurrences dans un corpus de 17 MB .Les mots ont été choisies selon leur nombre d'occurrences dans le corpus.

Le corpus utilisé pour Senseval2 a été extrait de BNC-2. Toutes les instances à désambiguïser appartenaient à une des 3 catégories grammaticales : nom, verbe ou adjectif. Le tableau 2.3 montre la structure du corpus de test, par catégorie grammaticale, nombre d'instances et nombre de mots à désambiguïser.

Noms		**Verbes**		**Adjectifs**		**total**	
Instances	*Mots*	*Instances*	*Mots*	*Instances*	*Mots*	*Instances*	*Mots*
1 754	29	1 806	29	768	15	4 328	73 (mots distincts)

TABLE 2.3 – Nombres d'instances et de mots à désambiguïser pour Senseval2. Lexical sample [2]

2.4.2.2 anglais - tous les mots

La section tous les mots de Senseval2 consistait, à désambiguïser tous les mots appartenant aux catégories grammaticales (nom, verbe, adjectif et adverbe). Comme performances de base ont été considérés les résultats d'un système qui choisit le sens le plus fréquent, sans tenir compte du contexte. Dans le cas de l'anglais, un tel système obtenait 57% de précision et rappel. Parmi les 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).

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 et de Senseval3, discutent la contribution de l'information de type Lesk (définitions + exemples d'usage) à la désambiguïsation sémantique. Notre tâche s'inscrit dans la section tous les mots

2.4.3 Senseval 3

L'epreuve Senseval 3 a repri les mêmes taches que celles dans Senseval 2 mais dans un cadre multilingue et avec d'autres tâches pour différentes langues (espagnol, chinois, roumain, catalan, etc.).

D'autres variantes de l'algorithme de Lesk ont été proposées telles que la comparaison floue de (Sidorov et Gelbukh [27]) qui a apporté un taux d'erreurs de 13% par rapport à

17% de Lesk originel et 29% du choix du sens le plus fréquent, la combinaison des sens proposée par (Banerjee et Pedersen [2]) avec 31.7% de précision par rapport à 22.6% de Lesk simple et 16% de Lesk utilisant les définitions dans la tâche approche lexical (*Sample Task*).

Conclusion

Dans ce chapitre, nous avons présenté l'algorithme de LESK et sa variante simplifiée avec leurs critéres de désambiguïsation, on a ainsi détaillé la campagne d'évaluation Senseval avec ses trois versions qui sera utilisée pour l'évaluation de nos implémentations.

Chapitre 3

Expérimentations et résultats

Introduction

Plusieurs variantes de 'algorithme de Lesk ont été proposées, quelques unes sont basées sur la façon de calculer le score (pondéré / non pondéré), d'autres sont basées sur le type de description des sens utilisé (définition, relations, définition et relations, etc.).

Ce chapitre présente les différentes variantes de l'algorithme de Lesk (Lesk original, Lesk simplifié, etc.) implémentés durant ce travail, les résultats obtenus pour chaque variante et pour d'autres approches calculant les mesures de similarité entre les sens des mots. Ces mesures se basent soit sur l'approche de LESK soit sur les distances sémantiques (WU AND PALMER).

Le chapitre se présente principalement en deux sections, une pour la description des corpus de test fournis dans Senseval 2 et 3 utilisés et la deuxième pour la description des résultats et performances obtenus par nos systèmes ainsi une études comparatives sous formes d'histogrammes entre les différentes méthodes implémentées.

3.1 Corpus de test

3.1.1 Senseval 2

Les données de test de Senseval2 ont été téléchargées du site officiel de Senseval. Le

corpus de test consiste en trois articles traitant des sujets différents et un total de 2260 mots à désambiguïser. Ces articles proviennent du Penn Treebank Text (un corpus de phrases arborées). L'inventaire des sens a été construit à l'aide des synsets de WordNet 1.7.

Les données sont structurées en format XML avec un système de balises qui mettent en évidence les instances des mots à désambiguïser dans leurs formes originales (noms, verbes, adjectifs, adverbes) en leurs assignant des étiquettes d'identification par document (d), phrase (s) et par mot de la phrase (t). Chaque instance à désambiguïser possède un attribut qui fournit la catégorie grammaticale (pos), sa forme canonique (lemma) et le numéro du sens (wnsn) de l'instance dans le dictionnaire WordNet, si cet attribut possède une valeur 0 cela veut dire que ce mot n'existe pas dans WordNet, et donc ne seront pas traités. Ces attributs sont très importants pour la mise en forme des données et pour la tâche de désambiguïsation. La figure 3.1 présente une phrase (*History, after all, is not on his side.*) du corpus :

```
<s snum=13>
<wf cmd=done id=d00.s14.t00 pos=NN lemma=history wnsn=1 lexsn=1:28:00::>History</wf>
<punc>,</punc>
<wf cmd=ignore pos=IN>after</wf>
<wf cmd=ignore pos=DT>all</wf>
<punc>,</punc>
<wf cmd=ignore pos=VBZ>is</wf>
<wf cmd=ignore pos=RB>not</wf>
<wf cmd=ignore pos=IN>on</wf>
<wf cmd=ignore pos=PRP$>his</wf>
<wf cmd=done id=d00.s14.t09 pos=NN lemma=side wnsn=10 lexsn=1:10:00::>side</wf>
<punc>.</punc>
</s>
```

FIGURE 3.1 – un extrait du corpus de test Senseval2

3.1.2 Senseval 3

Le corpus de test fournis lors de la campagne de Senseval 3 possède les mêmes caractéristiques que celui de Senseval 2. Il contient 1937 instances à désambiguïser pour trois articles traitant des sujets différents.

Le tableau 3.1 présente quelques données statistiques sur le nombre d'instances à désambiguïser pour chaque catégorie grammaticale (Part Of Speech, POS) des deux corpus. Il est remarquable sur le tableau que le nombre d'instances pour les adverbes est très faible, cette catégorie ne sera pas traitée dans nos implémentations.

Corpus	Noms	Verbes	Adjectifs	Adverbes
Senseval 2	1 057 (47%)	509 (23%)	417 (18%)	277 (12%)
Senseval 3	884 (46%)	719 (37%)	322 (17%)	12 (0.6%)

TABLE 3.1 – Le nombre d'instances réparties en catégories grammaticales où les instances sont définies dans WordNet

3.1.3 Les exceptions

Durant nos expérimentations, on a remarqué que les deux corpus contiennent quelques problèmes d'incompatibilité avec WordNet. Cette incompatibilité est illustré dans des cas où la catégorie grammaticale attribué par le corpus de l'instance à désambiguïser n'est pas définie dans WordNet, par exemple, l'instance *« fellow »* est définie comme adjectif (pos=JJ) dans le corpus Senseval 3, alors qu'elle est définie comme nom dans WordNet. L'autre cas est celui des mots (indéfinies) qui ne sont pas définis dans WordNet et ayant comme attribut *« wnsn=0 »*.

Le tableau 3.2 illustre le nombre d'exceptions ainsi que le nombre d'instances indéfinies dans WordNet.

Corpus	Exception	Indéfinies
Senseval 2	3	201
Senseval 3	2	124

TABLE 3.2 – Nombre d'exceptions et d'instances indéfinies dans Senseval 2 et Senseval 3

3.1.4 Quelques données statistiques

Cette section présente quelques données statistiques sur les deux corpus de test sous forme de tableaux.

Le tableau 3.3 illuste le pourcentage des instance ayant un seul sens (monosémique) dans les deux corpus

Corpus	Noms(%)	Verbes(%)	Adjectifs (%)	Adverbes(%)	Tous(%)
Senseval 2	22.8	2.3	19.4	37.2	20.54
Senseval 3	19.1	5.3	21.7	100	16.53

TABLE 3.3 – Pourcentages des instances monosémiques pour chaque catégorie grammaticale

Le tableau 3.4 illuste le nombre de mots disctincts et du nombre total de de leurs instances

Corpus	Nombre d'instances	Nombre de mots différents
Senseval 2	2 260	1 075
Senseval 3	1 937	952

TABLE 3.4 – Le nombre total d'instances et de mots différents

Le tableau 3.5 présente les instances pour chaque catégorie ayant un sens différent de celui le plus fréquent

Corpus	Noms	Verbes	Adjectifs	Adverbes
Senseval 2	295(27.9%)	302(59.3%)	129(30.9%)	48(17.3%)
Senseval 3	344(38.9%)	264(36.7%)	100(31%)	0

TABLE 3.5 – Nombre d'instances pour chaque catégorie grammaticale ayant un sens différent de celui du sens le plus fréquent.

Senseval 2	Senseval 3
gene (nom) : 60	be (verbe) : 137
cancer (nom) : 54	man (nom) : 17
ringer (nom) : 27	have (verbe) : 16
say (verbe) : 24	say (verbe) : 15
bell (nom) : 22	local (nom) : 13
not (adverbe) : 21	feel (verbe) : 12
cell (nom) : 21	State (nom) : 12

TABLE 3.6 – Liste des mots les plus fréquents

3.2 Environnement d'expérimentations

3.2.1 Présentation de WordNet

WordNet (voir annexe A) est une base de données lexicale développée par des linguistes du laboratoire des sciences cognitives de l'université de Princeton. Son but est de répertorier, classifier et mettre en relation de diverses manières le contenu sémantique et lexical de la langue anglaise.

La composante atomique sur laquelle repose le système entier est le synset *(synonym set)*. Un synset comporte la liste des synonymes exprimant le même concept, la définition du concept *(gloss)*, éventuellement des exemples d'usage, et les relations de ce concept avec les autres concepts. Ces caractéristiques sont présentées dans le Tableau 3.7.

Part of speech	Words	Concepts	TotalWord-Sense Pairs
Noun	114 648	79 689	141 690
Verb	11 306	13 508	24 632
Adjevtive	21 436	18 563	31 015
Adverb	4669	3664	5 808
Totals	152 059	115 424	203 145

TABLE 3.7 – Nombre de mots et de concepts de la base lexicographique WordNet 2.1 [15]

3.2.2 Présentation de Netbeans

NetBeans est un environnement de développement intégré (EDI), placé en open source par Sun en juin 2000. En plus de Java, NetBeans permet également de supporter différents autres langages, comme Python, C, C++, etc. Conçu en Java, NetBeans est disponible sous Windows, Linux, Solaris (sur x86 et SPARC), Mac OS X ou sous une version indépendante des systèmes d'exploitation (requérant une machine virtuelle Java). Un environnement Java Development Kit JDK (version 1.6) est requis pour les développements en Java.

3.3 Déroulement du processus de désambiguïsation

Le processus de la tâche de désambiguïsation suivi durant nos implémentations est constitué de trois grandes étapes :

3.3.1 Extraction et prétraitement des données

Durant la première étape du processsus de désambiguïsation, toutes les données nécessaires sont extraites à partir du corpus, les données étant séparrés dans des structures de données (des tableaux dans notre cas) où chaque tableau comporte un attribut pour toutes

les instances du corpus (c-à-d, un tableau pour les catégories grammaticales (POS), un tableau pour les formaes canoniques (lemma) et un tableau pour les étiquettes des sens attribuer par le corpus (wnsn)). Les informations nécessaires à la désambiguïsation sont extraites à apartir de WordNet et seront prétraités (tokenisation, élimination des mots vides, lemmatisation) pour qu'elles soient dans la même forme que les données extraite du corpus.

3.3.2 Désambiguïsation sémanique

Dans cette étape le choix de la méthode de désambiguïsation est fait (Algorithme de Lesk ou l'une de ses variantes, mesures de similarité), le choix de la taille de la feunêtre du contexte (on a coisit comme tailles : 2, 4, 6, 10, phrase comme contexte) et aprés la tâche de désambiguïsation (calcul des scores, attribution des sens aux descriptions éyant le plus grand score).

3.3.3 Analyse des résutats

Durant la dernière étape, les résultats obtenus seront comparrés avec les étiquettes de sens stockés dans le tableau contenant le wnsn, la précision est calculée et enfin une analyse des choix du sens (choix du sens le plus fréquent et choix du sens différent de celui le plus fréquent).

3.4 Expérimentations et résultats

Un texte arbitraire se compose de deux classes de mots une classe contenant les quatre catégories ouvertes (noms, verbes, adjectifs, adverbes) et un une contenant les catégories fermées. Durant nos expérimentations, les mots appartenant à la catégorie fermée doivent être éliminée du corpus de test afin de récupérer un contexte ayant seulement des mots pleins (appartenant à la catégorie ouverte), ils soivent être aussi éliminé des descriptions extraites de WordNet.

Cette section présente les différents variantes de l'algorithme de Lesk ainsi que d'autres

approches se basant sur le calcul de mesures de similarités entre les sens, implémentés dans ce travail.

3.4.1 Approches de Lesk

Les approches de Lesk font partie des méthodes se basant sur les connaissances en utilisant des ressources externes. WordNet est l'un de ces ressources, libre et gratuitement utilisables par le public.

Le principe des approches de Lesk est d'utiliser les informations fournies par un dictionnaire électronique pour la tâche de désambiguïsation. La section suivante présente l'algorithme original proposé par Lesk ainsi que des variantes implémentées dans notre système.

3.4.1.1 Algorithme de base de Lesk

3.4.1.1.a Principe

Avant de présenter l'idée de base de l'algorithme de Lesk, nous introduisons par un schéma les notations utilisées. La figure 3.3 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 $D(sj)$ dans le dictionnaire. Le mot *w* est représenté dans le dictionnaire par la réunion des définitions de ses sens, $E(w)$.

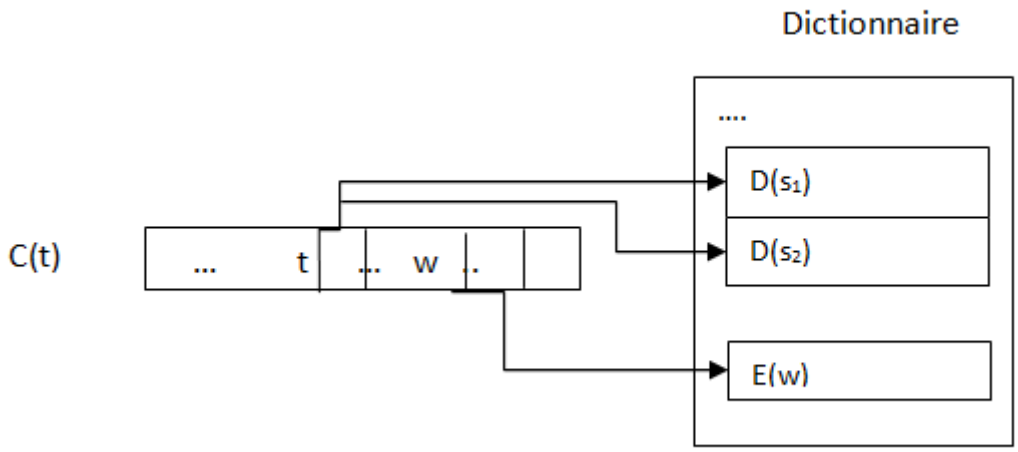

FIGURE 3.2 – Schéma de l'algorithme de Lesk de base [31]

L'interprétation de l'algorithme de Lesk que nous avons implémenté est décrite par le pseudo-code décrit dans la figure 3.4.

Les définitions $D(s_j)$ et $E(w))$ sont des sacs de mots qui n'appartiennent à aucune des 4 catégories fermées. 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 . 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 cet algorithme, les sacs de mots $D(s_j)$, $E(w))$ désignent des définitions + exemples d'usage extraits de WordNet. Les types de relation utilisée pour la désambiguïsation sont la synonymie et l'hyperonymie.

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 $sup = sup + |D(s_j) \cap E(w)|$
11. si $best\_score < sup$
12. $best\_score = sup$
13. $best\_candidate = s_j$
14. attribuer à t le sens donné par $best\_candidate$

FIGURE 3.3 – Pseudo-code de l'algorithme de base de Lesk [31]

3.4.1.1.b Résultats

Les précisions obtenues par notre système avec l'approche originelle de Lesk pour les deux corpus de test Senseval 2 et Senseval 3 sont respectivement illustrés dans les deux tableaux suivants (tableau 3.8 et tableau 3.9). On s'est basé seulement sur la précision comme métrique de performance puisqu'elle est presque équivoque au rappel (seulement 3 cas non traités pour Senseval 2 et 2 cas pour Senseval 3). Les colonnes 2-6 représentent les différentes tailles du contexte testées (par exemple, ± 2 signifie que le contexte contient 2 mots à gauche du mot cible et 2 à sa droite), les mots du contexte doivent appartenir aux 4 catégories ouvertes.

POS	± 2	±4	±6	±10	Phrase
Noms	45.87%	46.33%	43.43%	40.82%	42.42%
Verbes	33.84%	30.61%	29.70%	29.02%	31.01%
Adjectifs	50.36%	50.37%	48.98%	49.47%	53.17%
Adverbes	49.61%	41.96%	45.44%	37.74%	44.96%

TABLE 3.8 – Précision de l'algorithme original de Lesk. Senseval 2

POS	±2	±4	±6	±10	Phrase
Noms	38.97%	35.82%	34.43%	33.24%	34.83%
Verbes	27.75%	25.77%	23.79%	22.72%	26.70%
Adjectifs	53.41%	51.74%	49.92%	44.86%	51.64%

TABLE 3.9 – Précision de l'algorithme original de Lesk. Senseval 3

3.4.1.1.c Analyse des résultats

Une analyse des deux tableaux nous mène à conclure que l'augmentation de la taille du contexte, entraine une légère décroissance des performances sauf pour les adverbes, étant ces derniers sont éliminés lors de la campagne Senseval 3, ce qui confirme que : *« le faite de varier la taille du contexte ne produit pas en pratique des résultats très différents »* [15]. Nous pouvons également observer que les performances de l'algorithme pour les verbes sont les plus faibles, cela revient au degré de granularité qui atteint les 10 sens de moyenne (Senseval 2) et de 12 sens par moyenne (Senseval 3) par instance polysémique pour le corpus (*be : 13 sens, have : 19 sens, take : 42 sens, say : 11 sens*) pour le corpus [12]. On remarque aussi que les performances obtenues dans le corpus de Senseval 3 sont moins bonnes, cela est due à la taille du corpus (tableau 3.4).

Durant l'analyse de nos résultats, on a observé que la majorité des résultats positifs obtenus ont choisi comme sens le premier (sens par défaut), particulièrement pour les noms. Les tableaux 3.10 et 3.11 illustrent le pourcentage des résultats positifs obtenus en choisissant le sens possédant le plus grand score (numéro de sens (wnsn) différent de 1).

POS	±2	±4	±6	±10	Phrase
Noms	17.68%	21.21%	30.15%	25.85%	21.18%
Verbes	29.23%	36.95%	40.88%	40.24%	40.79%
Adjectifs	16.21%	21.95%	23.18%	24.18%	20.05%
Adverbes	16.67%	17.88%	19.30%	21.58%	21.94%

TABLE 3.10 – Pourcentage des résultats positifs ayant choisi un sens différent de celui par défaut, Lesk original. Senseval 2

POS	±2	±4	±6	±10	Phrase
Noms	17.94%	19.21%	23.64%	25.58%	19.06%
Verbes	27.80%	31.22%	31.01%	31.52%	27.04%
Adjectifs	24.30%	29.70%	28.82%	28.39%	25.09%

TABLE 3.11 – Pourcentage des résultats positifs ayant choisi un sens différent de celui par défaut, Lesk original. Senseval 3

On observe également que l'augmentation du contexte augmente les possibilités d'avoir des superpositions et donc défavorise le choix du sens par défaut.

Les résultats des autres approches implémentées dans ce travail seront présentés avec les mêmes formes des tableaux de cette approche.

3.4.1.2 Algorithme de Lesk simplifié

3.4.1.2.a Principe

Le principe de cette variante est similaire à celui de l'algorithme de base de Lesk, la différence réside dans le calcul des mots communs entre la définition $D(s_j)$ des sens du mot ambigu et les mots eux mêmes du contexte *w* et non leurs définitions. Soit $C(t))$ la fenêtre de contexte formée par le sac de mots *w*, en forme de base, la représentation de cet algorithme en pseudo-code est illustré dans la figure 3.5 :

1. pour chaque mot à désambiguïser t
2. $best\_score = 0$
3. $best\_candidate = s_1$
4. $sup = 0$
5. déterminer $C(t)$ le contexte de t
6. pour chaque sens candidat s_j de t
7. extraire du dictionnaire la définition $D(s_j)$
8. calculer le nombre de superpositions $sup = \left|D(s_j) \cap C(t)\right|$
9. si $best\_score < sup$
10. $best\_score = sup$
11. $best\_candidate = s_j$
12. attribuer à t le sens donné par $best\_candidate$

FIGURE 3.4 – Algorithme de Lesk, variante simplifiée [31]

3.4.1.2.b Résultats

Les résultats obtenus pour cette variante sont présentés dans les deux tableaux suivants (tableau 3.12, tableau 3.13) :

POS	±2	±4	±6	±10	Phrase
Noms	39.96%	41.0%	41.62%	42.63%	41.61%
Verbes	33.63%	33.41%	34.17%	33.09%	33.40%
Adjectifs	60.80%	60.80%	60.76%	59.78%	61.48%
Adverbes	68.86%	68.25%	68.21%	66.53%	67.09%

TABLE 3.12 – Précision de la variante Simplifiée de l'algorithme de Lesk. Senseval 2

POS	±2	±4	±6	±10	Phrase
Noms	40.24%	39.97%	40.24%	39.66%	40.65%
Verbes	37.15%	35.25%	33.41%	29.14%	33.9%
Adjectifs	49.94%	49.02%	49.57%	49.56%	49.86%

TABLE 3.13 – Précision de la variante Simplifiée de l'algorithme de Lesk. Senseval 3

3.4.1.2.c Analyse des résultats

Les performances pour la variante simplifiée de l'algorithme de Lesk ont été légère-

ment inferieurs à celles obtenues dans la version originelle de Lesk pour les traitements des noms, les meilleures performances sont obtenues par les adjectifs et les adverbes pour le corpus Senseval 2. Pour les verbes le niveau de précision s'est amélioré (37%) pour Senseval 3.

On observe également que le changement de la taille du contexte n'influence pas sur les performances du système. Limiter la taille du contexte à la phrase apparaît comme une bonne alternative puisque les performances obtenues se rapprochent des performances moyennes pour les différentes tailles du contexte testées.

Les tableaux 3.14 et 3.15 présentent une analyse des décisions prises par notre systèmes (sens par défaut / meilleur score).

POS	**±2**	**±4**	**±6**	**±10**	**Phrase**
Noms	16.81%	17.32%	18.13%	18.75%	17.67%
Verbes	21.71%	23.48%	25.32%	27.2%	23.93%
Adjectifs	10.31%	10.31%	11.47%	11.66%	13.04%
Adverbes	1.65%	1.67%	2.57%	3.55%	1.69%

TABLE 3.14 – Pourcentage des résultats positifs ayant choisi un sens différent de celui par défaut, Lesk simplifié. Senseval 2

POS	**±2**	**±4**	**±6**	**±10**	**Phrase**
Noms	13.58%	13.99%	14.22%	14.95%	14.31%
Verbes	20.34%	22.73%	25.02%	25.87%	22.90%
Adjectifs	13.26%	14.6%	14.88%	16.45%	15.38%

TABLE 3.15 – Pourcentage des résultats positifs ayant choisi un sens différent de celui par défaut, Lesk simplifié. Senseval 3

Les deux tableaux confirment que l'augmentation de la taille du contexte équilibre entre les décisions du choix du sens le plus fréquent (choix par défaut) et le choix du sens ayant le meilleur score. Les performances élevées des adverbes pour le corpus de Senseval 2 sont dues au choix du sens le plus fréquent par le système avec une moyenne qui dépasse les 95% des cas. Les deux tableaux confirment que l'augmentation de la taille du contexte équilibre entre les décisions du choix du sens le plus fréquent (choix par défaut) et le choix du sens ayant le meilleur score. Les performances élevées des adverbes pour le corpus de Senseval 2 sont dues au choix du sens le plus fréquent par le système

avec une moyenne qui dépasse les 95% des cas.

3.4.2 Les variantes proposées

3.4.2.1 Algorithme de Lesk simplifié utilisant les relations de synonymie

3.4.2.1.a Principe

Le principe de cette variante ressemble à celui de la variante simplifiée, le score se calcul par la somme des superpositions entre les mots synonymes $Syn(s_j)$ du sens s_j de *t* (plutôt que la définition) et les mots de son contexte $C(t)$. Le pseudo-code de cet algorithme est modifié par rapport aux deux précedents dans la méthode de calcul du score, il est présenté dans la figure 3.6 où $Syn(s_j)$ est l'ensemble des mots synonymes de *t*.

1. pour chaque mot à désambiguïser *t*
2. *best_score* = 0
3. *best_candidate* = s_1
4. *sup* = 0
5. déterminer *C(t)* le contexte de *t*
6. pour chaque sens candidat s_j de *t*
7. extraire du dictionnaire les termes synonymes $Syn(s_j)$
8. calculer le nombre de superpositions *sup* = $\left|Syn(s_j) \cap C(t)\right|$
9. si *best_score* < *sup*
10. *best_score* = *sup*
11. *best_candidate* = s_j
12. attribuer à *t* le sens donné par *best_candidate*

FIGURE 3.5 – Algorithme de Lesk, variante utilisant les synonymes

3.4.2.1.b Résultats

Les résultats de cette variante sont présentés dans les deux tableaux suivants (tableau 3.16, tableau 3.17) :

POS	±2	±4	±6	±10	Phrase
Noms	64.55%	64.55%	62.72%	62.60%	63.06%
Verbes	42.25%	42.04%	40.22%	39.82%	40.0%
Adjectifs	62.59%	62.59%	60.79%	60.78%	60.78%
Adverbes	65.66%	75.66%	72.83%	72.23%	72.28%

TABLE 3.16 – Précision de l'algorithme simplifié de Lesk en utilisant les synonymes. Senseval 2

POS	±2	±4	±6	±10	Phrase
Noms	62.67%	62.16%	61.89%	60.90%	61.87%
Verbes	49.25%	48.84%	47.83%	47.21%	49.09%
Adjectifs	62.2%	62.2%	62.2%	62.2%	62.2%

TABLE 3.17 – Précision de l'algorithme simplifié de Lesk en utilisant les synonymes. Senseval 3

3.4.2.1.c Analyse des résultats

Les résultats obtenus par cette variante sont largement meilleurs que ceux des variantes précédentes, ils atteignent un niveau de précision moyen de 60% pour Senseval 2. Cependant, cette performance revient au choix du sens le plus fréquent qui atteint souvent les niveaux de 100% (les mots réussis sont seulement ceux qui ont comme sens exacte, le premier sens). Il existe quelques cas traités avec réussite, mais ils sont très rares (15 seulement pour les deux corpus).

3.4.2.2 Algorithme de Lesk de base avec relations de synonymie

3.4.2.2.a Principe

Cette variante combine entre l'algorithme original (base) de Lesk et la variante précédente (relation de synonymie). Le calcul du score se fait par la somme des superpositions des mots de la définition $D(s_j)$ du terme *t* et les mots synonymes du sens du mot w_i de $Syn(w_i)$. La figure 3.7 présente le pseudo-code de cette variante.

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

FIGURE 3.6 – Algorithme de Lesk de base avec les relations de synonymie

3.4.2.2.b Résultats

Les résultats de cette approche sont présentés dans les tableaux 3.18 et 3.19 suivants :

POS	±2	±4	±6	±10	Phrase
Noms	56.68%	53.76%	52.2%	47.22%	53.06%
Verbes	34.90%	34.19%	34.13%	33.16%	34.11%
Adjectifs	55.22%	57.93%	55.22%	54.97%	58.09%
Adverbes	60.05%	59.34%	59.40%	55.41%	61.19%

TABLE 3.18 – Précision de l'algorithme original de Lesk en utilisant les synonymes. Senseval 2

3.4.2.2.c Analyse des résultats

Il est remarquable que les résultats obtenus de cette variantes pour les noms sont meilleurs que ceux obtenus dans les variantes implémentées précédemment à l'exception de celle utilisant les synonymes du terme *t* au lieu de sa définition qui été favorisé par le choix du sens le plus fréquent après une égalité du score pour tous les sens de *t*. les résultats obtenus pour les verbes sont légèrement meilleurs (32%-34%).

POS	±2	±4	±6	±10	Phrase
Noms	53.72%	47.88%	46.98%	43.09%	50.57%
Verbes	35.41%	32.76%	29.28%	25.25%	32.14%
Adjectifs	54.70%	54.58%	50.31%	46.34%	53.29%

TABLE 3.19 – Précision de l'algorithme original de Lesk en utilisant les synonymes. Senseval 3

Cette variante confirme que l'augmentation de la taille de la fenêtre du contexte diminue les performances. Les meilleurs résultats obtenus jusqu'à présent sont généralement pour les tailles du contexte moyennes (4, 6 et la phrase).

Comme pour les autres variantes, les tableaux 3.20 et 3.21 présentent la répartition des décisions prises pour les termes cibles.

POS	±2	±4	±6	±10	Phrase
Noms	9.51%	8.25%	11.08%	14.54%	9.67%
Verbes	18.40%	17.80%	22.92%	26.23%	21.03%
Adjectifs	8.14%	5.95%	6.82%	11.48%	7.67%
Adverbes	6.06%	4.03%	5.86%	7.35%	3.84%

TABLE 3.20 – Pourcentage des résultats positifs ayant choisi un sens différent de celui par défaut, Lesk original avec synonymes. Senseval 2

POS	±2	±4	±6	±10	Phrase
Noms	5.74%	7.60%	11.53%	16.0%	10.72%
Verbes	17.26%	22.21%	26.91%	29.83%	17.4%
Adjectifs	9.41%	13.07%	14.85%	21.27%	13.61%

TABLE 3.21 – Pourcentage des résultats positifs ayant choisi un sens différent de celui par défaut, Lesk original avec synonymes. Senseval 3

Il est clairement remarquable que l'augmentation de la taille du contexte augmente significativement les scores des sens, ce qui permet au système de sélectionner les sens ayant le meilleur score.

3.4.3 Approches utilisant des mesures de similarités

Les résultats obtenus lors de l'implémentation des approches basées sur l'lagorithme de Lesk qui utilisent des informations de types morphologiques nous ont poussés à faire

d'autres expérimentations avec des approches utilisant des informations sémantiques pour calculer la mesure de similarité entre concepts

D'autres approches implémentées dans ce travail sont des approches calculant les mesures de similarité entre les sens. La mesure de similarité étant le degré de relation de deux concepts, cela peut être utilisé pour déterminer le sens d'un mot à partir de son contexte. Plusieurs types de mesures ont été proposées, les plus connues entre elles et qu'on va implémenter dans ce travail sont celle de Lesk (*AdaptedLesk*) et celle de WU AND PALMER (*Wup*).

Dans notre travail, on a utilisé les implémentations des mesures de similarité fournie librement par le logiciel de Perl *« WordNet : : Similarity »* développées par Pedersen et al. en 2004.

3.4.3.1 Mesure de similarité de Lesk (Adapted Lesk)

3.4.3.1.a Principe

La mesure de similarité de Lesk fait partie des mesures se basant sur le gloss, proposée par Banerjee et Pedersen [2]. Elle combine les avantages du principe de superposition de gloss avec la structure hiérarchique des concepts, donc la mesure doit être calculée entre deux mots ayant la même catégorie grammaticale.

Le mécanisme du calcul du score est différent de celui de Lesk (qui calcule simplement le nombre de mots communs entre les définitions). Le mécanisme de lesk ne différencie pas entre une superposition de phrases et une superposition de mots, le calcul du score pour cette mesure de similarité se fait : pour deux définitions (gloss) données, la plus langue superposition (overlap) entre les deux définitions est détectée. L'overlap est enlevé et un unique marqueur est placé dans les deux chaines. Le processus est continu en mode récursif jusqu'il y aura plus d'overlap entre les deux chaines. Une superposition entre deux phrases de *n* mots est assignée d'un score n^2. La sommation des carrés de la longueur des overlaps est le score pour les paires des définitions.

$$\mathrm{p}airescore = \sum_{i=1}^{\#overlap} \mathrm{l}ength^2(overlap_i) \tag{3.1}$$

Par exemple, pour les deux synsets $s_1 = cat\#n\#7$ (*cat, pos= nom, wnsn= 7*) et $s_2 = dog\#n\#1$ et les overlaps *(claws, fissiped mamals)* utilisant une relation hypernym-hypernym. Le score entre s_1 et s_2 sera $1 + 2^2 = 5$.

Le grand inconvénient de cette mesure est le temps d'exécution. Dans notre implémentation, on n'a traité que le corpus Senseval 3 pour une fenêtre de contexte de deux mots pleins à droite et à gauche du mot cible.

3.4.3.1.b Résultats

Le tableau 3.22 présente les résultats obtenus par la mesure de similarité de Lesk (ALesk).

POS	**±2**
Noms	45.18%
Verbes	38.98%
Adjectifs	52.15%

TABLE 3.22 – Précision de la mesure ALesk pour un contexte égale à 2. Senseval 3

3.4.3.1.c Analyse des résultats

Les résultats obtenus pour cette mesure sont à la hauteur des implémentations précédentes, ils sont légèrement meilleurs pour la catégorie des verbes pour une fenêtre de contexte égale à 2. Le tableau 3.23 montre les pourcentages des choix du sens ayant un sens différent de celui le plus fréquent. Les décisions pour le sens pour les noms et les

POS	**±2**
Noms	18.59%
Verbes	16.05%
Adjectifs	5.86%

TABLE 3.23 – Pourcentage des résultats positifs ayant choisi un sens différent de celui par défaut, ALesk. Senseval 3

verbes n'étaient pas pour le sens le plus fréquent seulement (81%, 84% respectivement). Cependant, cette approche ne peut être évaluée pour une seule fenêtre du contexte.

3.4.3.2 Mesure de similarité de Wu et Palmer (Wup)

3.4.3.2.a Principe

La mesure de similarité proposé par WU AND PALMER citée dans [36] fait partie des mesures se basant sur le nombre d'arcs ou de noeuds entre deux concepts dans une hiérarchie de concepts. Un exemple d'hiérarchie des concepts de WordNet est illustré dans la figure 3.7.

Par exemple, dans la figure 3.8, le chemin entre *cheese* et *chocolate* est égale à 2 et entre *cheese* et *wood* à 7 indiquant que les concepts cheese et chocolate sont plus similaire que *cheese* et *wood*. La mesure de *Wu and Palmer* repose sur la notion du plus

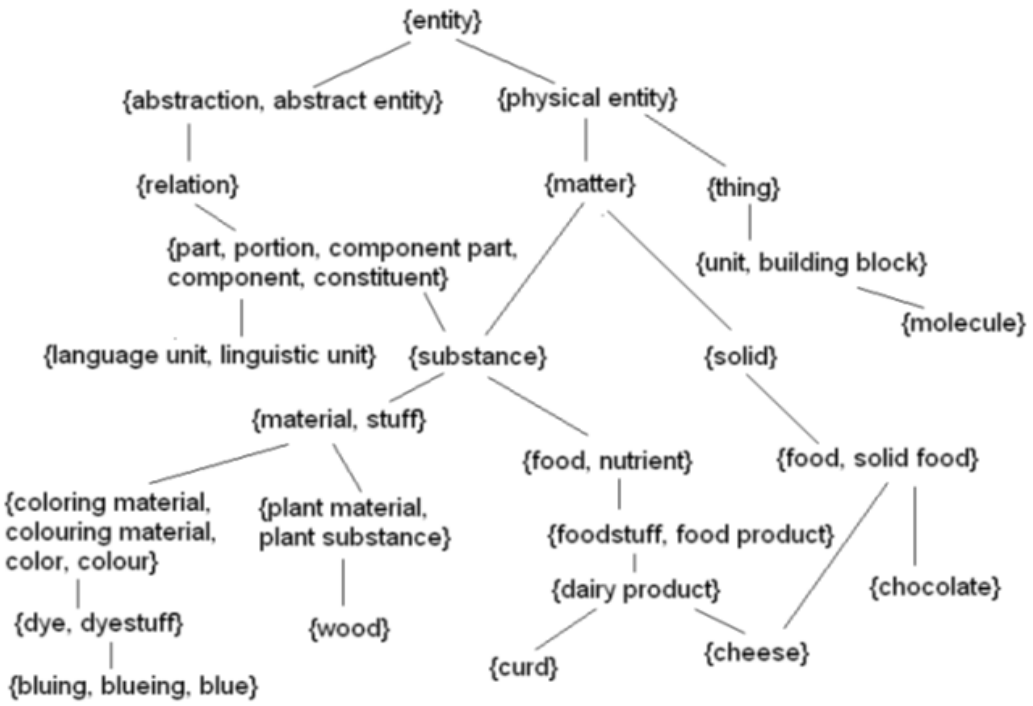

FIGURE 3.7 – Illustration d'une hiérarchie de concept de WordNet [12]

petit généralisant commun, c'est-à-dire le concept généralisant commun à $c_1$et c_2 le plus éloigné de la racine. Elle est définie par :

$$\mathrm{s}im(c_1, c_2) = \frac{2 \cdot depth(c)}{depth(c_1) + depth(c_2)} \tag{3.2}$$

La mesure de similarité sémantique implémentée dans ce travail calcul la similarité

entre chaque sens du mot polysémique et tous ses hypernymes avec les mots de son contexte et leurs hypernymes également. La décision est prise pour le sens ayant la plus grande valeur.

3.4.3.2.b Résultats

Les résultats obtenus par la mesure de Wu And Palmer sont illustré dans le tableau 3.24.

POS	**±2**	**±4**	**±6**	**±10**	**Phrase**
Noms	16.20%	19.41%	19.25%	15.87%	18.52%
Verbes	13.35%	21.64%	24.44%	23.16%	20.93%

TABLE 3.24 – Précision de la mesure Wup. Senseval 3

3.4.3.2.c Analyse des résultats

Les précisons obtenues pour cette méthodes sont loin d'être intéressantes, cela revient au grand nombre de calcul de la mesure de similarité entre les concepts durant le parcours de tous les hypernymes de chaque sens des mots du corpus.

3.5 Etude comparative des différentes implémentations

La dernière section porte sur une étude comparative sous forme d'histogrammes des différentes méthodes implémentées dans ce travail. Chaque histogramme illustre les précisions obtenues dans un corpus donné (Senseval 2 ou Senseval 3) pour une catégorie grammaticale donnée (noms, verbes, adjectifs, adverbes).

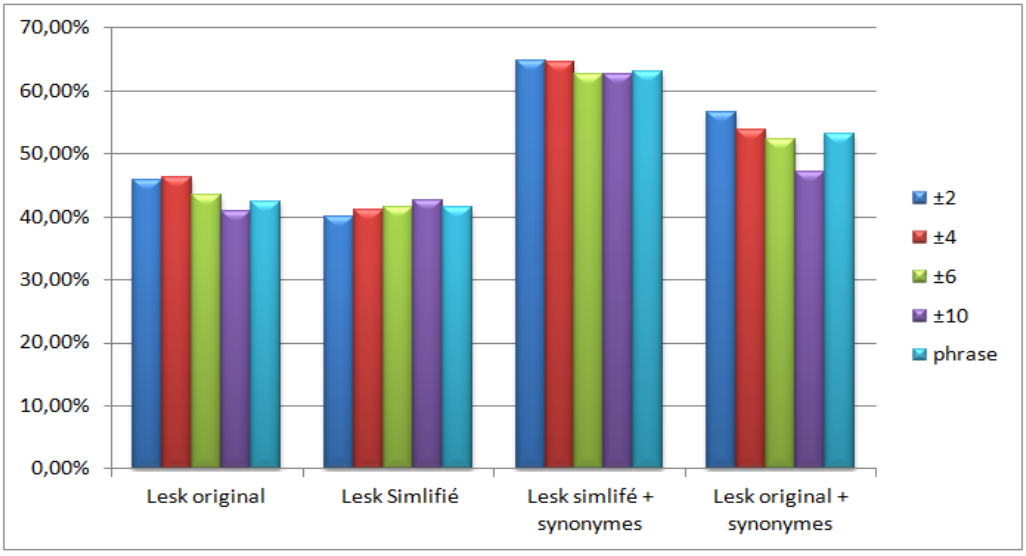

FIGURE 3.8 – Résultats des noms pour les différentes approches implémentées. Senseval 2.

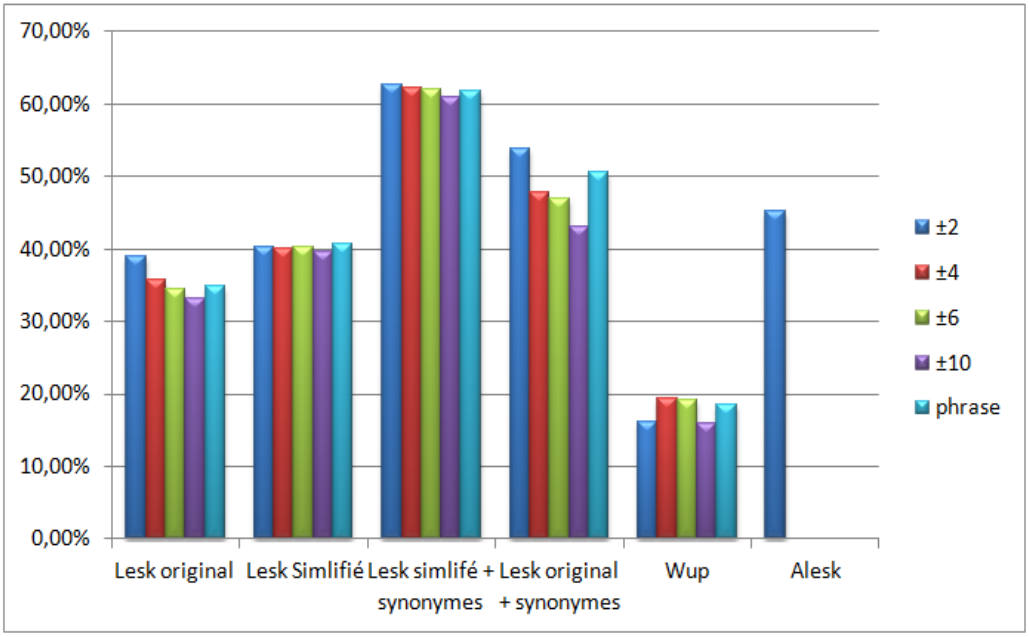

FIGURE 3.9 – Résultats des noms pour les différentes approches implémentées. Senseval 3.

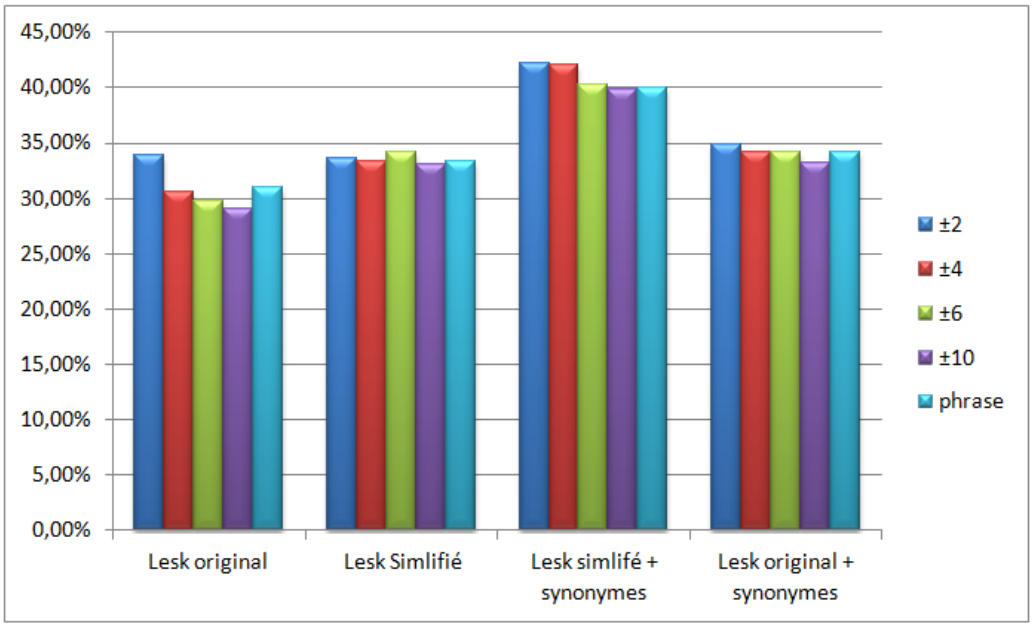

FIGURE 3.10 – Résultats des verbes pour les différentes approches implémentées. Senseval 2

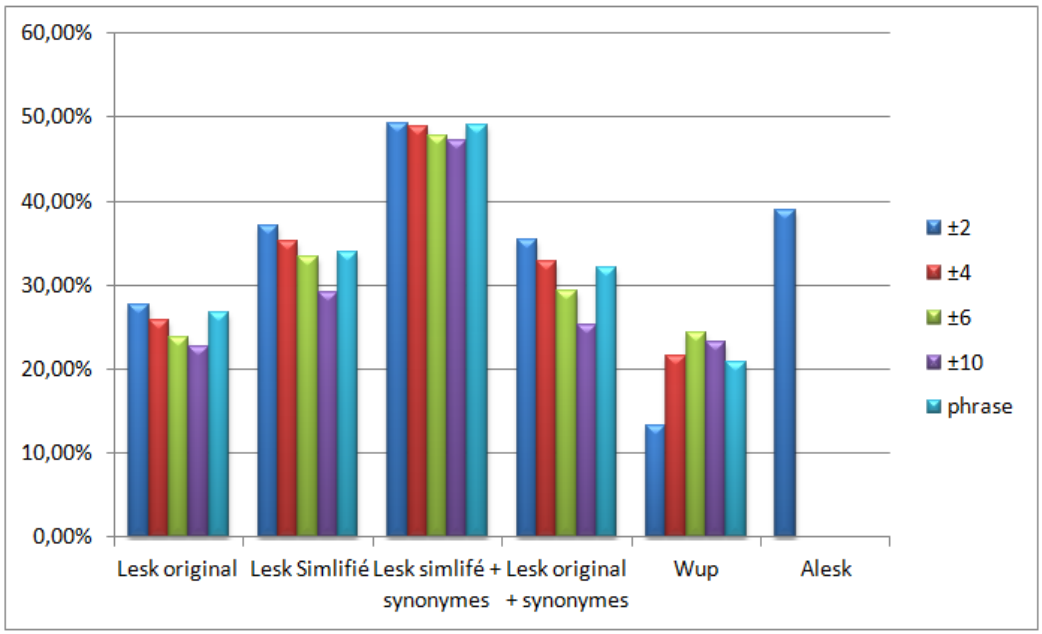

FIGURE 3.11 – Résultats des verbes pour les différentes approches implémentées. Senseval 3

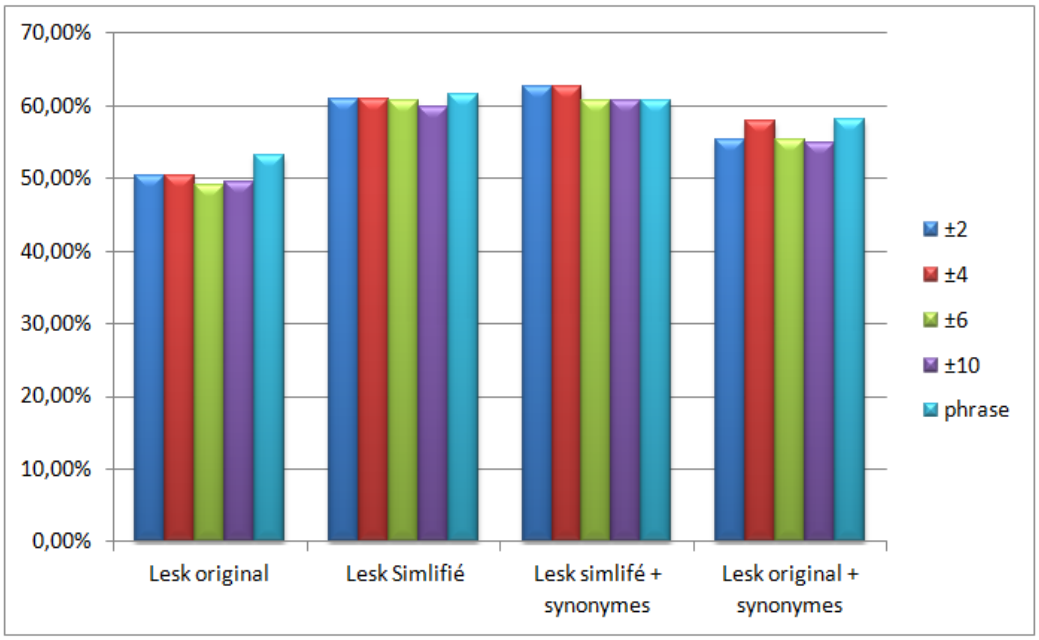

FIGURE 3.12 – Résultats des adjectifs pour les différentes approches implémentées. Senseval 2

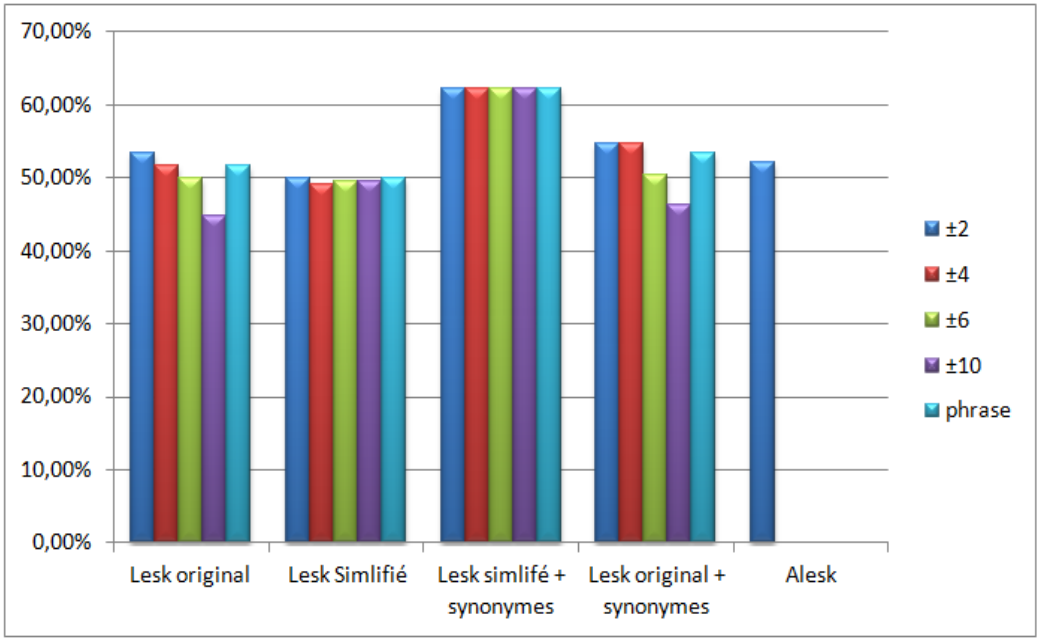

FIGURE 3.13 – Résultats des adjectifs pour les différentes approches implémentées. Senseval 3

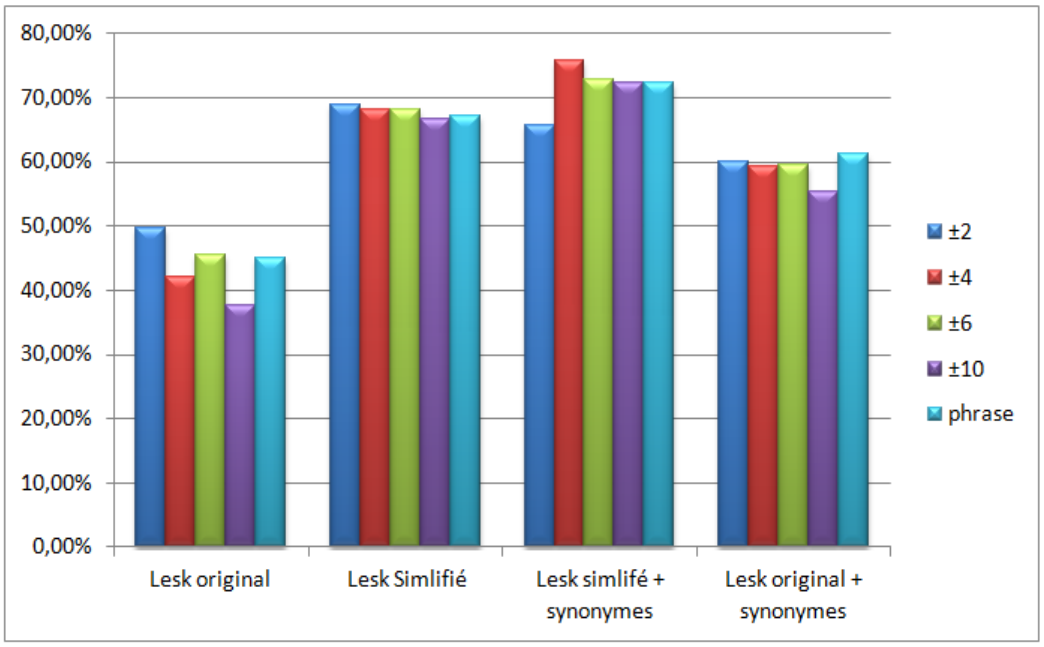

FIGURE 3.14 – Résultats des adverbes pour les différentes approches implémentées. Senseval 2

Conclusion

On a présenté durant ce chapitre les différentes approches implémentées dans ce projet en se basant sur l'algorithme de Lesk et un nombre de ces variantes ainsi que sur des mesures de similarité.

Les résultats obtenus dans nos implémentations montrent que l'augmentation de la taille de la fenêtre du contexte ne font que diminuer les performances, les meilleurs résultats obtenus étaienet pour des tailles de contexte égale à 4 et 6. Les meilleures performances ont étaient enregistrées par la variante original de Lesk utilsant des relations de synonymie des mots du contexte.

Il est difficile d'avoir des conclusions pour nos implémentations, il faut noter que ces performances n'ont été valable que pour les deux corpus de test fournis par *Senseval* et ils peuvent changer pour d'autres corpus de taille plus importantes

Conclusion générale

Notre étude porte sur l'étude de l'algorithme de LESK qui est le plus utilisé dans la désambiguïsation sémantique. Les expérimentations effectuées portaient sur la proposition de nouvelles variantes de cet algorithme. Ces variantes se basent sur l'utilisation de nouvelles méthodes de superpositions basées sur la relation de synonymie, ainsi que l'utilisation des distances sémantiques.

Afin d'avoir un caractère plus général à nos implémentations et à notre recherche, nos implémentations ont étaient testées sur des corpus de test provenant de Senseval2 et de senseval3. Les résultats des variantes proposées durant ce travail ont donné des performances meilleures que celle de la variante originelle de Lesk aisi que sa variante simplifiée.

Utiliser WordNet Domains pour déterminer le domaine du contexte en combinaison avec WordNet peut être une bonne alternative d'une façon que le dictionnaire peut fournir des exemples d'usages du domaine concerné.

Comme mentionné dans le chapitre précédent, il est difficile de généraliser les performances pour les deux corpus traités dans ce travail. Utiliser des corpus de volume important donne des résultats plus générales.

Bibliographie

[1] Laurent Audibert. *Outils d'exploration de corpus et désambiguïsation lexicale automatique*. PhD thesis, Université Provence, Marseille, France, 2003.

[2] Satanjeev Banerjee and Ted Pedersen. An adpted algorithm for word sens disambiguation using wordnet. *Proceeding of the Third International Conference on Intelligent Text Processing and Computational Linguistics*, pages 27–23, feb 2002.

[3] E. Black. An experiment in computational discrimination of english word senses. *IBM Journal of Research and Devlopment*, pages 185–194, feb 1988.

[4] E. Briscoe. Natural language and speech. *Lexical issues in natural language processing*, pages 39–68, 1991.

[5] 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*, 2002.

[6] Weiwei Guo and Mona Diab. Improvements to monolingual english word sense disambiguation. *Proceedings of the Workshop on Semantic Evaluations : Recent Achievements and Future Directions (SEW-2009)*, pages 64–69, jun 2009.

[7] A. Harley and D. Glennon. Sense tagging in action : Combining different test with additive weighings. *Association for Computational Linguistics Special Interest Group on the Lexicon (ACL-SIGLEX-1997)*, pages 74–78, jun 1997.

[8] N. Ide and J Véronis. Word sense disambiguation : The state of the art. *Computational Linguistics : Special Issue on Word Sense Disambiguation*, pages 1–40, 1998.

[9] Preiss Judita, Dehdari Jon, King Josh, and Mehay Dennis. Refining the most frequent sense baseline. *In Proceedings of the Workshop on Semantic Evaluations :Recent Achievements and Future Directions. Association for Computational Linguistics*, pages 10–18, jun 2009.

[10] A. Kilgarriff. Senseval : An exercise in evaluating word sense disambiguation programs. *8th International Congress on Lexicography (EURALEX-1998)*, pages 174–176, 1998b.

[11] Adam Kilgarriff and Joseph Rosenzweig. Framework and results for english senseval. *Computers and the Humanities*, pages 15–48, may 2000b.

[12] Varada Kolhatkar. *An Extended Analysis of a Method of All Words Sense Disambiguation*. PhD thesis, Faculty of the graduate school of the university of Minnesota, 2009.

[13] S. G. Kolte and S. G. Bhirud. Wordnet : A knowledge source for word sense disambiguation. *International Journal of Recent Trends in Engineering*, pages 213–217, Nov 2009.

[14] D. B. Lenat and R. Guha. *Building large knowledge-based systems : Representation and inference in the cyc project*. Massachusetts : Addison-Wesley.(ISBN : 0201517523), 1989.

[15] M. Lesk. Automatic sense disambiguation using machine readable dictionaries : How to tell a pine cone from an ice cream cone. *Special Interest Group for DOCumentation*, pages 24–26, Nov 1986.

[16] Rada Mihalcea and Ehsanul Faruque. Senselearner : Minimally supervised word sense disambiguation for all words in open text. *Association for Computational Linguistics*, pages 155–158, jul 2004.

[17] G. Miller, R. Beckwith, D. Fellbaum, C.AND Gross, and K. Miller. Wordnet : An on-line lexical database. *International Journal of Lexicography*, pages 235–244, 1990.

[18] Roberto Navigli and Mirella Lapata. Graph connectivity measures for unsupervised word sense disambiguation. *In Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 1683–1688, 2007.

[19] H. T. Ng and Y. K Lee. Integrating multiple knowledge sources to disambiguate word sense : An exemplar-based approach. *34thAnnual Meeting of the Society for Computational Linguistics*, pages 40–47, 1996.

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

[21] Judita Preiss, Anna Korhonen, and Ted Briscoe. Subcategorization acquisition as an evaluation method for wsd. *Proceedings of LREC*, pages 1551–1556, 2002.

[22] H Schütze. Dimensions of meaning. *Supercomputing-1992*, pages 787–796, 1992.

[23] H Schütze. Automatic word sense discrimination. *Computational Linguistics : Special Issue on Word Sense Disambiguation*, pages 97–123, 1998.

[24] S. Y. Sedelow and W. A. J. Sedelow. Categories and procedures for content analysis in the humanities. *The Analysis of Communication Content*, pages 487–499, 1969.

[25] S. Y. Sedelow and W. A. J. Sedelow. Thesaurus knowledge representation. *University of Waterloo Conference on Lexicology*, pages 29–43, 1986.

[26] S. Y. Sedelow and W. A. J. Sedelow. Recent model-based and model-related. *studies of a large-scale lexical resource (roget's thesaurus). 14th International Conference on Computational Linguistics*, pages 1223–1227, 1992.

[27] Grigori Sidorov and Alexander Gelbukh. Word sense disambiguation in a spanish explanatory dictionary. *Proceedings TALN-2001*, pages 398–402, jul 2001.

[28] K. Sparck Jones. *Synonymy and semantic classification*. PhD thesis, University of Cambridge, Cambridge, England, 1964.

[29] K. Sparck Jones. *Synonymy and semantic classification*. PhD thesis, Edinburgh, England : Edinburgh University Press, 1986.

[30] M. Stevenson and Y Wilks. Computational linguistics. *The interaction of knowledge sources in word sense disambiguation*, pages 321–349, jul 2001.

[31] F Vasilescu. *Désambiguïsation de corpus monolingues par des approches de type Lesk*. PhD thesis, Montréal, Canada, Université de Montréal, 2003.

[32] E. M. Voorhees. Using wordnet to disambiguate word senses for text retrieval,16thannual international conference on research and development in information retrieval. *Association for Computing Machinery Special Interest Group on Information Retrieval*, pages 171–180, 1993.

[33] Jean Véronis. Sense tagging : does it make sense ? *Proceedings of the Corpus Linguistics 2001 Conference*, jul 2001.

[34] Y. Wilks and M. Stevenson. Word sense disambiguation using optimised combinations of knowledge sources. *36thAnnual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, pages 1398–1402, 1998.

[35] Y. A. Wilks, D. C. Fass, C. M. Guo, J. E. MacDonald, T. Plate, and B. A. Slator. Providing machine tractable dictionary tools. *Cambridge, Massachusetts : MIT Press*, 1990.

[36] D. Yarowsky. Word sense disambiguation using stati. *stical models of roget's categories trained on large corpora. 14th International Conference on Computational Linguistics*, pages 454–460, 1992.