Désambiguïsation sémantique dans le cadre de la simplification lexicale : contributions à un système d'aide à la lecture pour des enfants dyslexiques et faibles lecteurs

Mokhtar Boumedyen Billami

AIX-MARSEILLE UNIVERSITÉ
ED 356 COGNITION, LANGAGE, ÉDUCATION
LABORATOIRE D'INFORMATIQUE ET SYSTÈMES
LIS – UMR CNRS – 7020

Thèse présentée pour obtenir le grade universitaire de docteur

Discipline : Sciences du langage
Spécialité : Traitement automatique des langues

Mokhtar Boumedyen BILLAMI

Désambiguïsation sémantique dans le cadre de la simplification lexicale : contributions à un système d'aide à la lecture pour des enfants dyslexiques et faibles lecteurs

Soutenue le 15/11/2018 devant le jury composé de :

Olivier FERRET	CEA LIST – LVIC	Rapporteur
Mathieu LAFOURCADE	Université Montpellier 2 – LIRMM	Rapporteur
Cécile FABRE	Université Toulouse 2 – CLLE	Examinatrice
Laurent PRÉVOT	Aix-Marseille Université – LPL	Examinateur
Núria GALA PAVIA	Aix-Marseille Université – LPL	Directrice de thèse
Johannes ZIEGLER	Aix-Marseille Université – LPC	Co-directeur de thèse

Résumé

La lecture est fondamentale pour tout ce qu'un enfant doit apprendre pendant son parcours scolaire. D'après des rapports nationaux (MJENR 2003) ou internationaux (PISA 2009), 20% à 30% des élèves français sont de faibles lecteurs et ont des difficultés pour comprendre les textes écrits, 5% à 10% sont des enfants dyslexiques. Ces lecteurs sont en grande difficulté face à des textes complexes ou avec un vocabulaire peu courant.

Ces dernières années, un nombre important de technologies ont été créées pour venir en aide aux personnes ayant des difficultés pour lire des textes écrits. Les systèmes proposés intègrent des technologies de la parole (lecture à « voix haute ») ou des aides visuelles (paramétrage et/ou mise en couleur des polices ou augmentation de l'espace entre lettres et lignes). Cependant, il est essentiel de proposer aussi des transformations sur le contenu afin d'avoir des substituts de mots plus simples et plus fréquents. Cela permettra de rendre les textes plus accessibles et plus faciles à lire et à comprendre. Le but de cette thèse est de contribuer à un système d'aide à la lecture permettant de proposer automatiquement une version simplifiée d'un texte donné tout en gardant le même sens des mots.

Le travail présenté adresse le problème de l'ambiguïté sémantique (très courant en traitement automatique des langues) et vise à proposer des solutions pour la désambiguïsation sémantique à l'aide de méthodes non supervisées et à base de connaissances provenant de ressources lexico-sémantiques. Dans un premier temps, nous proposons un état de l'art sur les méthodes de désambiguïsation sémantique et les mesures de similarité sémantique (essentielles pour la désambiguïsation sémantique). Par la suite, nous comparons divers algorithmes de désambiguïsation sémantique afin d'identifier le meilleur. Enfin, nous présentons nos contributions pour la création d'une ressource lexicale pour le français proposant des synonymes désambiguïsés et gradués en fonction de leur niveau de difficulté de lecture et compréhension. Nous montrons que cette ressource est utile et peut être intégrée dans un module de simplification lexicale de textes.

Mots clés : désambiguïsation sémantique, simplification lexicale, traitement automatique des langues, enfants dyslexiques, faibles lecteurs.

Abstract

Reading is fundamental to everything that a child needs to learn during his school career. According to national reports (MJENR 2003) or international reports (PISA 2009), 20% to 30% of French pupils are poor readers and have difficulties to understand the written texts, 5% to 10% are dyslexic children. These readers are very troubled when reading complex texts or texts with unusual vocabulary.

In recent years, a large number of technologies have been created to help people who have difficulty when reading written texts. The proposed systems integrate speech technologies (reading aloud) or visual aids (setting and/or coloring of fonts or increasing the space between letters and lines). However, it is essential to also propose transformations on the texts' content in order to have simpler and more frequent substitutes. This will make the texts more accessible and easier to read and understand. The purpose of this thesis is to contribute to develop a reading aid system that automatically provides a simplified version of a given text while keeping the same meaning of words.

The presented work addresses the problem of semantic ambiguity (quite common in natural language processing) and aims to propose solutions for word sense disambiguation (WSD) by using unsupervised and knowledge-based approaches from lexico-semantic resources. First, we propose a state of the art of the WSD approaches and semantic similarity measures which are crucial for this process. Thereafter, we compare various algorithms of WSD in order to get the best of them. Finally, we present our contributions for creating a lexical resource for French that proposes disambiguated and graduated synonyms according to their level of difficulty to be read and understood. We show that our resource is useful and can be integrated in a lexical simplification of texts module.

Keywords: word sense disambiguation, lexical simplification, natural language processing, dyslexic children, poor readers.

Remerciements

Mes remerciements, les plus vifs, ma profonde gratitude et mes respects s'adressent à Núria Gala, ma directrice de thèse, pour m'avoir accueilli au sein de l'équipe TALEP (Traitement Automatique du Langage Écrit et Parlé) du pôle Science des Données (SD), pour avoir pris sur son temps chaque fois que je sollicitais ses précieux conseils et pour m'avoir apporté les moyens d'arriver au bout de ce long et difficile chemin.

Je remercie particulièrement Johannes Ziegler, mon co-directeur de thèse, pour m'avoir fait découvrir le domaine de la psycholinguistique et les travaux de recherche qui se font dans le Laboratoire de Psychologie Cognitive (LPC). Je le remercie aussi pour les nombreuses explications fournies sur les difficultés de lecture et de compréhension auprès des enfants dyslexiques et faibles lecteurs lors de la lecture de textes écrits.

Je remercie ensuite tous mes collègues de l'équipe TALEP pour l'ambiance toujours conviviale et les nombreuses discussions intéressantes durant mes quatre années de thèse.

Mes remerciements s'adressent spécialement à Thomas François (membre du CENTAL, Centre de traitement automatique du langage à l'Université Catholique de Louvain (UCL)) pour nos collaborations précieuses ainsi que pour son aide et ses différents conseils.

Je tiens à remercier aussi Olivier Ferret et Mathieu Lafourcade qui m'ont fait l'honneur de rapporter ma thèse et avoir accordé du temps à une lecture attentive et détaillée de mon manuscrit. Je remercie également les examinateurs, Professeur Cécile Fabre (Université Toulouse 2) et Professeur Laurent Prévot (Université d'Aix-Marseille) d'avoir accepté de participer à mon jury de thèse et échangé leurs points de vue.

Des remerciements qui n'ont pas de limite, qui n'ont pas de mots pour les exprimer, je les adresse aux trois bougies qui illuminent ma vie : ma mère, mon père et ma femme Meryem, qui m'ont toujours soutenu dans mes choix et qui m'ont toujours encouragé à aller de l'avant. Que Dieu leur donne une longue vie pleine de santé et de joie. Une belle pensée aussi à mes frères et à ma belle famille, en souhaitant beaucoup de succès à tous ceux qui sont sur le chemin du savoir.

Table des matières

Liste des figures

Liste des tableaux

Introduction

Motivation et contexte de travail

Le domaine de la lisibilité et de la simplification de textes est en plein essor en traitement automatique des langues (TAL) (Collins-Thompson, 2014 ; François, 2015). Dans ce domaine, l'objectif principal est d'estimer le degré de difficulté d'un texte donné et d'en proposer une version simplifiée, adaptée aux besoins d'un public particulier : apprenants d'une langue, personnes avec peu d'instruction ou atteintes d'une pathologie du langage comme la dyslexie.

La dyslexie est un trouble du développement qui entraîne des difficultés de lecture et de compréhension des textes écrits. Ces difficultés se traduisent par un décodage lent et laborieux : plus les mots sont longs, plus le temps de lecture augmente et ceci de façon exponentielle (Ziegler *et al.*, 2003). Par conséquent, un enfant dyslexique lit en un an ce qu'un normo-lecteur lit en deux jours (Cunningham et Stanovich, 1998). Il s'agit d'un cercle vicieux parce que lire avec fluidité implique beaucoup d'entraînement et d'exposition aux textes écrits (Ziegler *et al.*, 2014). Par ailleurs, le manque de fluidité et la lenteur à lire nuisent à la compréhension des textes (les enfants sont plus concentrés sur le décodage des mots et des phrases) : cela a des conséquences au niveau du succès scolaire de l'enfant car la lecture est fondamentale pour tout ce que l'enfant doit apprendre pendant son parcours scolaire. La figure 0.1 illustre le cercle vicieux des troubles d'apprentissage (Vaivre-Douret et Tursz, 1999).

Ces dernières années, un nombre important de technologies ont été créées pour venir en aide aux personnes en difficulté [1]. Concrètement, pour le public dyslexique, les systèmes proposés intègrent des technologies de la parole (lecture « à voix haute », par exemple le système Gothit [2]) ou des aides visuelles (paramétrage et/ou mise en couleur des polices (Rello *et al.*, 2014) ou augmentation de l'espace entre lettres et lignes (Zorzi *et al.*, 2012)). Une autre solution possible, qui sollicite des techniques complexes en TAL, est l'application de transformations sur le contenu textuel.

1. Pour la lecture chez les dyslexiques (Sitbon *et al.*, 2010), pour la lecture chez les aphasiques ayant un trouble du langage (Carroll *et al.*, 1998) ou pour la lecture chez les adultes avec déficiences intellectuelles (Huenerfauth *et al.*, 2009).

2. *Ghotit Real Writer & Reader software* (http://www.ghotit.com).

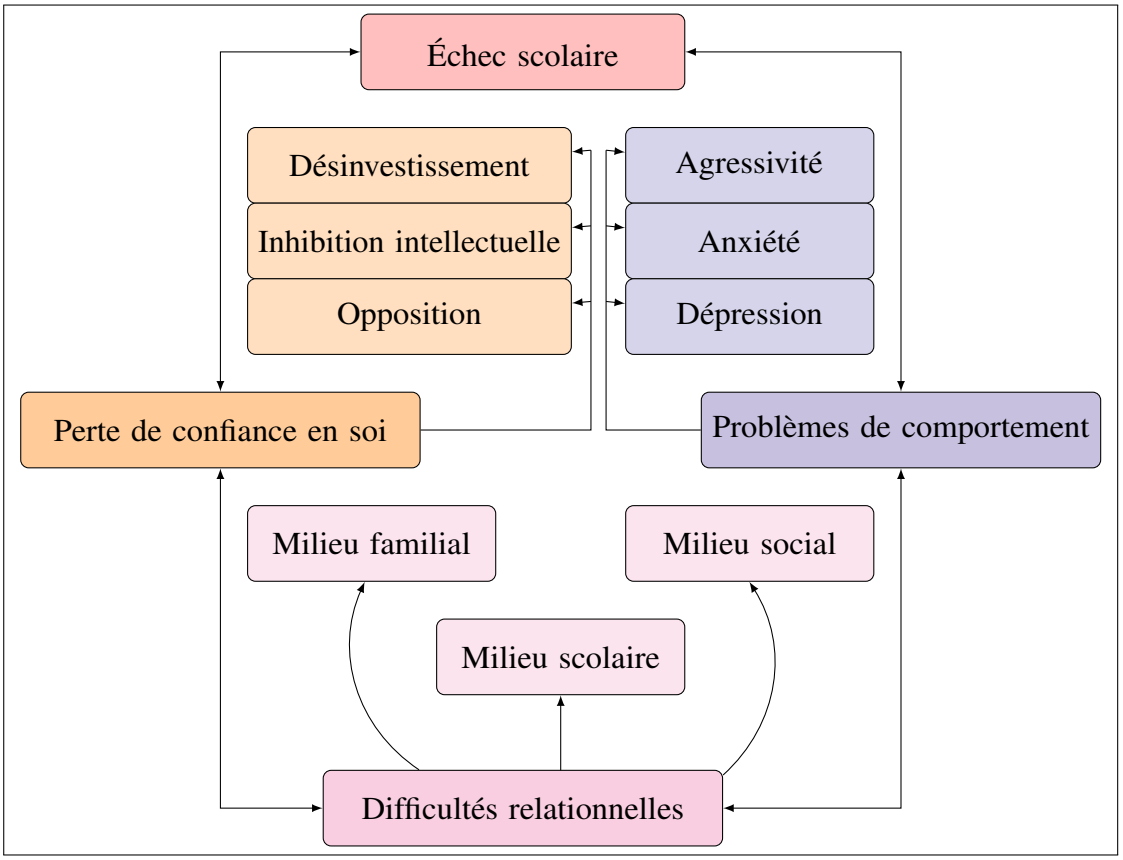


Figure 0.1. – Le cercle vicieux des troubles d'apprentissage

Parmi les travaux menés sur les transformations de textes, nous pouvons citer le travail de Sitbon et Bellot (2008) qui ont proposé une méthode d'adaptation des requêtes dans un système de recherche d'information basée sur la phonétique (l'inconsistance graphème-phonème est cruciale dans des langues comme le français où la différence entre langue orale et langue écrite est importante (Brunswick, 2010)). D'autre part, Rello *et al.* (2013) ont, quant à eux, mené plusieurs expériences pour l'espagnol qui montrent que la longueur des mots, ainsi que leur fréquence, sont décisives pour la lisibilité et la compréhension des textes chez les dyslexiques. Simplifier les mots des textes, ***tout en conservant le sens***, s'avère ainsi une approche prometteuse à laquelle nous nous sommes proposés de contribuer dans le cadre de cette thèse qui participe au projet ANR ALECTOR[3] (*A**ide à la LECT**ure pour amélioR**er l'accès aux documents pour enfants dyslexiques et faibles lecteurs*[4]). Ce projet aborde le défi de la simplification automatique de textes (SAT).

3. L'Agence Nationale de la Recherche finance le projet ALECTOR (http://www.agence-nationale-recherche.fr/Projet-ANR-16-CE28-0005).

4. https://alectorsite.wordpress.com

D'après Siddharthan (2014), la simplification est le processus de transformation d'un texte en un texte équivalent mais plus accessible à lire et à comprendre pour un public donné. Par exemple, les enfants dyslexiques représentent une cible très importante. D'après des rapports nationaux (MJENR [5] 2003) ou internationaux (PISA [6] 2009), 20% à 30% des élèves français ont des difficultés pour comprendre les textes écrits, 5 à 10% sont des enfants dyslexiques. Ces lecteurs sont en grande difficulté face à des textes complexes ou avec un vocabulaire peu courant.

La simplification automatique de textes est un domaine relativement récent en TAL (Gonzalez-Dios *et al.*, 2017 ; Paetzold et Specia, 2017 ; Shardlow, 2014 ; Siddharthan, 2014) et il y a encore peu de travaux menés pour le français (Brouwers *et al.*, 2014 ; Gala *et al.*, 2018 ; Seretan, 2012). En général, la simplification peut être mise en œuvre soit par l'ajout d'informations (reformulations, explications, définitions, *etc.*), soit par la réduction de la complexité linguistique. La plupart des systèmes de simplification existants se basent essentiellement sur cette dernière approche qui consiste à transformer le texte initial en un équivalent plus simple (Shardlow, 2014).

D'après Gala *et al.* (2018), il y aurait quatre niveaux de simplification de textes : (1) niveau lexical ; (2) niveau grammatical ; (3) niveau syntaxique ; et (4) niveau discursif. La figure 0.2 présente ces différents niveaux de simplification avec une brève description de chacun.

L'objectif de la simplification est que les lecteurs en difficulté puissent avoir la possibilité de s'entraîner à la lecture et ainsi améliorer leurs compétences, avant de se détacher progressivement des versions simplifiées de textes. En effet, la simplification de textes peut être vue comme une "béquille" permettant aux enfants dyslexiques et faibles lecteurs de s'entraîner à la lecture. La lecture de textes simplifiés améliore la qualité et la vitesse (avec un nombre réduit des mots mal lus) ainsi que la compréhension (Nandiegou et Reboul, 2018). Les textes mieux lus seront alors source de renouvellement de confiance chez ces lecteurs qui, de ce fait, retrouveront ou découvriront le plaisir de la lecture. Par ailleurs, il est à noter que la simplification n'appauvrit pas les textes : au contraire, elle en propose un substitut adapté aux difficultés sans en altérer le sens.

La simplification lexicale de textes représente l'une des étapes les plus importantes de la simplification. Elle peut mener à une amélioration de la vitesse de lecture. Cette amélioration est primordiale car les enfants en difficulté, lisant plus lentement que leurs camarades sans difficultés, peuvent avoir tendance à prendre du retard dans l'exécution des consignes ou la résolution d'un exercice. La simplification leur permettrait donc de réduire l'écart avec les normo-lecteurs, et ainsi les soulager de la pression liée à cet écart. La simplification lexicale vise à choisir des substituts équivalents généralement plus courts, plus fréquents et plus réguliers. Cependant, il s'avère que les substituts les plus courts sont aussi

5. *Le site du **M**inistre de la **J**eunesse, de l'**É**ducation **N**ationale et de la **R**echerche.*
6. ***P**rogramme **I**nternational pour le **S**uivi des **A**cquis des élèves.*

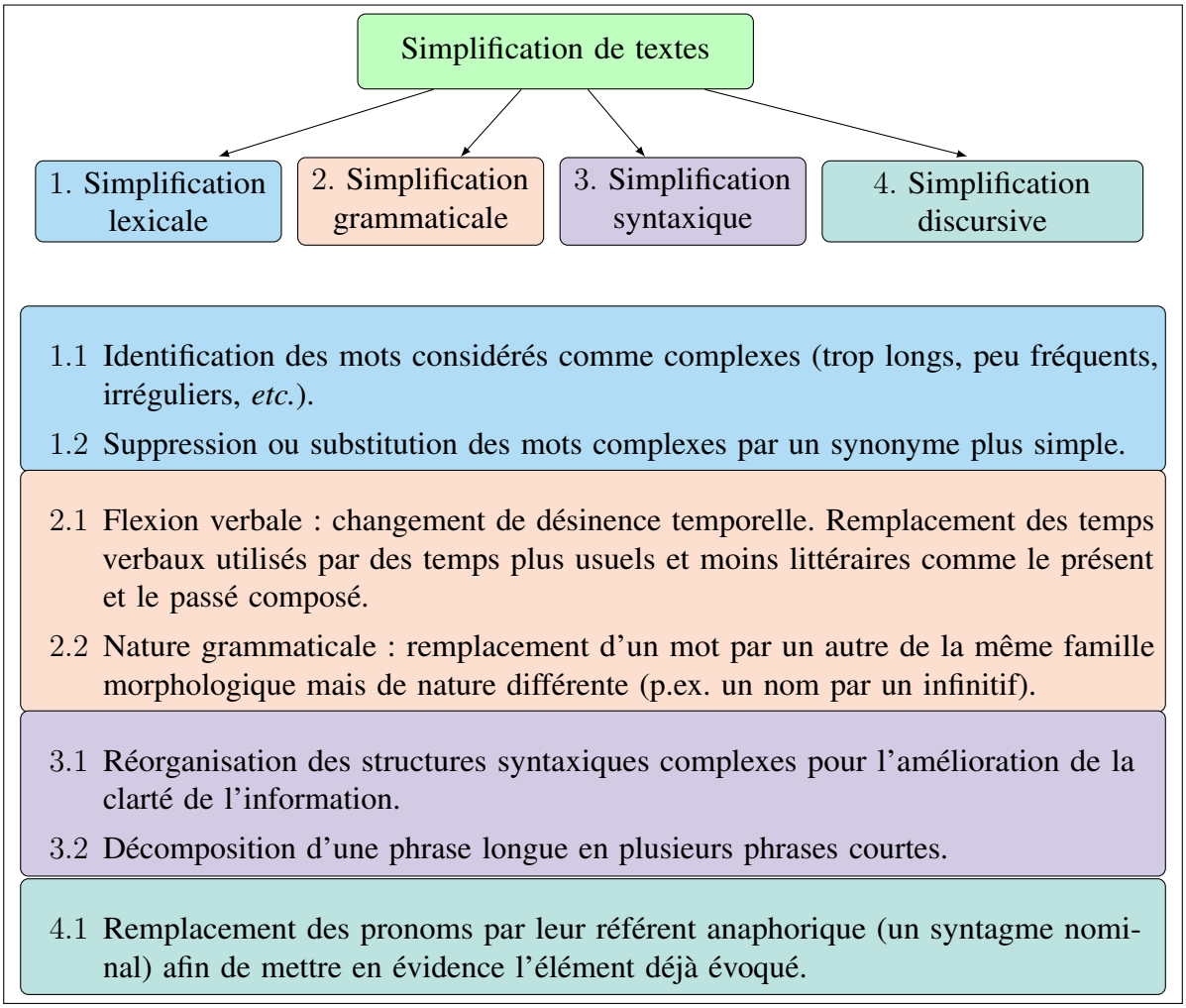


Figure 0.2. – Les différents niveaux de la simplification de textes

des mots ayant plusieurs sens. Par exemple, l'adjectif *glacial* correspond aux sens des mots décrits ci-dessous :

1. *froid*, *polaire*
2. *sec*, *insensible*
3. *inhospitalier*

Dans cet exemple et pour un texte donné, il est souhaitable de distinguer le sens propre ('basse température') (1), du sens figuré qualifiant une personne ou un comportement (2), ou du sens lieu, ambiance (3). Le mot *glacial* est un mot polysémique et son niveau de difficulté en termes de lecture et compréhension peut s'avérer supérieur par rapport au niveau de difficulté des mots *sec* et *froid*.

Désambiguïsation sémantique

Le processus de simplification lexicale se heurte à un problème de taille en TAL, celui de la **désambiguïsation sémantique** (indispensable à d'autres applications que nous décrivons ci-après). Il s'agit d'une tâche intermédiaire qui ne constitue pas une fin en soi, mais est indispensable à un niveau ou à un autre pour accomplir la plupart des tâches du TAL (Wilks et Stevenson, 1996). Elle consiste à sélectionner automatiquement le sens le plus approprié d'un mot en contexte (Ide et Véronis, 1998 ; Navigli, 2009). La désambiguïsation sémantique est essentielle pour l'amélioration de plusieurs applications (Kilgarriff, 1997 ; Navigli, 2009). Nous présentons, ci-dessous, celles ayant le plus d'intérêt à utiliser un système de désambiguïsation sémantique.

Traduction automatique (*Machine Translation* – MT)

La traduction automatique est la première des applications ayant considéré la désambiguïsation sémantique comme une tâche intermédiaire fondamentale (Weaver, 1949). Il s'agit donc d'un domaine de recherche par excellence où il est crucial de lever l'ambiguïté sémantique des mots afin d'aboutir à des traductions correctes (Carpuat et Wu, 2007 ; Vickrey *et al.*, 2005) . Par exemple, la traduction en anglais du mot français *glacial* est *icy* ou *bitter* selon s'il s'agit du froid ou d'une personne blessée (ou en colère).

Recherche d'information (*Information Retrieval* – IR)

Lever l'ambiguïté des mots d'une requête utilisateur fournie à un moteur de recherche peut permettre d'affiner le résultat retourné (Schütze et Pedersen, 1995 ; Zhong et Ng, 2012). Par exemple, si nous cherchons des textes traitant le sujet : 'les rayons laser', il faut ignorer les textes traitant les sujets suivants : 'les rayons de soleil', 'les rayons de magasin' ou encore 'les rayons de bicyclette'.

La plupart des moteurs de recherche n'utilisent pas explicitement la sémantique pour supprimer les documents, d'une base documentaire donnée, qui ne sont pas pertinents par rapport à une requête utilisateur. Concrètement, une désambiguïsation de tous les mots présents dans la base documentaire, associée à une éventuelle désambiguïsation des mots de la requête, permettrait d'éliminer les documents contenant les mêmes mots de la requête mais utilisés avec des significations différentes [7] et de retrouver des documents exprimant la même signification avec des libellés différents [8].

7. Cela a pour conséquence d'augmenter le nombre de documents pertinents retrouvés rapporté au nombre de documents total proposé par le moteur de recherche. Ce rapport fait référence en recherche d'information (IR) à la précision.

8. Cela a pour conséquence d'augmenter le nombre de documents pertinents retrouvés au regard du nombre de documents pertinents que possède la base documentaire. Ce rapport fait référence en recherche d'information (IR) au rappel.

Lexicographie

La désambiguïsation sémantique et la lexicographie (*i.e.*, la réalisation de dictionnaires) peuvent certainement bénéficier l'une de l'autre. D'une part, la désambiguïsation sémantique peut aider à fournir des groupements de sens empiriques et indices statistiques contextuels pour des nouveaux sens ou des sens existants, comme elle peut aider à créer de nouveaux dictionnaires plus lisibles (Richardson *et al.*, 1998). D'autre part, un lexicographe peut fournir de meilleurs inventaires de sens et des corpus annotés sémantiquement dont le bénéfice sera pour l'utilisation des méthodes de désambiguïsation sémantique.

Traitement de la parole (*Speech Processing – SP*)

La phonétisation correcte des mots en synthèse de la parole demande une tâche de désambiguïsation. Cette tâche est également utilisée en reconnaissance de la parole pour la segmentation des mots et pour la discrimination d'homophones. Ces derniers représentent des mots qui se prononcent de manière identique mais dont le sens est différent, par exemple *bar* et *barre*, *mer* et *maire* ou *auteur* et *hauteur*. Ferrand (1999) a décrit les caractéristiques de 640 homophones pour le français.

Substitution lexicale

La substitution lexicale est une tâche qui, ces dernières années, a reçu un intérêt majeur au sein de la communauté du TAL (Fabre *et al.*, 2014 ; McCarthy et Navigli, 2009). Le principe consiste à remplacer un mot-cible par un substitut potentiel tout en gardant le même sens du mot-cible par rapport au contexte dans lequel il apparaît.

La substitution lexicale reflète non seulement les capacités des systèmes de désambiguïsation sémantique à choisir le bon sens, mais peut également être utilisée pour comparer les ressources lexicales. Elle a le potentiel d'être elle-même bénéfique pour d'autres applications (par exemple, dans le cadre d'une simplification automatique de textes).

Positionnement et objectifs de ce travail de thèse

Le travail de cette thèse se positionne dans le cadre d'une recherche pluridisciplinaire en TAL, linguistique-informatique et psycholinguistique. Le but de ce travail est de contribuer à une étape essentielle de la simplification de textes : la simplification lexicale. En effet, un des *défis* les plus importants de cette étape est de pouvoir remplacer des mots par des équivalents plus accessibles d'un point de vue de leur compréhension, tout en conservant le même sens en contexte.

Nos objectifs s'inscrivent dans l'élaboration d'un outil de simplification automatique de textes destiné à un public d'enfants dyslexiques et faibles lecteurs. Il s'agit d'un dispositif d'aide à la lecture permettant de proposer semi-automatiquement

une version simplifiée d'un texte donné tout en gardant le même sens des mots. L'idée derrière cette simplification est bien évidemment que le texte soit plus accessible en termes de lecture et compréhension. Si nous reprenons l'exemple de l'adjectif '*glacial*', pour choisir un mot équivalent plus simple, il faut tenir compte du contexte dans lequel '*glacial*' apparaît. De ce fait, pour simplifier il est essentiel de lever d'abord l'ambiguïté des mots en tenant compte du contexte.

Selon le modèle de simplification lexicale proposé par SHARDLOW (2014), quatre étapes sont essentielles (*cf.* figure 0.3) :

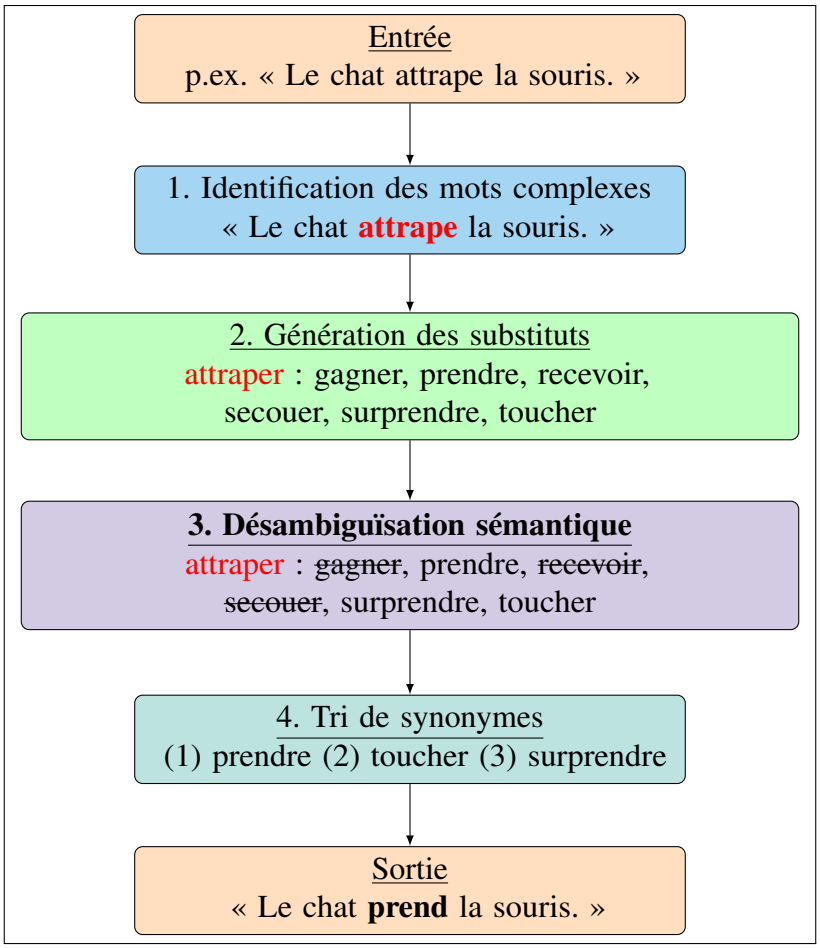


Figure 0.3. – Processus de simplification lexicale : application à la substitution du verbe complexe '*attraper*'

1. Identification des mots 'complexes' à remplacer dans un texte donné. Ces mots complexes sont des mots-cibles à remplacer par des substituts équivalents plus courts, plus fréquents et plus réguliers.
2. Génération d'une liste de synonymes pour chacun des mots-cibles.
3. Identification des synonymes correspondant au sens des mots-cibles.
4. Parmi la liste de synonymes, choix des mots 'simples' pour remplacer les mots-cibles.

Le deuxième point décrit une notion très importante, à savoir : *la synonymie*. En effet, il s'agit d'une relation lexicale sémantique d'équivalence entre signifiés. La synonymie exacte (ou absolue) étant rarissime, on considère comme synonymes deux unités lexicales ayant une « *valeur sémantique suffisamment proche pour que l'une puisse être utilisée à la place de l'autre pour exprimer sensiblement la même chose.* » (Polguère, 2002). Deux unités lexicales recouvrant (par inclusion ou intersection) la même notion sont donc des synonymes, par exemple *glacial* et *froid* dans le sens 'basse température' ou *glacial* et *inhospitalier* dans le sens 'lieu, ambiance'.

La figure 0.3, ci-dessus, présente une application du modèle de Shardlow (2014) sur la phrase : « *Le chat attrape la souris* ». Nous supposons que le verbe *attraper* est identifié comme étant complexe. Nous supposons aussi que la liste des synonymes du verbe est celle représentée dans la deuxième étape de la figure. La prochaine étape consiste à lever l'ambiguïté du verbe par rapport au contexte dans lequel il apparaît (ici, la phrase). Enfin, il ne reste plus qu'à remplacer le verbe *attraper* par un équivalent plus simple.

Dans le cadre de cette thèse, nous nous intéressons aux deux dernières étapes et essentiellement à la troisième, à savoir : la désambiguïsation sémantique. Concrètement, nos objectifs sont les suivants :

— Étude des méthodes de désambiguïsation sémantique existantes en TAL.

— Développement de différents modèles de désambiguïsation sémantique et identification du meilleur afin de choisir le bon sens en contexte parmi une liste de sens candidats. Cela permettrait ainsi de sélectionner un synonyme équivalent plus adapté et plus pertinent.

— Contribution à l'enrichissement d'une ressource lexicale proposant non seulement des synonymes désambiguïsés mais intégrant aussi des informations sur le niveau de difficulté de lecture et de compréhension d'un mot-sens.

— Développement d'une application mobile permettant une lecture de textes sur des tablettes afin de mener des tests de lecture pour étudier le bénéfice de la simplification lexicale auprès de lecteurs ayant des difficultés de lecture et de compréhension.

Organisation des chapitres

Ce travail de thèse est organisé en sept chapitres :

— Les deux premiers chapitres sont consacrés aux travaux d'état de l'art (chapitre 1 pour la désambiguïsation sémantique et chapitre 2 pour les approches utilisées pour mesurer la similarité sémantique entre mots et sens de mots, essentielles pour la désambiguïsation).

— Les chapitres 3 à 7 présentent nos contributions, notamment :

  — Le développement de méthodes de désambiguïsation sémantique à partir de différentes ressources lexico-sémantiques existantes.

  — La création d'une ressource lexicale pour le français, appelée ReSyf, permettant d'avoir des synonymes désambiguïsés et gradués en fonction de leur niveau de difficulté.

Ci-dessous, une brève description de chaque chapitre :

Le chapitre 1 est consacré à l'état de l'art en désambiguïsation sémantique. Nous présentons ici notre étude sur les types de sources de connaissances utilisés pour réaliser la désambiguïsation sémantique, à savoir : les ressources lexico-sémantiques et les corpus de données. Nous donnons un aperçu des approches de désambiguïsation et montrons l'importance de la représentation sémantique de mots et de sens pour lever l'ambiguïté des mots polysémiques.

Le chapitre 2 propose une classification des mesures de similarité sémantique et donne un aperçu des méthodologies d'évaluation de ces mesures.

Dans le chapitre 3, nous présentons nos premiers systèmes de désambiguïsation sémantique en nous basant sur des données provenant de corpus de données et de ressources lexico-sémantiques.

Le chapitre 4, quant à lui, propose une nouvelle approche pour la création de représentations sémantiques de mots et de sens. Ces représentations peuvent être utilisées pour la tâche de désambiguïsation sémantique.

Dans le chapitre 5, nous utilisons les représentations sémantiques, créées et validées dans le chapitre précédent, pour effectuer une nouvelle désambiguïsation sémantique. De nouveaux systèmes de désambiguïsation sont développés ici en utilisant nos propres représentations sémantiques.

Le chapitre 6 décrit une ressource lexicale pour le français proposant des synonymes désambiguïsés et gradués en fonction de leur niveau de difficulté en termes de lecture et compréhension. Cette ressource peut être utilisée pour la tâche de simplification lexicale et cela après avoir mené une désambiguïsation sémantique.

Dans le dernier chapitre (*cf.* chapitre 7), nous présentons une application Android que nous avons développée et qui fonctionne sur n'importe quel appareil mobile (smartphone ou tablette) ayant le système d'exploitation mobile Android.

Cette application présente des textes, phrase par phrase et permet le chronométrage et l'enregistrement de la lecture via le micro de l'appareil. Elle a été utilisée dans le cadre de l'étude du bénéfice de la simplification lexicale de textes auprès d'enfants présentant des troubles de la lecture (Nandiegou et Reboul, 2018).

Nous terminons ce mémoire avec un bilan du travail réalisé, avant d'aborder les différentes perspectives envisageables. Enfin, les différentes publications que nous avons réalisées durant cette thèse sont présentées à la fin de ce mémoire, après la bibliographie et les annexes.

Désambiguïsation sémantique : état de l'art

1.1. Description de la tâche de désambiguïsation sémantique

La désambiguïsation sémantique est l'une des tâches les plus populaires en TAL et en intelligence artificielle (IA). C'est une *tâche intermédiaire* qui ne constitue pas une fin en soi, mais est indispensable à un niveau ou à un autre pour accomplir la plupart des tâches du TAL (Wilks et Stevenson, 1996). Ainsi, la désambiguïsation sémantique suscite de l'intérêt depuis les premiers jours du TAL (Ide et Véronis, 1998). Elle est essentielle pour l'amélioration de plusieurs applications telles que la recherche d'information (Schütze et Pedersen, 1995 ; Zhong et Ng, 2012), la traduction automatique (Carpuat et Wu, 2007 ; Vickrey *et al.*, 2005) et la substitution lexicale (McCarthy et Navigli, 2009).

Schématiquement, il s'agit de choisir quel est le sens le plus approprié pour chaque mot d'un texte. La plupart des systèmes de désambiguïsation sémantique existants s'appuient sur deux grandes étapes (Navigli, 2009) : (1) représentation de l'ensemble des sens d'un mot ; et (2) choix du sens le plus proche du mot par rapport à son contexte. La première étape repose sur l'utilisation de ressources lexico-sémantiques telles que les dictionnaires ou les réseaux sémantiques. Ide et Véronis (1998) ont montré que la meilleure possibilité d'identifier le sens d'un mot ambigu est de se référer à son contexte.

L'annotation ou l'étiquetage sémantique des mots en sens nécessite une résolution de la polysémie en contexte (Audibert, 2003). La discrimination du sens des mots est une tâche centrale dans les applications du TAL. Cette tâche est repérée comme l'une des difficultés principales d'après Weaver (1949). Les difficultés de la désambiguïsation sont nombreuses, l'une, et non des moindres, reste l'établissement de la liste de sens pour chaque mot dont les propriétés distributionnelles permettent une utilisation dans le cadre de systèmes automatiques (Pierrel, 2000).

La désambiguïsation peut être de deux types :

(a) Désambiguïsation ciblée, seulement sur un mot particulier dans un texte.

(b) Désambiguïsation complète, pour tous les mots pleins d'un texte.

Les mots pleins, ou ce que nous appelons aussi mots à *classe ouverte*, peuvent être des noms, verbes, adjectives ou adverbes. Il y a deux critères importants pour choisir l'algorithme de désambiguïsation d'un mot donné :

(1) Mesure de similarité qui dépend des contraintes de la base de connaissances et du contexte applicatif.

(2) Temps d'exécution de l'algorithme.

Pour rédiger cet état de l'art, nous sommes partis de l'état de l'art proposé par Ide et Véronis (1998) pour les travaux antérieurs à 1998 et de l'état de l'art de Navigli (2009) auxquels nous avons incorporé les publications récentes dans le domaine.

Dans ce chapitre, après avoir présenté en section 1.2 les ressources utilisées comme sources de connaissances pour la représentation des sens de mots et l'aide à la désambiguïsation, nous nous intéressons aux approches de désambiguïsation sémantique existantes et les travaux réalisés (section 1.3). Nous présentons ensuite les méthodologies d'évaluation en section 1.4 avant de terminer par synthétiser les informations importantes dans la section *Conclusion* (*cf.* section 1.5).

1.2. Ressources utilisées comme sources de connaissances

Tous les systèmes de désambiguïsation sémantique, quelle que soit l'approche qu'ils adoptent, utilisent les connaissances présentes dans le contexte et dans les ressources lexico-sémantiques (Agirre et Martinez, 2001 ; Agirre et Stevenson, 2007). Dans les sous-sections suivantes, nous examinons brièvement ces sources de connaissances utilisées pour résoudre le problème de l'ambiguïté sémantique.

1.2.1. Ressources lexico-sémantiques

Les ressources lexico-sémantiques sont des sources de connaissances constituées d'informations produites indépendamment du contexte d'un mot-cible à désambiguïser. Elles peuvent être structurées comme des bases de données et des ontologies, non structurées comme des listes de mots, ou se trouver quelque part entre les deux. Les sources les plus utilisées sont les suivantes :

(a) Dictionnaires électroniques pour décrire en machine les dictionnaires traditionnels. Ces dictionnaires fournissent au minimum les catégories grammaticales possibles d'un mot donné ainsi que la définition (ou *glose*) de cha-

cun de ses sens. Pour ne citer que quelques uns, nous trouvons le TLFi [1], *Trésor de la Langue Française informatisé* (Dendien et Pierrel, 2003) et le Wiktionnaire [2] (Navarro *et al.*, 2009), un dictionnaire construit d'une manière collaborative.

(b) Thésaurus distributionnels décrivant des listes de référence lexicale. Ces thésaurus regroupent les mots en fonction des relations lexicales et sémantiques – le plus souvent la relation de *synonymie* (Ferret, 2014a).

(c) Réseaux lexico-sémantiques permettant de lier les concepts et leurs lexicalisations via une taxonomie des relations lexicales et sémantiques. Parmi ceux-ci, le plus connu est le WordNet de Princeton pour la langue anglaise [3] (Fellbaum, 1998), qui est décrit avec d'autres réseaux plus en détail ci-dessous. Ces réseaux peuvent fournir tout ou une partie des mêmes informations que les dictionnaires et les thésaurus ; ce qui les distingue est leur structure bien définie en tant que graphes dirigés ou non dirigés.

(d) Encyclopédies permettant de fournir de longues descriptions de textes pour les entrées lexicales mais peu d'informations linguistiques. L'encyclopédie la plus utilisée pour la désambiguïsation sémantique étant Wikipédia [4] (Mihalcea, 2006).

WordNet

WordNet est une ressource lexicale de large couverture, développée depuis plus de 30 ans pour la langue anglaise. WordNet est utilisable librement, y compris pour un usage commercial, ce qui en a favorisé une diffusion très large. Plusieurs autres ressources linguistiques ont été constituées (manuellement ou automatiquement) à partir de, en extension à, ou en complément à WordNet. Des programmes issus du monde de l'IA ont également établi des passerelles avec WordNet. La ressource WordNet pour la langue anglaise, ou PWN : Princeton WordNet (Fellbaum, 1998), est une base de données lexicale développée depuis 1985 par des linguistes du laboratoire des sciences cognitives de l'Université de Princeton. C'est un réseau sémantique, qui se fonde sur une théorie psychologique du langage.

WordNet décrit des termes, mots simples – *Single Words* ou expressions polylexicales – *Multiword Expression* (*MWE*), selon quatre catégories grammaticales différentes : *noms, adjectifs, adverbes* et *verbes*. Il est structuré en *synsets* (*concepts*). Chaque *synset* correspond à un ensemble de termes, que nous pouvons qualifier de synonymes entre eux, et représente un sens décrit par une définition. La première version diffusée de WordNet remonte à juin 1991. Son but est de répertorier, classifier et mettre en relation de diverses manières

1. http://atilf.atilf.fr
2. http://redac.univ-tlse2.fr/lexiques/wiktionaryx.html
3. http://wordnet.princeton.edu
4. https://fr.wikipedia.org/wiki/Wikipédia:Accueil_principal

le contenu sémantique et lexical de la langue anglaise. Le système se présente sous la forme d'une base de données électronique qu'on peut télécharger sur un système local. Des interfaces de programmation sont disponibles pour de nombreux langages. S'il n'est pas exempt de critiques (*granularité très fine, absence de relations paradigmatiques, etc.*), WordNet n'en reste pas moins l'une des ressources du TAL les plus populaires. Pour sa version 3.0, WordNet propose un ensemble de 117 658 *synsets* et 155 287 termes uniques [5].

Pour le français, l'EuroWordNet [6] (Vossen, 1998) constitue la première traduction française de WordNet. C'est une ressource d'une couverture limitée qui demande des améliorations significatives avant de pouvoir être utilisée (Jacquin *et al.*, 2007).

WOLF

WOLF : WordNet libre pour le français est une ressource inspirée du WordNet anglais de Princeton. WOLF représente une seconde traduction du WordNet pour le français. C'est une ressource initialement construite à l'aide de corpus parallèles (Sagot et Fišer, 2008) et étendue avec différentes techniques telles que la nominalisation d'événements (Apidianaki et Sagot, 2012) et la désambiguïsation de mots inter-langues (Hanoka et Sagot, 2012). La ressource WOLF est distribuée sous une licence libre compatible avec la LGPL (Lesser General Public License) et c'est aujourd'hui le WordNet français standard. Cette ressource est en cours de validation manuelle et est librement consultable [7]. WOLF contient l'intégralité des synsets du PWN d'abord en version 2 puis en version 3 bien que nombreux *synsets* restent vides dans WOLF. Cela permet d'avoir toute la structure arborescente des *synsets* du PWN dans WOLF. Actuellement, plus que la moitié des *synsets* est déjà décrite en français.

BabelNet

Navigli et Ponzetto (2012) ont proposé un réseau sémantique multilingue, nommé BabelNet [8], permettant de fournir des sens lexicographiques et des entités encyclopédiques. BabelNet a été créé en intégrant automatiquement la plus grande encyclopédie multilingue – c'est-à-dire Wikipédia – avec WordNet (Fellbaum, 1998). La construction de cette ressource s'est faite en deux grandes étapes : (1) mise en correspondance entre les pages de Wikipédia et les sens de WordNet ; et (2) système de traduction automatique, basé sur l'application de traduction en ligne de Google, pour recueillir une grande quantité de concepts multilingues et la compléter par les traductions manuellement éditées dans Wikipédia. La construction de BabelNet a permis de couvrir les sens manquants

5. http://wordnet.princeton.edu/wordnet/man/wnstats.7WN.html
6. http://projects.illc.uva.nl/EuroWordNet
7. https://gforge.inria.fr/projects/wolf
8. http://babelnet.org

dans WordNet. Le résultat est une ressource multilingue qui fournit des entrées lexicalisées multilingues, reliées entre elles avec une grande quantité de relations sémantiques.

De la même façon que WordNet, BabelNet regroupe les mots en différentes langues par groupes de synonymes appelés *Babel synsets*. Pour chaque *Babel synset*, BabelNet fournit des définitions textuelles en plusieurs langues, obtenues à partir de WordNet et Wikipédia. À l'heure actuelle, BabelNet intègre non seulement Wikipédia mais aussi plus de dix autres ressources, parmi lesquelles il y a Wiktionnaire, Wikidata, OmegaWiki et Open Multilingual WordNet, une collection de WordNets disponibles dans différentes langues. À la différence de WordNet qui offre une seule définition par sens, BabelNet permet d'offrir plusieurs définitions pour plusieurs langues.

La version 2.0 de BabelNet couvrait 50 langues, y compris toutes les langues européennes. Actuellement, la version 4.0 couvre 284 langues et contient près de 15.8 millions de *synsets*. Le réseau sémantique comprend toutes les relations lexico-sémantiques de WordNet (*hyperonymie et hyponymie, méronymie et holonymie, antonymie et synonymie, etc.*). Pour la langue française, BabelNet contient actuellement $4\ 141\ 338$ *synsets* et $5\ 301\ 989$ termes uniques dont $177\ 894$ termes sont polysémiques.

JeuxDeMots

Lafourcade (2007) a proposé JeuxDeMots, un réseau lexical contributif où les acteurs clés sont de *simples internautes* qui jouent à travers une interface présentée sous forme d'un jeu en ligne[9]. La base lexicale de ce réseau est en constante évolution, sa structure s'appuie sur les notions de nœuds et de relations entre nœuds. Chaque nœud représente une unité lexicale décrivant un terme. Les relations entre les nœuds sont typées et pondérées. Certaines de ces relations correspondent à des fonctions lexicales portant sur le vocabulaire lui-même (comme la relation d'*idée associée* et de *synonymie*) ou sur des relations sémantiques (comme la relation de *raffinement sémantique* qui décrit les sens possibles d'un terme ambigu, la relation d'*hyperonymie* évoquant des termes génériques et d'*hyponymie* évoquant des termes spécifiques). Il existe un autre type de relation décrit dans JeuxDeMots portant sur la prédiction de ce que peut faire un sujet ou ce qui peut être fait avec un objet.

La validation de la qualité des données collectées pour la construction de la base lexicale est fournie par les joueurs. Plus précisément, des relations proposées d'une manière anonyme par un joueur sont validées par d'autres joueurs, tout aussi anonymement. Les relations entre les unités lexicales sont pondérées. La pondération s'effectue de la façon suivante : plus une instance d'une relation est proposée, plus son poids est important, ceci tout en respectant certaines règles du jeu qui n'acceptent pas les relations taboues. Si nous nous référons

9. http://jeuxdemots.org

aux données collectées datant de Janvier 2018[10], la base lexicale contenait 180 390 253 instances de relations, 3 025 485 termes ayant au moins une relation sortante ($terme_A \rightarrow terme_B$) et 2 429 836 termes ayant au moins une relation entrante ($terme_A \leftarrow terme_B$).

Le tableau 1.1 liste toutes les ressources lexico-sémantiques présentées ci-dessus avec le nom des auteurs, la date de publication de la première version, le numéro de la dernière version et sa date de publication, le nombre de sens et le nombre de langues couvertes pour chaque ressource.

Ressource	**Auteur(s)**	**Date de publication**	**Dernière version**	**Nombre de sens**	**Nombre de langues**
WORDNET de Princeton FELLBAUM (1998)	*Christiane, Fellbaum*	Juin 1991	3.0 (Décembre 2006) et une version ultérieure 3.1	117 658 *synsets*	1 (anglais)
EUROWORDNET VOSSEN (1998)	*Piek, Vossen*	Mars 1996 (début du projet)	Juin 1999 (achèvement du projet)	15 132 : allemand ; 23 370 : espagnol ; 22 745 : français ; 40 428 : italien ; 7 678 : estonien ; 44 015 : néerlandais ; et 12 824 : tchèque	7 (allemand, espagnol, français, italien, estonien, néerlandais et tchèque)
WOLF SAGOT et FIŠER (2008)	*Benoît, Sagot et Darja, Fišer*	Mai 2008	1.0*b*4 (Janvier 2012)	59 091 *synsets* en français parmi 117 658 du WORDNET 3.0 pour l'anglais	1 (français)
BABELNET NAVIGLI et PONZETTO (2012)	*Roberto, Navigli et Simone Paolo, Ponzetto*	Décembre 2012	4.0 (Février 2018)	15 788 626 (*Babel synsets*)	284 (français inclus)
JEUXDEMOTS LAFOURCADE (2007)	*Mathieu, Lafourcade*	Juillet 2007	4.1.0 (la mise à jour de la base est mensuelle)	70 234 (raffinements sémantiques pour le français) [Janvier 2018]. Actuellement, seulement le français est maintenu	10 langues (anglais, arabe, français, espagnol, japonais, khmer, portugais, thaï, vietnamien et comorien)

Table 1.1. – Ressources lexico-sémantiques utilisées comme sources de connaissances

10. http://www.jeuxdemots.org/JDM-LEXICALNET-FR/?C=M;O=D

1.2.2. Corpus de données

Nous pouvons distinguer deux grands types de corpus de données utilisés pour la tâche de désambiguïsation sémantique : (a) corpus annotés sémantiquement en sens ; et (b) corpus non annotés (ou corpus bruts). Les corpus annotés en sens sont des textes dans lesquels certains mots pleins ou tous les mots pleins ont été annotés avec des étiquettes de sens provenant d'un inventaire de sens particulier. Ceux-ci comprennent SemCor (Miller *et al.*, 1993), Open Mind Word Expert (Mihalcea et Chklovski, 2003) et les différents corpus Senseval/SemEval décrits dans la sous-section 1.4.4. Les corpus bruts sont de grandes collections de documents qui manquent d'annotations en sens, bien que certains contiennent d'autres types d'annotation. Parmi ces corpus, il y a le BNC (British National Corpus) (Burnard, 2007), Europarl [11] (European Parliament Proceedings Parallel Corpus) (Koehn, 2005) et ANC (American National Corpus) (Ide et Suderman, 2004). D'autres, tels que le corpus WaCky (Baroni *et al.*, 2009), sont automatiquement récoltés depuis le Web.

1.3. Approches pour la désambiguïsation sémantique

Il existe plusieurs méthodes de désambiguïsation sémantique, deux catégories majoritaires peuvent être distinguées :

1. Les méthodes dirigées par les données (*cf.* sous-section 1.3.2), où l'on trouve les méthodes supervisées, semi-supervisées et non supervisées (Bakx, 2006 ; Navigli, 2009).
   (a) Les méthodes supervisées s'appuient sur un corpus d'apprentissage réunissant des exemples d'instances désambiguïsées de mots.
   (b) Les méthodes semi-supervisées, quant à elles, s'appuient sur un corpus de petite taille et permettent d'arriver à la construction d'un corpus de grande taille.
   (c) Les méthodes non supervisées exploitent les résultats de méthodes d'acquisition de sens d'une manière automatique.
2. Méthodes basées sur les connaissances (*cf.* sous-section 1.3.3), nécessitant une modélisation étendue des informations lexico-sémantiques ou encyclopédiques (Lafourcade, 2011 ; Navigli, 2009 ; Tchechmedjiev, 2012).

La figure 1.1 illustre ces différentes méthodes. Avant de pouvoir présenter en détail ces approches de désambiguïsation, il nous semble essentiel de décrire à quoi correspond la représentation sémantique des mots et des sens.

11. http://www.statmt.org/europarl

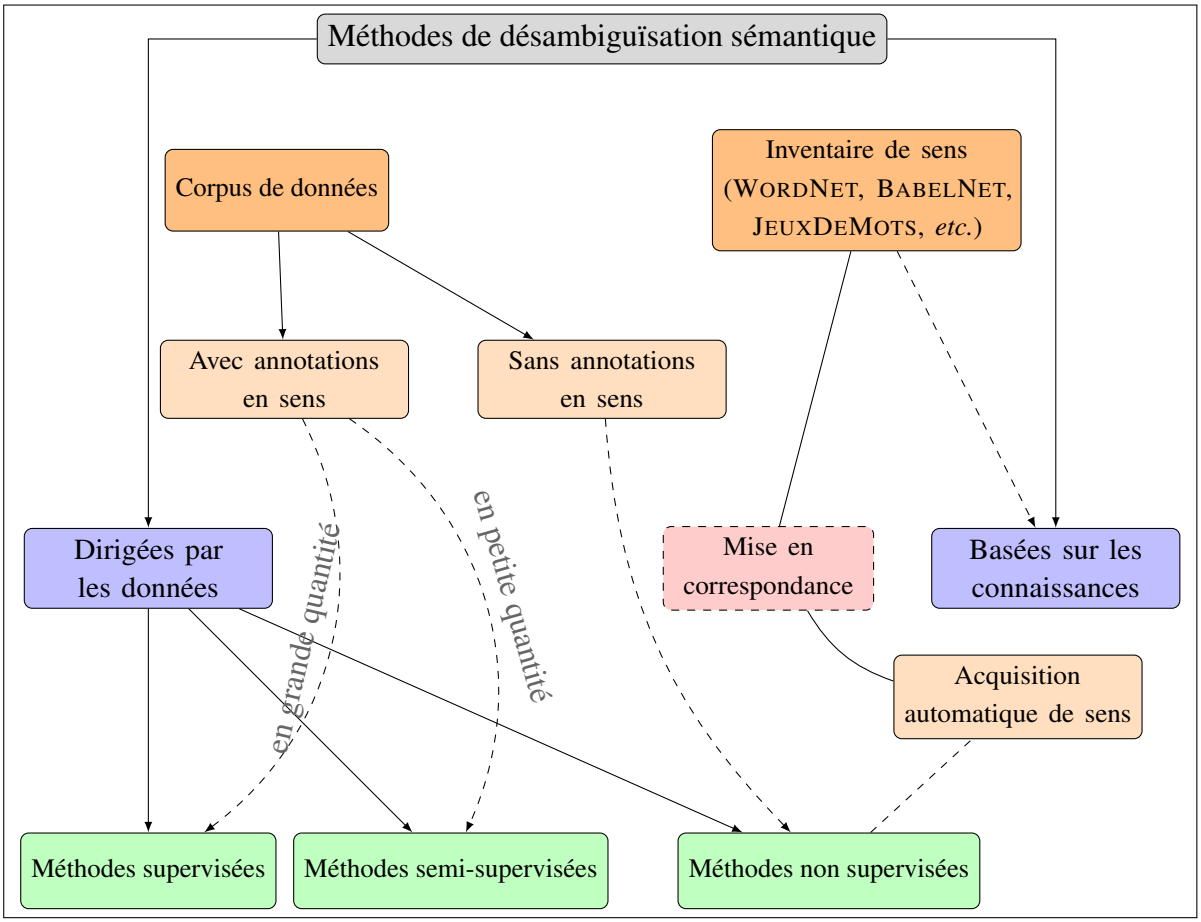


Figure 1.1. – Méthodes de désambiguïsation sémantique

1.3.1. Représentation sémantique de mots et de sens

Il y a deux grandes catégories d'approches pour construire des représentations sémantiques de mots et de sens : (1) à base de modèles distributionnels (Baroni *et al.*, 2014 ; Turney et Pantel, 2010) ; et (2) à base de ressources lexico-sémantiques (Camacho-Collados *et al.*, 2015 ; Wu et Giles, 2015 ; Zesch *et al.*, 2008). Par exemple, NASARI (Camacho-Collados *et al.*, 2015) est un modèle à base des ressources BabelNet et Wikipédia. Aussi, les *Word embeddings* ou ce que nous appelons « *plongements de mots* » (Mikolov *et al.*, 2013a ; Pennington *et al.*, 2014) sont des exemples de modèles distributionnels.

NASARI

Il s'agit d'une approche permettant la modélisation de concepts et d'entités nommées *via* l'attribution d'une représentation sémantique des sens de mots. NASARI (Camacho-Collados *et al.*, 2015), *a Novel Approach to a Semantically-*

Aware Representation of Items, est un modèle permettant de représenter les items lexicaux (mots ou sens) comme des vecteurs dans un espace sémantique. Il y a deux types de modèle NASARI : (a) à base de mots où les dimensions des vecteurs de représentation sémantique correspondent à des mots ; et (b) à base de sens où les dimensions correspondent à des sens. Le calcul des pondérations dans ces vecteurs repose sur l'utilisation de la spécificité lexicale de Lafon (1980), une mesure statistique utilisée principalement pour l'extraction de termes. NASARI utilise les correspondances (sens BabelNet, article Wikipédia) : les entrées de NASARI représentent l'identifiant d'un sens de BabelNet possédant une correspondance dans WordNet et le titre d'un article de Wikipédia s'il en existe un. Sachant que NASARI repose sur les données de Wikipédia, il ne propose des vecteurs sémantiques que pour les noms (la catégorie grammaticale la plus couverte par BabelNet).

Modèles distributionnels

À ce jour, les modèles distributionnels sont le paradigme prédominant pour la modélisation des mots. Ce paradigme repose sur l'hypothèse suivante : « les mots dont les distributions sont similaires sont sémantiquement proches » (Harris, 1954). Ces modèles visent à construire un vecteur pour chaque mot en fonction des différents contextes dans lesquels il peut apparaître, capturant généralement des informations sémantiques et syntaxiques de mots. Les techniques les plus connues reposent sur l'analyse statistique des données textuelles (ADT) en prenant en considération les cooccurrences pour la création des représentations vectorielles de mots. Les modèles classiques considèrent le contexte comme un sac de mots (Deerwester *et al.*, 1990 ; Salton *et al.*, 1975) tandis que les modèles les plus sophistiqués tiennent compte, par exemple, des dépendances syntaxiques (Lin, 1998b). Les poids dans les vecteurs à base de cooccurrences sont habituellement calculés à base de tf-idf, Term Frequency-Inverse Document Frequency (Jones, 1972) ou de la pmi, Pointwise Mutual Information (Evert, 2005). Le modèle d'espace vectoriel VSM (Vector Space Model) (Turney et Pantel, 2010), lui-même fondé sur les bases de la statistique des cooccurrences, est un modèle distributionnel pour lequel le poids associé à chaque dimension du vecteur indique la pertinence ou l'importance de cette dimension pour le mot en entrée.

Les représentations distributionnelles de mots ont été tout d'abord proposées par Rumelhart *et al.* (1988) et ont été utilisées avec succès dans les modèles de langage (Bengio *et al.*, 2006 ; Mnih et Hinton, 2009) ainsi que dans nombreuses tâches du TAL telles que l'apprentissage des représentations de mots (Mikolov, 2012), la reconnaissance d'entités nommées (Turian *et al.*, 2010), l'étiquetage de rôles sémantiques (Collobert *et al.*, 2011) ou la désambiguïsation sémantique (Camacho-Collados *et al.*, 2016). Les représentations distributionnelles sont très utiles pour les tâches du TAL car elles peuvent être utilisées

comme entrées entières d'algorithmes ou comme traits dans des applications telles que la recherche d'information (Manning *et al.*, 2008) et la classification de documents (Sebastiani, 2002). Le principal avantage est que les représentations de mots similaires sont proches dans l'espace vectoriel défini.

Word embeddings

Les dernières années ont vu une augmentation spectaculaire de la popularité des représentations vectorielles continues de mots, appelées *word embeddings* ou « *plongements de mots* », principalement en raison de leur capacité à capturer des informations sémantiques à partir de quantités massives de contenu textuel. En conséquence, de nombreuses tâches du TAL ont essayé de tirer parti du potentiel de ces modèles distributionnels, souvent appliqués avec succès (Baroni et Lenci, 2010 ; Mikolov *et al.*, 2013a ; Sahlgren, 2008). Il s'agit de projeter les mots selon un modèle de langage dans un espace dans lequel les relations sémantiques entre ces mots peuvent être observées ou mesurées.

Les *embeddings* représentent les mots comme ils peuvent représenter aussi les sens ou les textes dans un espace vectoriel continu à *n* dimensions. Les *embeddings* capturent des informations syntactico-sémantiques telles que les régularités de langage. Ces relations sont capturées par un vecteur de décalage d'une relation spécifique. La capacité des *embeddings* à capturer des connaissances a été exploitée dans plusieurs applications telles que la traduction automatique (Mikolov *et al.*, 2013c), l'analyse des sentiments (Socher *et al.*, 2013) et la désambiguisation sémantique (Chen *et al.*, 2014).

Un *embedding* est une représentation d'un objet topologique tel qu'un collecteur, un graphe ou un champ dans un certain espace de telle sorte que sa connectivité ou ses propriétés algébriques soient préservées (Insall *et al.*, 2018). Il a été tout d'abord présenté par Bengio *et al.* (2003). Un mot *embedding* peut se voir comme un mapping qui, grâce à la fonction de mise en correspondance $M : \{V \rightarrow \mathbb{R}^n : w \mapsto \vec{w}\}$, projette le mot w provenant du vocabulaire V dans un espace continu $\mathbb{R}$ à n dimensions ($\mathbb{R}^n$).

Contrairement aux techniques distributionnelles traditionnelles telles que l'analyse sémantique latente (LSA : Latent Semantic Analysis) présentée par Landauer et Dumais (1997) ou l'allocation latente de Dirichlet (LDA : Latent Dirichlet Allocation) présentée par Blei *et al.* (2003), Bengio *et al.* (2003) ont conçu un réseau de neurones à propagation avant (*feed-forward*) capable de prédire un mot w étant donné les mots précédents menant au mot w. Collobert et Weston (2008) ont présenté un modèle beaucoup plus profond composé de plusieurs couches pour l'extraction de traits, dans le but de construire une architecture générale pour les tâches du TAL. Une percée majeure est survenue lorsque Mikolov *et al.* (2013a) ont mis en avant un algorithme efficace pour l'entraînement des *embeddings* et qui se nomme Word2Vec [12].

12. https://code.google.com/archive/p/word2vec

WORD2VEC construit un réseau de neurones dont le but est de projeter les mots d'une langue (contenus dans une fenêtre sémantique définie) dans un espace de représentation vectorielle. Chaque mot est représenté par un vecteur, de taille modérée, qui correspond à une projection du mot dans un espace où les distances modélisent les relations inter-mots. Cette projection permet de tirer profit des mots selon leurs sens dans une région de l'espace sémantique proche. Par exemple, « *Paris* » et « *Londres* » partagent l'idée de « *capitale d'un Pays* ». Il y a deux types du modèle WORD2VEC : le premier repose sur une architecture fondée sur les sacs-de-mots continus (*continuous bag of words* ou CBOW), le deuxième repose sur une architecture fondée sur les SKIP–GRAM. Ces architectures sont manipulées par un réseau de neurones. Le modèle CBOW cherche à prédire un mot selon son contexte alors que le modèle SKIP–GRAM cherche à prédire un contexte sachant un mot.

Un modèle similaire à WORD2VEC a été présenté par PENNINGTON *et al.* (2014), développé à Stanford et qui se nomme GLOVE [13] (GLOBAL VECTORS FOR WORD REPRESENTATION), mais au lieu d'utiliser des traits latents pour représenter les mots, GLOVE fait une représentation explicite produite à partir du calcul statistique sur les occurrences de mots. GLOVE repose sur une utilisation de cooccurrences entre les termes (le nombre de fois qu'un terme apparaît en concomitance avec un autre). En croisant les probabilités de cooccurrences, il se veut capable de reproduire le même fonctionnement de WORD2VEC.

SCHNABEL *et al.* (2015) ont présenté une étude sur les méthodes d'évaluation des techniques d'*embeddings*. Dans les évaluations classiques, les résultats montrent des ordres différents de ces techniques d'*embeddings* remettant en question l'hypothèse qu'il existe une seule représentation vectorielle optimale. Ces évaluations se répartissent en deux catégories principales : l'évaluation extrinsèque et l'évaluation intrinsèque. Dans l'évaluation extrinsèque, les *word embeddings* sont utilisés en tant que traits d'entrée pour une tâche en aval afin de mesurer les changements dans les métriques de performance spécifiques à cette tâche. Les évaluations intrinsèques, quant à elles, testent directement les relations syntaxiques ou sémantiques entre les mots (BARONI *et al.*, 2014 ; MIKOLOV *et al.*, 2013b). Ces tâches impliquent généralement un ensemble présélectionné de mots pour une requête et de mots-cibles liés sémantiquement. Les évaluations sont effectuées en compilant un score agrégé, comme un coefficient de corrélation, pour mesurer la qualité des *embeddings*. SCHNABEL *et al.* (2015) ont proposé de nouvelles méthodes d'évaluation permettant de comparer directement les *embeddings* à des requêtes spécifiques. Ces nouvelles méthodes permettent de réduire les biais et fournissent des jugements pertinents de manière rapide et précise grâce au *crowdsourcing*, une technique d'enrichissement par la foule [14].

13. https://nlp.stanford.edu/projects/glove

14. Le *crowdsourcing* peut être représenté par un modèle de résolution de problèmes à l'ère d'Internet qui consiste, pour une personne ou une organisation, à lancer un appel à solutions auprès du public, les

1.3.2. Approches dirigées par les corpus de données

Après avoir décrit les différentes approches pour représenter les mots et les sens de mots, nous décrivons dans cette sous-section les différentes approches dirigées par les données pour la désambiguïsation sémantique (cf. figure 1.1).

Approches supervisées

Ces méthodes sont basées sur l'hypothèse que le contexte d'un mot polysémique peut fournir suffisamment de preuves pour sa désambiguïsation. Puisque l'annotation manuelle des occurrences de mots en sens est un processus difficile et long, connu sous le nom de *goulot d'étranglement de l'acquisition des connaissances* (Pilehvar et Navigli, 2014), les méthodes supervisées ne sont pas évolutives et nécessitent la répétition d'un effort comparable pour chaque nouvelle langue. Actuellement, les systèmes de désambiguïsation sémantique les plus performants sont basés sur un apprentissage supervisé.

Une annotation de mots d'un corpus avec des sens désambiguïsés provenant d'un inventaire de sens (par exemple, WordNet) est extrêmement coûteuse. À l'heure actuelle, très peu de corpus annotés sémantiquement sont disponibles pour l'anglais ; à notre connaissance, rien n'existe pour le français. Le consortium de données linguistiques (LDC : Linguistic Data Consortium [15]) a distribué un corpus contenant approximativement $200\ 000$ phrases en anglais issues du corpus *Brown* et *Wall Street Journal* dont toutes les occurrences de 191 lemmes ont été annotées avec WordNet (Ng et Lee, 1996). Le corpus SemCor (Miller *et al.*, 1993) reste le plus grand corpus annoté manuellement en sens (352 textes avec $234\ 136$ instances de sens de mots). Cependant, ces corpus contiennent peu de données pour être utilisés avec des méthodes statistiques. Ng (1997) estime que, pour obtenir un système de désambiguïsation à large couverture et de haute précision, nous avons probablement besoin d'un corpus d'environ $3,2$ millions d'instances de sens de mots. L'effort humain pour construire un tel corpus d'apprentissage peut être estimé à 27 années pour une annotation d'un mot par minute par personne (Edmonds, 2000). Il est clair qu'avec une telle ressource à portée de main, les systèmes supervisés seraient beaucoup plus performants. Plus récemment, Pasini et Camacho-Collados (2018) ont proposé un court survol sur les corpus annotés en sens en passant par ceux qui sont annotés manuellement, semi-automatiquement ou encore totalement d'une manière automatique.

Comme analysé par Lee et Ng (2002), les systèmes de désambiguïsation classiques utilisent généralement un ensemble fixe de traits pour modéliser le contexte d'un mot. Le premier trait est basé sur les mots entourant le mot-cible. Il s'agit généralement du contexte local sous la forme d'un tableau binaire, où

intéressés pouvant y répondre en présentant des propositions ou des plans de leur propre initiative, souvent après avoir collaboré à distance et en ligne avec d'autres personnes qui ont une idée semblable.

15. https://www.ldc.upenn.edu

chaque position représente l'occurrence d'un mot particulier. Les étiquettes de catégories grammaticales (POS : Parts of Speech) des mots voisins ont également été largement utilisées. Les collocations locales représentent un autre trait standard qui capture les séquences ordonnées de mots pouvant apparaître autour du mot-cible (Firth, 1957). Bien qu'elles ne soient pas très populaires, les relations syntaxiques ont également été étudiées en tant que traits syntaxiques (Stetina *et al.*, 1998).

Le système IMS (*It Makes Sense*) de Zhong et Ng (2010) est un bon représentant pour cette catégorie des méthodes de désambiguïsation. IMS fournit une plateforme extensible et flexible permettant l'utilisation non seulement de différents traits syntaxiques et sémantiques mais aussi des techniques de classification. Par défaut, IMS utilise trois ensembles de traits : (1) étiquettes POS des mots environnants, avec une fenêtre de trois mots de chaque côté, restreinte par la limite de la phrase ; (2) ensemble de mots qui apparaissent dans le contexte du mot-cible après suppression des mots outils ; et (3) collocations locales composées de 11 traits autour du mot-cible. IMS utilise une machine à vecteurs de support linéaire (SVM : Support Vector Machine) comme classificateur.

D'autres traits plus sophistiqués ont également été étudiés : les modèles sémantiques distributionnels, tels que l'analyse sémantique latente (Van de Cruys et Apidianaki, 2011), l'allocation latente de Dirichlet (Cai *et al.*, 2007) ainsi que les *word embeddings* (Iacobacci *et al.*, 2016 ; Rothe et Schütze, 2015 ; Taghipour et Ng, 2015 ; Zhong et Ng, 2010). Durant les dernières années, des efforts ont été faits pour tirer parti de l'intégration des *embeddings* afin d'améliorer les systèmes de désambiguïsation sémantique supervisés. Taghipour et Ng (2015) ont montré que les performances des systèmes supervisés conventionnels peuvent être améliorés en utilisant les *embeddings* comme de nouveaux traits. Dans la même direction, Rothe et Schütze (2015) ont entraîné des *embeddings* en mélangeant des mots et des sens, et en introduisant un ensemble de traits basés sur des calculs dans les représentations résultantes. Iacobacci *et al.* (2016) ont proposé des méthodes grâce auxquelles les *word embeddings* peuvent être exploités dans des systèmes état-de-l'art de désambiguïsation sémantique supervisée. Ils ont aussi effectué une analyse approfondie de la manière dont les différents paramètres de ces modèles affectent les performances des systèmes de désambiguïsation. Ils ont ainsi étudié les différentes techniques de combinaison des *embeddings*.

Approches semi-supervisées

Pour ces méthodes, un petit corpus annoté manuellement est généralement utilisé comme point de départ pour arriver à la création d'un corpus plus grand annoté sémantiquement. Des travaux basés sur ces méthodes ont été présentés par Mihalcea et Faruque (2004). Une deuxième option consiste à utiliser une approche à base de corpus bilingues alignés sur les mots, basée sur l'hy-

pothèse qu'un mot ambigu dans une langue pourrait être sans ambiguïté dans le contexte d'une seconde langue, contribuant ainsi à annoter le sens dans la première langue (Ng et Lee, 1996).

Des efforts ont été fournis pour annoter sémantiquement des corpus en utilisant des méthodes de *boostrapping*. Hearst (1991) a proposé un algorithme, *CatchWord*, pour une classification des noms qui comprend une phase d'apprentissage au cours de laquelle plusieurs occurrences de chaque nom sont manuellement annotées. Les informations statistiques extraites du contexte de ces occurrences sont ensuite utilisées pour lever l'ambiguïté d'autres occurrences. Si une autre occurrence peut être désambiguïsée avec certitude, le système acquiert automatiquement des informations statistiques de ces nouvelles occurrences désambiguïsées, améliorant ainsi ses connaissances progressivement. Hearst (1991) indique qu'une première série d'au moins 10 occurrences est nécessaire pour la procédure, et que 20 ou 30 occurrences sont nécessaires pour une haute précision.

Approches non supervisées

Ces méthodes sont basées sur l'hypothèse que les sens similaires se produisent dans des contextes similaires. Il est donc possible de regrouper les usages de mots en fonction de leur signification commune et d'induire des sens. Ces méthodes conduisent à la difficulté de mettre en correspondance les sens induits dans un inventaire de sens et elles nécessitent toujours une intervention manuelle afin d'effectuer une telle mise en correspondance. Pour ne citer que quelques exemples, ces méthodes ont été étudiées par Agirre *et al.* (2006), Brody et Lapata (2009), Manandhar *et al.* (2010), Van de Cruys et Apidianaki (2011) et Marco et Navigli (2013).

Reisinger et Mooney (2010) ont proposé un modèle à base d'un espace vectoriel multi-prototype permettant dans un premier temps de mettre les contextes de chaque mot dans des clusters et ensuite chaque cluster génère un vecteur prototype distinct pour un mot en faisant la moyenne sur tous les vecteurs de contexte dans le cluster. Huang *et al.* (2012) ont suivi cette idée mais ont introduit des vecteurs à distribution continue basés sur des modèles utilisant les réseaux de neurones. Ces deux modèles conduisent à une induction de sens non supervisée en regroupant des contextes de mots. Il reste difficile pour ces modèles de déterminer le nombre de clusters pour chaque mot, une limite qui n'existe pas lorsque nous utilisons une base de connaissances comme WordNet, BabelNet ou JeuxDeMots. Les modèles à base de clusters ne peuvent pas être utilisés d'une manière directe pour effectuer une désambiguïsation sémantique puisqu'il reste difficile de faire le lien entre un sens et un cluster.

1.3.3. Approches basées sur les ressources lexico-sémantiques

Ces méthodes fonctionnent indépendamment des données annotées dans les corpus et peuvent exploiter la structure des réseaux sémantiques pour identifier les significations les plus appropriées. Elles permettent d'obtenir une large couverture et une bonne performance en utilisant des connaissances structurées rivalisant ainsi les méthodes supervisées.

Les approches fondées sur les connaissances exploitent largement les ressources lexico-sémantiques. Cependant, il a été montré par Cuadros et Rigau (2006) que les quantités d'informations lexicales et sémantiques contenues dans de telles ressources sont généralement insuffisantes pour avoir de très hautes performances en désambiguïsation. De ce fait, beaucoup de travaux ont été proposés pour étendre automatiquement les ressources existantes. Par exemple, le travail de Suchanek *et al.* (2008) permettant d'inclure des liens de Wikipédia à WordNet afin d'intégrer une utilisation complète de l'heuristique du premier sens de WordNet et avoir une représentation plus riche pour ce sens. Ponzetto et Navigli (2009), quant à eux, ont proposé dans un premier temps une mise en correspondance à base de graphes entre les catégories de Wikipédia et les sens de WordNet. Ensuite, ils ont proposé une mise en correspondance intégrale entre les pages de Wikipédia et les sens de WordNet (Ponzetto et Navigli, 2010).

L'une des approches les plus classiques de cette catégorie consiste à estimer la proximité sémantique entre chaque sens candidat par rapport à chaque sens de chaque mot appartenant au contexte [16] du mot à désambiguïser. En d'autres termes, il s'agit de donner des scores locaux et de les propager au niveau global. Une application de cette méthode exhaustive est proposée par Pedersen *et al.* (2003). Nous pouvons imaginer la rapide explosion combinatoire (complexité exponentielle) que retourne cette approche exhaustive. Il est possible de se retrouver facilement avec un temps de calcul très long alors que le contexte qu'il s'agit d'utiliser est petit. Par exemple, pour une phrase de 10 mots avec 10 sens en moyenne, il y aurait 10^{10} combinaisons possibles (séquences de 10 sens, un sens pour chacun des 10 mots). Le calcul exhaustif est donc très compliqué à réaliser dans des conditions réelles et, surtout, rend impossible l'utilisation d'un contexte de taille importante. Pour diminuer le temps de calcul, il est possible d'utiliser une fenêtre autour du mot afin de réduire le temps d'exécution d'une combinaison mais le choix d'une fenêtre de taille quelconque peut mener à une perte de cohérence globale de la désambiguïsation. Plusieurs solutions, autres que la méthode exhaustive, ont été proposées. Par exemple, des approches à base de corpus pour diminuer le nombre de combinaisons à examiner comme la recherche des chaînes lexicales compatibles (Vasilescu *et al.*, 2004) ou en-

16. Il peut s'agir d'une phrase, d'un paragraphe ou d'un texte brut.

core des approches issues de l'intelligence artificielle comme le recuit simulé [17] (Cowie *et al.*, 1992) et les algorithmes à colonies de fourmis (Guinand et Lafourcade, 2010 ; Schwab *et al.*, 2011) ou encore les algorithmes génétiques (Gelbukh *et al.*, 2003). Tchechmedjiev (2012) fournit plus de détails pour ces méthodes.

Le contexte du mot à désambiguïser est délimité par une fenêtre textuelle qui se situe à gauche ou à droite ou des deux côtés et dont la taille peut varier. Les fenêtres peuvent être délimitées soit à l'aide de séparateurs de phrases ou de paragraphes, soit à l'aide de « n-grammes » qui permettent d'observer un certain nombre $(n-1)$ de mots entourant le mot polysémique dans le texte. La définition de la taille de la fenêtre textuelle est liée à celle de la distance optimale entre les mots ambigus et les indices contextuels pouvant servir à leur désambiguïsation (Audibert, 2007). Selon Yarowsky (1993), une grande fenêtre est nécessaire pour lever l'ambiguïté des noms alors que seulement une petite fenêtre suffit pour le cas des verbes ou des adjectifs. Dans un cadre d'analyse distributionnelle de données, plusieurs recherches sont faites sur la construction automatique de thésaurus à partir de cooccurrences de mots provenant d'un corpus de grande taille. Pour chaque mot-cible en entrée, une liste ordonnée de voisins les plus proches (*nearest neighbours*) lui est attribuée. Les voisins sont ordonnés en fonction de la similarité distributionnelle qu'ils ont avec le mot-cible. Lin (1998a) a proposé une méthode pour mesurer la similarité distributionnelle entre deux mots (un mot-cible et son voisin). McCarthy *et al.* (2004) ont proposé un modèle de désambiguïsation qui tient compte de l'utilisation des voisins distributionnels.

Plusieurs modèles de représentation sémantique supposent que chaque mot possède un seul vecteur sémantique. Ceci est généralement problématique car l'ambiguïté sémantique est omniprésente, ce qui est aussi le problème de la désambiguïsation sémantique. Chen *et al.* (2014) ont proposé un modèle unifié permettant à la fois une représentation et une désambiguïsation des sens de mots. Chaque sens a sa propre représentation. Ce modèle assume, d'une part, qu'une meilleure qualité de représentation des sens (WSR – Word Sense Representation) capture de riches informations permettant d'améliorer la désambiguïsation sémantique. D'autre part, une désambiguïsation d'une meilleure qualité permet de fournir des corpus fiables pouvant être utilisés pour l'apprentissage des représentations des sens. Le développement de ce modèle se réalise en trois grandes étapes : (1) initialisation des vecteurs de mots et vecteurs de sens ; (2) l'application d'un algorithme de désambiguïsation ; et (3) apprentissage des vecteurs de sens à partir d'occurrences pertinentes.

La première étape consiste à se servir d'un modèle neuronal Word2Vec de type Skip–gram entraîné sur un corpus de données textuelles pour apprendre des représentations vectorielles continues de mots. La représentation vectorielle des sens est basée sur les définitions (gloses de WordNet). Le vecteur de

17. Méthode d'optimisation stochastique classique fondée sur les principes physiques du refroidissement des métaux qui a été appliquée à la désambiguïsation.

chaque sens d'un mot-cible est le vecteur moyen après concaténation des vecteurs de mots de la définition. Le modèle ne prend que les mots pleins de la définition hors le mot-cible ayant un score de similarité positif et non nul avec le mot-cible.

La deuxième étape consiste à appliquer un des deux algorithmes de désambiguïsation proposés et qui sont à base de connaissances provenant de WordNet : (a) l'algorithme L2R (*left to right*) ou (b) l'algorithme S2C (*simple to complex*). La différence principale de ces deux algorithmes est dans l'ordre des mots. L'algorithme L2R désambiguïse les mots de gauche à droite dans l'ordre naturel de la phrase tandis que l'algorithme S2C désambiguïse les mots avec peu de sens en premier. L'avantage de l'utilisation de S2C est que la désambiguïsation des mots avec peu de sens peut être utile pour la désambiguïsation des autres mots. Comme la représentation d'un sens se construit à partir des mots de la définition, la représentation du contexte peut se faire de même à partir des vecteurs de mots. Les deux algorithmes reposent sur le principe suivant : chaque sens de mot possède un score et le sens ayant le meilleur score est celui retourné en sortie. Si la différence entre le score obtenu par le meilleur sens et le score du sens se trouvant en deuxième position est supérieure au seuil $\varepsilon = 0.1$, le vecteur du contexte est mis à jour en remplaçant le vecteur du mot polysémique traité par le vecteur du sens correspondant.

La troisième étape consiste à réentraîner le modèle Skip–gram sur le même corpus utilisé lors de l'étape 1 mais cette fois-ci pour apprendre à la fois des représentations de mots et de sens. La stratégie de mise à jour implémentée dans l'étape 2 est utilisée ici pour valider ou non l'existence d'un sens pour un mot polysémique donné.

Moro *et al.* (2014) ont développé un système de désambiguïsation, nommé Babelfy [18], à base de connaissances provenant du réseau sémantique BabelNet. Babelfy est un système état-de-l'art qui utilise l'intégralité de la structure de BabelNet. Cette structure inclut non seulement des sens lexicographiques mais aussi des entités encyclopédiques. En plus de la désambiguïsation des noms communs, verbes, adjectifs et adverbes, Babelfy permet la détection et la désambiguïsation d'entités nommées dans toutes les langues couvertes par BabelNet (271 langues lors de l'utilisation de la version 3.0 de BabelNet).

La tâche de désambiguïsation d'entités nommées (aussi appelée *Entity Linking*), et comme citée par Daher *et al.* (2017), est une tâche qui consiste à faire automatiquement le lien entre des entités trouvées dans un texte et des entités connues, présentes dans la base de connaissances utilisée (Ling *et al.*, 2015 ; Shen *et al.*, 2015). Par exemple, pour l'entité nommée *New York*, BabelNet dans sa version 4.0 propose plus de 20 sens différents. Parmi ces sens, on trouve : *Ville de New York, l'état de New York, titre d'une chanson, nom d'un album, nom de l'épisode d'une série télévisée, nom d'un magazine, etc.*

18. http://babelfy.org

1.4. Méthodologies d'évaluation

Après avoir décrit en section 1.3 les différentes approches pour la désambiguïsation sémantique, nous décrivons dans cette section les différentes méthodologies d'évaluation pour les systèmes de désambiguïsation sémantique.

Ces méthodologies se divisent en deux catégories. La première, connue sous le nom d'*évaluation extrinsèque*, *in vivo* ou *de bout en bout*, mesure la contribution du système à la performance globale de certaines applications plus larges, telles que la traduction automatique. La deuxième catégorie est celle des *évaluations intrinsèques* ou *in vitro*, où les systèmes sont testés en tant qu'applications autonomes à l'aide de repères spécialement construits. Les méthodes d'évaluation extrinsèques sont effectuées dans le contexte d'une application particulière du monde réel. Elles sont considérées comme une évaluation plus appropriée de l'utilité ultime d'un système (Edmonds et Kilgarriff, 2002). Dans l'évaluation intrinsèque, que nous présentons dans le reste de la section, les annotations de sens appliquées par les systèmes automatisés sont directement comparées aux annotations appliquées par les humains sur les mêmes données. Les évaluations intrinsèques sont populaires parce qu'elles sont faciles à définir et à mettre en œuvre (Palmer *et al.*, 2007). Cependant, elles ont un certain nombre d'inconvénients : elles sont liées à un inventaire de sens particulier ; un ensemble d'étiquettes de sens appliquées manuellement à partir d'un inventaire ne peut pas être utilisé pour évaluer les systèmes de désambiguïsation sémantique utilisant un inventaire différent. Les évaluations intrinsèques ont également été critiquées comme étant artificielles dans la mesure où leurs résultats peuvent ne pas correspondre aux performances du système dans des tâches réelles (Ide et Véronis, 1998).

Dans ce qui suit, nous présentons tout d'abord à quoi correspond un corpus de référence (*cf.* sous-section 1.4.1). Ensuite, nous décrivons les mesures pour évaluer les systèmes de désambiguïsation sémantique (*cf.* sous-section 1.4.2) et les systèmes *Baseline* (*cf.* sous-section 1.4.3) avant de terminer par décrire les campagnes d'évaluation existantes (*cf.* sous-section 1.4.4).

1.4.1. Corpus de référence

Un prérequis pour l'évaluation intrinsèque est un ensemble de données – *i.e.*, un corpus de textes dans lequel des annotateurs humains ont fournit des étiquettes de sens aux mots. Ces textes peuvent se représenter par un document unique ou une grande collection de phrases isolées ; le but est d'avoir un nombre suffisant d'occurrences de mots étiquetées (également connues sous le nom d'éléments, d'instances ou d'exemples) pour permettre une évaluation statistiquement significative. Les étiquettes de sens que les annotateurs humains appliquent aux éléments du corpus sont des renvois aux significations répertoriées dans un inventaire de sens particulier. Idéalement, une seule étiquette de sens

est appliquée à chaque élément, bien que de nombreux corpus permettent aux annotateurs d'appliquer plusieurs libellés de sens pour les cas où la distinction de sens n'est pas claire. Certains corpus fournissent également des étiquettes de sens spéciales pour marquer les instances dont la signification n'est pas fournie par l'inventaire de sens. L'évaluation peut varier quant à la façon de traiter ces étiquettes « non assignables ». Certains les traitent comme des étiquettes de sens ordinaires ; d'autres les excluent simplement et les notent comme des instances non attribuables.

1.4.2. Mesures d'évaluation

L'évaluation des systèmes de désambiguïsation sémantique est généralement effectuée en termes de mesures d'évaluation empruntées au domaine de la recherche d'information (PALMER *et al.*, 2007), que nous présentons ci-après.

Soit $M = (w_1, \ldots, w_n)$ l'ensemble de test de taille n décrivant les occurrences de mots polysémiques annotées en sens dans un corpus de référence ; soit A_r une fonction de référence permettant d'associer à chaque occurrence de mot $w_i \in M$ $(i \in \{1, \ldots, n\})$ le(s) sens le(s) plus approprié(s) à partir d'un inventaire de sens I – ***i.e.***, $A_r(w_i) \subseteq Sens_I(w_i)$; Soit $A_s \in Sens_I(w_i) \cup \varepsilon$ la fonction système qui permet d'associer à chaque occurrence de mot $w_i \in M$ le sens retourné automatiquement par un système de désambiguïsation sémantique. Lorsque le système ne fournit aucun sens, c'est la réponse ε qui est mise en sortie. Les mesures d'évaluation peuvent se décrire comme ci-dessous :

Couverture

Il s'agit du pourcentage d'éléments de l'ensemble de test pour lesquels le système de désambiguïsation a fourni des affectations de sens. La fonction C décrite dans l'équation 1.1 retourne la couverture.

$$C = \frac{\text{Affectations réalisées}}{\text{Total d'affectations à réaliser}} = \frac{|\{i \in \{1, \ldots, n\} : A_s(w_i) \neq \varepsilon\}|}{n} \tag{1.1}$$

Précision

Cette mesure permet de fournir la qualité des réponses données par le système de désambiguïsation. Il s'agit de déterminer le pourcentage d'éléments pour lesquels le système a fourni des affectations de sens correctes. La fonction P décrite dans l'équation 1.2 retourne la précision.

$$P = \frac{\text{Affectations correctes}}{\text{Affectations réalisées}} = \frac{|\{i \in \{1, \ldots, n\} : A_s(w_i) \in A_r(w_i)\}|}{|\{i \in \{1, \ldots, n\} : A_s(w_i) \neq \varepsilon\}|} \tag{1.2}$$

Rappel

Il s'agit du pourcentage d'éléments de l'ensemble de test pour lesquels le système de désambiguïsation a fourni des affectations de sens correctes. La fonction R décrite dans l'équation 1.3 retourne le rappel.

$$R = \frac{\text{Affectations correctes}}{\text{Total d'affectations à réaliser}} = \frac{|\{i \in \{1, \ldots, n\} : A_s(w_i) \in A_r(w_i)\}|}{n} \quad (1.3)$$

À partir des fonctions 1.2 et 1.3, nous avons $R \leqslant P$. Si $C = 100\%$ alors $P = R$. En désambiguïsation sémantique, le rappel fait référence au taux d'exactitude (*accuracy rate*) qui se détermine par la même mesure.

F–mesure

Il s'agit d'une mesure qui détermine la moyenne harmonique pondérée de la précision et du rappel. La fonction F_1 décrite dans l'équation 1.4 retourne cette moyenne.

$$F_1 = \frac{2PR}{P + R} \quad (1.4)$$

Il est à noter que $F_1 = P = R$ lorsque $P = R$.

1.4.3. Systèmes « *Baseline* »

Un test de performance pour tout algorithme de désambiguïsation se fait en comparaison par rapport à une baseline, par exemple la baseline du sens le plus fréquent dans un corpus (MFS : Most Frequent Sense) considérée comme la baseline la plus forte (Agirre et Edmonds, 2007) ou WFS (WordNet First Sense) pour décrire le premier sens dans le réseau sémantique WordNet (Miller *et al.*, 1990). Il n'est cependant pas facile de battre cette baseline. Plus clairement, si les chercheurs en désambiguïsation sémantique avaient accès aux valeurs du MFS, leur effort pour améliorer cette heuristique repousserait les frontières de la désambiguïsation. Cependant, pour obtenir les valeurs du MFS, c'est-à-dire le nombre d'occurrences de chaque sens pour chaque mot, il est nécessaire d'avoir à disposition un corpus annoté en une grande quantité de sens comme le corpus SemCor (Miller *et al.*, 1993) pour l'anglais. Il n'est pas possible pour la plupart des langues d'avoir un tel corpus même si des ressources sémantiques ayant le rôle d'un inventaire de sens sont disponibles. Créer des corpus de telle taille pour toutes les langues est très coûteux et irréalisable de nos jours, si on tient compte du temps à fournir et la somme d'argent nécessaire à investir.

Bhingardive *et al.* (2015) ont proposé une méthode non supervisée pour la détection du MFS à partir d'un corpus non annoté et cela en exploitant les *word*

embeddings. La méthode s'appelle UMFS–WE (Unsupervised Most Frequent Sense Detection using Word Embeddings). L'idée développée est de comparer le *word embedding* d'un mot-cible avec ses *sense embeddings*. C'est ainsi que le sens prédominant est celui ayant la similarité la plus élevée avec le mot-cible. WordNet a été utilisé comme inventaire de sens pour les expériences effectuées. Le vecteur du *sense embedding* est définit comme le vecteur moyen d'un ensemble de mots décrivant le sens. L'obtention de ce sac de mots a été réalisée par l'application de l'algorithme de Lesk étendu proposé par Banerjee et Pedersen (2003), que nous détaillons dans le chapitre 2, permettant de prendre non seulement les mots décrits dans le sens lui-même mais aussi les mots décrivant les sens reliés par une relation d'hyperonymie et d'hyponymie. Bhingardive *et al.* (2015) ont observé un gain de performance significatif par rapport au WFS pour la langue Hindou. Aussi pour la langue anglaise, ils ont amélioré la baseline MFS basée sur le corpus SemCor pour les mots dont la fréquence est supérieur à 2. La méthode qu'ils proposent est indépendante de la langue et du domaine utilisé.

McCarthy *et al.* (2007) ont proposé, quant à eux, une approche non supervisée pour trouver le sens prédominant à l'aide d'un thésaurus. Ils ont utilisé différentes mesures de similarité basées sur WordNet. Leur approche surpasse la baseline MFS basée sur SemCor pour les mots dont la fréquence d'apparition dans SemCor est inférieure à 5.

1.4.4. Campagnes d'évaluation

Plusieurs campagnes d'évaluation ont été organisées pour évaluer la performance des algorithmes de désambiguïsation : Senseval–1 (Kilgarriff et Rosenzweig, 2000), Senseval–2 (Edmonds, 2002), Senseval–3 (Mihalcea et Edmonds, 2004) pour l'anglais et RomansEval, désambiguïsation sémantique des sens pour des langues romanes telles que le français (Segond, 2000) et l'italien (Calzolari et Corazzari, 2000). La suite des travaux de désambiguïsation a été explorée dans des campagnes successives qui ont eu lieu tous les trois ans entre 1998 et 2010 et annuellement depuis 2012. Par exemple, SemEval–2007 (Navigli *et al.*, 2007), SemEval–2013 (Navigli *et al.*, 2013) et SemEval–2015 (Moro et Navigli, 2015). Pour chaque campagne d'évaluation, des corpus de données ont été fournis.

L'évaluation intrinsèque peut être utilisée pour deux variantes de désambiguïsation : *all-words disambiguation* (AWD), les systèmes sont censés fournir une annotation en sens pour chaque mot plein dans un texte donné. Dans l'autre variante, *lexical sample disambiguation* (LSD), les systèmes reçoivent un ensemble fixe de lemmes comme échantillon lexical et sont chargés de désambiguïser toutes leurs occurrences dans un document ou une collection de textes courts. AWD est la tâche de référence pour l'évaluation de la désambiguïsation sémantique, car elle nécessite un inventaire de sens ayant une large couverture

et un effort considérable pour produire l'ensemble de données annotées manuellement. Il est également plus difficile d'appliquer des méthodes de désambiguïsation supervisées à tous les scénarios possibles pour lesquels les mots polysémiques peuvent apparaître. Cela demande un nombre suffisant d'exemples annotés manuellement pour chaque mot. Cependant, AWD est une tâche plus naturelle qui lie les distributions de mots et de sens se trouvant dans des textes du monde réel. En revanche, la tâche LSD permet de produire plus facilement des données de test, car toutes les instances d'un lemme donné peuvent être étiquetées en même temps (désambiguïsation ciblée) plutôt que d'avoir des annotations séquentielles d'un mot au mot qui le suit. Comme les ensembles d'échantillons lexicaux contiennent généralement un nombre minimum d'occurrences par lemme, ils sont particulièrement adaptés aux systèmes de désambiguïsation supervisés. Pour la tâche LSD, il est courant de sélectionner les lemmes de manière à assurer une distribution particulière à travers la catégorie grammaticale, la fréquence des mots, la polysémie, le domaine ou d'autres caractéristiques d'intérêt.

L'un des obstacles majeurs d'une désambiguïsation sémantique pour atteindre de bons résultats est la granularité fine des inventaires de sens. Dans SENSEVAL–3, les systèmes ayant participé à la tâche *English All-Words* (EAW) ont atteint une performance autour de 65% (SNYDER et PALMER, 2004) avec une utilisation de WORDNET comme inventaire de sens. Une performance de $72,9\%$ a été obtenue sur la tâche *English Lexical Sample* (ELS). WORDNET est une ressource possédant une granularité fine dont la distinction des sens est difficile à reconnaître par les annotateurs humains (EDMONDS et KILGARRIFF, 2002).

Une désambiguïsation avec un inventaire de sens à granularité forte (ou plus optimale) a alors été proposée dans SEMEVAL–2007 sur les mêmes tâches de SENSEVAL–3 (EAW et ELS). Les résultats ont été meilleurs : $82 - 83\%$ pour EAW et $88,7\%$ pour ELS. Cela montre que la granularité de l'inventaire de sens a un impact décisif lorsque nous souhaitons atteindre des performances dans les $80 - 90\%$.

Le tableau 1.2 présente des statistiques sur les corpus de données pour la langue anglaise proposés dans SENSEVAL–1, SENSEVAL–2, SENSEVAL–3, SEMEVAL–2007 et SEMEVAL–2010. Pour la plupart de ces corpus, WORDNET est utilisé comme inventaire de sens. Pour SENSEVAL–1 et pour la tâche ELS, HECTOR (ATKINS, 1992) est utilisé comme inventaire de sens (un dictionnaire produit par des lexicographes de la *Presse Universitaire d'Oxford*). Pour SEMEVAL–2007 et pour la tâche ELS, la ressource ONTONOTES (HOVY *et al.*, 2006) est utilisée comme inventaire de sens. Pour ces campagnes, deux corpus ont été proposés pour la tâche ELS : un pour l'apprentissage et un pour le test. Le tableau 1.2 décrit pour chaque corpus le nombre total d'occurrences de mots annotés (*tokens*) et le nombre de mots uniques annotés (*types*).

Le tableau 1.3 présente des statistiques sur les différents ensembles de données multilingues proposés dans SEMEVAL–2013 et SEMEVAL–2015.

Tâche de désambiguïsation sémantique	Inventaire de sens	Corpus d'apprentissage		Corpus de test	
		Tokens	Types	Tokens	Types
SENEVAL–1 ELS	HECTOR	13 127	30	8 451	35
SENEVAL–2 EAW	WORDNET 1.7	–	–	2 473	> 1 082
SENEVAL–2 ELS	WORDNET 1.7	8 611	73	4 328	73
SENEVAL–3 EAW	WORDNET 1.7	–	–	2 041	> 960
SENEVAL–3 ELS	WORDNET 1.7	7 860	57	3 944	57
SEMEVAL–2007 EAW (granularité fine)	WORDNET 2.1	–	–	466	> 327
SEMEVAL–2007 ELS (granularité forte)	ONTONOTES	22 281	100	4 851	100
SEMEVAL–2007 EAW (granularité forte)	WORDNET 2.1	–	–	2 269	1 183
SEMEVAL–2010 EAW	WORDNET 3.0	–	–	1 632	8 157

Table 1.2. – Ensembles de données proposés pour la tâche de désambiguïsation sémantique monolingue, traitant la langue anglaise, dans Senseval/SemEval

Langue	Instances	Mots singuliers	Expressions polylexicales	Entités nommées	Nombre moyen de sens par instance	Nombre moyen de sens par lemme
	BABELNET (SEMEVAL–2013)					
Allemand	1 467	1 267	21	176	1.00	1.05
Anglais	1 931	1 604	127	200	1.02	1.09
Espagnol	1 481	1 103	129	249	1.15	1.19
Français	**1 656**	**1 389**	**89**	**176**	**1.05**	**1.15**
Italien	1 706	1 454	211	41	1.22	1.27
	WIKIPÉDIA (SEMEVAL–2013)					
Allemand	1 156	957	21	176	1.07	1.08
Anglais	1 242	945	102	195	1.15	1.16
Espagnol	1 103	758	107	248	1.11	1.10
Français	**1 039**	**790**	**72**	**175**	**1.18**	**1.14**
Italien	1 977	869	85	41	1.20	1.18
	WORDNET (SEMEVAL–2013)					
Anglais	1 644	1 502	85	57	1.01	1.10
	BABELNET (SEMEVAL–2015)					
Anglais	1 261	1 094	81	86	8.1	7.6
Espagnol	1 239	1 088	67	84	6.8	6.8
Italien	1 225	1 085	66	74	6.1	5.9

Table 1.3. – Ensembles de données proposés pour la tâche de désambiguïsation sémantique multilingue dans SemEval–2013 et SemEval–2015

Pour ces campagnes, seulement des corpus de test sont fournis. Trois inventaire de sens ont été utilisés, à savoir : BABELNET, WIKIPÉDIA et WORDNET.

Le tableau 1.3 décrit le nombre d'instances annotées manuellement pour chaque langue. Ces instances sont des termes représentant soit des mots simples, soit des expressions polylexicales. Il est à noter que seulement des noms (noms communs et entités nommées) ont été proposés comme instances à désambiguïser pour le corpus SemEval–2013 alors que le corpus SemEval–2015 propose une annotation sémantique manuelle pour l'ensemble des mots pleins. Aussi, un corpus en langue française est disponible seulement dans SemEval–2013 alors qu'il n'a pas été proposé pour la campagne qui a suivi. Dans le tableau 1.3, les deux dernières colonnes, présentant le nombre moyen de sens, décrivent le nombre moyen d'annotations en sens effectuées par les annotateurs.

1.5. Conclusion

Dans ce chapitre, nous avons formellement défini la tâche de désambiguïsation sémantique des sens de mots. Nous avons étudié les types de sources de connaissances utilisés pour réaliser cette tâche, à savoir : les ressources lexico-sémantiques et les corpus de données. Nous avons donné un aperçu des approches de désambiguïsation et montré l'importance de la représentation sémantique de mots et de sens. Nous avons vu que les méthodes supervisées nécessitent un corpus d'apprentissage rassemblant une grande quantité d'exemples. Les méthodes basées sur les connaissances, quant à elles, ne nécessitent pas d'avoir de tels corpus et cela n'empêche pas qu'elles soient compétitives vis-à-vis des méthodes supervisées. Enfin, nous avons décrit comment les systèmes de désambiguïsation sont évalués, en présentant les mesures utilisées pour évaluer leur performance ainsi que les corpus de référence disponibles et proposés durant les campagnes d'évaluation.

Dans le chapitre suivant, nous explorons les différentes approches utilisées pour mesurer la similarité sémantique entre mots et sens de mots. Comme nous l'avons mentionné au tout début de ce chapitre, la mesure de similarité sémantique est un critère important pour le choix de l'algorithme de désambiguïsation sémantique.

Mesures de similarité sémantique

2.1. Introduction

L'intelligence artificielle fédère de nombreux domaines scientifiques dans le but de développer des machines capables d'aider les humains à prendre des décisions pour l'exécution de traitements complexes. La plupart de ces traitements exigent des compétences cognitives élevées que ce soit pour des processus d'apprentissage ou des processus de décision. Pour le TAL, l'un des principaux objectifs de ces recherches est de donner aux machines la capacité de déterminer la relation sémantique qui existe entre des termes de la façon dont les êtres humains définissent cette relation.

Au cours des dernières années, de nombreux chercheurs de différents domaines ont développé et étudié la notion de mesure sémantique : il s'agit d'un outil mathématique utilisé principalement pour estimer la force de la relation sémantique entre les unités lexicales à travers des descriptions numériques portant leurs significations. Cette mesure peut se voir sous deux angles (Turney et Pantel, 2010) :

(a) Mesure de proximité sémantique proche de la *synonymie*. Par exemple, les mots *outil* et *instrument* sont des synonymes et partagent le même hyperonyme *matériel*.

(b) Mesure de proximité sémantique incluant, sans restrictions, différentes relations sémantiques (souvent appelée *semantic relatedness*). Par exemple, *outil* a pour hyperonyme *matériel* comme il a pour hyponyme *marteau*.

Pour le TAL, mesurer la similarité sémantique entre les mots est essentielle pour de nombreuses applications telles que la substitution lexicale (Biran *et al.*, 2011 ; Fabre *et al.*, 2014 ; McCarthy et Navigli, 2009) ou l'enrichissement sémantique de requêtes (Voorhees, 1994). De ce fait, elle a reçu un intérêt considérable qui a eu comme conséquence le développement d'une vaste gamme d'approches pour en déterminer une mesure. Ainsi, plusieurs types de mesures de similarité existent. Les mesures de similarité sémantique utilisent différentes représentations obtenues à partir d'informations provenant soit de ressources

lexico-sémantiques, soit de gros corpus de données ou bien des deux. Intuitivement, les mots *vert* et *couleur*, qui se réfèrent au sens COULEUR et dont il existe une relation sémantique d'*hyperonymie–hyponymie* entre les deux, ont une similarité sémantique plus forte que celle entre *vert* et *vers* même s'il existe dans ce cas-là une similarité formelle (graphique) très élevée. Les mesures de similarité sémantique sont essentielles non seulement pour les mots mais également pour les sens ou les textes. Elles peuvent être utilisées pour la désambiguïsation sémantique (NAVIGLI, 2009) ou l'alignement et l'intégration de différentes ressources lexico-sémantiques (MATUSCHEK et GUREVYCH, 2013). Pour les textes, elles permettent par exemple d'évaluer la qualité de sortie des systèmes de traduction automatique (LAVIE et DENKOWSKI, 2009) ou de recherche d'information (OTEGI *et al.*, 2015). Des travaux existants ont montré qu'il est possible d'ajuster ou d'étendre des approches utilisées pour un niveau de granularité à un autre. Par exemple, les mesures au niveau du mot ont été ajustées pour mesurer la similarité entre les textes (CORLEY et MIHALCEA, 2005) alors que les mesures au niveau du sens ont été étendues au niveau du mot en supposant que la similarité entre deux mots est celle de leurs sens les plus proches (BUDANITSKY et HIRST, 2006).

Dans ce chapitre, nous présentons tout d'abord dans la section 2.2 une classification des mesures de similarité sémantique existantes ; cette classification distingue les deux approches principales correspondant aux mesures à base de corpus et mesures à base de ressources lexico-sémantiques. Une troisième classe est ajoutée consistant à combiner les mesures provenant des deux précédentes classes. Nous nous intéressons ensuite aux méthodologies d'évaluation (*cf.* section 2.3) en présentant les listes de référence disponibles et les mesures d'évaluation avant de terminer par proposer, dans un résumé, les informations les plus importantes de ce chapitre (*cf.* conclusion, section 2.4).

2.2. Classification des mesures de similarité sémantique

Le but des mesures sémantiques est de capturer la force de l'interaction sémantique entre les mots, sens ou textes en fonction de leur description. Par exemple, les mots *fournaise* et *four* sont-ils plus sémantiquement liés que les mots *fournaise* et *instrument* ? La plupart des êtres humains seraient d'accord qu'ils le sont. Cela a été prouvé par exemple dans les expériences de JOUBARNE et INKPEN (2011) utilisant l'accord inter-annotateurs sur les évaluations de similarité sémantique.

La classification des mesures sémantiques tient compte de plusieurs aspects, parmi lesquels :

(a) Type d'éléments que la mesure vise à comparer (par exemple : mots, sens ou textes).

(b) Sources de connaissances à utiliser pour extraire la sémantique requise par la mesure.

(c) Hypothèses à prendre en compte lors de la comparaison.

(d) Forme canonique adoptée pour la représentation des éléments à comparer.

Nous pouvons classer les mesures de similarité sémantique en fonction des sources de connaissances à utiliser. Nous présentons tout d'abord dans la sous-section 2.2.1 les mesures à base de corpus. Ensuite, nous présentons dans la sous-section 2.2.2 les mesures basées sur des ressources lexico-sémantiques avant de terminer par celles pouvant utiliser les deux sources de connaissances (*cf.* sous-section 2.2.3).

La figure 2.1 illustre les différentes approches utilisées par les mesures sémantiques. Il est à noter qu'il existe deux points de vue possibles pour l'étude des éléments à comparer [1] : (1) démarche onomasiologique où on part des sens des termes vers les différentes réalisations de ces termes ; et (2) démarche sémasiologique où on part du signe des termes vers leur concept. Pour la première démarche, les éléments à comparer sont des éléments lexicaux, c'est-à-dire des termes, phrases ou textes. Nous pouvons inclure dans cette démarche les gloses (définitions) des sens puisqu'elles sont représentées comme des phrases. Pour la deuxième démarche, les éléments à comparer sont des concepts ou instances de concepts provenant de ressources structurées comme les ontologies ou les réseaux lexico-sémantiques. L'utilisation de la structure hiérarchique des réseaux lexico-sémantiques nous permet d'effectuer une comparaison entre les sens des termes.

2.2.1. Mesures à base de corpus

Les mesures sémantiques basées sur les corpus de données permettent la comparaison d'unités lexicales à partir de l'analyse de textes. Ces mesures reposent sur l'analyse statistique de l'usage des mots dans les textes, c'est-à-dire, l'analyse des occurrences de mots et les contextes dans lesquels ils se produisent. Ces mesures ne peuvent pas être réduites à des formules mathématiques uniques. Elles se réfèrent plutôt à des architectures complexes de traitements qui sont utilisées à la fois pour (a) extraire la sémantique des unités lexicales comparées ; et (b) comparer ces unités en analysant leur sémantique. Elles tirent parti d'une grande variété d'algorithmes, ce qui fait qu'elles représentent un vaste domaine d'étude.

Ces mesures sont souvent désignées comme des mesures distributionnelles (Mohammad et Hirst, 2012) puisqu'elles sont basées sur l'hypothèse distributionnelle de Harris (1954) : « les mots dont les distributions sont similaires

1. Les éléments à comparer peuvent être des termes, phrases, textes ou concepts.

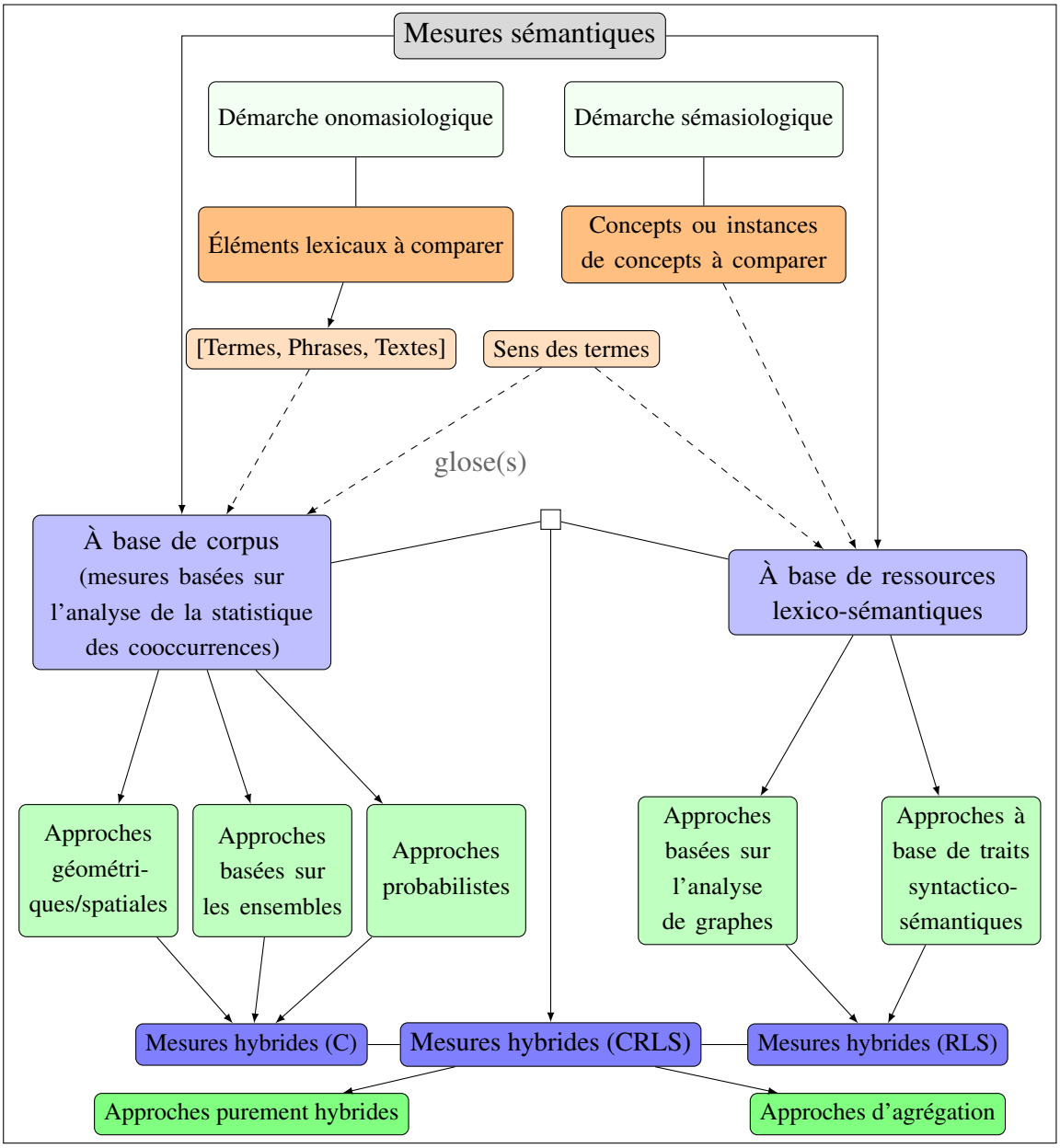


Figure 2.1. – Approches à base de corpus et de ressources lexico-sémantiques utilisées par les mesures sémantiques

sont sémantiquement proches », une hypothèse centrale de la sémantique distributionnelle. Cependant, il existe des mesures à base de corpus dont l'hypothèse distributionnelle n'est pas considérée comme la racine de l'approche. Par exemple, les mesures basées sur l'analyse des résultats fournis par les systèmes de recherche d'information. Mihalcea *et al.* (2006) et Panchenko (2013) ont proposé différentes classifications de ces mesures. Achananuparp *et al.* (2008) ont proposé un état de l'art sur les travaux associés.

Une grande partie de l'état de l'art liée aux mesures à base de corpus se concentre sur la comparaison de deux mots d'une paire donnée ; Des études approfondies ont été proposées par Curran (2003), Sahlgren (2008), Mohammad et Hirst (2012) et Panchenko (2013). Plusieurs contributions ont également été proposées pour comparer des paires de phrases ou de textes (Buscaldi *et al.*, 2013 ; Corley et Mihalcea, 2005 ; Ramage *et al.*, 2009), faisant ainsi référence à la tâche de similarité sémantique entre textes (STS, Semantic Textual Similarity). Cependant, la plupart des mesures pour comparer des phrases ou des textes sont des extensions de mesures définies principalement pour comparer des mots, ou reposent sur des approches qui sont également utilisées pour comparer des mots tels que l'analyse sémantique latente (Lintean *et al.*, 2010) ou l'allocation latente de Dirichlet (Blei *et al.*, 2003). D'autre part, Kamp *et al.* (2014) ont étudié la notion de compositionnalité qui est essentielle pour adapter les modèles à base de mots aux modèles à base de sens ou de textes.

Les représentations sémantiques des unités lexicales, comme nous les avons décrite dans le chapitre 1 (*cf.* sous-section 1.3.1), sont d'une importance majeure pour la définition des mesures sémantiques à base de corpus. Ces représentations sont des objets mathématiques décrivant, par exemple, des ensembles de mots, des vecteurs ou des distributions de probabilités. Elles proviennent du modèle sémantique développé à partir d'un corpus de données. Le modèle sémantique est la pièce maîtresse à partir de laquelle les représentations sont extraites. Par définition, l'utilisation d'une mesure sémantique consiste à comparer la représentation sémantique de deux éléments différents en se basant sur des formules mathématiques. Dans ce qui suit, nous ne nous intéressons pas à lister toutes les mesures à base de corpus proposées dans la littérature mais plutôt à présenter les notions centrales sur lesquelles s'appuient les mesures les plus utilisées. Les modèles à base de corpus s'appuient principalement sur le contexte dans lequel les occurrences de mots apparaissent. Le contexte est d'une importance majeure pour capturer le sens d'un mot à travers l'analyse des relations syntactico-sémantiques. En effet, la signification d'un mot est généralement considérée comme interprétable seulement dans le cadre de l'utilisation d'un contexte donné.

Afin de mesurer la similarité sémantique entre deux mots en se basant sur un modèle distributionnel, il existe trois principales approches décrites par Harispe *et al.* (2015) :

(a) Approche géométrique/spatiale permettant d'évaluer les positions relatives de deux mots dans l'espace sémantique défini par les vecteurs de contextes. Par exemple, la mesure Cosinus.

(b) Approche basée sur les ensembles permettant d'analyser le chevauchement de l'ensemble des contextes dans lesquels les mots apparaissent.

Par exemple, l'indice (ou coefficient) de Dice (DICE, 1945) et l'indice (ou coefficient) de Jaccard (JACCARD, 1901).

(c) Approche probabiliste basée sur des modèles probabilistes et des mesures proposées par la théorie de l'information. Par exemple, la mesure WEIGHTED OVERLAP (PILEHVAR *et al.*, 2013) et les mesures de LIN (1998a) ; LIN (1998b).

COSINUS

De base, COSINUS est une fonction trigonométrique qui dans un triangle rectangle permet de mesurer l'angle, le rapport de la longueur du côté adjacent par la longueur de l'hypoténuse. COSINUS est la mesure la plus utilisée actuellement pour mesurer la similarité dans un espace d'*embedding* (MIKOLOV *et al.*, 2013a). Elle permet de calculer la similarité entre deux vecteurs V_1 et V_2 en calculant le rapport entre le produit scalaire et la norme des deux vecteurs. La fonction Sim_{Cos} décrite dans l'équation 2.1 retourne COSINUS.

$$Sim_{Cos}(V_1, V_2) = \frac{V_1 \,.\, V_2}{\|V_1\| \, \|V_2\|} \tag{2.1}$$

Indice de Dice

DICE (1945) a proposé une mesure permettant de calculer la similarité entre deux ensembles de mots. Soient A et B deux ensembles finis, la fonction Sim_{Dice} décrite dans l'équation 2.2 retourne l'indice de Dice.

$$Sim_{Dice}(A, B) = \frac{2 \times |A \cap B|}{|A| + |B|} \tag{2.2}$$

$|A|$ et $|B|$ sont le nombre d'éléments des ensembles A et B respectivement et $|A \cap B|$ est le nombre d'éléments qui se trouvent à la fois dans les deux ensembles A et B. L'indice peut varier de 0 (quand A et B sont disjoints) à 1 (quand A et B sont égaux).

Indice de Jaccard

JACCARD (1901) a proposé une mesure de même type que la mesure de Dice, c'est-à-dire, applicable sur des ensembles d'éléments. L'indice de Jaccard consiste à déterminer le rapport entre la taille de l'intersection des ensembles considérés et la taille de l'union de ces ensembles. La fonction $Sim_{Jaccard}$ décrite dans l'équation 2.3 retourne l'indice de Jaccard.

$$Sim_{Jaccard}(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{2.3}$$

Weighted Overlap

Pilehvar *et al.* (2013) ont proposé une mesure de similarité, nommée Weighted Overlap (WO), permettant de calculer la similarité entre deux listes ordonnées en comparant le classement des dimensions. Nous supposons que les éléments de chaque liste sont classés selon leur poids d'importance, du plus fort vers le plus faible. Soit D l'ensemble des dimensions non nulles qui apparaissent à la fois dans les deux vecteurs V_1 et V_2. Soit $r_d(V)$ la fonction qui renvoie le rang de la dimension d dans le vecteur V. La fonction Sim_{WO} décrite dans l'équation 2.4 retourne WO.

$$Sim_{\text{WO}}\ (V_1, V_2) = \frac{\sum_{d \in D} \left(r_d(V_1) + r_d(V_2)\right)^{-1}}{\sum_{i=1}^{|D|} (2i)^{-1}} \tag{2.4}$$

Le dénominateur est un facteur de normalisation qui garantit une valeur maximale de 1. La fonction retourne une valeur minimale de 0, cette valeur se produit lorsqu'il n'y a pas de chevauchement entre les deux vecteurs, c'est-à-dire $|D| = 0$. Elle retourne la valeur de 1 lorsqu'il y a une parfaite correspondance au niveau du classement des dimensions partagées.

Mesure de Lin

Lin (1998a) a proposé une mesure de similarité entre mots en utilisant des dépendances syntaxiques extraites automatiquement après analyse d'un corpus de données. Ces dépendances syntaxiques devront être stockées et indexées. Cette méthode repose sur l'utilisation d'un ensemble de relations grammaticales de dépendances syntaxiques. L'ensemble peut représenter une liste définie de relations comme il peut représenter toutes les relations extraites lors de l'analyse du corpus. La mesure de Lin permet de mesurer le degré de cooccurrence entre deux mots. La seule condition pour que la mesure fonctionne est que les deux mots à comparer doivent partager la même partie du discours (POS ou catégorie grammaticale). Prenons l'exemple suivant : « *Le chat attrape la souris* » . Nous pouvons avoir les triplets [2] de dépendance syntaxique suivants : (*attraper, sujet, chat*), (*attraper, objet, souris*), (*chat, déterminant, le*) et (*souris, déterminant, la*). Nous pouvons voir les triplets comme des traits syntaxiques : pour le triplet (*attraper, sujet, chat*), le nom *chat* possède le trait syntaxique *sujet (attraper)*. La fonction Sim_{Lin} décrite dans l'équation 2.5 retourne la similarité distributionnelle de Lin entre deux mots w_1 et w_2.

$$Sim_{\text{Lin}}(w_1, w_2) = \frac{2 \times I\big(F(w_1) \cap F(w_2)\big)}{I\big(F(w_1)\big) + I\big(F(w_2)\big)} \tag{2.5}$$

$F(w_i)$ représente l'ensemble des traits syntaxiques du mot w_i. $F(w_i) \cap F(w_j)$

2. Un triplet de dépendance syntaxique se compose d'un gouverneur, d'un nom de la relation syntaxique et d'un dépendant.

décrit l'ensemble des traits syntaxiques partagés entre w_i et w_j. $I(S)$ est le flux d'information contenu dans les traits de S, avec $I(S) = -\sum_{f \in S} log\ P(f)$ où $P(f)$ est la probabilité estimée par le pourcentage des mots possédant le trait syntaxique f parmi l'ensemble des mots ayant la même catégorie grammaticale des mots w_1 et w_2. La mesure de LIN prend une valeur entre 0 et 1. Elle retourne 1 si w_1 et w_2 partagent les mêmes traits syntaxiques et retourne 0 si aucun trait syntaxique du mot w_1 n'est partagé avec les traits syntaxiques du mot w_2.

Même si les mesures sémantiques à base de corpus sont non supervisées et ne requièrent pas de connaissances provenant de ressources lexico-sémantiques, il est tout de même indispensable que les mots à comparer doivent se produire plusieurs fois dans le corpus. Aussi, les résultats obtenus dépendent directement de la qualité du corpus utilisé. D'autre part, il est tout à fait possible de combiner différentes mesures à base de corpus, ce à quoi nous faisons référence dans la figure 2.1 par mesures hybrides (C : Corpus).

2.2.2. Mesures basées sur des ressources lexico-sémantiques

Les mesures de cette catégorie utilisent explicitement les connaissances définies dans les ressources représentant non seulement des unités lexicales à comparer (mots ou sens de mots) mais aussi des concepts et leurs instances définis, par exemple, dans des ontologies. Ces mesures sont généralement utilisées pour exploiter des relations sémantiques non ambiguës ou des concepts définis dans des taxonomies. WORDNET est l'une des ressources les plus utilisées par ces mesures. Les réseaux sémantiques peuvent être considérés comme étant des graphes permettant ainsi de structurer des nœuds ayant des caractéristiques similaires dans des classes ordonnées. La figure 2.2 décrit un exemple de représentation à base de graphe tiré du réseau JEUXDEMOTS. Comme nous l'avons mentionné dans le chapitre 1 (*cf.* sous-section 1.2.1), JEUXDEMOTS propose différentes relations lexico-sémantiques.

Dans ce qui suit, nous présentons des algorithmes largement utilisés pour mesurer la similarité sémantique entre mots et sens ainsi qu'une fonction d'activation proposée par LAFOURCADE (2011) permettant d'utiliser les connaissances provenant de JEUXDEMOTS.

Algorithme de Lesk

Les premières approches de désambiguïsation sémantique à base de connaissances provenant de ressources lexico-sémantiques avaient tendance à utiliser des dictionnaires traditionnels comme inventaire de sens et étaient orientées vers la tâche de désambiguïsation d'un échantillon lexical. Cette tâche a été mentionnée dans le chapitre 1 (*cf.* sous-section 1.4.4). L'avènement des dictionnaires électroniques dans les années 1980 a conduit les premiers systèmes de

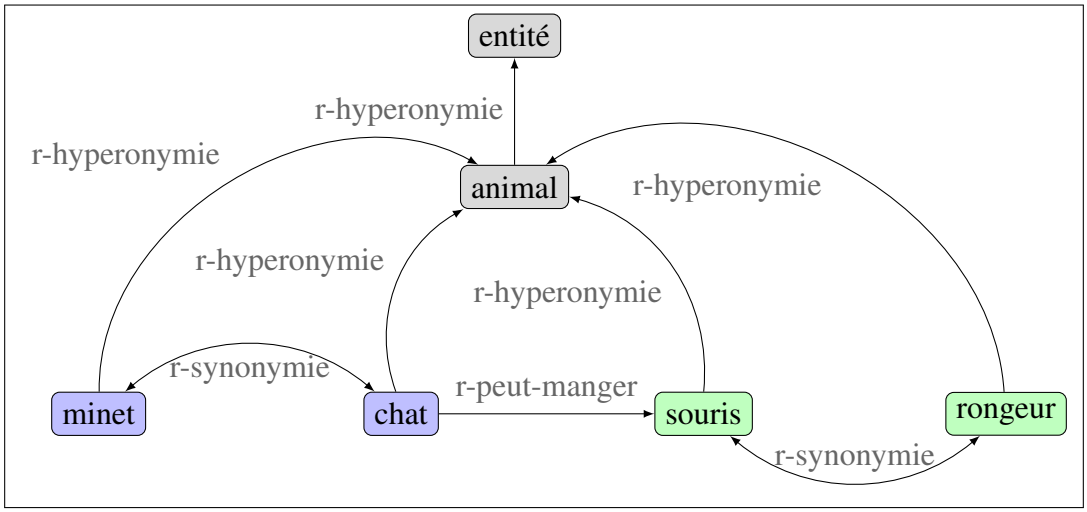


Figure 2.2. – Exemple de représentation à base de graphe de certains nœuds du réseau JeuxDeMots

désambiguïsation à traiter tous les mots polysémiques couverts par ces dictionnaires. Probablement, la première de ces tentatives fut celle de Lesk (1986), dont le système ne nécessite rien de plus que les mot(s) cible(s) en contexte et un dictionnaire électronique.

L'idée derrière cet algorithme est que deux mots se trouvant dans un même contexte peuvent être simultanément désambiguïsés. Étant donné la description (glose(s)) de chaque sens pour chaque mot du contexte, le principe consiste à trouver le maximum de chevauchement entre chaque combinaison de sens. Soit $T = \{w_1, \ldots, w_n\}$ une liste de mots pleins utilisée pour décrire un contexte, soit $D : L \rightarrow S$ une fonction dictionnaire permettant d'associer à chaque mot w appartenant au lexique L du dictionnaire un ensemble de sens candidats ($D(w) \subseteq S$). D'autre part, chaque sens $s \in S$ a une description $G(s) = (g_1, g_2, \ldots, g_m)$, qui comme T est aussi une liste de mots pleins. Les mots de l'ensemble $G(s)$ peuvent être considérés comme des traits sémantiques. La fonction $Lesk$ décrite dans l'équation 2.6 retourne le nombre de mots en commun entre deux sens.

$$Lesk(s_1, s_2) = \mid G(s_1) \cap G(s_2) \mid \tag{2.6}$$

Pour désambiguïser une paire de mots w_i et w_j, il suffit de trouver la paire de sens permettant d'avoir le maximum de chevauchement. La fonction Sim_{Lesk} décrite dans l'équation 2.7 retourne ce maximum.

$$Sim_{Lesk}(w_i, w_j) = \underset{s_k \in D(w_i), s_z \in D(w_j)}{\textbf{arg max}} \; Lesk(s_k, s_z) \tag{2.7}$$

La méthode s'adapte facilement pour désambiguïser plus de deux mots. Soit $T^{'} = \{t_1, \ldots, t_m\}$ un ensemble faisant partie de T ($T^{'} \subseteq T$). La fonction $Lesk\ (T^{'})$ décrite dans l'équation 2.8 retourne la meilleure combinaison de sens.

$$Lesk\ (T^{'}) = \underset{s_{t_1} \in D(t_1), \ldots, s_{t_m} \in D(t_m)}{\textbf{arg max}} \sum_{i=1}^{m} \sum_{\substack{j=1 \\ i \neq j}}^{m} Lesk(s_{t_i}, s_{t_j}) \tag{2.8}$$

Afin d'avoir une idée de comment fonctionne un algorithme de Lesk, prenons l'exemple de la phrase : « *Le chat attrape la* ***souris*** » où nous cherchons à désambiguïser le mot *souris*. Nous supposons que le dictionnaire utilisé par l'algorithme contient les entrées pour *souris, chat* et *attraper* comme décrit dans le tableau 2.1.

– souris	
Sens 1 :	Petit mammifère rongeur omnivore de la famille des Muridés.
Sens 2 :	Périphérique pour ordinateur, système de pointage.
– chat	
Sens 1 :	Mammifère carnivore félin de taille moyenne, au museau court et arrondi, domestiqué, apprivoisé ou encore à l'état sauvage.
Sens 2 :	Communication informelle entre plusieurs personnes sur le réseau Internet, par échange de messages affichés sur leurs écrans.
– attraper	
Sens 1 :	Prendre (un animal) à une trappe, à un piège ou à quelque chose de semblable.

Table 2.1. – Description des sens des mots *souris, chat* et *attraper*

L'algorithme a besoin d'au moins deux mots en entrée pour fonctionner. Cette condition est remplie puisque la phrase contient trois mots pleins, à savoir deux noms communs (*chat* et *souris*) et un verbe (*attraper*). Pour simplifier l'exemple, supposons que le dictionnaire dispose d'un seul sens pour le verbe *attraper*. L'algorithme compare les définitions pour chaque combinaison possible de sens du mot *souris* avec les mots *attraper* et *chat* : {$(souris_1, chat_1)$, $(souris_1, chat_2)$, $(souris_1, attraper_1)$, $(souris_2, chat_1)$, $(souris_2, chat_2)$, $(souris_2, attraper_1)$}. Parmi ces couples, $chat_1$ et $souris_1$ partagent comme mots pleins le mot *mammifère* qui reste le seul point en commun vis-à-vis des autres combinaisons. Dans cet exemple, le seul sens du verbe *attraper* ne partage pas de mots avec les sens de *souris*. Au final, la similarité entre *souris (animal)* et *chat (animal)* est plus forte que la similarité entre *souris (périphérique)* et *chat (communication)*.

Bien que l'implémentation de l'algorithme reste simple, il a rapporté une précision de 50 à 70% dans les campagnes d'évaluation SENSEVAL/SEMEVAL (*cf.* sous-section 1.4.4, chapitre 1). Après sa proposition par LESK (1986), l'algo-

rithme a connu une large palette de variantes qui ont été utilisées comme référence pour une comparaison avec des systèmes plus sophistiqués. Cependant, des imprécisions dans la version originale de l'algorithme laissent indéterminés des détails aussi importants. Par exemple, la comptabilisation de plusieurs occurrences d'un même mot, la lemmatisation des gloses, *etc*. De ce fait, il est probable d'avoir deux implémentations de l'algorithme qui ne fonctionnent pas exactement de la même manière.

Variante de Lesk simplifié

Kilgarriff et Rosenzweig (2000) ont proposé une variante permettant de comparer chaque sens candidat d'un mot-cible w_c directement avec le contexte dans lequel il apparaît. Cette variante consiste à analyser le chevauchement entre le contexte (excepté w_c) et chaque définition de sens candidat. La fonction $Lesk_{w_c}$ décrite dans l'équation 2.9 retourne le score pour cette variante.

$$Lesk_{w_c} = \underset{s_k \in D(w_c)}{\textbf{arg max}} \mid G(s_k) \cap T \mid \tag{2.9}$$

Il est à noter que l'algorithme de Lesk est très sensible aux mots présents dans les définitions. Une absence des mots importants dans les définitions rend l'algorithme incapable de lever l'ambiguïté. Plusieurs améliorations ont été proposées pour remédier à ce problème en cherchant soit à ajouter, si cela est possible, les exemples proposés avec les sens (Kilgarriff et Rosenzweig, 2000) ou plutôt à étendre l'algorithme en ajoutant les définitions des sens reliés par des relations sémantiques (Banerjee et Pedersen, 2003).

Kilgarriff et Rosenzweig (2000) ont trouvé que l'inclusion des exemples de sens pour la variante de Lesk simplifié conduit à des performances significativement meilleures que l'utilisation seulement des définitions de sens. Soit $E = \{e_1, \ldots, e_n\}$ un ensemble d'exemples associé à un sens s. Les fonctions 2.6, 2.7 et 2.9 deviennent respectivement 2.10, 2.11 et 2.12.

$$Lesk^{'}(s_1, s_2) = \mid (G(s_1) \cup E(s_1)) \cap (G(s_2) \cup E(s_2)) \mid \tag{2.10}$$

$$Sim_{Lesk'}(w_i, w_j) = \underset{s_k \in D(w_i), s_z \in D(w_j)}{\textbf{arg max}} Lesk^{'}(s_k, s_z) \tag{2.11}$$

$$Lesk^{'}_{w_c} = \underset{s_k \in D(w_c)}{\textbf{arg max}} \mid (G(s_k) \cup E(s_k)) \cap T \mid \tag{2.12}$$

Algorithme de Lesk étendu

Une autre variante de l'algorithme de Lesk a été proposée par Banerjee et Pedersen (2003) qui ont observé que, là où il existe une ressource lexico-sémantique permettant de fournir des relations entre sens, celles-ci peuvent être utilisées pour augmenter le nombre de définitions avec celles des sens associés.

Ils ont utilisé WordNet comme inventaire de sens. Soit R un ensemble de relations y compris la description du sens (G) : $R = \{Glose, Hyperonymie, Hyponymie, Meronymie, Holonymie\}$ [3]. La fonction $Lesk_{Étendu}$ décrite dans l'équation 2.13 retourne le score de Lesk étendu entre deux sens selon l'ensemble de relations R. Pour que la fonction soit symétrique, une paire de relations (r_1, r_2) est conservée seulement et seulement si la paire inverse (r_2, r_1) existe dans la ressource. L'ensemble R comme présenté par Banerjee et Pedersen (2003) n'est pas obligatoirement fixe, ce qui peut créer d'autres variantes.

$$Lesk_{Étendu}(s_1, s_2) = \sum_{(r_1,r_2)\in R^2} | G(r_1(s_1)) \cap G(r_2(s_2)) | \tag{2.13}$$

Dans cet algorithme, la manière de calculer le recouvrement entre les mots des définitions est différente. Le calcul est basé sur le principe relevé par la loi de Zipf (1949), qui met en évidence une relation quadratique entre la longueur d'une phrase et sa fréquence d'occurrence dans un corpus. De ce fait, n mots qui apparaissent l'un à côté de l'autre dans une séquence portent plus d'informations que s'ils étaient séparés.

D'autre part, Ponzetto et Navigli (2010) ont combiné l'algorithme de Lesk étendu avec la variante de Lesk simplifié pour avoir un algorithme simplifié étendu. Dans ce dernier, les définitions sont concaténées et comparées directement avec le contexte du mot-cible.

Mesure de similarité globale pour la désambiguïsation sémantique

Navigli (2009) a proposé une mesure globale qui s'utilise pour lever l'ambiguïté d'un mot polysémique.

Si w_c est le mot-cible à désambiguïser et T est l'ensemble des mots du contexte dans lequel apparaît le mot w_c alors la fonction $\hat{S}$ décrite dans l'équation 2.14 retourne le score du sens candidat retourné.

$$\hat{S} = \underset{s\in S(w_c)}{\arg\max} \sum_{\substack{w_i\in T \\ w_i\neq w_c}} \text{MAX } Score(s, s^{'}) \tag{2.14}$$

- $S(w_i)$ est l'ensemble des sens du mot w_i.
- $s^{'} \in S(w_i)$ $avec$ $i = 1 \ldots n$ et $i \neq c$.
- $S(w_c)$ est l'ensemble des sens du mot-cible w_c.
- $Score(s, s^{'})$ est la fonction utilisée pour mesurer la similarité entre les deux sens s et $s^{'}$.

3. Une description détaillée de ces relations est disponible sur le site du WordNet de Princeton : https://wordnet.princeton.edu/man/wngloss.7WN.html#sect4.

Fonction d'activation

Lafourcade (2011) a proposé une fonction à base du réseau lexical JeuxDeMots permettant de mesurer la similarité entre deux nœuds du réseau. Cette fonction est appelée par activation parce qu'elle permet de vérifier s'il existe une connexion directe entre les deux nœuds. Soient w_1 et w_2 deux termes décrits dans JeuxDeMots, soit R un ensemble de relations contenant au moins une relation r. La valeur $w_1[w_2]$ est le poids de la dimension w_2 dans le vecteur V_{w_1} du terme w_1. De même pour $w_2[w_1]$ qui présente le poids de la dimension w_1 dans le vecteur V_{w_2} du terme w_2. La fonction Act décrite dans l'équation 2.15 représente la fonction d'activation.

$$\begin{aligned} Act &: \mathbb{R} \times \mathbb{R} \times \mathbb{N}^+ \rightarrow \mathbb{R} \\ Act(w_1, w_2, R) &= max(w_1[w_2],\ w_2[w_1], Sim\ (w_1, w_2)) \end{aligned} \tag{2.15}$$

Avec Sim une mesure de similarité sémantique entre les deux termes w_1 et w_2. Il est à noter que la fonction Act se définit sur l'ensemble $\mathbb{R}$ comme elle peut être restreinte sur l'ensemble $\mathbb{R}^+$, c'est-à-dire, l'ensemble de poids des dimensions des vecteurs à comparer est positif – *i.e.*, $Act : \mathbb{R}^+ \times \mathbb{R}^+ \times \mathbb{N}^+ \rightarrow \mathbb{R}^+$.

Comme pour les mesures à base de corpus, il est tout à fait possible de combiner différentes mesures à base de ressources lexico-sémantiques, ce que nous faisons référence dans la figure 2.1 par mesures hybrides (RLS : Ressources Lexico–Sémantiques).

2.2.3. Mesures basées sur les deux types de ressources

Dans les deux sous-sections précédentes, nous avons étudié deux types de mesures sémantiques : (1) basées sur les corpus permettant de comparer des mots, phrases ou textes (statistique des cooccurrences) ; et (2) basées sur les connaissances pouvant être utilisées pour comparer des mots ou sens de mots provenant de ressources lexico-sémantiques. Des mesures hybrides existent permettant de combiner les deux types de mesures, ce que nous faisons référence dans la figure 2.1 par CRLS. Par exemple, la ressource Wikipédia peut être utilisée par ce nouveau type de mesure puisqu'elle représente à la fois un corpus riche en textes et une organisation conceptuelle riche en connaissances. La plupart du temps, les mesures hybrides sont de base une combinaison de plusieurs mesures (Panchenko et Morozova, 2012). Harispe *et al.* (2015) ont décrit deux grandes approches utilisées par ces mesures :

(a) Mesures purement hybrides : définition de stratégies tirant parti de l'analyse des données provenant des deux sources de connaissances. Par exemple, la mesure de Resnik (1995) basée sur la théorie de l'information.

(b) Mesures agrégées : définition d'agrégation combinant des mesures sémantiques basées sur les corpus, sur les ressources lexico-sémantiques et même les mesures purement hybrides. L'idée derrière est que les scores des mesures sont agrégés selon une fonction mathématique. Par exemple, les fonctions *minimum, maximum, médiane, etc.*

Panchenko et Morozova (2012) ont démontré le rôle important que peut jouer la combinaison des mesures à base de corpus et de ressources lexico-sémantiques pour l'amélioration des performances des systèmes d'évaluation de mesures de similarité.

2.3. Méthodologies d'évaluation

Toute évaluation d'une mesure sémantique vise à distinguer les avantages et les inconvénients de cette mesure dans une application donnée, c'est-à-dire, la comparaison pouvant s'effectuer entre différentes mesures permet de déterminer le degré de pertinence d'une mesure à une autre et selon un contexte d'utilisation spécifique. En effet, dans l'absolu, il est difficile de généraliser et dire qu'une mesure sémantique est la meilleure dans tous les contextes d'utilisation possibles. On ne peut que dire qu'une mesure est meilleure qu'une autre si certaines conditions sont remplies. Par exemple, la mesure de Lin (Lin, 1998a) définie dans la sous-section 2.2.1 peut être meilleure qu'un Cosinus pour la comparaison de deux mots selon une analyse d'un corpus donné mais cette comparaison de mesures peut avoir lieu seulement et seulement si les deux mots à comparer partagent la même catégorie grammaticale vu que la méthode de Lin ne fonctionne que lorsque les deux mots à comparer sont de la même nature grammaticale.

Cette section présente deux éléments principaux pour l'évaluation des mesures sémantiques. Tout d'abord, nous présentons dans la sous-section 2.3.1 les listes de référence décrivant des jeux de données pour permettre la comparaison des mesures sémantiques. Ensuite, dans la sous-section 2.3.2, nous présentons les mesures d'évaluation.

2.3.1. Listes de référence

Mesurer la similarité sémantique entre les mots ou sens est un domaine qui a reçu beaucoup d'attention depuis plusieurs années. Des jeux de données ont été construits pour l'évaluation de la similarité, que ce soit pour l'anglais (Finkelstein *et al.*, 2001 ; Rubenstein et Goodenough, 1965), pour l'allemand (Mohammad *et al.*, 2007) ou pour le français (Joubarne et Inkpen, 2011).

Le tableau 2.2 présente dans un ordre chronologique les listes de référence état-de-l'art les plus utilisées pour la comparaison des mesures sémantiques.

Référence	Type d'évaluation et possibilité d'accès à la ressource	Nombre de paires
RUBENSTEIN et GOODENOUGH (1965) [RG–65 (Anglais)]	Similarité sémantique entre noms communs	65
MILLER et CHARLES (1991) [MC–30]	Similarité sémantique entre noms communs	30
LANDAUER et DUMAIS (1997) [TOEFL]	Évaluation du degré de la synonymie entre noms communs, verbes et adjectifs. Liste TOEFL : *Test of English as a Foreign Language*	80
FINKELSTEIN *et al.* (2001) [WORDSIM353]	Évaluation de la relation sémantique. Ressource : http://www.cs.technion.ac.il/∼gabr/resources/data/wordsim353	153 et 200. Total : 353
TURNEY (2001) [ESL]	Évaluation du degré de la synonymie entre noms communs, verbes et adjectifs. Liste ESL : *English as a Second Language*, une liste similaire à TOEFL	50
MOHAMMAD *et al.* (2007) [RG–65 (Allemand)]	Version allemande de la liste RG–65. Ressource : https://www.ukp.tu-darmstadt.de/data/semantic-relatedness/german-relatedness-datasets	65
PEDERSEN *et al.* (2007)	Évaluation de la relation sémantique entre mots du domaine médical	29
JOUBARNE et INKPEN (2011) [RG–65 (Français)]	Version française de la liste RG–65 avec un nouveau jugement humain. Ressource : http://www.site.uottawa.ca/∼mjoub063/wordsims.htm	65
SCHWARTZ et GOMEZ (2011) [CONCEPTSIM]	Similarité sémantique entre sens provenant de WORDNET (*synsets*). Désambiguïsation des paires de noms communs des listes RG–65, MC–30 et WORDSIM353. Ressource : http://www.seas.upenn.edu/∼hansens/conceptSim	65 (RG–65) ; 28 (MC–30) ; 97 (WORDSIM353). Total : 190
HUANG *et al.* (2012) [SCWS]	Évaluation de la relation sémantique entre mots en contexte. Liste SCWS : *Stanford's Contextual Word Similarities*. Ressource : https://nlp.stanford.edu/∼ehhuang/SCWS.zip	2 003
LUONG *et al.* (2013) [RW]	Évaluation de la relation sémantique entre mots rares. Liste RW : *The Stanford Rare Word Similarity Dataset*. Ressource : https://nlp.stanford.edu/∼lmthang/morphoNLM	2 034
BAKER *et al.* (2014)	Évaluation de la relation sémantique entre verbes. Ressource : http://ie.technion.ac.il/∼roiri	143
BRUNI *et al.* (2014) [MEN *test Collection*]	Évaluation de la relation sémantique entre mots. Ressource : http://clic.cimec.unitn.it/∼elia.bruni/MEN.html	3 000
HILL *et al.* (2014) [SIMLEX–999]	Similarité sémantique entre mots. Ressource : http://www.cl.cam.ac.uk/∼fh295/simlex.html	999

Table 2.2. – Listes de référence état-de-l'art utilisées pour la comparaison des mesures sémantiques

La plupart des listes ci-dessus sont basées sur des évaluations humaines et sont composées de paires de termes ou sens pour lesquelles les humains ont été invités à attribuer des scores de similarité sémantique pour évaluer la relation lexicale de *synonymie* ou des relations sémantiques telles que l'*hyperonymie* et l'*hyponymie*. Les instructions fournies aux participants varient d'un jeu de données à l'autre. Par exemple, l'échelle utilisée dans la liste proposée par Rubenstein et Goodenough (1965) est de $[0, 4]$ [4] alors que celle utilisée pour la liste proposée par Finkelstein *et al.* (2001) est de $[0, 10]$ [5]. Généralement, la qualité d'un ensemble de données est évaluée en analysant l'accord entre les participants, c'est-à-dire, comment les scores des participants sont corrélés.

La grande majorité des listes présentées dans le tableau 2.2 sont en anglais. Certaines ont été traduites manuellement ou automatiquement dans d'autres langues ou mises en correspondance avec des bases de connaissances (par exemple, la liste ConceptSim proposée par Schwartz et Gomez (2011) utilise WordNet). Dans ce cas, les paires de mots sont (manuellement) mises en correspondance avec des paires non ambiguës de sens afin d'être utilisées pour évaluer les mesures sémantiques fondées sur les connaissances.

Par ailleurs, il existe des listes dédiées à des domaines spécifiques (la liste proposée par Pedersen *et al.* (2007) pour le domaine médical) ou pour le traitement des mots rares (la liste proposée par Luong *et al.* (2013)). Le projet SML [6] (*The Semantic Measures Library*), présentant une librairie *open source Java*, de base propose différentes mesures sémantiques pour calculer la similarité entre différents éléments linguistiques (mots, sens, phrases ou même textes) mais fournit aussi différentes informations sur les listes de référence.

La liste proposée par Rubenstein et Goodenough (1965) est une liste très populaire et très utilisée, souvent appelée RG–65. Elle contient un ensemble de 65 paires de noms communs pour la langue anglaise. Le but de la création de cette liste était d'étudier la similarité sémantique et contextuelle pour l'ensemble des paires.

Joubarne et Inkpen (2011), quant à eux, ont fournit le même jeu RG–65 pour la langue française. Ils sont passés par une traduction des paires de l'anglais vers le français et ont établi un autre jugement humain. La traduction a été réalisée par l'utilisation d'une combinaison du dictionnaire Larousse français-anglais [7], Le Grand dictionnaire terminologique (GDT [8]) maintenu par l'Office québécois de la langue française, un couple de locuteurs natifs et un traducteur humain. Le jugement humain a fait appel à 18 évaluateurs qui ont le français comme langue maternelle.

4. De 0 (les deux éléments d'une paire donnée ne sont pas liés) à 4 (les deux éléments d'une paire donnée sont complètement liés).
5. De 0 (non liés) à 10 (complètement liés).
6. http://www.semantic-measures-library.org/sml
7. http://www.larousse.fr/dictionnaires/francais-anglais
8. http://www.granddictionnaire.com

Certains jeux de données comme celui de FINKELSTEIN *et al.* (2001), proposant un ensemble de 353 paires, ont été décomposés en deux-sous ensembles parce que la similarité sémantique telle qu'elle est reste toujours soumise à de multiples facteurs. Par exemple, au sens large, les personnes âgées et les adolescents n'associeront probablement pas le même score de similarité sémantique entre les deux mots *smartphone* et *ordinateur* mais si le regard tend vers le type de relation existant entre les mots, l'objectif de mesurer la similarité sémantique entre ces mots aura été atteint (MILLER et CHARLES, 1991). Par conséquent, il est essentiel de bien définir le type d'évaluation.

2.3.2. Mesures d'évaluation

Dans la littérature, deux approches peuvent être distinguées pour évaluer et comparer les mesures sémantiques. L'une des deux approches est intrinsèque (comparaison directe aux annotations humaines) et l'autre est extrinsèque (évaluation d'un système qui en dépend). Dans la plupart des évaluations, l'approche intrinsèque est la plus utilisée, c'est-à-dire l'approche basée sur le calcul du score de similarité et/ou du degré de la relation sémantique. D'autre part, les mesures sémantiques peuvent être évaluées indirectement en analysant les résultats de traitement qui en dépendent.

L'évaluation intrinsèque définit une classe de mesures qui sont différentes de celles proposées dans la section 2.2. Ces nouvelles mesures sont de nature statistique et s'appuient plutôt sur le classement absolu ou relatif des entités dans les vecteurs. Les mesures sémantiques sont dans ce cas là évaluées pour leur capacité à imiter les évaluations humaines. Les exemples type pour ces mesures d'évaluation : la corrélation de Spearman (ρ) et de Pearson (r), qui calculent la dépendance statistique du classement de deux variables. Pearson mesure la corrélation linéaire de deux variables en fonction des différences dans leurs valeurs, tandis que Spearman considère le classement relatif des valeurs des deux variables. Soit L la liste de référence contenant l'ensemble des n paires évaluées par les annotateurs, soit V_1 le vecteur contenant les valeurs attendues (la valeur souhaitée pour chaque paire de L), soit V_2 le vecteur contenant les valeurs obtenues par l'utilisation d'une mesure sémantique donnée. La fonction $r\ (V_1, V_2)$ décrite dans l'équation 2.16 retourne la corrélation de Pearson.

$$r\ (V_1, V_2) = \frac{\sum_{i=1}^{n} (V_{1i} - \bar{V_1})\ (V_{2i} - \bar{V_2})}{\sqrt{\sum_{i=1}^{n} (V_{1i} - \bar{V_1})^2} \sqrt{\sum_{i=1}^{n} (V_{2i} - \bar{V_2})^2}} \tag{2.16}$$

La fonction $\rho\ (V_1, V_2)$ décrite dans l'équation 2.17 retourne la corrélation de Spearman.

$$\rho\ (V_1, V_2) = 1 - \frac{6\ \sum_{i=1}^{n} (V_{1i} - V_{2i})^2}{n\ (n^2 - 1)} \tag{2.17}$$

Des adaptations de cette corrélation sont souvent envisagées pour évaluer d'une manière indirecte (évaluation extrinsèque) la qualité des mesures sémantiques, c'est-à-dire, en utilisant des données qui ne se réfèrent pas directement aux attentes humaines concernant les notions mesurées. Dans ce cas, la qualité d'une mesure est souvent évaluée indirectement en évaluant un système qui en dépend. Par exemple, un système de désambiguïsation sémantique ou de classification de documents.

Si nous prenons le système de classification de documents, l'objectif principal du système est de mesurer le degré d'appartenance d'un document à une classe. Cela consiste à mesurer la similarité de ce document avec le centre de gravité de cette classe. D'autre part, la classe à laquelle un document sera affecté est la classe pour laquelle l'élément a le plus haut degré d'appartenance. Par cette approche, les systèmes sont généralement évalués en utilisant une formule traditionnelle de la précision (*cf.* équation 2.18).

$$Précision_T = \frac{VP}{VP \ + \ FP} = \frac{vrais\ positifs}{vrais\ positifs \ + \ faux\ positifs} \tag{2.18}$$

Pour la classification de documents, la précision présente le rapport entre les documents pertinemment récupérés et les documents récupérés.

2.4. Conclusion

Dans ce chapitre, nous avons proposé une classification des mesures de similarité sémantique et nous avons montré que deux grandes classes peuvent être distinguées si nous nous basons sur le type de sources de connaissances à utiliser, à savoir : les corpus de données et les ressources lexico-sémantiques. Nous avons étudié les différentes approches pour chaque type de données et nous avons montré que ces approches peuvent être fusionnées pour produire de nouvelles mesures sémantiques. Ensuite, nous avons donné un aperçu sur les méthodologies d'évaluation. Nous avons vu qu'un effort considérable a été effectué pour proposer des jeux de données afin de permettre la comparaison des mesures sémantiques. À la fin de ce chapitre, nous avons décrit les mesures d'évaluation utilisées principalement pour cette comparaison.

Dans le chapitre suivant, nous présentons nos premières approches de désambiguïsation sémantique à base de connaissances provenant du réseau sémantique BABELNET et utilisant des mesures sémantiques à base de corpus et à base de ressources lexico-sémantiques. Nous rappelons que les mesures sémantiques représentent un critère important pour le choix de l'algorithme de désambiguïsation sémantique.

Désambiguïsation sémantique à base de connaissances provenant du réseau sémantique BABELNET

3.1. Motivation

La désambiguïsation sémantique consiste à choisir quel est le sens le plus approprié pour chaque mot d'un texte (IDE et VÉRONIS, 1998 ; NAVIGLI, 2009). Nous rappelons que l'objectif principal de cette thèse est de proposer un système de désambiguïsation sémantique pour l'aide à la substitution lexicale. Comme nous l'avons mentionné dans le chapitre 1, les systèmes de désambiguïsation sémantique les plus performants sont basés sur un apprentissage supervisé (*cf.* sous-section 1.3.2). Ces systèmes ont besoin d'un corpus de grande taille annoté manuellement en sens. Pour le français, et à notre connaissance, aucun corpus de ce genre n'existe. De ce fait, proposer un système de désambiguïsation supervisé permettant de lever l'ambiguïté de tous les mots pleins d'un texte sans un corpus d'apprentissage est inenvisageable. Avec cette contrainte, une désambiguïsation à base de connaissances pour le français serait plus convenable à ce jour (*cf.* chapitre 1, sous-section 1.3.3).

Une des approches classiques d'une désambiguïsation à base de connaissances consiste à estimer la similarité sémantique entre les sens de deux mots puis de l'étendre à l'ensemble des mots du texte. La méthode la plus directe donne un score de similarité à toutes les paires de sens de mots puis choisit la chaîne de sens qui retourne le meilleur score. On imagine la complexité exponentielle liée à cette approche exhaustive, nous nous retrouvons facilement avec un temps de calcul très long alors que le contexte qu'il est possible d'utiliser est petit. En d'autres termes, il y aurait $\prod_{w \in T} N_w$ combinaisons (séquences de sens) à évaluer, avec N_w le nombre de sens du mot w et T l'ensemble des mots du contexte. Par exemple, pour une phrase de 10 mots avec 10 sens en moyenne, il y aurait 10^{10} combinaisons possibles (séquences de 10 sens, un sens pour chacun des 10 mots). Considérons la phrase suivante : « *Dans une petite ville,*

un marchand de fruits possédait un magasin situé juste au-dessus d'une cave profonde. [1] », *petit* (adjectif) a 22 sens dans BABELNET (NAVIGLI et PONZETTO, 2012) dans sa version 4.0 [2], *ville* (nom) 18, *marchand* (nom) 9, *fruit* (nom) 10, *posséder* (verbe) 5, *magasin* (nom) 14, *situé* (adjectif) 1, *juste* (adverbe) 5, *cave* (nom) 10 et *profond* (adjectif) 22, il y a alors 2 744 280 000 combinaisons de sens possibles à analyser.

Le calcul exhaustif est donc très compliqué à réaliser dans des conditions réelles et, surtout, rend impossible l'utilisation d'un contexte plus important. Pour diminuer le temps de calcul, on peut utiliser une fenêtre autour du mot afin de réduire le temps d'exécution d'une combinaison mais le choix d'une fenêtre de taille quelconque peut mener à une perte de cohérence de la désambiguïsation. Dans ce chapitre, nous proposons une approche de désambiguïsation qui utilise le réseau sémantique BABELNET pour réduire le nombre de combinaisons tout en gardant une cohérence au niveau de la désambiguïsation. Nous avons choisi d'utiliser BABELNET parce qu'il offre un très grand nombre de *synsets* et couvre plusieurs mots polysémiques. Comme nous l'avons mentionné dans le chapitre 1, BABELNET utilise plusieurs ressources lexicographiques et encyclopédiques telles que WORDNET (FELLBAUM, 1998), WIKIPÉDIA, WIKTIONNAIRE, WIKIDATA, OMEGAWIKI, *etc.* (*cf.* sous-section 1.2.1). Aussi, il a l'avantage d'être multilingue. Il peut être utilisé par un système de désambiguïsation permettant de lever l'ambiguïté des mots provenant de différents textes écrits dans différentes langues. L'approche que nous proposons a été testée sur deux langues différentes, à savoir : le français et l'anglais.

Ce chapitre est organisé comme suit : nous présentons tout d'abord dans la section 3.2 notre approche de désambiguïsation faisant face au problème de l'approche exhaustive (complexité exponentielle du nombre de combinaisons à évaluer). Ensuite, une grande section (*cf.* section 3.3) est consacrée à l'évaluation intrinsèque des systèmes issus de cette approche. Nous présentons dans la sous-section (3.3.1) les différents corpus utilisés avant de présenter deux variantes de l'évaluation : (1) évaluation sur des échantillons lexicaux (*cf.* sous-section 3.3.2) ; et (2) évaluation sur l'ensemble des mots annotés (*cf.* sous-section 3.3.3). Cette dernière évaluation a été réalisée sur un corpus écrit en langue anglaise. Nous terminons ce chapitre avec une conclusion qui résume les aspects les plus importants (*cf.* section 3.4).

3.2. Approche de désambiguïsation à base de connaissances

Désambiguïser tous les mots pleins d'un corpus dont le contexte représente une phrase, un paragraphe ou tout un texte brut, est une tâche qui demande

1. Phrase tirée du corpus IREST (*International Reading Speed Texts*) correspond à des textes standards pour des tests de vitesse de lecture : http://www.vision-research.eu/index.php?id=641.
2. http://babelnet.org

beaucoup de temps si on se base sur un algorithme exhaustif simple. La clé de notre approche de désambiguïsation est l'observation des voisins de chaque mot polysémique dans le texte : au lieu de comparer chaque sens d'un mot à désambiguïser avec tous les sens de tous les mots qui se trouvent dans le texte, nous faisons une comparaison uniquement avec les sens des voisins sélectionnés au moyen d'une similarité distributionnelle. D'une part, ces voisins fournissent souvent des indices sur le sens le plus probable d'un mot dans un texte. D'autre part, cela nous permet de diminuer le temps d'exécution de l'algorithme et de ne pas perdre une cohérence au niveau de la désambiguïsation de tous les mots du texte. Il s'agit de garder les mots ayant de forts liens sémantiques afin de retourner le sens le plus spécifique (le plus adéquat) à chaque mot pour le contexte utilisé.

Ce que nous proposons dans un premier temps est d'utiliser une méta-heuristique d'optimisation combinatoire qui consiste à choisir les voisins les plus proches par sélection distributionnelle autour de chaque mot à désambiguïser. Un travail proche du nôtre a été proposé par McCarthy *et al.* (2004). Ce travail repose sur l'utilisation des voisins distributionnels et consiste à trouver le sens prédominant dans l'intégralité d'un texte donné. L'approche utilisée par McCarthy *et al.* (2004) est aussi non supervisée et consiste à défier les *baselines* de la désambiguïsation que nous avons présentées dans le chapitre 1 (*cf.* sous-section 1.4.3).

Dans cette section, nous présentons d'abord notre stratégie de sélection des voisins distributionnels depuis le contexte du mot à désambiguïser (*cf.* sous-section 3.2.1). Ensuite, nous décrivons l'algorithme de désambiguïsation à proprement dit (*cf.* sous-section 3.2.2).

3.2.1. Sélection des voisins distributionnels

Nous utilisons des mesures de similarité distributionnelle pour le choix des voisins les plus proches. La similarité distributionnelle est une mesure indiquant le degré de cooccurrence entre un mot-cible et son voisin apparaissant dans des contextes similaires. Par exemple, dans un texte décrivant l'équipement d'un *ordinateur de bureau* placé dans une salle de la maison, les voisins *écran, clavier* et *disque* ont une similarité distributionnelle plus forte avec le mot *souris (périphérique)* que les mots *maison* et *salle*.

Nous utilisons deux approches totalement différentes à base d'analyse distributionnelle. L'une d'elles repose sur la méthode de Lin (1998a) et l'autre repose sur l'utilisation des *word embeddings* :

(1) La première consiste à réaliser une analyse syntaxique en dépendances permettant d'extraire un ensemble de traits syntaxiques pour chaque mot analysé. Cette méthode vise à déterminer la similarité distributionnelle entre un mot polysémique et chacun de ses voisins, en se référant aux traits syntaxiques qu'ils partagent.

(2) La deuxième approche consiste à utiliser le modèle WORD2VEC proposé par MIKOLOV *et al.* (2013a). La similarité distributionnelle, dans ce cas-là, consiste à comparer le vecteur du mot polysémique à désambiguïser et le vecteur de chacun de ses voisins.

Nous utilisons la fonction Sim_{LIN} (*cf.* fonction 2.5, chapitre 2) pour mesurer la similarité à l'aide de la méthode proposée par LIN (1998a). D'autre part, nous utilisons la fonction COSINUS (*cf.* fonction 2.1, chapitre 2) pour mesurer la similarité à l'aide des *word embeddings*. Pour l'entraînement des *embeddings*, nous utilisons le modèle SKIP–GRAM. Nous nous intéressons à la sélection d'un contexte réduit en terme de taille et permettant de retourner un certain nombre k des mots les plus pertinents par rapport au contexte d'origine.

3.2.2. Algorithme de désambiguïsation par sélection distributionnelle et à base de connaissances provenant de BABELNET

Notre méthode de désambiguïsation sémantique prend en considération des critères distributionnels. Cette méthode repose sur l'hypothèse suivante : « plus la similarité distributionnelle entre les voisins est forte plus la probabilité d'avoir le sens le plus proche est grande ». Nous pouvons voir notre méthode comme un processus à deux niveaux :

(1) Le premier niveau sélectionne les voisins les plus proches au moyen d'une similarité distributionnelle.

(2) Le deuxième niveau permet de lever les ambiguïtés au moyen d'une similarité sémantique.

La similarité distributionnelle entre le mot à désambiguïser et chacun des voisins sélectionnés est plus forte que celle du mot à désambiguïser et chacun des autres mots du contexte. La similarité sémantique utilisée tient compte des traits sémantiques provenant des définitions des sens. Ces traits sémantiques représentent les mots pleins des définitions.

La similarité distributionnelle est utilisée pour déterminer un score entre chaque mot à désambiguïser et l'ensemble des mots pleins du texte. Cela a pour but de retourner les k meilleurs voisins qui ont le plus grand score de similarité. Si la mesure de similarité est celle de (LIN, 1998a), le contexte est limité aux mots qui partagent la même catégorie grammaticale du mot à désambiguïser. Dans le cas contraire, si la mesure de similarité est basée sur les *word embeddings*, le partage de la catégorie grammaticale n'est pas obligatoire. Dans ce cas, tous les mots pleins avec toutes les catégories grammaticales sont pris en compte.

Après avoir choisi les voisins distributionnels, nous adaptons la méthode structurelle proposée par NAVIGLI (2009) et qui est décrite formellement dans l'équation 2.14 (*cf.* chapitre 2 sous-section 2.2.2). Soient w_c un mot-cible à désambiguïser, $N_{w_c} = \{N_1, N_2, \ldots, N_k\}$ l'ensemble des k voisins les plus proches de w_c,

le sens $s^{'} \in S(N_i)$ où $S(N_i)$ est l'ensemble des sens du voisin N_i et $S(w_c)$ est l'ensemble des sens du mot-cible w_c. La fonction $\hat{S}'$ décrite dans l'équation 3.1 retourne le sens choisi par l'algorithme de désambiguïsation pour le mot-cible w_c. La fonction $Score(s, s^{'})$ retourne le score de similarité entre les sens s et $s^{'}$.

$$\hat{S}' = \underset{s \in S(w_c)}{\arg\max} \sum_{N_i \in N_{w_c} : N_i \neq w_c} \text{MAX } Score(s, s^{'}) \tag{3.1}$$

Pour mesurer la similarité sémantique entre deux sens, nous utilisons l'algorithme de LESK (1986), que nous appelons LeskBase par la suite (*cf.* chapitre 2, sous-section 2.2.2, équation 2.6). Pour cette utilisation, nous tenons compte de la forme lemmatisée des mots pleins et d'une comptabilisation d'occurrences d'un même mot. Nous utilisons aussi l'algorithme de Lesk étendu proposé par BANERJEE et PEDERSEN (2002) (*cf.* chapitre 2, sous-section 2.2.2, équation 2.13) mais dans une version simplifiée [3]. Nous utilisons aussi la variante permettant de comparer directement chaque sens candidat avec le contexte du mot à désambiguïser (*cf.* chapitre 2, sous-section 2.2.2, équation 2.9). Nous appelons cette variante LeskVariante tout au long de ce chapitre. Il est à noter que la sélection des voisins distributionnels ne concerne pas LeskVariante. Cette dernière prend en considération tout le contexte du mot à désambiguïser.

Pour comparer deux sens ou un sens candidat avec un contexte, nous utilisons du texte provenant des définitions (gloses) des sens et du contexte du mot à désambiguïser. Nous rappelons que BABELNET propose plusieurs définitions pour un sens donné et cela pour différentes langues. Nous tenons compte de toutes les définitions de la langue du corpus d'évaluation utilisé. Aussi, BABELNET utilise un système de traduction automatique pour l'enrichissement de sa base. Nous pouvons nous retrouver avec des sens qui ne proposent aucune définition pour une langue donnée (par exemple, le français). Dans ce cas-là, nous prenons en compte la liste des synonymes se trouvant dans les *Babel synsets*.

Le principe des algorithmes à base de Lesk est de compter le nombre de mots pleins partagés entre les deux ensembles de mots à comparer. Afin d'avoir cette liste de mots pleins partagés, nous réalisons une analyse morphologique de chaque texte pour obtenir la forme lemmatisée de chaque mot plein. C'est cette forme qui est mise en comparaison. Nous utilisons l'analyseur TREETAGGER [4] pour obtenir les mots pleins provenant des gloses de sens.

Au niveau de la comparaison des deux ensembles de mots, on peut facilement se retrouver avec des définitions de sens trop concises et il est difficile d'obtenir des distinctions de similarité fines. Pour ces cas, nous nous servons

3. Pour des raisons calculatoires, nous avons préféré utiliser une version simplifiée de l'algorithme de Lesk étendu en faisant une comparaison seulement entre les mots et non pas entre des séquences de mots comme décrit dans la version originale. Les relations sémantiques prises en compte sont les suivantes : {*hyperonymie, hyponymie, méronymie, holonymie*}. Nous tenons compte de la(les) glose(s) du sens mis en comparaison.

4. http://www.cis.uni-muenchen.de/∼schmid/tools/TreeTagger

de l'heuristique suivante une fois obtenu le score final de chaque sens candidat :

Dans le cas où deux sens ou plus possèdent le meilleur score de similarité, le sens retourné est celui qui a le plus grand nombre de connexions sémantiques avec les autres sens du réseau BabelNet.

Cette information est fournie dans BabelNet. Le plus souvent, le sens d'un mot qui a le plus de connexions sémantiques est le plus général. Par exemple, le sens *souris (genre de rongeur)* possède $1\ 453$ connexions sémantiques contre $1\ 244$ pour le sens *souris (informatique)* selon la version 4.0 du réseau. Aussi, le sens *avocat (homme de loi)* possède $2\ 208$ connexions sémantiques contre 210 pour le sens *avocat (fruit)*.

3.3. Évaluation intrinsèque de la désambiguïsation sémantique

Dans Senseval–1 et Senseval–2, des variantes de l'algorithme de Lesk ont été considérées soit comme des approches de base soit comme des systèmes complets. Dans Senseval–1, la plupart des systèmes de désambiguïsation ayant participé à la tâche *English All-Words* (EAW) ont été surclassés par une LeskVariante. D'un autre côté, durant Senseval–2, les algorithmes LeskBase et Lesk étendu ont été surclassés par la plupart des systèmes ayant participé à la tâche *English Lexical Sample* (ELS).

Les expériences que nous avons menées ont été réalisées sur deux corpus différents pour deux langues : français et anglais. Le contexte de chaque mot à désambiguïser représente le paragraphe. La première évaluation est menée sur le corpus français en utilisant seulement l'approche distributionnelle à base de traits syntaxiques (Billami, 2015). Pour cette évaluation, le corpus que nous utilisons et qui est décrit dans la sous-section 3.3.1 est de petite taille ($6\ 235$ occurrences de mots sur l'ensemble des textes). La deuxième évaluation que nous présentons a été menée sur un corpus de référence pour l'anglais (*i.e.* le corpus SemCor (Miller *et al.*, 1993)) qui est de taille plus importante et pour lequel nous avons utilisé les deux approches d'analyse distributionnelle (*cf.* sous-section 3.2.1) (Billami et Gala, 2016). Nous avons ainsi comparé l'utilisation de notre algorithme de désambiguïsation avec le système état-de-l'art Babelfy (Moro *et al.*, 2014).

Pour les expériences sur le corpus français, nous avons fait une évaluation sur un échantillon lexical (*i.e.* LSD : *lexical sample disambiguation*) contenant des noms et des verbes. Pour les expériences sur le corpus anglais SemCor, nous avons choisi, d'une part, de faire des expériences sur un échantillon de mots polysémiques (*i.e. English Lexical Sample*) contenant des noms, adjectifs et verbes et, d'autre part, de faire des expériences sur l'ensemble des mots polysémiques (*i.e. English All-Words*). Nous présentons pour ces expériences

sur l'anglais une comparaison entre la sélection des voisins distributionnels et les voisins les plus proches linéairement. Les résultats, que nous présentons par la suite, montrent que la sélection des voisins distributionnels est bien meilleure.

3.3.1. Description des corpus de travail et des corpus d'évaluation

Dans cette sous-section, nous décrivons les corpus sur lesquels nous avons réalisé une analyse distributionnelle afin de choisir les voisins les plus proches pour un mot ambigu dans un contexte donné. Nous faisons référence à ces corpus par "corpus de travail". Nous présentons ensuite les corpus d'évaluation pour chacune des deux langues utilisées.

Corpus de travail pour le français

Nous avons à disposition un ensemble de trois corpus de différents genres :

1. Collection de l'agence française de presse – AGENCE FRANCE PRESSE (AFP) [5].
2. Collection d'articles d'un journal local français : L'EST RÉPUBLICAIN (EST REP) [6].
3. Collection d'articles issue de la ressource encyclopédique libre WIKIPÉDIA [7].

L'ensemble des données de ces trois corpus est décrit dans le tableau 3.1. Ces différents corpus ont été analysés automatiquement par la chaîne de traitement MACAON [8] (NASR *et al.*, 2011) pour la lemmatisation du corpus, l'annotation en parties du discours ainsi que pour l'extraction des dépendances syntaxiques.

Corpus	Phrases	Tokens
AGENCE FRANCE PRESSE (AFP)	2 041 146	**59 914 238**
L'EST RÉPUBLICAIN (EST REP)	**2 998 261**	53 913 288
WIKIPÉDIA	1 592 035	33 821 460
Total	**6 631 442**	**147 648 986**

Table 3.1. – Données du premier corpus de travail utilisé pour l'extraction des triplets de dépendance syntaxique

Après l'analyse automatique en dépendances syntaxiques sur les données du corpus de travail, nous réalisons un filtrage sur les relations grammaticales

5. http://www.afp.com/fr
6. http://www.estrepublicain.fr
7. WIKIPÉDIA : https://fr.wikipedia.org.
8. MACAON : chaîne de traitement permettant d'effectuer des tâches standard du TAL.

extraites. Nous ne tenons pas compte des relations entre deux mots dont l'un est une unité lexicale sémantiquement vide. Par exemple, les conjonctions de coordination (*et, ou, ni, mais, car, or, donc*) et les déterminants.

Après avoir réalisé ce filtrage, nous disposons d'un ensemble de relations de cooccurrences, par exemple la relation *objet–de* et *sujet–de*. Ainsi, nous avons un ensemble de triplets de cooccurrences (w, r, x) associés avec leur fréquence d'apparition où r est une relation grammaticale et x est une cooccurrence associée avec le mot w selon la relation r. Par exemple, les triplets de dépendance syntaxique dans la phrase « *leurs regards recouvraient les eaux du fleuve* » retournés par la chaîne de traitement MACAON sont : (*regard, det, leur*), (*recouvrir, suj, regard*), (*recouvrir, obj, eau*), (*eau, det, le*), (*eau, dep, de*) et (*de, obj, fleuve*) [9]. Comme nous l'avons mentionné dans le chapitre 2 lors de la description de la mesure de Lin, nous pouvons voir les triplets comme des traits syntaxiques : pour le triplet (*recouvrir, suj, regard*), *regard* a pour trait syntaxique *suj (recouvrir)*.

Nous réalisons un deuxième filtrage en tenant compte seulement des traits syntaxiques apparaissant au moins 5 fois dans le corpus. Au final, nous obtenons $2\ 754\ 686$ triplets différents de dépendance syntaxique correspondant à $31\ 774$ noms uniques et $5\ 421$ verbes uniques. Ces triplets sont stockés et indexés après extraction de $12\ 785\ 450$ cooccurrences de mots.

Sur un ensemble de 30% sélectionné aléatoirement depuis la base de triplets extraits et pour lequel nous avons obtenu $22\ 168$ noms différents, la probabilité d'avoir le trait syntaxique *suj (border)* est de $\frac{38}{22\ 168}$ parce que seulement 38 noms uniques sont utilisés comme *sujet* pour le verbe *border*. La quantité d'information pour ce trait est de 6.37. Si on prend l'exemple du nom *fleuve*, ce nom possède le trait *suj (border)* comme il possède le trait *obj (connaître)*. La probabilité d'avoir *obj (connaître)* est de $\frac{582}{22\ 168}$. La quantité d'information retournée est de 3.64. Dans ce cas, le trait *suj (border)* est plus informatif que le trait *obj (connaître)*.

Corpus de travail pour l'anglais

Nous utilisons le corpus Europarl [10], European Parliament Proceedings Parallel Corpus (Koehn, 2005). Nous avons choisi ce corpus car il s'agit d'un corpus parallèle. Il couvre 21 langues dont le français et l'anglais. Le tableau 3.2 présente des statistiques sur le corpus pour l'anglais (plus de 2 millions de phrases et près de 54 millions de mots). Nous utilisons la chaîne de traitement MateTools [11] (Bohnet et Nivre, 2012) pour la lemmatisation du corpus, l'annotation en parties du discours ainsi que pour l'extraction des dépendances syntaxiques. Le système utilisé permet de coupler l'analyse morphologique et

9. La relation *det* est spécifique à un nom et son déterminant ; *suj* est la relation entre un verbe et son sujet ; *obj* est la relation entre un verbe et son objet ou autres ; la dernière relation est *dep* pour présenter une relation générique par défaut.

10. Nous utilisons la version 7 du corpus (http://www.statmt.org/europarl).

11. https://code.google.com/archive/p/mate-tools

l'analyse en dépendances avec des arbres non projectifs. Les modèles MATE-TOOLS que nous utilisons sont entraînés sur les données de CONLL SHARED TASK 2009 [12] (HAJIČ *et al.*, 2009).

Corpus	Phrases	Tokens
EUROPARL	**2 218 201**	**53 974 751**

Table 3.2. – Données du deuxième corpus de travail utilisé principalement pour la sélection des voisins distributionnels

Pour l'entraînement des *embeddings*, nous utilisons une alternative du projet WORD2VEC [13] fait en langage de programmation JAVA par l'équipe MEDALLIA [14] pour l'intégrer dans notre programme principal de désambiguïsation sémantique, fait en JAVA. L'entraînement du réseau de neurones est réalisé sur le corpus EUROPARL avec un prétraitement en avant. Par exemple, la phrase tirée du corpus « *In short, the issue is an important one.* » est remplacée par « *in_IN short_Adj ,_, the_DT issue_N be_V an_DT important_Adj one_N ._.* ». Pour le paramétrage du réseau de neurones, nous avons choisi une fenêtre d'une taille de 20 mots (la fréquence d'apparition des mots est d'un minimum égal à 5, les dimensions des vecteurs sont de 300, le nombre d'exemples négatifs est de 7 avec une utilisation de l'alternative *softmax* hiérarchique). Contrairement à la méthode de LIN (1998a), l'utilisation des *word embeddings* nous permet d'avoir l'avantage de comparer des mots ayant différentes parties du discours.

Corpus d'évaluation pour le français

Nous travaillons sur deux corpus différents : corpus IREST [15] contenant 10 textes et un corpus brut contenant 20 textes pour un total de 30 textes. Le corpus IREST correspond à des textes standards créés et élaborés pour servir dans des tests de vitesse de lecture tandis que le deuxième corpus correspond à des textes de lecture pour enfants en école primaire. Ces corpus proviennent du domaine général et ne sont pas des corpus de spécialité. Nous avons $6\,235$ occurrences de mots ($4\,139$ occurrences de mots pleins) et une moyenne de 208 occurrences (138 occurrences de mots pleins) par texte. Les textes sont lemmatisés et annotés en parties du discours par la chaîne de traitement MACAON. Le travail de désambiguïsation mené pour cette partie d'évaluation ne concerne que des unités monolexicales (les expressions polylexicales n'ont pas été prises en compte).

12. http://ufal.mff.cuni.cz/conll2009-st
13. https://github.com/medallia/Word2VecJava
14. http://engineering.medallia.com
15. http://www.vision-research.eu/index.php?id=641

Le tableau 3.3 résume d'une part le nombre de mots pleins du corpus d'évaluation couverts ou non par BabelNet pour chaque catégorie grammaticale (nom commun : NC, adjectif : ADJ, adverbe : ADV et verbe : V) et cela par nombre de types (mots différents) et tokens (ensemble total de mots). Le total (T) est aussi décrit dans le tableau. D'autre part, les taux de couverture obtenus sont présentés. La couverture globale présente le rapport entre les mots couverts par BabelNet et l'ensemble des mots du corpus d'évaluation. Le pourcentage des mots polysémiques présente le rapport entre les mots polysémiques couverts par BabelNet et l'ensemble des mots couverts par BabelNet (l'ensemble des mots monosémiques et polysémiques).

POS	Mots polysémiques		Mots monosémiques		Mots non couverts		Nombre total		**% Couverture globale**		**% Mots polysémiques**	
	Tokens	Types	Tokens	Types	Tokens	Types	Tokens	Types	**Tokens**	**Types**	**Tokens**	**Types**
NC	**1 660**	**590**	**130**	**39**	51	28	**1 841**	**657**	**97.23**	**95.74**	92.74	**93.8**
ADJ	353	164	28	19	**165**	**48**	546	231	69.78	79.22	92.65	89.62
ADV	375	68	79	6	33	14	487	88	93.22	84.09	82.6	91.89
V	1 135	327	31	30	99	47	1 265	404	92.17	88.37	**97.34**	91.6
T	**3 523**	**1 149**	**268**	**94**	**348**	**137**	**4 139**	**1 380**	**91.59**	**90.07**	**92.93**	**92.44**

Table 3.3. – Taux de couverture des mots pleins du corpus français d'évaluation par le réseau sémantique BabelNet

Nous avons la meilleure couverture globale pour les noms sur les tokens et les types. Nous atteignons 97.23% en tokens contre 92.17% pour les verbes et 95.74% en types contre 88.37% pour les verbes. La couverture en tokens des verbes polysémiques est plus grande par rapport à la couverture des noms polysémiques (97.34% contre 92.74%) [16]. Pour les mots non couverts, les erreurs d'annotations grammaticales (POS) retournées par MACAON représentent la quasi-totalité des cas (seulement 69.78% des tokens pour les adjectifs sont couverts).

Corpus d'évaluation pour l'anglais

Le test et l'évaluation de notre méthode portent sur le corpus SemCor. Il contient $234\ 136$ instances de sens de mots contenues dans 352 textes, dont 80% proviennent du Corpus Brown et 20% proviennent du roman The Red Badge of Courage pour lesquels les mots à classe ouverte ont été annotés sémantiquement avec WordNet. Au total, SemCor contient $11\ 860$ paragraphes et $37\ 176$ phrases. La lemmatisation et l'annotation des mots en parties du discours sont

16. Nous avons une occurrence par verbe monosémique (30 types pour 31 tokens) et une couverture de 327 verbes polysémiques contre 30 verbes monosémiques.

fournies avec le corpus. De ce fait, nous n'avons pas besoin de refaire ce traitement.

Le corpus SemCor utilise WordNet (Fellbaum, 1998) comme inventaire de sens. Nous avons montré au tout début de ce chapitre notre intérêt d'utiliser BabelNet [17] (Navigli et Ponzetto, 2012). BabelNet couvre plusieurs mots polysémiques et propose plus d'informations que WordNet sur les sens. Il utilise en plus de WordNet d'autres ressources afin d'enrichir ses sens telles que Wikipédia, Wiktionnaire, Wikidata, OmegaWiki, Open Multilingual WordNet, *etc.*

Comme BabelNet permet d'avoir les correspondances de ses sens avec ceux de WordNet et comme SemCor utilise WordNet comme inventaire de sens, nous pouvons utiliser BabelNet pour annoter les mots en sens dans SemCor. Un avantage qu'offre BabelNet est qu'il permet de différencier un concept d'une entité nommée. Dans toutes nos expériences, nous nous intéressons à la désambiguïsation des mots dont les sens décrivent des concepts. BabelNet propose une mise en correspondance de ses *Babel synsets* avec les sens de la version 3.0 de WordNet, nous avons donc utilisé la version 3.0 de SemCor. Toutefois, dans SemCor 3.0, certains mots sont annotés avec des sens provenant de la version 1.6 de WordNet et n'ont pas une correspondance avec les sens de BabelNet. Il s'agit de $1\ 728$ sens uniques provenant de WordNet 1.6 correspondant à $8\ 721$ occurrences de mots annotés.

En termes d'annotation sémantique avec BabelNet, nous avons à disposition $25\ 881$ sens uniques correspondant à $225\ 415$ occurrences de mots annotés dont seulement $224\ 370$ sont annotés comme étant des concepts. Le tableau 3.4 résume le nombre de tokens et types annotés comme concepts avec BabelNet. Parmi ces $224\ 370$ occurrences, 699 occurrences sont annotées avec plus d'un sens. Nous avons une proportion de $96.28\%(\frac{225\ 415}{234\ 136})$ de mots annotés manuellement dans SemCor qui sont couverts par BabelNet et de $95.83\%(\frac{224\ 370}{234\ 136})$ de cas que nous traitons pour la tâche de désambiguïsation sur l'ensemble des mots annotés.

POS	Tokens	Types
NC	85 957	**11 119**
ADJ	31 132	4 836
ADV	18 947	1 501
V	**88 334**	4 665
T	**224 370**	**22 121**

Table 3.4. – Nombre de tokens et types du corpus d'évaluation SemCor annotés avec des concepts par le réseau sémantique BabelNet

17. Nous avons utilisé la version 2.5.1 de BabelNet pour toutes nos expériences.

Notre système de désambiguïsation annote en sens toutes les occurrences sur lesquelles nous prenons une référence (224 370 occurrences de mots dont 191 146 occurrences sont pour des mots polysémiques [18]). Afin de mesurer la performance de notre système, nous faisons une évaluation seulement sur ces 191 146 occurrences de mots.

3.3.2. Évaluation sur des échantillons lexicaux

Nous présentons dans cette sous-section une évaluation sur un échantillon lexical pour le français et pour l'anglais. Pour chaque évaluation, nous présentons d'abord l'ensemble des mots que nous avons choisi, ce que nous appelons notre "jeu de test". Ensuite, les résultats des expériences menées sont décrits.

Jeu de test pour le français

Nous avons choisi les données de notre jeu de test selon leur niveau d'ambiguïté. Nous avons à disposition un corpus d'évaluation pour lequel il est difficile de sélectionner des mots polysémiques selon leur fréquence d'apparition (peu fréquent, fréquent et très fréquent). Le tableau 3.5 ci-dessous résume les informations quantitatives utilisées pour la sélection des mots polysémiques du jeu de test.

POS	Candidat	Fréquence d'apparition	Nombre de sens	Niveau d'ambiguïté
Noms	*fleuve*	3	3	**Peu ambigu**
	fée	8	3	
	pêcheur	4	4	**Ambigu**
	plante	15	5	
	castor	4	9	**Très ambigu**
	souris	10	9	
Verbes	*planter*	2	3	**Peu ambigu**
	naître	7	3	
	obliger	2	5	**Ambigu**
	taire	9	5	
	troubler	2	7	**Très ambigu**
	parler	6	10	

Table 3.5. – Liste des noms et verbes du jeu de test pour le français : fréquence d'apparition et niveau d'ambiguïté par utilisation du réseau sémantique BabelNet

18. BABELNET propose au moins deux sens.

La sélection des mots du jeu de test s'est portée sur le niveau d'ambiguïté (peu ambigu, ambigu et très ambigu). Nous prenons deux mots polysémiques pour chaque niveau d'ambiguïté et cela pour les noms et les verbes. Nous considérons les mots qui ont moins de 4 sens comme peu ambigus (*cf.* tableau 3.5), les mots qui ont entre 4 et 6 sens comme ambigus et les mots qui ont plus de 6 sens comme très ambigus.

Résultats des expériences menées pour le français

Pour mesurer la performances de notre algorithme de désambiguïsation (avec les différentes mesures à base de Lesk comme cité dans la sous-section 3.2.2), nous utilisons le taux d'exactitude (*accuracy rate*). L'évaluation de notre algorithme est effectuée sur des données dont la couverture des sens par BABELNET est de 100%. Ce taux d'exactitude est calculé pour chaque mot du jeu de test et pour chaque mesure de Lesk testée. Il présente le rapport entre le nombre d'occurrences correctement désambiguïsées et le nombre total d'occurrences d'un mot. L'ensemble des taux d'exactitude obtenus est résumé dans le tableau 3.6. Notre jeu de test contient 44 occurrences pour 6 noms et 28 occurrences pour 6 verbes (un total de 72 occurrences sur 12 mots différents). Nous avons affecté manuellement [19] à chaque occurrence de mot le bon sens proposé dans BABELNET. Notre évaluation porte, d'une part, sur le niveau d'ambiguïté des mots polysémiques et, d'autre part, sur le nombre de voisins distributionnels à sélectionner ($k-$plus proches voisins, $k - PPV$). Notre choix s'est porté sur trois valeurs différentes, $k \in \{3, 5, 7\}$. Nous avons choisi aussi de prendre en compte différentes versions du corpus de travail afin de mesurer le degré de confiance de notre approche en sélectionnant aléatoirement une partie de l'ensemble des triplets de dépendance syntaxique ($30\%V1$ pour une première version, $30\%V2$ pour une deuxième version, $50\%V1$ et $50\%V2$) ou la totalité des triplets de dépendance. Les résultats du tableau 3.6 sont obtenus en utilisant une première sélection de 30% sur l'ensemble des triplets (*i.e.* $30\%V1$) et un contexte réduit à 5 voisins les plus proches au moyen d'une similarité distributionnelle.

Les résultats retournés par l'algorithme de Lesk étendu sont intéressants en comparaison avec les autres algorithmes ou ce que BABELFY retourne sur l'ensemble des noms étudiés. Lesk étendu retourne le bon sens pour les noms peu ambigus. Pour les noms ambigus, il retourne le bon sens sur toutes les occurrences de *plante*, en revanche, il se trompe sur toutes les occurrences de *pêcheur* parce qu'il y a deux sens pour lesquels le score retourné par nos méthodes est le même.

19. Deux locuteurs ont annoté manuellement les 72 occurrences de mots en sens. Il s'agit de moi-même et Mme Nùria GALA. Tous les sens proposés par les deux locuteurs sont des sens attendus.

Jeu de test	LeskBase	LeskÉtendu	LeskVariante	BABELFY
fleuve	100	100	0	100
fée	0	100	100	87.5
pêcheur	0	0	0	0
plante	86.67	100	80	100
castor	100	100	100	100
souris	30	100	0	30
planter	100	100	100	100
naître	0	85.71	85.71	100
obliger	50	100	50	100
taire	*erreur* POS	*erreur* POS	*erreur* POS	–
troubler	0	0	0	0
parler	16.67	100	16.67	50

Table 3.6. – Taux d'exactitude obtenus par les méthodes de Lesk de base, Lesk étendu et par sélection aléatoire de 30% (V1) sur l'ensemble des triplets pour les 5 plus proches voisins et comparaison avec la variante de Lesk et Babelfy sur les données du jeu de test

Ci-dessous, une description des deux sens de *pêcheur* avec un extrait d'un texte où *pêcheur* n'est pas désambiguïsé correctement.

— Sens 1 : « *la pêche est l'activité consistant à capturer des animaux aquatiques dans leur milieu naturel* ».

— Sens 2 : « *personne dont la profession est d'attraper des poissons* ».

Sur un extrait de texte :

— « ... *il fut recueilli par un vieux* ***pécheur*** *de saumons.* ... ».

Le bon sens de *pêcheur* est le deuxième mais le premier est retourné par nos méthodes vu qu'il possède plus de connexions sémantiques ($1\ 576$ contre 355). Pour les noms très ambigus, Lesk étendu ne retourne pas de mauvais sens contrairement à BABELFY. Sur quelques textes décrivant la *souris* comme *souris (genre de rongeur)*, BABELFY retourne une entité nommée *MouseHunt* décrivant un long métrage de *Gore Verbinski*.

Pour les verbes, il est difficile de juger la sensibilité du taux d'exactitude au niveau d'ambiguïté. D'une part, nous avons des erreurs POS (par exemple, le verbe *taire* ne se trouve sur aucun des textes utilisés mais il apparaît après analyse (utilisation de MACAON) comme verbe), d'autre part, le manque des définitions en français dans BABELNET. Dans certains cas, le sens le plus fort dans le réseau pour le verbe étudié est retourné même s'il n'est pas le plus proche du verbe selon le contexte dans lequel il apparaît et cela malgré l'utilisation des synonymes qui représentent aussi un très petit nombre dans les *Babel synsets*.

Sur l'ensemble, nous remarquons que l'algorithme de Lesk étendu est beaucoup plus régulier par rapport à LeskBase ou LeskVariante. Le meilleur taux

d'exactitude que nous obtenons sur les mots étudiés est celui retourné par l'algorithme de Lesk étendu (90.91% pour les noms et 57.14% pour les verbes). Lesk étendu est meilleur par rapport à BABELFY et LeskVariante pour la désambiguïsation des noms (taux d'exactitude de 72.73% par BABELFY et 54.55% par LeskVariante) ainsi que pour la désambiguïsation des verbes (taux d'exactitude de 50% par BABELFY et 35.71% par LeskVariante).

Dans le tableau 3.7, nous présentons les résultats obtenus par variation du nombre des voisins les plus proches et par sélection aléatoire ou non d'un ensemble de dépendances syntaxiques. Nous remarquons que la variation de l'ensemble des triplets de dépendance syntaxique apporte des résultats différents pour la désambiguïsation (différence légère pour les noms mais forte pour les verbes). Les voisins d'un mot étudié changent à chaque fois où on utilise un ensemble de triplets différent. Si nous prenons l'exemple de *plante*, nous avons sur un texte les voisins (*bande, feuille, oiseau*) par sélection de $30\%V1$ sur l'ensemble des triplets alors que nous obtenons un autre ensemble de voisins (*animal, insecte, oiseau*) par sélection de $30\%V2$ sur les triplets.

Algorithme de	**Noms**					**Verbes**				
Lesk étendu	$k=3$	$k=5$	$k=7$	Moyenne	ÉcartType	$k=3$	$k=5$	$k=7$	Moyenne	ÉcartType
Depends30%V1	**90.91**	**90.91**	84.09	88.64	3.21	50	57.14	**60.71**	55.95	4.45
Depends30%V2	84.09	84.09	81.82	83.33	1.07	42.86	39.29	**60.71**	47.62	9.37
Depends50%V1	75	**90.91**	75	80.3	7.5	25	28.57	28.57	27.38	1.68
Depends50%V2	75	75	84.09	78.03	4.29	35.71	32.14	**60.71**	42.85	12.71
Depends100%	84.09	77.27	77.27	79.54	3.21	32.14	**60.71**	**60.71**	51.19	13.47
Moyenne	81.82	83.64	80.45	81.97	1.3	37.14	43.57	54.28	45	7.07
ÉcartType	**6.1**	**6.65**	**3.69**	**3.76**	–	**8.63**	**13.05**	**12.86**	**9.80**	–

Table 3.7. – Taux d'exactitude obtenus par application de l'algorithme de Lesk étendu et par sélection de différents ensembles de triplets pour différents nombres de voisins distributionnels les plus proches (k–PPV)

Pour les noms et sur la variation des k voisins les plus proches, nous obtenons le meilleur taux d'exactitude (90.91%) pour $k \in \{3, 5\}$ par rapport au cas où $k = 7$.

Cela signifie qu'un petit nombre de voisins est nécessaire pour retourner le bon sens pour les noms, ce qui est tout le contraire pour les verbes où le meilleur taux d'exactitude retourné est atteint lorsque $k = 7$. Les résultats montrent qu'on obtient un bon degré de confiance pour les noms (écart type de 3.76) par contre un degré de confiance faible pour les verbes (écart type de 9.80).

Les figures 3.1 et 3.2 présentent les résultats obtenus par utilisation de l'algorithme de Lesk étendu respectivement sur les noms et les verbes.

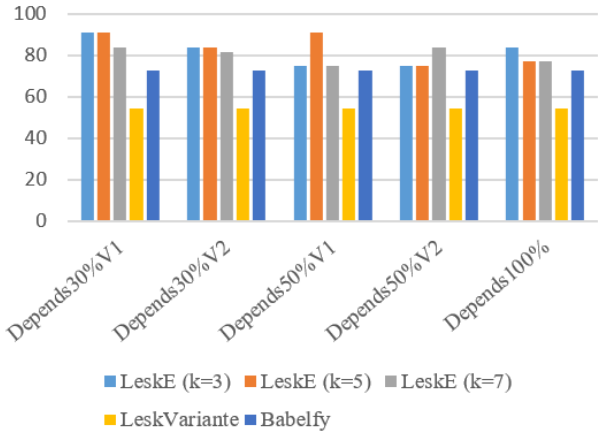


Figure 3.1. – Taux d'exactitude obtenus sur les **noms** du jeu de test pour le français par utilisation de l'algorithme de Lesk étendu

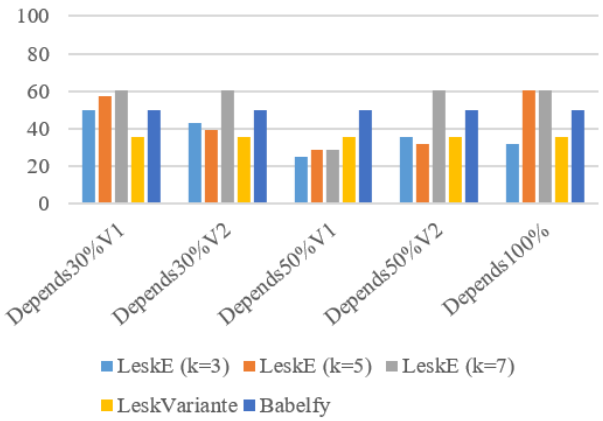


Figure 3.2. – Taux d'exactitude obtenus sur les **verbes** du jeu de test pour le français par utilisation de l'algorithme de Lesk étendu

Les figures 3.3 et 3.4 présentent les résultats obtenus pour les différents algorithmes utilisés et BABELFY sur un ensemble précis de dépendances syntaxiques.

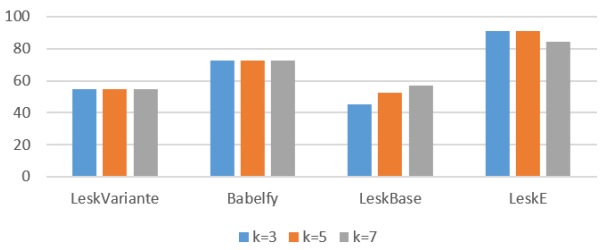


Figure 3.3. – Taux d'exactitude obtenus sur les **noms** du jeu de test pour le français par sélection aléatoire de 30% sur les dépendances syntaxiques

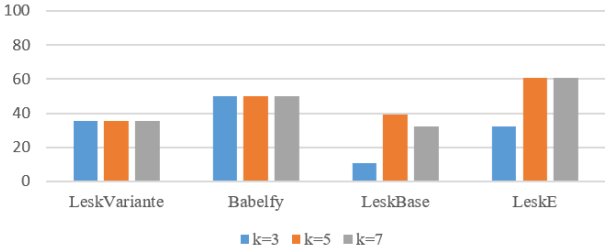


Figure 3.4. – Taux d'exactitude obtenus sur les **verbes** du jeu de test pour le français par utilisation de l'ensemble des dépendances syntaxiques

Le meilleur taux d'exactitude retourné pour les noms est de 90.91% contre 60.71% pour les verbes. La meilleure combinaison retourne 77.78% ($k = 5$ et $30\%V1$ de l'ensemble des triplets de dépendance syntaxique).

L'algorithme de Lesk étendu retourne le meilleur résultat par rapport à la LeskBase et BABELFY pour les noms et cela sur toutes les variations utilisées pour obtenir un ensemble de triplets de dépendance syntaxique. Pour les verbes, une sélection aléatoire d'une partie des triplets de dépendance apporte à notre approche des scores inférieurs par rapport à ce que nous obtenons pour les noms.

Nous faisons l'hypothèse que l'utilisation d'une autre mesure de similarité sémantique dans le cas où il y a moins de traits sémantiques (mots pleins des gloses) peut améliorer nos résultats.

Jeu de test pour l'anglais

Les mots du jeu de test pour l'anglais sont choisis selon le niveau d'ambiguïté et la fréquence d'apparition. Nous avons fait une sélection de quatre mots pour les catégories grammaticales noms, adjectifs et verbes.

— Liste des noms = {*argument, disc, paper, plan*}.

— Liste d'adjectifs = {*black, narrow, valid, wet*}.

— Liste des verbes = {*add, begin, note, operate*}.

Le tableau 3.8 présente les informations quantitatives sur les mots du jeu de test. Nous avons à disposition un ensemble de 12 mots représentant un total de 1 022 occurrences dans SEMCOR.

POS	Candidat	Fréquence d'apparition	Nombre de sens	Niveau d'ambiguïté
Noms	*disc*	11	6	**Ambigu**
	plan	68	6	
	paper	71	7	**Très ambigu**
	argument	39	10	
Adjectifs	*valid*	5	2	**Peu ambigu**
	narrow	20	5	**Ambigu**
	wet	21	6	
	black	44	14	**Très ambigu**
Verbes	*note*	105	4	**Ambigu**
	add	196	6	
	operate	82	7	**Très ambigu**
	begin	360	11	

Table 3.8. – Liste des noms, adjectifs et verbes du jeu de test pour l'anglais : fréquence d'apparition et niveau d'ambiguïté par utilisation du réseau sémantique BabelNet

Résultats des expériences menées pour l'anglais

Comme pour l'évaluation sur le français, le taux d'exactitude est utilisé ici pour mesurer la performance du système. Nous avons vu les performances de notre système pour le français. Cependant, le jeu de test était de petite taille. Pour l'anglais, le jeu de test est de taille plus importante. D'autre part, nous prenons ici tous les triplets de dépendance syntaxique extraits depuis le corpus de travail.

Nous utilisons le paramètre k pour choisir le nombre de voisins à sélectionner. Pour ces expériences, le choix s'est porté sur une valeur entre 2 et 7. Aussi, nous faisons une comparaison avec les voisins les plus proches linéairement. Si k est pair, nous sélectionnons $\frac{k}{2}$ voisins linéaires de la droite du mot à désambiguïser et $\frac{k}{2}$ de sa gauche. Si k est impair, la valeur entière $\frac{k}{2}$ représente le nombre de voisins de gauche et $\frac{k}{2}+1$ représente le nombre de voisins de droite. Dans le cas où le mot à désambiguïser est au début ou à la fin du paragraphe donné, nous nous assurons de prendre le bon nombre k en tenant compte de plus de voisins d'un côté à l'autre. Nous avons présenté dans la sous-section 3.2.1 deux mesures distributionnelles, la première repose sur les traits syntaxiques et la deuxième sur les *word embeddings* (WORD2VEC ou W2V). Dans ces expériences, nous utilisons une autre mesure distributionnelle, ALL [20], qui représente une combinaison des deux mesures.

La figure 3.5 présente les résultats sur l'ensemble des mots du jeu de test par application de la méthode de LIN.

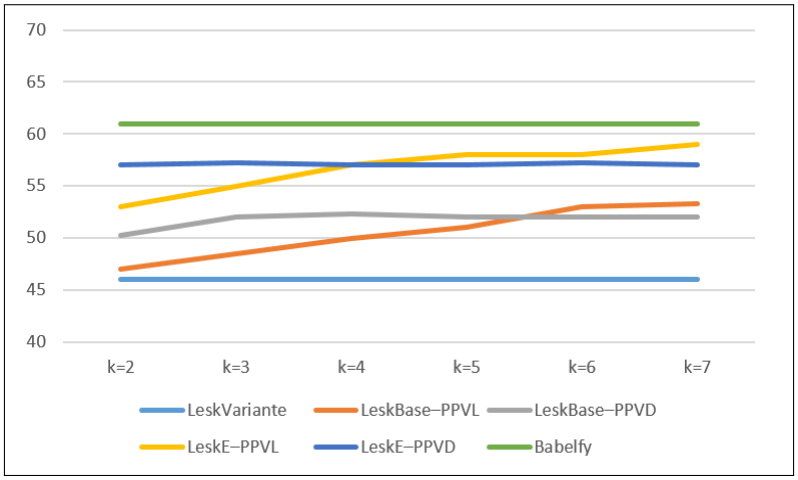


Figure 3.5. – Taux d'exactitude obtenus sur l'ensemble des mots du jeu de test extraits du corpus anglais SemCor ; comparaison entre la sélection des voisins linéaires et des voisins distributionnels (application de la méthode d'analyse distributionnelle de Lin)

20. Nous utilisons une moyenne entre les deux mesures et prenons en compte des voisins partageant la même partie du discours avec le mot à désambiguïser.

LESKBASE–PPVL présente l'application de l'algorithme LeskBase sur les plus proches voisins linéairement, LESKE–PPVL fait référence à l'application de l'algorithme de Lesk étendu sur ces mêmes voisins, LESKBASE–PPVD pour l'algorithme LeskBase en utilisant les voisins distributionnels et LESKE–PPVD pour l'application de l'algorithme de Lesk étendu sur ces voisins distributionnels. Le meilleur taux d'exactitude que nous atteignons sur l'ensemble des mots du jeu de test est de 58% contre 61% pour BABELFY. L'algorithme de Lesk étendu reste le meilleur pour ces expériences. Avec son utilisation, nous avons l'avantage d'avoir besoin d'un petit nombre de voisins distributionnels contrairement à l'algorithme LeskBase où nous avons besoin d'un nombre de voisins plus important.

Nous avons identifié que la mesure de LIN permet de sélectionner des voisins distributionnels qui donnent de bonnes performances de désambiguïsation et que la combinaison des deux mesures distributionnelles ne retourne pas des résultats meilleurs.

Nous développons cette remarque plus en détail dans la sous-section suivante sur la désambiguisation de l'ensemble des mots pleins du corpus SEMCOR. D'autre part, l'utilisation d'un petit nombre de voisins distributionnels avec l'algorithme de Lesk étendu comme le montre la figure 3.5 permet d'améliorer les performances de désambiguïsation.

3.3.3. Évaluation sur tous les mots pleins du corpus

Le tableau 3.9 présente les résultats obtenus sur l'ensemble des mots polysémiques traités dans le corpus SEMCOR ($191\ 146$ occurrences de mots). Ce tableau tient compte de toutes les méthodes d'analyse distributionnelle et toutes les variantes de notre système de désambiguïsation en sélectionnant quatre voisins. Sur ce même tableau, nous présentons une comparaison avec la sélection de quatre voisins les plus proches linéairement.

Algorithmes /	Lesk	LeskBase–	LeskE–	LeskBase–PPVD			LeskE–PPVD		
POS	**Variante**	**PPVL**	**PPVL**	LIN	W2V	ALL	LIN	W2V	ALL
Noms	40.7%	40.6%	45.3%	41.4%	**40.0%**	41.3%	**47.3%**	44.7%	47.2%
Adjectifs	48.3%	45.3%	44.8%	48.0%	43.3%	47.9%	**49.4%**	**40.4%**	**49.4%**
Adverbes	45.9%	**47.6%**	42.0%	45.8%	46.4%	45.7%	42.3%	**41.7%**	42.3%
Verbes	**34.4%**	**25.2%**	30.0%	29.9%	28.0%	30.0%	30.2%	26.1%	30.2%

Table 3.9. – Taux d'exactitude obtenus selon différents algorithmes de désambiguïsation ($k = 4$) pour une évaluation sur le corpus anglais SemCor

En termes de comparaison des méthodes d'analyse distributionnelle sur ces expériences, il s'avère que l'utilisation de la méthode de LIN est plus rentable et meilleure, pour le choix des voisins distributionnels, que l'utilisation de la méthode W2V. Nous remarquons aussi que la combinaison des deux méthodes ne retourne pas des résultats aussi bons que la simple utilisation de la méthode de LIN. Sans surprise, l'algorithme de Lesk étendu retourne les meilleurs résultats pour les noms et les adjectifs. Pour le cas des verbes, nous pouvons avoir les meilleurs résultats par simple utilisation de LeskVariante.

Pour le type des voisins à utiliser (distributionnels *vs* linéaires), le voisin distributionnel permet au système de désambiguïsation sémantique d'être plus performant et cela pour les noms, verbes et adjectifs. Une exception est à noter pour les adverbes où les résultats sont plus ou moins proches selon l'algorithme utilisé. Sur l'ensemble des valeurs possibles de k, nous avons remarqué que l'utilisation d'un petit nombre de voisins distributionnels peut nous mener à atteindre les meilleurs résultats.

Nous avons mené une autre expérience seulement sur les occurrences de mots annotés manuellement avec le sens le plus fort dans BABELNET. Nous aurions pu imaginer, avec l'heuristique utilisée, qu'on peut atteindre des résultats dans les $80 - 90\%$ mais ce n'est pas le cas. Nous atteignons seulement 75% pour les noms et 67% sur l'ensemble des mots pleins avec une utilisation de la méthode de LIN. Nous avons remarqué sur l'ensemble que BABELNET propose plusieurs sens qui ne sont pas raffinés et contient des incohérences, celles-ci se traduisent souvent par des anomalies dans SEMCOR.

3.4. Conclusion

Dans ce chapitre, nous avons présenté un système de désambiguïsation à base de connaissances par sélection distributionnelle des voisins, en se basant sur un corpus de travail, et traits sémantiques provenant du réseau sémantique BABELNET. Le système que nous avons proposé permet de réduire la complexité exponentielle qu'engendre l'approche la plus directe de désambiguïsation, à savoir l'approche exhaustive. Notre système possède deux niveaux de traitement : (1) niveau permettant de sélectionner les voisins les plus proches au moyen d'une similarité distributionnelle ; et (2) niveau permettant de lever les ambiguïtés au moyen d'une similarité sémantique.

Nous avons proposé une évaluation sur deux corpus pour deux langues différentes : français et anglais. Les résultats que nous avons obtenus sur le français ont été validés sur le corpus anglais SEMCOR qui possède un nombre plus important d'occurrences de mots annotés. Nous avons comparé la sélection des voisins distributionnels avec la sélection des voisins les plus proches linéairement. Sur l'ensemble des expériences menées pour l'anglais, nous avons remarqué que l'utilisation d'un nombre de 2 à 4 voisins distributionnels permet d'atteindre une meilleure désambiguïsation que la simple utilisation des voisins les plus proches linéairement. Cependant, cette approche de désambiguïsation basée sur BABELNET possède plusieurs limites liées à la ressource :

1. BABELNET possède une granularité très fine de sens et ne propose pas une représentation structurée des sens d'un mot donné. Par exemple, pour *souris (animal)*, BABELNET propose de nombreux sens sans une structure hiérarchique, parmi lesquels : '*espèce de petit rongeur*', '*genre de rongeur*' et '*rongeur*' alors qu'il s'agit d'un même concept. Cela rend la désambiguïsation encore plus difficile même si nous avons désambiguïsé correctement le nom *souris* pour notre expérience sur l'échantillon lexical français.
2. Le manque de définitions dans les sens de BABELNET. Cela est constaté le plus souvent pour les verbes français.
3. BABELNET repose sur l'utilisation d'un système de traduction automatique pour enrichir sa base de connaissances. Nous avons remarqué que l'utilisation des synonymes pour les sens n'ayant pas de définition pour une langue donnée a permis d'avoir des synonymes d'une autre langue sans avoir fait la demande. Ceci en peut créer d'autres ambiguïtés puisque des mots peuvent être monosémiques dans une langue et polysémiques dans une autre langue.
4. L'utilisation des synonymes n'aide pas le système de désambiguïsation à trouver le bon sens. En effet, en plus d'avoir des synonymes dans une autre langue que celle qui est souhaitée, nous nous retrouvons le plus souvent avec un petit nombre de synonymes et dans plusieurs cas avec un seul mot (mot-cible) dans un *Babel synset* donné.

Dans le chapitre suivant, nous décrivons un modèle de représentation de sens et de mots à base d'une autre ressource lexico-sémantique, à savoir JeuxDeMots (Lafourcade, 2007). Contrairement à BabelNet où les relations lexico-sémantiques sont décrites uniquement entre les sens, JeuxDeMots permet d'avoir les relations couvertes par BabelNet et, en plus, d'autres relations entre les mots et sens. Aussi, JeuxDeMots permet d'avoir une structure hiérarchique des sens, c'est-à-dire, le sens le plus général d'un concept donné se trouve dans le premier niveau et les sens les plus spécifiques du même concept se trouvent dans les niveaux supérieurs, contrairement à BabelNet où tous les sens d'un même concept se trouvent sur un même niveau.

Création et validation de signatures sémantiques de mots et de sens à base du réseau lexical JeuxDeMots

4.1. Motivation

L'intégration de la notion de similarité sémantique entre les mots et/ou sens est essentielle dans différentes applications du TAL. Comme nous l'avons présenté dans le chapitre 2, elle a reçu un intérêt considérable qui a eu comme conséquence le développement d'une vaste gamme d'approches pour en déterminer une mesure. Ainsi, plusieurs types de mesures de similarité existent. Les mesures de similarité sémantique utilisent différentes représentations obtenues à partir d'informations soit dans des ressources lexicales, soit dans de gros corpus de données ou bien dans les deux. Dans ce chapitre, nous nous intéressons à la création de signatures sémantiques décrivant des représentations vectorielles de mots et de sens à partir du réseau lexical JeuxDeMots proposé par Lafourcade (2007).

D'une manière générale, une signature sémantique peut être considérée comme une forme spéciale de représentation du modèle d'espace vectoriel (VSM, Vector Space Model) (Turney et Pantel, 2010). De la même façon que la représentation d'un élément linguistique à base du modèle VSM, le poids associé à une dimension dans une signature sémantique indique la pertinence ou l'importance de cette dimension pour l'élément linguistique. La différence principale est dans la manière dont les poids sont calculés. Dans une représentation à base du modèle VSM, chaque dimension correspond habituellement à un mot individuel dont le poids est souvent calculé sur les bases de la statistique des co-occurrences, tandis que dans une signature sémantique un élément linguistique est représenté comme une distribution de probabilités sur toutes les entités du réseau lexical utilisé (dans notre cas, JeuxDeMots) où les poids sont estimés

sur la base des propriétés structurelles de ce réseau. Nous avons choisi d'utiliser JeuxDeMots parce que c'est une ressource qui est disponible pour le français et qui a l'avantage de proposer des raffinements sémantiques selon une structure hiérarchique, ce que BabelNet ne propose pas. En d'autres termes, JeuxDeMots propose une granularité plus optimale des sens par rapport à BabelNet.

Dans ce chapitre, nous présentons tout d'abord dans la section 4.2 notre méthode de création de signatures de mots et de sens. Nous décrivons les relations lexico-sémantiques sur lesquelles nous nous basons (*cf.* sous-section 4.2.1) ainsi que les similarités sémantiques que nous utilisons pour comparer les signatures sémantiques (*cf.* sous-section 4.2.2). Nous présentons ensuite dans la section 4.3 l'évaluation de la qualité des signatures sémantiques créées. Cette évaluation utilise deux variantes : (1) évaluation intrinsèque sur les mesures de similarité sémantique (*cf.* sous-section 4.3.1) ; et (2) évaluation extrinsèque sur la substitution lexicale (*cf.* sous-section 4.3.2). Enfin, dans la conclusion, nous résumons les points les plus importants de ce chapitre (*cf.* section 4.4).

4.2. Création de signatures sémantiques de mots et de sens

La méthode que nous proposons s'appuie tout d'abord sur les propriétés individuelles de chaque élément linguistique qui se définit comme étant un nœud dans le réseau lexical JeuxDeMots, cela afin de décrire les nœuds liés à travers une représentation sémantique. Ce qui nous ramène par la suite à comparer les éléments linguistiques en termes de leurs représentations. Ces dernières sont typées, pondérées et appelées des signatures sémantiques.

Les signatures sémantiques peuvent être utilisées, par exemple, pour déterminer si la similarité sémantique entre *fournaise* et *four* est plus forte que la similarité entre *fournaise* et *instrument* ou pour savoir si le synonyme *prix* du mot *intérêt* représente un substitut pertinent par rapport au synonyme *avantage* dans un contexte décrivant le sens FINANCE pour le mot *intérêt*. Dans ce qui suit, nous présentons les différentes relations que nous utilisons pour la génération des signatures sémantiques ainsi que les mesures de similarité à utiliser pour leur évaluation.

4.2.1. Utilisation des relations lexico-sémantiques provenant de JeuxDeMots

Nous construisons différentes signatures pour chaque entrée lexicale dans JeuxDeMots. Une signature peut dépendre d'une seule relation r comme elle peut dépendre d'une combinaison de relations appartenant à un ensemble R. Les dimensions dans une signature sont les nœuds liés par relations sortantes à l'entrée lexicale. Dans le cas où $|R| = 1$, le poids d'une dimension indique

l'importance de cette dimension par rapport à la seule relation r pour une signature S. Si $|R| \geqslant 2$ alors le poids d'une dimension est la somme des poids de cette dimension sur l'ensemble des relations appartenant à R. Dans la ressource, les pondérations sont dans $\mathbb{R}$, c'est-à-dire qu'elles peuvent être positives comme elles peuvent être négatives. Nous prenons en considération seulement les pondérations positives (restriction à $\mathbb{R}^+$). La taille d'une signature n'est pas fixe et peut varier selon les liens sortants (instances des relations) de l'entrée lexicale vers les autres nœuds du réseau. Une signature peut être vue comme un vecteur où les dimensions inexistantes sont des dimensions ayant une valeur nulle.

Nous nous intéressons aux relations listées ci-dessous. Chacune est décrite, d'une part, avec un exemple et, d'autre part, avec son nombre d'instances dans le réseau lexical. Nous avons mentionné dans le chapitre 1 que JeuxDeMots est une ressource qui est en **constante évolution** depuis sa création en 2007 (*cf.* sous-section 1.2.1). Nous nous référons aux données collectées datant de Janvier 2017 et Janvier 2018 pour fournir le nombre d'instances de chaque relation dans le réseau[1]. Nous montrons que le réseau est devenu de plus en plus riche en connaissances. Dans la section 4.3, nous présentons une étude comparative entre l'utilisation de chaque version de la base lexicale.

— ***Synonyme*** : quels sont les termes décrivant un sens identique ou proche ? – relation lexicale (p.ex. *outil* → *instrument*) – **746 128** / **833 259** instances ($\approx +11.7\%$ de progression).

— ***Acception*** : quels sont les termes évoquant les sens possibles de la cible ? – relation associative (p.ex. *outil* → *dispositif*) – **93 899** / **98 692** instances ($\approx +5.1\%$ de progression).

— ***Domaine*** : quels sont les domaines auxquels peut appartenir la cible ? – relation sémantique (p.ex. *outil* → *mécanique*) – **670 055** / **1 009 863** instances ($\approx +50.7\%$ de progression).

— ***Agent*** : que peut faire ce sujet ? – relation prédicative (p.ex. *outil* ← *fonctionner*) – **1 055 271** / **1 498 818** instances ($\approx +42.0\%$ de progression).

— ***Patient*** : que peut-on faire avec cet objet ? – relation prédicative (p.ex. *outil* ← *fabriquer*) – **65 397** / **72 135** instances ($\approx +10.3\%$ de progression).

— ***Hyperonyme (Générique)*** : quels sont les termes associés aux génériques de la cible ? – relation sémantique (p.ex. *outil* → *matériel*) – **3 155 763** / **4 492 305** instances ($\approx +42.4\%$ de progression).

— ***Hyponyme (Spécifique)*** : quels sont les termes associés aux spécifiques de la cible ? – relation sémantique (p.ex. *outil* → *marteau*) – **712 125** / **782 852** instances ($\approx +9.9\%$ de progression).

1. http://www.jeuxdemots.org/JDM-LEXICALNET-FR/?C=M;O=D

— ***Idée associée*** : quels sont les termes que nous pouvons associer librement à la cible ? – relation associative (p.ex. *outil* → *bricolage*) – **17 768 142** / **105 133 268** instances (une progression de près de **6** fois plus) [2].

D'autre part, le lien entre les termes polysémiques et leurs sens est défini par la relation *Raffinement sémantique* :

— ***Raffinement sémantique*** : quels sont les sens ou raffinements sémantiques de la cible ? – relation sémantique (p.ex. *outil* → *outil* (*instrument*)) – **65 029** / **70 234** instances ($\approx +8.0\%$ de progression).

Un terme monosémique n'a pas de raffinement sémantique. Nous utilisons cette relation de *Raffinement sémantique* pour connaître les sens possibles d'un terme polysémique donné. Prenons un autre exemple : actuellement, le réseau propose pour le nom *barrage* 7 raffinements sémantiques : (1) *barrage (ouvrage d'art)* ; (2) *barrage (tir de barrage)* ; (3) *barrage (match de barrage)* ; (4) *barrage (rocher)* ; (5) *barrage (barrière)* ; (6) *barrage (droit de péage)* ; et (7) *barrage (équitation)*. Le cinquième raffinement sémantique *barrage (barrière)* possède lui-même un autre raffinement sémantique : *barrage (barrière, police)* (Lafourcade, 2011, p. 125). Ce dernier raffinement se présente dans un deuxième niveau hiérarchique alors que les 7 autres raffinements sont tous sur un même premier niveau. Ces raffinements sémantiques sont aussi des nœuds dans le réseau et peuvent avoir des liens vers des mots comme vers d'autres sens. Nous utilisons la relation de *Raffinement sémantique* principalement pour connaître les sens possibles, nous ne l'utilisons pas pour créer des dimensions dans les signatures de mots et de sens.

Nous construisons, d'une part, une signature pour chaque relation listée ci-dessus et cela pour les mots et les raffinements sémantiques. D'autre part, cinq autres signatures sont construites en utilisant une combinaison de relations. La première utilise une combinaison de deux relations, à savoir, *Synonyme* et *Acception*. La deuxième combine deux relations, à savoir, *Hyperonyme* et *Hyponyme*. La troisième combine aussi deux relations, à savoir, *Agent* et *Patient*. La quatrième ne privilège aucune relation, toutes les relations ont un coefficient égal à 1, tandis que la cinquième, donne une importance supérieure à certaines relations :

• **Signature par combinaison de relations sans coefficient** : nous utilisons toutes les relations listées ci-dessus dont le poids de chaque dimension pour chaque relation est représenté avec un même coefficient.

2. Il s'agit de la relation contenant le plus grand nombre d'instances puisqu'elle englobe tous les termes faisant penser à une entrée lexicale donnée. Aussi, il s'agit de la relation qui grandit le plus rapidement.

• **Signature par combinaison de relations avec coefficient** : toutes les relations listées ci-dessus sont utilisées dont le poids de chaque dimension pour chaque relation est multiplié par un coefficient différent : {"*Domaine*" : 6, "*Acception*" : 5, "*Hyperonyme*" : 4, "*Hyponyme*" : 4, "*Synonyme*" : 3, "*Agent*" : 2, "*Patient*" : 2, "*Idée associée*" : 1}.

Les signatures sont par la suite normalisées. Plusieurs normes ont été présentées dans la littérature telles que la norme euclidienne, la norme 1 qui présente la somme des valeurs absolues des dimensions ou la norme infinie. La norme 1 est la norme adéquate au sens où nous voulons garder l'aspect de distribution de probabilité. Pour la normalisation de nos signatures, nous avons choisi d'utiliser la norme infinie. Elle se définit par la fonction : $\|\vec{X}\|_\infty = max(|x_1|, |x_2|, \ldots, |x_n|)$. Elle a l'avantage de comparer proportionnellement tous les poids des dimensions à la dimension possédant le poids maximal. Cette dimension aura une valeur égale à 1 et toutes les autres dimensions auront une valeur appartenant à l'intervalle $]0, 1]$. Un seuil est fixé à 0.01 pour la validation des dimensions des signatures. Toutes les dimensions ayant une valeur inférieure au seuil sont ignorées.

Si nous supposons que la fonction $Sim(A, B, R)$ retourne la valeur normalisée du poids de la dimension B dans la signature S_A du terme A construite à partir de l'ensemble des relations appartenant à R et si nous prenons la signature de l'entrée lexicale *intérêt* avec comme relations seulement la relation *Synonyme*, la fonction $Sim($*intérêt, prix, Synonyme*$)$ retourne une valeur de 0.98 et la fonction $Sim($*intérêt, avantage, Synonyme*$)$ retourne une valeur de 0.93, cela en utilisant les données de la base datant de Janvier 2018.

4.2.2. Similarité entre les signatures sémantiques

Une fois que nous avons obtenu une signature sémantique pour chaque entrée lexicale se trouvant dans JeuxDeMots, nous pouvons calculer la similarité entre deux entrées lexicales en comparant leurs signatures sémantiques correspondantes. Nous adoptons deux techniques pour cette comparaison :

(1) Cosinus ; et (2) Weighted Overlap (Pilehvar *et al.*, 2013).

Ces deux techniques sont décrites dans le chapitre 2 (*cf.* section 2.2). Soit S_A et S_B deux signatures sémantiques à comparer. Ces signatures décrivent des représentations vectorielles pour deux éléments A et B, respectivement. Les équations 4.1 et 4.2 décrivent les fonctions Cosinus et Weighted Overlap (WO), respectivement, pour comparer ces signatures.

$$Sim_{Cos}(S_A, S_B) = \frac{S_A \,.\, S_B}{\|S_A\| \, \|S_B\|} \tag{4.1}$$

$$Sim_{WO}\,(S_A, S_B) = \frac{\sum_{d \in D} \left(r_d(S_A) + r_d(S_B)\right)^{-1}}{\sum_{i=1}^{|D|} (2i)^{-1}} \tag{4.2}$$

Nous utilisons en plus de ces deux techniques quatre fonctions d'activation, que nous appelons aussi configurations, pour comparer nos signatures sémantiques. Les fonctions Act_1 et Act_2 numérotées en (4.3) et (4.4), respectivement, retournent une valeur de 1 si $A \in S_B$ ou $B \in S_A$.

$$Act_1(A, B, R) = \begin{cases} 1 \; si \; A \in S_B \; ou \; B \in S_A \\ Sim_{Cos}(S_A, S_B) \; sinon \end{cases} \tag{4.3}$$

$$Act_2(A, B, R) = \begin{cases} 1 \; si \; A \in S_B \; ou \; B \in S_A \\ Sim_{WO}(S_A, S_B) \; sinon \end{cases} \tag{4.4}$$

Les deux fonctions Act_3 et Act_4 numérotées en (4.5) et (4.6), respectivement, prennent une forme similaire à la fonction d'activation proposée par Lafourcade (2011), p. 21-22 (*cf.* chapitre 2, sous-section 2.2.2).

$$Act_3(A, B, R) = max(A[B],\ B[A], Sim_{Cos}(S_A, S_B)) \tag{4.5}$$

$$Act_4(A, B, R) = max(A[B],\ B[A], Sim_{WO}(S_A, S_B)) \tag{4.6}$$

La différence entre Act_1, Act_2 et Act_3, Act_4 décrites ci-dessous est dans la manière de calculer la similarité au cas où $A \in S_B$ ou $B \in S_A$. Si nous reprenons l'exemple des mots *intérêt* et *prix* avec *Synonyme* comme relation, nous avons $S_{intérêt}[prix] = 0.98$ et $S_{prix}[intérêt] = 0.27$. Les fonctions Act_1 et Act_2 retournent une valeur égale à 1 puisque *prix* $\in \text{Synonymes}_{intérêt}$ comme *intérêt* $\in \text{Synonymes}_{prix}$ tandis que la fonction Act_3 retourne $max(0.98, Sim_{Cos}(S_{intérêt}, S_{prix}))$ et la fonction Act_4 retourne $max(0.98, Sim_{WO}(S_{intérêt}, S_{prix}))$.

4.3. Évaluation de la qualité des signatures sémantiques

Après avoir obtenu les signatures sémantiques, nous en avons évalué la qualité. L'évaluation de ces signatures est réalisée sur deux tâches différentes (Billami et Gala, 2017) :

(a) Évaluation intrinsèque portant sur les mesures de similarité sémantique (*cf.* sous-section 4.3.1).

(b) Évaluation extrinsèque portant sur la substitution lexicale (*cf.* sous-section 4.3.2).

Nous comparons nos représentations vectorielles avec deux modèles provenant des systèmes de l'état de l'art. Le premier repose sur des représentations de sens, à savoir NASARI (Camacho-Collados *et al.*, 2015) tandis que le deuxième repose sur des représentations de mots dans un espace vectoriel continu, *word embeddings* ou *plongements de mots* (*cf.* chapitre 1, sous-section 1.3.1).

Afin de mesurer la similarité sémantique entre mots par utilisation de NASARI, nous nous basons sur l'algorithme de CAMACHO-COLLADOS *et al.* (2015), p. 570. Cet algorithme propose une mesure égale à 1 si les deux mots à comparer sont synonymes [3]. Il intègre la mesure WEIGHTED OVERLAP comme mesure de similarité entre sens. Nous utilisons une deuxième instance de cet algorithme avec la mesure COSINUS.

Afin de mesurer la similarité sémantique entre mots par utilisation des *word embeddings*, nous utilisons un jeu de vecteurs basé sur une variante de l'algorithme GLOVE (PENNINGTON *et al.*, 2014) proposée par l'équipe Alpage pour le français et que nous appellerons par la suite DEPGLOVE [4]. Nous utilisons la mesure COSINUS comme mesure de similarité.

4.3.1. Évaluation intrinsèque : mesures de similarité sémantique

Nous évaluons la qualité de nos signatures sémantiques par rapport à un jugement humain. Pour cela, nous calculons la corrélation entre les scores retournés par les mesures décrites dans la sous-section 4.2.2 et les annotations humaines. L'idée est de voir si les scores obtenus sont fortement corrélés avec ceux donnés par les humains. Nous avons opté pour l'utilisation de la liste de référence RG–65 pour le français (JOUBARNE et INKPEN, 2011) qui décrit une liste de paires de mots. Elle présente une traduction avec un autre jugement humain pour le jeu de données original RG–65 créé pour l'anglais (RUBENSTEIN et GOODENOUGH, 1965).

Pour l'évaluation de la qualité des signatures de sens, nous nous basons sur l'hypothèse de BUDANITSKY et HIRST (2006) : « la similarité entre deux mots est celle de leurs sens les plus proches ». La fonction Sim_{Sens} décrite dans l'équation 4.7 retourne le score de similarité entre les sens les plus proches de deux mots w_1 et w_2.

$$Sim_{Sens}(w_1, w_2) = \underset{s_i \in Sens\ (w_1),\ s_j \in Sens\ (w_2)}{\textbf{arg max}} Sim\ (s_i, s_j) \tag{4.7}$$

$Sens\ (w_1)$ décrit l'ensemble des sens du mot w_1 et $Sens\ (w_2)$ décrit l'ensemble des sens du mot w_2. Si $w_t\ (t \in \{1,\ 2\})$ est polysémique alors $Sens\ (w_t)$ représente l'ensemble des raffinements sémantiques du premier niveau, sinon si le mot n'a pas de raffinements sémantiques alors la signature du mot w_t est celle qui est prise dans la comparaison. La fonction $Sim\ (s_i, s_j)$ est une fonction qui

3. L'algorithme utilise WIKTIONNAIRE comme base de synonymes. Cette ressource est intégrée dans BABELNET. Nous avons fait le choix d'utiliser les mêmes ressources que CAMACHO-COLLADOS *et al.* (2015) utilisent pour l'algorithme. Nous avons utilisé la version 3.7 de BABELNET comme base de synonymes (http://babelnet.org/download).

4. http://alpage.inria.fr/depglove/process.pl

joue le rôle de Cosinus, WO ou l'une des quatre configurations que nous avons proposées.

Nous avons pris la liste des paires telle qu'elle est fournie par Joubarne et Inkpen (2011)[5]. Parmi les 65 paires traduites, la traduction directe pour chaque mot de deux paires a retourné le même mot. Il s'agit de la paire *(cock, rooster)* traduite en *(coq, coq)* et de la paire *(cemetery, graveyard)* traduite en *(cimetière, cimetière)*. Ces deux paires ne sont pas utilisées pour l'évaluation. Le tableau 4.1 décrit l'ensemble de noms communs se trouvant dans RG–65. Nous présentons dans le tableau 4.2 les résultats de corrélations obtenus par utilisation des différentes signatures de mots. Le tableau 4.3 présente le même type de résultats par utilisation des signatures de sens ; quant à la liste des 63 paires, avec les scores du jugement humain et les scores retournés automatiquement par utilisation de nos signatures, elle figure dans l'annexe A.

asile	*bois*	*coussin*	*fournaise*	*grimace*	*moine*	*outil*	*signature*
asylum	*cimetière*	*dîner*	*frère*	*grue*	*monticule*	*périple*	*sorcier*
auto	*colline*	*esclave*	*fruit*	*instrument*	*nourriture*	*refuge*	*sourire*
autographe	*coq*	*ficelle*	*garçon*	*joyau*	*oiseau*	*rivage*	*trip*
automobile	*corde*	*forêt*	*gars*	*magicien*	*oracle*	*sage*	*verre*
bijou	*côte*	*four*	*goblet*	*midi*	*oreiller*	*serf*	*voyage*

Table 4.1. – Les 48 mots du vocabulaire correspondant au jeu de données RG–65 pour le français

Le vocabulaire du jeu de données n'a pas une couverture parfaite sur l'ensemble des signatures. Par exemple, les mots *asylum* et *goblet* ne sont associés à aucun nœud dans le réseau lexical et n'ont aucune signature. La raison est qu'ils représentent une mauvaise traduction en français ou que la traduction n'a jamais eu lieu. Le nom *asylum* peut se traduire par *refuge* ou *asile* et le nom *goblet* s'écrit *gobelet* en français. Nous appelons *asylum* et *goblet* des OOV (*Out-Of-Vocabulary*). Il y a seulement 46 mots sur 48 (95.83%) ayant des signatures pour les types suivants : {*synonyme, combinaison de synonyme avec acception (r_synonyme_acception), idée associée, combinaison de toutes les relations sans coefficient (r_traits_égaux), combinaison de toutes les relations avec coefficient (r_traits_avec_coeff)*}.

Nous constatons sur les résultats du tableau 4.2 que l'utilisation des données de la base lexicale de JeuxDeMots datant de Janvier 2018 donne de meilleurs résultats que l'utilisation des données de Janvier 2017 et cela pour toutes les techniques et fonctions d'activation utilisées. Sans surprise, le nombre élevé de données de la base lexicale datant de Janvier 2018 permet d'obtenir de meilleurs résultats.

5. http://www.site.uottawa.ca/∼mjoub063/wordsims.htm

Type de signature	Date	Couverture (%)	COS	WO	$\mathbf{Act_1}$	$\mathbf{Act_2}$	$\mathbf{Act_3}$	$\mathbf{Act_4}$
r_synonyme	01/18	**95.83**	0.73	0.59	**0.88**	0.85	**0.89**	0.85
	01/17	**95.83**	0.73	0.59	**0.88**	0.85	**0.89**	0.85
r_acception	01/18	79.17	0.39	0.36	0.77	0.79	0.76	0.78
	01/17	77.08	0.38	0.34	0.80	0.80	0.79	0.80
r_synonyme_acception	01/18	**95.83**	0.73	0.73	**0.88**	**0.88**	0.87	**0.87**
	01/17	**95.83**	0.72	**0.74**	**0.88**	**0.88**	0.87	**0.87**
r_domaine	01/18	87.5	0.41	0.36	0.53	0.48	0.50	0.42
	01/17	81.25	0.33	0.27	0.43	0.39	0.40	0.32
r_agent	01/18	58.33	0.37	0.48	0.37	0.48	0.37	0.48
	01/17	58.33	0.23	0.35	0.23	0.35	0.23	0.35
r_patient	01/18	60.42	0.51	0.57	0.51	0.57	0.51	0.57
	01/17	60.42	0.51	0.47	0.51	0.47	0.51	0.47
r_agent_patient	01/18	72.92	0.34	0.45	0.34	0.45	0.34	0.45
	01/17	72.92	0.25	0.33	0.25	0.33	0.25	0.33
r_hyperonyme	01/18	85.42	0.28	0.36	0.57	0.63	0.48	0.54
	01/17	85.42	0.22	0.29	0.46	0.51	0.43	0.48
r_hyponyme	01/18	89.58	0.33	0.36	0.54	0.53	0.49	0.48
	01/17	87.5	0.33	0.36	0.45	0.45	0.44	0.44
r_hyperonyme_hyponyme	01/18	91.67	0.22	0.26	0.52	0.54	0.46	0.48
	01/17	89.58	0.18	0.25	0.44	0.48	0.43	0.47
r_idée_associée	01/18	**95.83**	**0.77**	**0.74**	**0.88**	**0.88**	0.84	0.84
	01/17	**95.83**	0.68	0.67	**0.88**	**0.88**	0.82	0.81
r_traits_avec_coeff	01/18	**95.83**	0.61	0.61	0.86	0.86	0.83	0.82
	01/17	**95.83**	0.56	0.65	0.86	0.86	0.81	0.81
r_traits_égaux	01/18	**95.83**	0.72	0.73	0.87	0.87	0.85	0.85
	01/17	**95.83**	0.66	0.70	0.87	0.87	0.85	0.85

Table 4.2. – Corrélations de Pearson obtenues selon différentes signatures de mots avec différentes techniques et configurations

Les fonctions d'activation permettent d'avoir de meilleures corrélations que l'utilisation des techniques Cosinus et WO. Pour l'utilisation de ces techniques, les meilleures corrélations sont celles où les relations suivantes sont utilisées : *Idée associée* et *r_synonyme_acception*.

Type de signature	Date	COS	WO	$\mathbf{Act_1}$	$\mathbf{Act_2}$	$\mathbf{Act_3}$	$\mathbf{Act_4}$
r_synonyme	01/18	**0.32**	0.33	0.52	0.52	0.52	0.52
	01/17	0.22	0.22	0.45	0.45	0.45	0.45
r_domaine	01/18	0.23	0.23	0.23	0.23	0.23	0.23
	01/17	0.20	0.22	0.20	0.22	0.20	0.22
r_agent	01/18	0.29	0.36	0.29	0.36	0.29	0.36
	01/17	0.23	0.28	0.23	0.28	0.23	0.28
r_patient	01/18	0.13	0.13	0.13	0.13	0.13	0.13
	01/17	0.19	0.21	0.19	0.21	0.19	0.21
r_agent_patient	01/18	0.22	0.30	0.22	0.30	0.22	0.30
	01/17	0.18	0.22	0.18	0.22	0.18	0.22
r_hyperonyme	01/18	0.15	0.16	0.21	0.22	0.21	0.21
	01/17	0.14	0.15	0.26	0.27	0.20	0.21
r_hyponyme	01/18	0.19	0.16	0.27	0.27	0.28	0.28
	01/17	0.13	0.17	0.27	0.27	0.27	0.27
r_hyperonyme_hyponyme	01/18	0.12	0.15	0.22	0.22	0.22	0.22
	01/17	0.12	0.15	0.27	0.27	0.22	0.22
r_idée_associée	01/18	0.29	0.35	0.56	0.58	**0.56**	0.59
	01/17	0.29	0.37	0.56	0.59	**0.56**	0.60
r_traits_avec_coeff	01/18	0.25	**0.44**	0.55	**0.65**	0.46	0.58
	01/17	0.22	0.31	0.57	0.60	0.45	0.49
r_traits_égaux	01/18	0.28	0.42	0.56	0.64	0.53	**0.62**
	01/17	0.26	0.34	**0.58**	0.61	0.52	0.56

Table 4.3. – Corrélations de Pearson obtenues selon différentes signatures de sens avec différentes techniques et configurations

Pour l'utilisation des fonctions d'activation, les signatures de mots à base de synonymie permettent d'obtenir les meilleures corrélations (il est à noter que cette relation propose des éléments lexicaux qui vont au-delà de la définition stricte de synonymie).

Nous obtenons une corrélation de 0.89 par utilisation de la troisième fonction d'activation et 0.88 par utilisation de la première fonction. La combinaison de la relation de *Synonyme* avec *Acception* ne retourne pas une corrélation meilleure (Act_1 et Act_3) que la simple utilisation de la relation de *Synonyme*. La raison est que nous obtenons par cette combinaison des signatures avec un plus grand nombre de dimensions. Le chevauchement entre les signatures n'est pas plus

grand que l'utilisation d'une seule relation. Il en est de même pour la combinaison des relations *Hyperonyme* et *Hyponyme*. Cependant, ces deux dernières relations ne permettent pas d'obtenir une meilleure couverture. JeuxDeMots (à ce jour) ne propose aucun terme générique ou spécifique pour les mots suivants : {*asylum, autographe, goblet, trip*}. Pour la relation *Idée associée*, la corrélation obtenue reste relativement supérieure par rapport aux autres relations ou combinaison de relations hors *r_synonyme* et *r_synonyme_acception*.

Les résultats obtenus pour l'évaluation sur les signatures de sens (*cf.* tableau 4.3) sont moins bons par rapport à l'utilisation des signatures de mots. Cela est lié, d'une part, à la fonction Sim_{Sens} définie dans l'équation 4.7 et, d'autre part, à la taille des signatures de sens. En effet, les signatures de mots que nous avons obtenues sont de taille plus importante que les signatures de sens. La meilleure corrélation obtenue par utilisation des signatures de sens est de 0.65 (deuxième fonction d'activation et signatures par combinaison de relations avec coefficient). Pour ces expériences, utiliser la fonction WO à la place d'un Cosinus est plus rentable pour la comparaison de deux signatures de sens. Cela en raison du petit nombre de dimensions dont tiennent compte ces signatures. La mesure cosinus a tendance à retourner des scores relativement faibles lorsque les dimensions des deux signatures à comparer sont petites, contrairement à la mesure WO qui n'est pas affectée par le nombre de dimensions.

Pour la comparaison avec les systèmes de l'état de l'art, nous utilisons les signatures construites avec les types suivants :

$\rightarrow$ *r_traits_avec_coeff, r_traits_égaux, r_idée_associée et r_synonyme*.

Il est à noter que même les modèles NASARI et DEPGLOVE ne permettent pas d'avoir une couverture parfaite. Par exemple, NASARI ne fournit aucune représentation de sens pour le nom *trip* et DEPGLOVE ne fournit aucun vecteur dans son espace vectoriel continu pour les noms *asylum* et *goblet*. Cela nous amène à mettre à l'écart trois paires contenant au moins l'un de ces noms [6] pour permettre une comparaison sur un même ensemble de paires couvertes par tous les modèles. Le tableau 4.4 présente les résultats de corrélation obtenus pour cette comparaison. Cette comparaison s'effectue par utilisation de la première fonction d'activation.

> Les résultats obtenus montrent clairement que les signatures de mots et l'utilisation de la première configuration permettent d'avoir une corrélation largement supérieure aux systèmes à base de corpus.

Une corrélation de Pearson (r) de 0.88 est obtenue sur les 60 paires pour la signature typée avec la relation *Idée associée* contre 0.82 pour NASARI à base

6. Rappelons que deux paires ont été déjà ignorées à cause de la traduction qui a retourné le même mot pour les deux éléments de la paire, ce qui nous ramène à garder 60 paires au final.

de la mesure *WO* ou 0.48 pour DEPGLOVE. Pour la corrélation de Spearman (ρ), nous avons obtenu aussi une valeur de 0.88 pour *r_traits_égaux* contre une valeur de 0.78 pour $\text{NASARI}_{\text{WO}}$, 0.77 pour $\text{NASARI}_{\text{COS}}$ et 0.50 pour DEPGLOVE.

Système	**Date**	**Représentation sémantique**	**Corrélation de Pearson** $(\mathbf{r})$	**Corrélation de Spearman** (ρ)
r_idée_associée	01/18	Mots	**0.88**	0.85
	01/18	Sens	0.55	0.55
r_traits_avec_coeff	01/18	Mots	0.86	0.87
	01/18	Sens	0.55	0.57
r_traits_égaux	01/18	Mots	0.87	**0.88**
	01/18	Sens	0.55	0.60
r_synonyme	01/18	Mots	**0.88**	0.75
	01/18	Sens	0.52	0.46
DEPGLOVE	10/16	Mots	0.48	0.50
$\text{NASARI}_{\text{COS}}$	04/15	Sens	0.80	0.77
$\text{NASARI}_{\text{WO}}$	04/15	Sens	0.82	0.78

Table 4.4. – Corrélations de Pearson et Spearman obtenues selon différentes signatures avec utilisation de la première configuration, comparaison avec les résultats obtenus par NASARI et DEPGLOVE sur un ensemble de 60 paires couvertes par tous les systèmes

Joubarne et Inkpen (2011) ont utilisé deux mesures de similarité sémantique à base de corpus : (1) Pointwise Mutual Information (pmi) ; et (2) Second Order Co-occurrence Pointwise Mutual Information (soc-pmi). Le principe de la pmi est d'estimer si l'apparition simultanée de deux mots A et B est supérieure à la probabilité d'apparition *à priori* des deux mots indépendamment ; quant à la soc–pmi, il s'agit du même principe en tenant compte des mots communs apparaissant dans le voisinage de A et B selon une fenêtre contextuelle définie à la base. Les corrélations de Pearson obtenues entre ces mesures et les 18 évaluateurs humains pour l'ensemble des 63 paires sont de 0.29 pour la pmi et de 0.17 pour la soc–pmi.

4.3.2. Évaluation extrinsèque : substitution lexicale

La substitution lexicale est une tâche qui, ces dernières années, a reçu un intérêt majeur au sein de la communauté du TAL. D'abord, une première campagne d'évaluation, SemEval–2007, a vu le jour pour l'anglais (McCarthy et Navigli, 2009) ; ensuite une adaptation de cette dernière a été présentée pour

le français (Fabre *et al.*, 2014) dans l'atelier de Sémantique Distributionnelle[7] (SemDis), basée sur des données issues du corpus français FRWAC (Baroni *et al.*, 2009). Le principe est de remplacer un mot-cible par un substitut potentiel tout en gardant le même sens du mot-cible par rapport à un contexte donné.

La substitution lexicale a un double intérêt pour l'évaluation de la similarité sémantique :

(a) Elle représente une évaluation extrinsèque pour laquelle la similarité sémantique à un rôle prépondérant pour que des différences la concernant puissent être observées vis-à-vis de la tâche de substitution.

(b) Le niveau contextuel est pris en compte.

Cette tâche se décompose elle-même en deux sous-tâches (Fabre *et al.*, 2014 ; Ferret, 2014b) :

1. Génération de candidats substituts pour le mot-cible à remplacer.
2. Choix de l'un des candidats en fonction du contexte.

Le jeu d'évaluation fourni dans SemDis comporte 30 unités lexicales (10 noms, 10 verbes et 10 adjectifs). Pour chaque mot-cible, 10 phrases différentes ont été proposées (300 phrases au total). Pour chaque phrase, il est possible de fournir jusqu'à 10 substituts au maximum classés par ordre décroissant de préférence. Les données ont été fournies par la suite avec des annotations manuelles. Le tableau 4.5 décrit les mots-cibles à substituer.

Noms	**Verbes**	**Adjectifs**
affection, capacité, couverture, débit, direction, don, espace, intérêt, montée, vaisseau	*arrêter, commander, entraîner, éplucher, essuyer, faucher, fonder, interpréter, maintenir, taper*	*aisé, compris, grossier, hermétique, incorrect, mince, modeste, obscur, riche, vaseux*

Table 4.5. – Les 30 mots-cibles pour la tâche de substitution lexicale

Pour la première sous-tâche qui consiste à générer des candidats substituts, nous prenons les signatures de mots construites à base de la relation de *Synonyme*. Pour une entrée lexicale donnée, les dimensions de sa signature représentent des substituts potentiels. Nous avons fait le choix de présélectionner les candidats en tenant compte seulement des synonymes ayant un poids d'importance supérieur ou égal à la valeur de 0.8, cela afin de tenir compte seulement des termes représentant des synonymes stricts.

Pour la deuxième sous-tâche et dans cette partie, notre objectif est non pas de développer un modèle sophistiqué de substitution lexicale mais plutôt de comparer l'utilisation de nos représentations sémantiques avec un modèle utilisant

7. https://www.irit.fr/semdis2014/fr/task1.html

un algorithme comme celui décrit par FERRET (2014b). Cet algorithme consiste à mesurer la similarité entre chaque candidat substitut et l'ensemble de mots pleins de la phrase contenant le mot-cible à remplacer, hors ce dernier. Par la suite, nous appellerons cet algorithme *Sub_Lex*. Pour des raisons calculatoires et afin d'éviter une explosion combinatoire, nous évaluons ici seulement les signatures de mots. Nous n'utilisons pas la fonction Sim_{Sens} décrite dans l'équation 4.7 mais plutôt directement nos techniques et configurations sur les signatures de mots. Afin d'obtenir l'ensemble de mots pleins, nous avons réalisé une analyse morpho-syntaxique avec l'outil TALISMANE [8] (URIELI, 2013) sur l'ensemble des phrases du corpus SEMDIS.

Mesures d'évaluation

Il s'agit de mesures utilisées dans SEMEVAL–2007 pour la tâche de substitution lexicale, à savoir la mesure *best* et la mesure *oot (out of ten)* [9].

- **best** : le système est évalué par rapport à la première substitution proposée. Le meilleur score renvoie le substitut choisi majoritairement par les annotateurs.
- **oot** *(out of ten)* : le système est évalué par rapport à tous les substituts proposés (dans la limite de 10). Le meilleur score obtenable correspond au nombre maximum de réponses couvertes par les annotateurs.

Résultats d'expérimentation

Nous avons testé les mesures COSINUS, WEIGHTED OVERLAP ainsi que les quatre configurations sur trois signatures à base des types suivants : {*idée associée, combinaison de toutes les relations sans coefficient, combinaison de toutes les relations avec coefficient*}. Le tableau 4.6 présente les résultats que nous obtenons. Le tableau 4.7 présente une comparaison de nos meilleurs résultats avec ceux retournés par les systèmes ayant participé à l'atelier SEMDIS ainsi qu'avec l'utilisation du modèle DEPGLOVE.

Nous avons comparé l'utilisation de nos représentations avec DEPGLOVE seulement. Nous ne pouvons pas appliquer l'algorithme de FERRET (2014b) en utilisant NASARI car ce dernier propose des représentations vectorielles de sens seulement pour les noms. Son utilisation, dans ce cas, permet d'évaluer seulement les noms et réduit le contexte en comparant un candidat substitut seulement avec les noms du contexte.

Les résultats du tableau 4.6 sont classés par catégorie grammaticale et par ordre décroissant du score *best* sur l'ensemble des mots à substituer (*Total*) du corpus SEMDIS. Il en est de même pour le tableau 4.7. Ce dernier regroupe

8. http://redac.univ-tlse2.fr/applications/talismane.html

9. Pour comprendre mieux le fonctionnement de ces mesures, une description détaillée est proposée par FABRE *et al.* (2014), p. 201 et peut être consultée.

Système	Date	best				oot			
		Nom	*Adj.*	*Verbe*	*Total*	*Nom*	*Adj.*	*Verbe*	*Total*
JDM_TraitsÉgaux_FctAct4	01/18	.069	.067	.115	**.084**	.282	.316	**.348**	**.315**
	01/17	.071	.063	.098	.077	.269	.331	.325	.308
JDM_TraitsÉgaux_FctAct3	01/18	.067	**.075**	.107	.083	.275	.336	.319	.310
	01/17	.078	.059	.100	.079	.261	.339	.323	.308
JDM_TraitsÉgaux_FctAct1	01/18	.077	.059	.112	.083	.268	.337	.316	.307
	01/17	.075	.053	.099	.076	.258	.340	.322	.307
JDM_TraitsIdéeAssociée_FctAct3	01/18	.058	.065	.122	.082	.282	.333	.321	.312
	01/17	.060	.063	.095	.073	.270	.318	.323	.304
JDM_TraitsÉgaux_FctAct2	01/18	.073	.056	**.119**	.082	.273	**.349**	.311	.311
	01/17	.068	.053	.111	.077	.269	.332	.322	.308
JDM_TraitsAvecCoeff_FctAct2	01/18	.075	.051	.115	.080	.264	.343	.314	.307
	01/17	.066	.052	.096	.071	.264	.334	.324	.307
JDM_TraitsAvecCoeff_FctAct3	01/18	.076	.065	.097	.079	.260	.326	.315	.300
	01/17	.078	.070	.075	.074	.260	.340	.322	.307
JDM_TraitsAvecCoeff_FctAct1	01/18	.079	.045	.111	.078	.260	.327	.317	.301
	01/17	.077	.056	.094	.076	.259	.341	.326	.309
JDM_TraitsIdéeAssociée_FctAct1	01/18	.063	.055	.113	.077	.274	.335	.322	.310
	01/17	.060	.047	.097	.068	.268	.318	.326	.304
JDM_TraitsIdéeAssociée_FctAct4	01/18	.064	.053	.112	.077	.274	.331	.314	.306
	01/17	.067	.046	.093	.069	**.283**	.317	.344	**.315**
JDM_TraitsIdéeAssociée_FctAct2	01/18	.068	.050	.108	.075	.266	.333	.321	.307
	01/17	**.081**	.054	.092	.076	.277	.317	.346	.313
JDM_TraitsAvecCoeff_FctAct4	01/18	.062	.064	.096	.074	.263	.341	.315	.306
	01/17	.058	.058	.066	.061	.261	.334	.327	.307
*JDM_TraitsIdéeAssociée*_COS	01/18	.054	.047	.106	.069	.271	.326	.314	.304
	01/17	.047	.043	.090	.060	.277	.337	.294	.302
*JDM_TraitsIdéeAssociée*_WO	01/18	.068	.052	.088	.069	.268	.327	.311	.302
	01/17	.060	.037	.086	.061	.265	.342	.312	.306
*JDM_TraitsÉgaux*_COS	01/18	.047	.054	.092	.064	.261	.331	.311	.301
	01/17	.040	.038	.078	.052	.258	.330	.308	.299
*JDM_TraitsÉgaux*_WO	01/18	.043	.052	.086	.060	.272	.341	.313	.309
	01/17	.046	.043	.080	.056	.277	.333	.320	.310
*JDM_TraitsAvecCoeff*_COS	01/18	.045	.034	.091	.057	.249	.320	.300	.289
	01/17	.040	.033	.080	.051	.243	.324	.308	.292
*JDM_TraitsAvecCoeff*_WO	01/18	.042	.041	.077	.053	.259	.328	.313	.300
	01/17	.048	.030	.072	.050	.261	.330	.315	.302
baseline_jdmsyn	01/18	.034	.006	.055	.031	.231	.253	.273	.252
	01/17	.029	.006	.051	.029	.247	.258	.303	.269

Table 4.6. – Résultats pour la tâche de substitution lexicale selon différentes signatures avec différentes mesures

d'autres systèmes pour lesquels certains utilisent le même algorithme que le notre (notés avec *Sub_Lex*). Nous utilisons une *baseline*, *baseline_jdmsyn*, consistant à renvoyer les 10 premiers synonymes d'un mot-cible par ordre d'importance depuis les signatures à base de la relation *Synonyme*. Durant l'atelier SemDis,

Système	best				oot			
	Nom	*Adj.*	*Verbe*	*Total*	*Nom*	*Adj.*	*Verbe*	*Total*
JDM_TraitsÉgaux_FctAct4	.069	.067	.115	**.084**	**.282**	.316	**.348**	**.315**
JDM_TraitsÉgaux_FctAct3	.067	**.075**	.107	.083	.275	.336	.319	.310
JDM_TraitsÉgaux_FctAct1	**.077**	.059	.112	.083	.268	.337	.316	.307
JDM_TraitsÉgaux_FctAct2	.073	.056	**.119**	.082	.273	**.349**	.311	.311
DEPGLOVE ***(Sub_Lex)***	.017	.053	.033	.034	.242	.331	.280	.284
Proxteam_JDM_Syn	.110	.106	.075	.097	.398	.429	.379	.402
CEA_list-word_cos_sent ***(Sub_Lex)***	.075	.074	.076	.075	.195	.245	.268	.236
Proxteam_AxeParaProx_JDM_Syn	.055	.054	.087	.065	.311	.396	.363	.357
Alpage_WoDiS	.054	.072	.061	.063	.191	.211	.213	.205
Proxteam_LM	.052	.040	.061	.051	.233	.166	.237	.212
baseline_Campagne	.044	.040	.052	.045	.294	.336	.344	.325
CEA_list-fredist_cos_sent ***(Sub_Lex)***	.032	.028	.060	.040	.181	.225	.303	.236
CEA_list-isc_cos_w2	.030	.041	.041	.037	.243	.281	.329	.284
CEA_list-isc_cos_sent ***(Sub_Lex)***	.025	.034	.040	.033	.233	.287	.340	.287
CEA_list-isc_l2_sent ***(Sub_Lex)***	.004	.012	.015	.010	.163	.230	.300	.231

Table 4.7. – Comparaison de nos meilleurs résultats pour la tâche de substitution lexicale avec ceux obtenus en utilisant DEPGLOVE et les systèmes ayant participé à l'atelier SemDis

une *baseline* a été proposée, *baseline_Campagne* . Elle consiste d'abord à sélectionner dans le dictionnaire DicoSyn (Ploux et Victorri, 1998) l'ensemble des synonymes pour un mot-cible en ne prenant en compte que les mots singuliers, puis de prendre les dix premiers synonymes selon un ordre de fréquence décroissant dans le corpus FRWAC.

Il apparaît clairement dans le tableau 4.6 que l'utilisation des signatures à base de combinaison des différentes relations avec un même coefficient [10] rend performant l'algorithme implémenté. D'autre part, cet algorithme surpasse la *baseline* par utilisation de nos trois signatures sur toutes les configurations. Pour les systèmes proposés par Ferret (2014b), à savoir, les quatre systèmes décrits dans le tableau 4.7 et notés avec CEA_list-.* et (*Sub_Lex*), l'utilisation de nos configurations et signatures sémantiques reste globalement meilleure que l'utilisation de ses représentations sémantiques à base du modèle neuronal Skip–gram dont l'une des différences secondaires, hors le modèle, avec DEPGLOVE est dans le corpus utilisé pour l'entraînement des *word embeddings*.

Il existe un seul système parmi tous les systèmes décrits dans Fabre *et al.* (2014) qui surpasse les performances globales de ce que nous proposons. Il retourne un *best* de .097 (*cf. Proxteam_JDM_Syn*) contre notre meilleur système (.084). Le système *Proxteam_JDM_Syn* repose sur des balades aléatoires dans

10. Il s'agit des systèmes *JDM_TraitsÉgaux_FctAct**i*** avec $\mathbf{i} \in \{1, 2, 3, 4\}$.

des graphes construits à partir de corpus et différentes ressources lexicales. Il est à noter que les résultats obtenus pour cette tâche dépendent à la fois des ressources et des algorithmes utilisés.

4.4. Conclusion

Dans ce chapitre, nous avons décrit une approche à base du réseau lexical JeuxDeMots permettant de créer des signatures sémantiques de mots et de sens. Nous avons évalué l'ensemble des signatures sur la tâche de mesures de similarité sémantique en utilisant le jeu de données RG–65. L'évaluation de la qualité des signatures de mots a été aussi évaluée sur la tâche de la substitution lexicale en utilisant le corpus SemDis. Pour cette deuxième tâche, nous avons utilisé un algorithme consistant à mesurer la similarité sémantique entre chaque candidat substitut et l'ensemble des mots pleins du contexte contenant le mot-cible à remplacer, hors ce dernier. Notre approche repose sur l'utilisation de plusieurs relations définies dans le réseau JeuxDeMots. Nous avons utilisé différentes fonctions pour mesurer la similarité sémantique et nous avons démontré que les résultats obtenus en utilisant notre approche surpassent dans certains cas les résultats obtenus en utilisant des systèmes de l'état de l'art comme GLOVE ou NASARI.

Dans le chapitre suivant, nous allons utiliser ces signatures sémantiques de mots et de sens pour la tâche de désambiguïsation sémantique. Nous montrons que l'idée d'avoir des représentations sémantiques à la fois pour les mots et les sens permet d'améliorer plusieurs aspects des systèmes de désambiguïsation présentés dans le chapitre précédent.

Désambiguïsation sémantique à base de signatures sémantiques créées à partir du réseau lexical JeuxDeMots

5.1. Motivation

La désambiguïsation sémantique est une tâche qui consiste à choisir pour chaque mot polysémique son sens le plus proche par rapport au contexte dans lequel il apparaît (Navigli, 2009). Ce contexte peut représenter une phrase, un paragraphe ou tout un texte. Nous avons mentionné dans le chapitre 1 que les systèmes de désambiguïsation les plus performants sont basés sur un apprentissage supervisé (*cf.* sous-section 1.3.2). Cependant, pour atteindre une très haute performance, ces systèmes ont besoin d'un corpus de grande taille annoté manuellement en sens. L'effort à fournir pour avoir un tel corpus est considérable. Aussi, le temps qu'il faut pour le produire est très long. À notre connaissance, aucun corpus de ce type existe pour le français. Nous nous intéressons, dans ce chapitre, à une approche de désambiguïsation à base de connaissances (*cf.* chapitre 1, sous-section 1.3.3) provenant du réseau lexical JeuxDeMots (Lafourcade, 2007) qui apporte non seulement des solutions aux limites que nous avons rencontrées lorsque nous avons utilisé BabelNet (Navigli et Ponzetto, 2012) (*cf.* chapitre 3) mais aussi une réduction au niveau de la complexité de l'algorithme de désambiguïsation.

Les méthodes de désambiguïsation à base de connaissances exploitent la structure graphique des ressources lexico-sémantiques pour identifier les significations les plus appropriées des mots selon un contexte donné Navigli (2009). Comme nous l'avons mentionné dans le chapitre 1, l'une des approches les plus classiques de cette catégorie consiste à estimer la proximité sémantique entre chaque sens candidat par rapport à chaque sens de chaque mot appartenant au contexte du mot à désambiguïser ($\prod_{w \in T} N_w$ combinaisons (séquences de sens)

à évaluer, avec N_w le nombre de sens du mot w et T l'ensemble de mots du contexte). Pour l'exemple d'une phrase de 10 mots avec 10 sens en moyenne, il y a 10^{10} combinaisons possibles à évaluer. Nous avons proposé dans le chapitre 3 une première solution pour la réduction du temps d'exécution d'une combinaison en comparant chaque sens candidat seulement avec les sens de mots provenant du contexte et sélectionnés au moyen d'une similarité distributionnelle.

Dans ce chapitre, nous utilisons les signatures sémantiques de mots et de sens que nous avons créées à partir du réseau lexical JeuxDeMots et dont nous avons évalué la qualité dans le chapitre précédent. Nous proposons une approche plus directe que l'approche classique citée ci-dessus. Au lieu de comparer chaque sens candidat à chaque sens des mots du contexte, nous comparons directement le vecteur de chaque sens candidat avec le vecteur de chaque mot du contexte. Dans ce cas-là, il y a $|T-1|\ .\ \sum_{w \in T} N_w$ paires de (sens, mot) à traiter, avec N_w le nombre de sens du mot polysémique w et T l'ensemble de mots du contexte. Considérons la phrase suivante tirée du corpus d'évaluation SemEval–2013 (Navigli *et al.*, 2013) : « *Il donne une grande flexibilité au processus, a dit John Coequyt, Représentant à Washington du Sierra Club.* » , *donner* (verbe) a 30 raffinements sémantiques dans JeuxDeMots, *grand* (adjectif) 8, *flexibilité* (nom) 4, *processus* (nom) 5, *avoir* (verbe) 7, *dire* (verbe) n'a pas actuellement de raffinements sémantiques dans JeuxDeMots, *John* (nom propre) 2, *Coequyt* (nom propre) n'a pas de raffinements sémantiques, *représentant* (nom) n'a pas de raffinements sémantiques, *Washington* (entité nommée) 3 et *Sierra Club* (entité nommée) n'a pas de raffinements sémantiques, il y a alors "seulement" 590 paires à traiter. Pour cet exemple, le nombre théorique de séquences de sens à évaluer est de $201\ 600$ combinaisons.

Ce chapitre est organisé comme suit : nous présentons tout d'abord dans la section 5.2 notre approche de désambiguïsation à base de connaissances provenant de JeuxDeMots. Ensuite, la section 5.3 est consacrée à l'évaluation intrinsèque des systèmes issus de cette approche. Nous présentons dans la sous-section 5.3.1 le corpus d'évaluation et dans la sous-section 5.3.2 les résultats obtenus par application des systèmes « *Baseline* » [1]. Cela avant de présenter les résultats d'évaluation de nos différents systèmes (*cf.* sous-sections 5.3.3 et 5.3.4). Nous terminons ce chapitre avec une conclusion qui résume les aspects les plus importants (*cf.* section 5.4).

1. Nous avons présenté les différents systèmes « *Baseline* » de désambiguïsation sémantique dans le chapitre 1 (*cf.* sous-section 1.4.3).

5.2. Désambiguïsation sémantique à base de connaissances provenant du réseau lexical JeuxDeMots

Désambiguïser tous les mots pleins [2] d'un corpus en se basant sur l'approche classique à base de connaissances est une tâche qui demande beaucoup de temps puisque l'algorithme à utiliser possède une complexité exponentielle très importante. Nous avons proposé dans le chapitre 3 une première solution à la réduction de cette complexité en utilisant les voisins les plus proches en contexte sélectionnés au moyen d'une similarité distributionnelle. Pour faire cette sélection, nous avons utilisé un corpus de travail de grande taille afin de soit extraire un ensemble de traits syntaxiques pour chaque mot plein analysé, soit entraîner des *word embeddings*. Dans ce chapitre, nous nous intéressons à un algorithme de désambiguïsation qui compare directement un vecteur de sens candidat à un vecteur de mot du contexte et qui peut faire une sélection de voisins en se basant non pas sur un corpus de travail mais plutôt sur une relation associative définie dans le réseau lexical JeuxDeMots, nommée *Inhibition*.

La relation *Inhibition* permet de retourner, pour une cible donnée, des termes qui ont tendance à être exclus par cette cible. Par exemple, le raffinement sémantique *chat (discussion)* inhibe (exclut) le mot *souris* et le raffinement sémantique *marteau (fou)* inhibe le mot *outil*. Cette relation est symétrique. Ci-dessous, nous décrivons le nombre d'instances de la relation selon les données de la base lexicale du réseau collectées de Janvier 2017 et Janvier 2018.

— ***Inhibition*** : quels sont les termes qui inhibent la cible ? – relation associative (p.ex. *outil* → *marteau* (*fou*)) – **258 613** / **436 869** instances ($\approx +68.9\%$ de progression).

Dans cette section, nous présentons notre méthode de désambiguïsation avec deux représentations différentes des vecteurs de mots et de sens. La première utilise directement des signatures sémantiques de mots et de sens (*cf.* sous-section 5.2.1). La deuxième utilise des *word embeddings* dont la construction des vecteurs de sens repose sur l'utilisation des signatures sémantiques de sens (*cf.* sous-section 5.2.2).

5.2.1. Désambiguïsation à base de signatures sémantiques de mots et de sens créées à partir de JeuxDeMots

Nous utilisons les signatures sémantiques de mots et de sens que nous avons créées et validées dans le chapitre 4. Nous avons choisi d'utiliser des signatures avec un type qui combine différentes relations : {"*Domaine*", "*Acception*", "*Hyperonyme*", "*Hyponyme*", "*Synonyme*", "*Agent*", "*Patient*", "*Idée associée*"} (*cf.*

2. Les mots pleins, ou ce que nous appelons aussi mots à *classe ouverte*, peuvent être des noms, verbes, adjectives ou adverbes.

r_traits_égaux[3], sous-section 4.2.1). Nous utilisons ce type de signatures pour l'évaluation de la désambiguïsation (*cf.* section 5.3). Ce type de signatures a donné de bons résultats pour l'évaluation de la substitution lexicale (*cf.* sous-section 4.3.2). D'autre part, il est possible de choisir un autre type de signatures. L'algorithme 1, ci-dessous, décrit en détail le système de désambiguïsation que nous avons proposé pour lever l'ambiguïté d'un mot-cible mot_c.

Algorithme 1 : Désambiguïsation sémantique d'un mot-cible mot_c par utilisation des signatures sémantiques de mots et de sens

---

Entrées :

$mot - cible\ (mot_c)$: mot à traiter

$raff\_sem\ (mot_c)$: ensemble de sens du premier niveau pour le mot-cible

$CXT(mot_c)$: liste de mots du contexte du mot-cible à désambiguïser, hors ce dernier

Sim : mesure de similarité sémantique entre un $sens_i \in raff\_sem\ (mot_c)$ et un mot $\in CXT(mot_c)$. $Sim \in \{\text{COS},\ \text{WO},\ Act_1,\ Act_2,\ Act_3,\ Act_4\}$

Résultat :

$\hat{\text{Sens}}_{\textbf{mot-cible}}$: sens du mot-cible ayant le meilleur score

Données :

S_{Type} : ensemble de signatures de mots et de sens dont les dimensions dépendent du type de signatures

$Rels_{Inhib}$: ensemble de paires de termes dont les éléments de chaque paire sont liés par une relation d'inhibition avec un poids positif non nul

Initialisation :

$Score_{raffs\_C} = \emptyset$ /* Fonction de hachage permettant d'associer à chaque sens candidat le score retourné par l'algorithme. */

pour chaque $sens_i \in raff\_sem\ (mot_c)$ **faire**

  $Score(sens_i) = 0$;

  pour chaque $voisin_j \in CXT(mot_c)$, *avec* $j \in \{1, \ldots, |CXT(mot_c)|\}$ **faire**

    si $(sens_i,\ voisin_j) \notin Rels_{Inhib}$ **alors**

      $Score(sens_i) = Score(sens_i) + Sim(S_{Type}(sens_i),\ S_{Type}(voisin_j))$;

  $Score_{raffs\_C} \leftarrow Score_{raffs\_C} \bigcup (sens_i,\ Score(sens_i))$;

si *(*$|\textbf{Best}\ (Score_{raffs\_C})| \geqslant 2$*)* **alors**

  $\hat{\text{Sens}}_{\textbf{mot-cible}} \leftarrow$ JDM–FS–LISTE$(mot_c, \textbf{Best}\ (Score_{raffs_C}))$ /* Fonction qui retourne le sens ayant le poids le plus fort dans JEUXDEMOTS à partir d'une liste donnée. */

sinon

  $\hat{\text{Sens}}_{\textbf{mot-cible}} \leftarrow \textbf{Best}\ (Score_{raffs\_C})$

---

3. Le poids de chaque relation pour une dimension donnée est représenté avec un même coefficient.

Nous avons à disposition comme entrées un mot-cible (mot_c) à désambiguïser et un ensemble de mots appartenant au contexte du mot_c ($CXT(mot_c)$). Depuis JeuxDeMots, nous prenons tous les raffinements sémantiques de premier niveau, s'ils existent, pour le mot_c comme sens candidats ($raff\_sem\ (mot_c)$). Comme nous l'avons mentionné dans le chapitre 4 (*cf.* sous-section 4.2.1), JeuxDeMots permet d'avoir toute une structure hiérarchique de raffinements sémantiques d'un mot polysémique donné. Ce qui donne une granularité plus optimale par rapport à ce que nous avons vu avec BabelNet où tous les sens sont représentés sur un même niveau.

Depuis les données collectées de la base lexicale du réseau datant de Janvier 2018, nous avons effectué une extraction de toutes les instances de la relation *Inhibition* ayant un poids de relation positif ($Rels_{Inhib}$). Le principe de l'algorithme de désambiguïsation sémantique consiste à comparer chaque sens candidat ($sens_i$) seulement avec les mots du contexte qui ne sont pas exclus par ce sens, c'est-à-dire, il n'existe pas une relation d'inhibition entre le sens candidat et chaque mot du contexte. Cette manière de procéder à la sélection des mots du contexte donne l'avantage aux sens qui excluent moins de mots d'avoir un score plus important.

Pour la comparaison des signatures, nous utilisons des techniques que nous avons déjà utilisées pour l'évaluation de la qualité des signatures ainsi que différentes fonctions d'activation décrites dans le chapitre 4 (*cf.* sous-section 4.2.2). Après obtention du score de chaque sens candidat, le système choisit le sens ayant le meilleur score. D'autre part, nous utilisons l'heuristique suivante :

Dans le cas où deux sens candidats ou plus ont le meilleur score de similarité, le sens retourné parmi ces sens est celui qui a le poids le plus important dans le réseau JeuxDeMots.

Il s'agit du poids de la relation *Raffinement sémantique* entre chaque sens candidat et le mot-cible (mot_c).

5.2.2. Désambiguïsation sémantique à base des *word embeddings*

Dans cette sous-section, nous présentons d'autres algorithmes que nous avons proposés pour la désambiguïsation sémantique. L'approche générale reste la même que celle proposée ci-dessus (*cf.* sous-section 5.2.1). La différence principale est dans les représentations vectorielles de mots et de sens à utiliser. Au lieu d'utiliser les signatures sémantiques de mots et de sens, nous utilisons des représentations vectorielles continues de mots et de sens à base des *word embeddings* ou « *plongements de mots* ».

Nous utilisons un jeu de vecteurs de mots proposé par FAUCONNIER (2015) [4] et basé sur le modèle WORD2VEC avec une utilisation du type CBOW, *continuous bag of words* (*cf.* chapitre 1, sous-section 1.3.1). L'entraînement des *embeddings* s'est effectué par utilisation des données issues du corpus français FRWAC (BARONI *et al.*, 2009) avec un prétraitement en avant comprenant la lemmatisation des mots du corpus. Ce dernier contient près de 1.6 milliard de mots. Les vecteurs de mots que nous utilisons ont 500 dimensions.

Pour les vecteurs de sens, nous avons fait le choix de construire pour chaque sens un vecteur centroïde défini à partir des vecteurs de tous les mots singuliers représentant des dimensions dans la signature sémantique du sens. L'algorithme 2, ci-dessous, décrit en détail le système de désambiguïsation.

Algorithme 2 : Désambiguïsation sémantique d'un mot-cible mot_c par utilisation des *word embeddings* (première variante)

Entrées :
$mot - cible\ (mot_c)$: mot à traiter
$raff\_sem\ (mot_c)$: ensemble de sens du premier niveau pour le mot-cible
$CXT(mot_c)$: liste de mots du contexte du mot-cible, hors ce dernier
Résultat :
$\mathbf{\hat{Sens}_{mot-cible}}$: sens du mot-cible ayant le meilleur score
Données :
$\mathrm{S_{Type}}$: ensemble de signatures de sens dont les dimensions dépendent du type de signatures
$Rels_{Inhib}$ `// Voir l'algorithme 1.`

Initialisation :
$Score_{raffs\_C} = \emptyset$ `// Voir l'algorithme 1.`
pour chaque $sens_i \in raff\_sem\ (mot_c)$ **faire**
    $Score(sens_i) = 0$;
    $V(sens_i) \leftarrow$ Vecteur centroïde défini à partir des vecteurs de tous les mots singuliers représentant les dimensions de la signature $S_{Type}(Sens_i)$
    pour chaque $voisin_j \in CXT(mot_c)$, *avec* $j \in \{1, \ldots, |CXT(mot_c)|\}$ **faire**
        si $(sens_i,\ voisin_j) \notin Rels_{Inhib}$ **alors**
            $Score(sens_i) = Score(sens_i) + Cosinus(V(sens_i),\ V(voisin_j))$;
    $Score_{raffs\_C} \leftarrow Score_{raffs\_C} \bigcup (sens_i,\ Score(sens_i))$;
si *(*$|\mathbf{Best}\ (Score_{raffs\_C})| \geqslant 2$*)* **alors**
    $\mathbf{\hat{Sens}_{mot-cible}} \leftarrow$ JDM–FS–LISTE$(mot_c,\ \mathbf{Best}\ (Score_{raffs_C}))$ `// Voir l'algorithme 1.`
sinon
    $\mathbf{\hat{Sens}_{mot-cible}} \leftarrow \mathbf{Best}\ (Score_{raffs\_C})$

---

4. http://fauconnier.github.io

Nous n'utilisons pas les expressions polylexicales – *Multiword Expression* (*MWE*) qui se trouvent dans la signature sémantique de chaque sens pour la construction du vecteur centroïde du sens (représentation vectorielle continue du sens). En effet, comme cité par Salehi *et al.* (2015), les expressions polylexicales sont des combinaisons de mots qui présentent une certaine idiomaticité (Baldwin et Kim, 2009), y compris l'idiomaticité sémantique. Leur sens est souvent non compositionnel, c'est-à-dire que la sémantique, par exemple, de *pomme de terre* ne peut pas être prédite à partir de la sémantique des mots *pomme* et *terre* ou de *examen clinique* à partir des mots *examen* et *clinique*[5].

La fonction $V(sens_i)$ décrite dans l'équation 5.1 retourne le vecteur centroïde (moyen) d'un $sens_i$ du mot-cible mot_c.

$$V(sens_i) = \frac{1}{|S'_{Type}(sens_i)|} \sum_{d \in S'_{Type}(sens_i)} V(d) \tag{5.1}$$

$S'_{Type}(sens_i)$ est la signature sémantique de $sens_i$ qui ne prend en compte que les mots singuliers comme dimensions. d est une dimension représentant un mot singulier appartenant à la signature sémantique de $sens_i$.

L'algorithme 2, ci-dessus, traite le même nombre de paires de (sens, mot) que l'algorithme 1. La mesure de similarité utilisée ici pour comparer un vecteur de sens à un vecteur de mot est un Cosinus.

Nous utilisons une deuxième variante de la méthode de désambiguïsation à base des *word embeddings*. Pour cette variante, nous nous inspirons de l'approche de désambiguïsation proposée par Chen *et al.* (2014). Le principe de leur approche de désambiguïsation consiste à comparer le vecteur de chaque sens candidat directement au vecteur du contexte. Ce dernier est le vecteur centroïde de tous les vecteurs de mots du contexte. Dans notre cas, pour chaque sens candidat, le vecteur du contexte est le vecteur centroïde des vecteurs de mots qui ne sont pas exclus par le sens. En général, l'application de cette méthode consiste à traiter $\sum_{w \in T} N_w$ paires de (sens, contexte), avec N_w le nombre de sens du mot polysémique w et T l'ensemble de mots polysémiques du contexte. Cela nous rappelle le même nombre de paires à traiter lorsque nous utilisons la variante de Lesk (Kilgarriff et Rosenzweig, 2000) (*cf.* chapitre 2, sous-section 2.2.2).

La fonction $V(Contexte)_{sens_i}$ décrite dans l'équation 5.2 retourne le vecteur centroïde du contexte du mot-cible mot_c pour le calcul du score de $sens_i$.

$$V(Contexte)_{sens_i} = \frac{1}{|C(sens_i)|} \sum_{w_j \in C(sens_i),\ w_j \neq mot_c} V(w_j) \tag{5.2}$$

$C(sens_i)$ est l'ensemble de mots singuliers appartenant au contexte du mot-cible mot_c, hors ce dernier, et qui ne sont pas exclus par le $sens_i$. Le mot w_j est un mot singulier appartenant à $C(sens_i)$.

5. Pour des travaux récents sur la prédiction de la compositionnalité, voir (Cordeiro *et al.*, 2016 ; Reddy *et al.*, 2011).

L'algorithme 3, ci-dessous, décrit en détail ce système de désambiguïsation.

Algorithme 3 : Désambiguïsation sémantique d'un mot-cible mot_c par utilisation des *word embeddings* (deuxième variante)

Entrées :
$mot - cible\ (mot_c)$: mot à traiter
$raff\_sem\ (mot_c)$: ensemble de sens du premier niveau pour le mot-cible
$CXT(mot_c)$: liste de mots du contexte du mot-cible, hors ce dernier
Résultat :
$\mathbf{\hat{Sens}_{mot-cible}}$: sens du mot-cible ayant le meilleur score
Données :
S_{Type} : ensemble de signatures de sens dont les dimensions dépendent du type de signatures
$Rels_{Inhib}$ `// Voir l'algorithme 1.`

Initialisation :
$Score_{raffs\_C} = \emptyset$ `// Voir l'algorithme 1.`
pour chaque $sens_i \in raff\_sem\ (mot_c)$ **faire**
    $Score(sens_i) = 0$;
    $C(sens_i) \leftarrow$ Ensemble de mots singuliers appartenant au contexte du mot-cible et qui n'inhibent pas le $sens_i$
    $V(sens_i)$ `// Voir l'algorithme 2.`
    $V(Contexte)_{sens_i} \leftarrow$ Vecteur centroïde défini à partir des vecteurs de tous les mots de l'ensemble $C(sens_i)$
    $Score(sens_i) = Cosinus(V(sens_i), V(Contexte)_{sens_i})$;
    $Score_{raffs\_C} \leftarrow Score_{raffs\_C} \bigcup (sens_i, Score(sens_i))$;
si $(|\mathbf{Best}\ (Score_{raffs\_C})| \geqslant 2)$ **alors**
    $\mathbf{\hat{Sens}_{mot-cible}} \leftarrow$ JDM–FS–LISTE$(mot_c, \mathbf{Best}\ (Score_{raffs_C}))$ `// Voir l'algorithme 1.`
sinon
    $\mathbf{\hat{Sens}_{mot-cible}} \leftarrow \mathbf{Best}\ (Score_{raffs\_C})$

5.3. Évaluation intrinsèque de la désambiguïsation sémantique

Nous présentons dans cette section l'évaluation de nos différents systèmes de désambiguïsation que nous avons décrits dans la section 5.2 – *i.e.*, application des algorithmes 1, 2 et 3.

Nous avons fait le choix d'évaluer la qualité de nos systèmes sur le corpus SEMEVAL–2013 (NAVIGLI *et al.*, 2013) que nous décrivons dans la sous-section 5.3.1. Nous présentons les résultats retournés par les systèmes « *Baseline* » dans la sous-section 5.3.2 avant de présenter les résultats de nos systèmes de désambiguïsation dans les sous-sections 5.3.3 et 5.3.4.

5.3.1. Description du corpus d'évaluation

Navigli *et al.* (2013) ont proposé un corpus multilingue annoté en sens pour la tâche de désambiguïsation sémantique AWD (*all-words disambiguation*) durant la campagne d'évaluation SemEval–2013. Trois inventaires de sens ont été utilisés, à savoir : BabelNet dans sa version 1.1.1, Wikipédia dont la base de l'encyclopédie date d'Octobre 2012 et WordNet dans sa version 3.0. Le corpus SemEval–2013 couvre 5 langues différentes, à savoir : allemand, anglais, espagnol, **français** et italien. Le corpus d'origine est en anglais et a été traduit dans les quatre autres langues. Durant les différentes campagnes d'évaluation Senseval/SemEval, des corpus en langue française ont été présentés seulement dans Senseval–1 (Segond, 2000) et SemEval–2013 (Navigli *et al.*, 2013). Pour Senseval–1, un dictionnaire spécifique a été utilisé comme inventaire de sens pour la tâche LSD (*lexical sample disambiguation*). Nous nous intéressons au corpus SemEval–2013 pour l'évaluation de nos systèmes de désambiguïsation.

Le corpus SemEval–2013 contient 13 articles tirés des jeux de données disponibles dans les éditions 2010, 2011 et 2012 de l'atelier sur la traduction automatique (WSMT, *Workshop on Statistical Machine Translation*[6]). Les articles couvrent différents domaines, allant du sport à l'actualité financière. La lemmatisation et l'annotation des mots en parties du discours sont fournies avec le corpus. Ce dernier propose des annotations en sens pour les noms communs et les entités nommées. Les noms communs peuvent être des mots singuliers ou des expressions polylexicales, par exemple, "*poids lourd*" pour spécifier un *camion* ou une *catégorie de poids en sports de combat*. Le tableau 5.1 décrit l'ensemble des instances annotées en sens dans le corpus SemEval–2013 pour le français et cela selon deux inventaires de sens : BabelNet et Wikipédia. Le réseau sémantique WordNet a été utilisé comme inventaire de sens que pour le corpus original en anglais.

Inventaire de sens	Instances	Mots singuliers	Expressions polylexicales	Entités nommées	Nombre moyen de sens par instance	Nombre moyen de sens par lemme
BabelNet	**1 656**	**1 389**	**89**	**176**	**1.05**	**1.15**
Wikipédia	1 039	790	72	175	1.18	1.14

Table 5.1. – Ensembles de données proposés pour la tâche de désambiguïsation sémantique pour le français dans SemEval–2013

Les deux dernières colonnes du tableau 5.1, présentant le nombre moyen de sens, décrivent le nombre moyen d'annotations en sens effectuées par l'annotateur. En effet, l'annotation a été réalisée par un seul locuteur natif pour le français. En général, pour l'ensemble des langues, un seul locuteur natif a annoté le corpus sauf pour l'italien où deux locuteurs natifs ont annoté différents

6. http://www.statmt.org/wmt12

sous-ensembles du corpus. Aussi, comme BabelNet couvre les sens lexicographiques de WordNet et encyclopédiques de Wikipédia, les annotateurs ont utilisé principalement BabelNet comme inventaire de sens et ont mis une correspondance, si elle existe, vers les sens de Wikipédia et WordNet.

Comme nous l'avons mentionné au début de ce chapitre, nous nous intéressons ici à une désambiguïsation en utilisant les raffinements sémantiques de termes décrits dans JeuxDeMots. Comme le corpus SemEval–2013 utilise principalement BabelNet comme inventaire de sens, il nous a été indispensable de réaliser un travail de mise en correspondance entre les sens de BabelNet fournis par le locuteur natif français et les raffinements sémantiques décrits dans le réseau lexical JeuxDeMots. D'autre part, la plupart des entités nommées se trouvant dans le corpus ne sont pas définies avec des raffinements sémantiques dans JeuxDeMots. De base, ce réseau ne présente pas une ressource encyclopédique et il est tout à fait normal que nous ne trouvons pas, par exemple, la liste complète des noms de joueurs d'un club sportif de football. Nous évaluons nos systèmes de désambiguïsation seulement sur les noms communs et nous excluons la désambiguïsation de l'ensemble des 176 entités nommées. Nous avons effectué une mise en correspondance sur un ensemble de $1\,480$ instances $(1\,656 - 176)$ [7]. Le tableau 5.2, ci-dessous, présente un extrait de texte tiré du corpus d'évaluation avec, d'une part, une annotation en sens par utilisation de BabelNet et, d'autre part, la mise en correspondance avec des raffinements sémantiques définis dans JeuxDeMots.

« … *Honnêtement, le* ***marché*** *est très calme, a relevé Mace Blicksilver, de Marblehead Asset Management. Il reste dans des* ***marges*** *étroites, le* ***volume*** *d'****échanges*** *est devenu très faible et je pense que cela va rester le* ***cas*** *jusqu'à la* ***fin*** *de l'****année*** … »
– marché : [BabelNet : *marché financier*] [JeuxDeMots : *marché (bourse)*]
– marge : [BabelNet : *pourcentage de gain*] [JeuxDeMots : *marge (bénéfice)*]
– volume : [BabelNet : *masse ou en gros*] [JeuxDeMots : *volume (quantité)*]
– échange : [BabelNet : *transfert réciproque entre organismes*] [JeuxDeMots : *échange (commerce)* ou *échange (échanger)*]
– cas : [BabelNet : *état actuel*] [JeuxDeMots : *cas (situation particulière)*]
– fin : [BabelNet : *extrémité*] [JeuxDeMots : *fin (limite)*]
– année : [BabelNet : *intervalle de temps, calendrier*] [JeuxDeMots : _]

Table 5.2. – Exemple de mise en correspondance entre des sens provenant de BabelNet et des raffinements sémantiques provenant de JeuxDeMots

7. La mise en correspondance a été effectuée par moi-même sur l'ensemble de sens fournis.

Sur l'ensemble des sept mots annotés sémantiquement en sens, nous avons pu réaliser une correspondance pour six mots entre les sens provenant de BabelNet et les raffinements sémantiques de JeuxDeMots. Actuellement, JeuxDeMots ne propose pas de raffinements sémantiques pour le nom '*année*'. Dans l'exemple ci-dessus, '*année*' correspond à "*un intervalle de temps défini conventionnellement dans le cadre d'un calendrier*". Pour une prochaine mise à jour de la base lexicale du réseau JeuxDeMots, nous pouvons déjà voir trois raffinements sémantiques : 'année (astronomie)', 'année (calendrier)' et 'année (décennie)'.

5.3.2. Systèmes « *Baseline* »

Avant de mesurer la performance de nos algorithmes de désambiguïsation décrits dans la section 5.2, nous utilisons deux systèmes de base parmi ceux que nous avons décrits dans le chapitre 1 (*cf.* sous-section 1.4.3) :

1. JDM–FS : JeuxDeMots First Sense similaire à WFS (WordNet First Sense) où l'idée consiste à retourner le sens (raffinement sémantique) ayant le poids le plus fort dans JeuxDeMots sans tenir compte du contexte.
2. MFS–WE : Most Frequent Sens using Word Embeddings. Nous nous inspirons de la méthode proposé par Bhingardive *et al.* (2015) qui utilise les *word embeddings* pour créer des représentations vectorielles continues de sens et compare le vecteur de chaque sens d'un mot-cible (mot_c) avec le vecteur de ce mot-cible. Le sens retourné est celui qui possède la similarité Cosinus la plus forte avec le mot-cible et cela à chaque fois sans tenir compte du contexte.

Afin de comprendre le fonctionnement de ces systèmes, prenons l'exemple du nom '*cas*'. Ce nom possède sept raffinements sémantiques dans JeuxDeMots. Le tableau 5.3, ci-dessous, liste l'ensemble de sept raffinements avec le score de chaque raffinement pour chaque système. La liste est triée par ordre décroissant des scores retournés par le premier système (*i.e.* JDM–FS).

Raffinement sémantique	**JDM–FS**	**MFS–WE**
cas (individu)	**1.0**	0.115
cas (situation particulière)	0.9574	0.0254
cas (grammaire)	0.9149	0.0624
cas (médecine)	0.8723	0.0531
cas (déjection)	0.6383	0.0996
cas (droit)	0.6383	0.2711
cas (postérieur)	0.617	**0.3217**

Table 5.3. – Les 7 raffinements sémantiques du nom '*cas*' avec leurs poids d'importance

Pour le système JDM–FS, le score de chaque raffinement sémantique est obtenu de la manière suivante :

Le poids de la relation *Raffinement sémantique* est utilisé. Il est ensuite normalisé avec la norme infinie qui est définie par la fonction : $\|\vec{X}\|_\infty = max(|x_1|, |x_2|, \ldots, |x_n|)$ avec x_i le ième raffinement sémantique du mot-cible (mot_c). Le raffinement ayant le poids maximum a un score égal à 1. Le score des autres raffinements présente le rapport entre le poids des raffinements et le poids maximum.

Pour le système MFS–WE, le score de chaque raffinement sémantique est obtenu de la manière suivante :

La mesure Cosinus est utilisée pour comparer le vecteur du mot-cible (mot_c) et le vecteur du raffinement sémantique. La construction du vecteur du raffinement sémantique est décrite dans la sous-section 5.2.2.

Pour le nom '*cas*' et suivant les résultats du tableau 5.3, le système JDM–FS retourne pour toutes les occurrences de '*cas*' le raffinement sémantique "*cas (individu)*" ; quant au système MFS-WE, il retourne le raffinement sémantique "*cas (postérieur)*". Dans le corpus SemEval–2013, 15 occurrences du nom '*cas*' ont été annotées sémantiquement en sens : 10 occurrences avec le sens "*cas (individu)*" et 5 occurrences avec le sens "*cas (situation particulière)*".

Le tableau 5.4 présente les résultats de désambiguïsation sémantique obtenus sur l'ensemble des instances du corpus SemEval–2013. Nous utilisons les mesures de précision, rappel et F-mesure comme mesures d'évaluation. Nous avons présenté ces mesures dans le chapitre 1 (*cf.* sous-section 1.4.2).

Système	Précision (%)	Rappel (%)	F–mesure (%)
JDM–FS	**53.1**	**32.2**	**40.1**
MFS–WE	50.7	30.7	38.3

Table 5.4. – Résultats de désambiguïsation sémantique par utilisation des systèmes « *Baseline* »

Les résultats montrent clairement la différence large entre la précision et le rappel. À l'état actuel [8], JeuxDeMots ne propose pas de raffinements sémantiques pour 580 occurrences de mots annotés en sens avec BabelNet parmi $1\,480$ instances dans le corpus. De ce fait, les systèmes de désambiguïsation ont retourné une réponse pour 900 instances. Sur l'ensemble de ces 900 instances, deux instances n'ont pas le bon raffinement sémantique dans JeuxDeMots. Il s'agit de l'absence de deux raffinements : '*accent (attention)*' et '*ligne (Internet)*'.

8. Nous avons utilisé les données de la base datant de Janvier 2018 que ce soit pour la génération des raffinements sémantiques candidats ou pour la construction de signatures sémantiques de mots et de sens.

5.3.3. Évaluation de l'algorithme de désambiguïsation à base de signatures sémantiques de mots et de sens

Pour l'évaluation de l'algorithme 1 de désambiguïsation sémantique (*cf.* sous-section 5.2.1), nous avons utilisé différentes mesures de similarité afin de comparer les signatures de sens candidats aux signatures de mots appartenant au contexte de chaque mot à désambiguïser, hors ce dernier. Ces mesures peuvent être soit des techniques de comparaison de deux représentations vectorielles telles que COSINUS ou WEIGHTED OVERLAP, soit des fonctions d'activation (*cf.* chapitre 4, sous-section 4.2.2).

D'autre part, nous avons fait le choix d'évaluer l'algorithme selon le contexte utilisé. Il peut s'agir de la phrase dans laquelle le mot à désambiguïser apparaît ou tout le document (texte intégral). Choisir le texte intégral comme contexte de chaque mot à désambiguïser est une heuristique déjà proposée par GALE *et al.* (1992) :

One Sense Per Discourse (un sens par discours ou document) : « *un mot est systématiquement référé avec le même sens dans un discours ou un document donné* » (GALE *et al.*, 1992).

Le tableau 5.5, ci-dessous, décrit la performance de l'algorithme de désambiguïsation sémantique selon la mesure de similarité utilisée et le contexte du mot à désambiguïser et cela sur l'ensemble de $1\,480$ instances de noms communs du corpus SEMEVAL–2013.

Mesure de similarité	Contexte	Précision (%)	Rappel (%)	F–mesure (%)
COSINUS (COS)	Phrase	40.6	24.8	30.8
	Texte	43.4	26.6	32.9
WEIGHTED OVERLAP (WO)	Phrase	42.0	25.7	31.9
	Texte	45.5	27.9	34.6
Activation$_1$ (Act$_1$)	Phrase	54.2	33.2	41.2
	Texte	**57.5**	**35.2**	**43.7**
Activation$_2$ (Act$_2$)	Phrase	53.1	32.5	40.3
	Texte	51.1	31.3	38.8
Activation$_3$ (Act$_3$)	Phrase	53.7	32.9	40.8
	Texte	50.8	31.1	38.6
Activation$_4$ (Act$_4$)	Phrase	52.5	32.2	39.9
	Texte	48.5	29.7	36.8

Table 5.5. – Performance du système de désambiguïsation sémantique, à base de signatures sémantiques de mots et de sens provenant de JeuxDeMots, par utilisation de différentes mesures de similarité

Les résultats obtenus montrent que l'utilisation de l'une des quatre fonctions d'activation rend l'algorithme de désambiguïsation meilleur que l'utilisation directe de la technique Cosinus ou Weighted Overlap (WO).

Nous rappelons que ces fonctions d'activation utilisent ces techniques et renvoient le score maximum entre le score obtenu par l'utilisation de ces techniques et les poids d'activation entre les éléments à comparer. Dans notre cas, un sens candidat à un mot du contexte.

Pour le corpus SemEval–2013, l'intégration de la première fonction d'activation permet au système de désambiguïsation de retourner la meilleure précision et cela quelque soit le contexte utilisé (57.5% pour l'utilisation du texte intégral et 54.2% pour l'utilisation de la phrase).

Comme nous l'avons mentionné dans la sous-section 5.3.2, le faible rappel obtenu revient à l'absence de raffinements sémantiques dans JeuxDeMots pour un ensemble d'occurrences de mots annotés en sens dans le corpus SemEval–2013. Par exemple, le nom 'début' n'a pas de raffinements sémantiques dans JeuxDeMots alors qu'il est polysémique. Nous pouvons faire la distinction entre deux sens et proposer deux raffinements sémantiques comme suit :

1. Moment où commence une activité ou un événement [raffinement sémantique : *début (commencement)*].
2. La première partie ou section de quelque chose [raffinement sémantique : *début (première apparition)*].

Le corpus SemEval–2013 propose 5 instances du nom 'début' : trois font référence au premier sens et les deux autres font référence au deuxième sens.

D'autre part, dans le chapitre 4, nous avons évalué la qualité des signatures sémantiques de mots et de sens par rapport à un jugement humain. Nous avons vu que la qualité des signatures de mots est meilleure que celle des signatures de sens. Avec l'évolution constante de la base du réseau lexical JeuxDeMots, nous pensons que nous devrions obtenir de meilleurs résultats de précision dans l'avenir lorsque nous utilisons l'algorithme 1 de désambiguïsation sémantique.

5.3.4. Évaluation des algorithmes de désambiguïsation à base des *word embeddings*

Nous présentons dans cette sous-section les résultats d'évaluation des algorithmes 2 et 3 (*cf.* sous-section 5.2.2). Ces algorithmes reposent sur l'utilisation des *word embeddings*.

Le tableau 5.6 décrit la performance des deux algorithmes de désambiguïsation sémantique. Nous faisons référence à l'algorithme 2 par WSD–WE (WORD SENSE DISAMBIGUATION USING WORD EMBEDDINGS) et à l'algorithme 3 par WSD–CVE (WORD SENSE DISAMBIGUATION USING CONTEXT VECTOR EMBEDDINGS). Nous rappelons que pour ces deux algorithmes, nous n'avons pas entraîné des *sense embeddings* et *context embeddings* mais simplement pris le vecteur centroïde défini à partir des vecteurs de tous les mots singuliers appartenant au sens et au contexte, respectivement. Les mots appartenant aux sens sont les dimensions des signatures sémantiques de sens.

Système	Contexte	Précision (%)	Rappel (%)	F–mesure (%)
WSD–WE	Phrase	44.9	27.5	34.1
	Texte	45.6	27.9	34.6
WSD–CVE	Phrase	45.4	27.8	34.5
	Texte	**48.5**	**29.7**	**36.8**

Table 5.6. – Performance des systèmes de désambiguïsation sémantique à base des *word embeddings*

Les résultats obtenus montrent que l'utilisation d'un vecteur centroïde décrivant le contexte d'un mot à désambiguïser retourne des résultats légèrement meilleurs que l'utilisation des vecteurs de mots du contexte. Cette remarque est vraie tant pour l'utilisation de la phrase que pour l'utilisation du texte intégral comme contexte.

Nous présentons dans le tableau 5.7 nos meilleurs résultats et une comparaison aux systèmes « *Baseline* ». Nous faisons référence à l'algorithme 1 par WSD–JDM (WORD SENSE DISAMBIGUATION USING SEMANTIC SIGNATURES FOR WORDS AND WORD SENSES).

Système	Contexte	Précision (%)	Rappel (%)	F–mesure (%)
WSD–JDM $(\mathbf{Act_1})$	Texte	**57.5**	**35.2**	**43.7**
WSD–CVE	Texte	48.5	29.7	36.8
JDM–FS	–	53.1	32.2	40.1
MFS–WE	–	50.7	30.7	38.3

Table 5.7. – Performance de nos meilleurs systèmes de désambiguïsation sémantique et comparaison avec les systèmes *Baseline*

Le système WSD–JDM avec application de la première fonction d'activation est le plus performant. Il donne une performance avec une marge de $+9\%$ de précision, $+5.5\%$ de rappel et $+6.9\%$ de F-mesure par rapport au système WSD–CVE, et cela par utilisation du texte intégral comme contexte.

L'application directe de la mesure COSINUS comme mesure de similarité ne permet pas au système WSD–JDM d'être plus performant que les systèmes à base des *word embeddings*. En effet, le système WSD–JDM (COSINUS) retourne une précision de 43.4% contre 48.5% pour WSD-CVE lorsque le texte intégral est utilisé comme contexte.

5.4. Conclusion

Dans ce chapitre, nous avons présenté une approche de désambiguïsation sémantique à base de connaissances provenant du réseau lexical JEUXDEMOTS. D'une part, des systèmes issus de cette approche utilisent les signatures sémantiques de mots et de sens créées à partir de JEUXDEMOTS. D'autre part, d'autres systèmes de la même approche utilisent les signatures sémantiques de sens et les *word embeddings*. Ces derniers systèmes manipulent des représentations vectorielles continues soit pour les sens et les mots, soit pour les sens et le contexte. Le temps d'exécution de ces systèmes est réduit par rapport au temps d'exécution du système issu de l'approche classique consistant à comparer chaque sens candidat d'un mot-cible à chaque sens des mots du contexte.

Nous avons proposé une évaluation de nos différents systèmes sur le corpus SEMEVAL–2013 pour le français et nous avons comparé la performance de l'application de notre approche à deux systèmes de base, à savoir : JDM–FS (ce système retourne le sens ayant le poids le plus grand dans JEUXDEMOTS sans tenir compte du contexte) et MFS–WE (ce système retourne le sens le plus fréquent par utilisation des *word embeddings* sans tenir compte du contexte). Pour l'utilisation de nos différents systèmes, le contexte du mot à désambiguïser a été d'abord défini comme étant la phrase et ensuite étendu au texte intégral. Nous avons remarqué que l'application de l'heuristique *One Sense Per Discourse* (un sens par texte) proposée par GALE *et al.* (1992) retourne de meilleurs résultats pour le corpus SEMEVAL–2013 et cela pour tous les systèmes qui utilisent directement la similarité distributionnelle COSINUS.

Dans le chapitre suivant, nous décrivons la création d'une ressource lexicale en français proposant des synonymes désambiguïsés et gradués en fonction de leur niveau de difficulté. Avoir à disposition une telle ressource est bénéfique pour un système de simplification lexicale qui utilise en amont un système de désambiguïsation sémantique.

RESYF : une ressource de synonymes désambiguïsés et gradués en fonction de leur niveau de difficulté

6.1. Introduction

Durant ces dernières années, de gros corpus de données ont été mis à la disposition de la communauté du TAL (BARONI *et al.*, 2009 ; KOEHN, 2005). Avec la maturité croissante de la linguistique de corpus et l'utilisation des techniques du TAL, les ressources lexicales avec des informations quantitatives ont fait des progrès remarquables. Il s'agit, par exemple, du réseau sémantique BABELNET (NAVIGLI et PONZETTO, 2012) ou WOLF (SAGOT et FIŠER, 2008), le WORDNET libre pour le français inspiré du WORDNET anglais de Princeton (FELLBAUM, 1998). La couverture de ces ressources a augmenté et une description des unités lexicales intégrant une lemmatisation et une annotation en partie du discours a été proposée. Les modèles statistiques ont donné des descriptions plus précises du lexique (en termes de fréquences, de modèles $n-$gramme, *etc.*).

Parmi la variété de ressources lexicales, il existe des ressources 'graduées', c'est-à-dire, des ressources où les unités lexicales ont un niveau de difficulté associé (ou bien une estimation de leur fréquence d'apparition dans un niveau d'apprentissage donné) (GALA, 2015). Pour l'anglais, par exemple, il y a le thésaurus en ligne [1] qui permet de filtrer les synonymes et les antonymes d'un mot donné par pertinence, longueur et complexité. Pour le français, par exemple, il y a MANULEX (LÉTÉ *et al.*, 2004) qui est utilisé principalement par les enseignants et les psycholinguistes pour identifier le niveau de difficulté des mots à l'école (contexte de l'apprentissage du français comme langue maternelle (L1)). MANULEX décrit les distributions de fréquence de $23\,812$ lemmes français répartis sur

1. http://www.thesaurus.com

trois niveaux d'enseignement primaire : 1ère année (CP), 2ème année (CE1) et 3ème à 5ème année (CE2 à la CM2). Les fréquences ont été estimées sur un corpus de matériels pédagogiques utilisés à ces trois niveaux. Par ailleurs, FLELEX (FRANÇOIS *et al.*, 2014) est un autre lexique gradué, similaire à MANULEX mais destiné aux apprenants du français comme langue étrangère (L2). Il a été construit à partir d'un corpus de textes pédagogiques classés selon les six niveaux de compétence définis par le CECR [2] (CONSEIL DE L'EUROPE, 2001), allant de A1 à C2. MANULEX et FLELEX sont deux lexiques qui peuvent être utilisés pour l'entraînement des modèles d'ordonnancement (classement) des unités lexicales en fonction de leur difficulté de lecture et compréhension. Par ailleurs, les 'grades' fournis par ces deux lexiques sont des niveaux statiques et établis à partir des fréquences d'apparition de mots en corpus. Aussi, ces deux lexiques ne permettent pas d'avoir des listes de synonymes explicites.

Dans ce chapitre, nous décrivons la construction d'une ressource pour le français permettant d'avoir des synonymes désambiguïsés et gradués dynamiquement en fonction de leur complexité (niveau de difficulté) pour des lecteurs en langue maternelle (L1). Pour une liste de n synonymes, les 'grades' sont de 1 à n. Chaque synonyme dans cette liste a un niveau de difficulté différent des niveaux de difficulté des autres synonymes. Cette ressource, appelée RESYF, vise à proposer des synonymes avec des niveaux de difficulté associés. La ressource pourra être utilisée lors de l'apprentissage du français (L1) ou intégrée à un système de simplification automatique de textes. La ressource RESYF a été développée dans le cadre du projet ALECTOR [3]. Le principe de RESYF est de proposer, pour chaque terme en entrée (mot singulier – *Single Word* ou expression polylexicale – *Multiword Expression* (*MWE*)), une liste de synonymes dont le tri de tous ces synonymes y compris le terme en entrée s'effectue en fonction de leur difficulté.

La synonymie est une relation lexicale sémantique d'équivalence entre signifiés. La synonymie exacte (ou absolue) étant rarissime, on considère comme synonymes deux unités lexicales ayant une « *valeur sémantique suffisamment proche pour que l'une puisse être utilisée à la place de l'autre pour exprimer sensiblement la même chose.* » (POLGUÈRE, 2002). Deux unités lexicales recouvrant (par inclusion ou intersection) la même notion sont donc des synonymes, par exemple *bleu* et *azur* dans le sens 'couleur' ou *bleu*, *contusion* et *ecchymose* dans le sens 'résultat d'un choc'.

La notion de difficulté est comprise ici comme une valeur qui situe le terme en entrée sur une échelle de complexité de lecture et de compréhension par rapport à des termes sémantiquement équivalents, par exemple : *bleu* ('résultat d'un choc') par rapport à *contusion* ou *ecchymose*. En ce qui concerne les ressources similaires par d'autres langues, à notre connaissance, seul le lexique CASSAURUS (BAEZA-YATES *et al.*, 2015) pour l'espagnol est similaire à RESYF.

2. Cadre Européen Commun de Référence pour les langues : apprendre, enseigner et évaluer (CECR).
3. https://alectorsite.wordpress.com

Cependant, les mots de ce lexique ne sont assignés qu'à deux classes, simples et complexes, ce qui est une vue très réductrice de la complexité lexicale.

Ce chapitre est organisé comme suit : nous présentons tout d'abord dans la section 6.2 nos différentes méthodologies d'acquisition des données représentant le vocabulaire de notre ressource. Ensuite, la section 6.3 décrit le modèle statistique utilisé pour ordonner les mots de la ressource (il repose sur la prise en compte combinée d'un large ensemble de variables linguistiques et psycholinguistiques). La section 6.4 est consacrée à l'évaluation de la qualité de la ressource ReSyf ; quant à la section 6.5, elle fournit des détails sur la ressource elle-même et sa disponibilité. Enfin, dans la conclusion, nous résumons les points les plus importants de ce chapitre (*cf.* section 6.6).

6.2. Méthodologies d'acquisition des données de ReSyf

La première version de ReSyf a été proposée par Gala *et al.* (2013). Il s'agissait d'une ressource graduée de synonymes mais non désambiguïsés sémantiquement. Le réseau de synonymes de JeuxDeMots (Lafourcade, 2007) avait été utilisé pour décrire les données de ReSyf et chaque mot de la ressource avait été attribué un des trois niveaux de difficulté de Manulex. Pour graduer les mots absents de Manulex, un modèle de classification avait été employé (Gala *et al.*, 2014).

Dans une version postérieure de la ressource, Gala *et al.* (2015) ont proposé des synonymes désambiguïsés et cela par utilisation du réseau sémantique BabelNet. ReSyf a constitué, ainsi, un premier pas vers un lexique gradué de synonymes, tout en conservant les sens des mots (et pas seulement les formes). Cependant, la ressource comportait quelques défauts. Tout d'abord, le recours à l'échelle à trois niveaux de Manulex limitait la finesse de discrimination des synonymes. Pour reprendre l'exemple de *bleu*, il s'est vu attribuer la classe 1 et ressort comme le synonyme le plus simple, *contusion* et *ecchymose* appartiennent tous les deux au niveau 3, sans qu'aucune distinction ne soit faite entre ces deux termes. Un second problème est que, pour une entrée donnée, ReSyf disposait d'une granularité trop fine de sens. Par exemple, pour *souris*, il existait de nombreux sens, parmi lesquels 'espèce de petit rongeur', 'genre de rongeur' et 'rongeur'. Un tel niveau de précision dans la désambiguïsation sémantique n'est pas souhaitable pour la ressource, étant donné les applications prévues (au niveau scolaire).

Une nouvelle version de ReSyf a vu le jour en 2017. Elle surmonte les deux faiblesses citées ci-dessus. Afin de ne pas recourir à l'échelle à trois niveaux de Manulex, François *et al.* (2016) ont proposé une méthode d'ordonnancement de synonymes que nous décrivons dans la section suivante (*cf.* section 6.3). Dans cette section, nous présentons nos contributions dans le processus de

constitution de la ressource lexicale de synonymes : ces derniers sont désambiguïsés (rassemblés par sens) avec une granularité plus optimale de sens.

Nous décrivons d'abord dans la sous-section 6.2.1 un premier travail de construction de la liste de synonymes à partir des sens de BABELNET. Ensuite, dans les sous-sections 6.2.2 et 6.2.3, nous décrivons un deuxième travail mené pour la construction de cette liste par utilisation du réseau lexical JEUXDEMOTS.

6.2.1. Utilisation des synonymes provenant de BABELNET

L'un des obstacles majeurs de la désambiguïsation sémantique est la granularité fine des inventaires de sens (NAVIGLI, 2009). Par exemple, dans WORDNET, les distinctions entre sens sont parfois difficiles à effectuer pour les annotateurs humains (EDMONDS et KILGARRIFF, 2002). Notre objectif est dès lors d'obtenir une ressource de synonymes pour le français qui soit caractérisée par une granularité sémantique plus optimale, car cela facilite alors le processus de distinction des sens en contexte.

La liste de termes que nous avons utilisée provient du réseau sémantique BABELNET[4]. Notre méthode repose sur l'utilisation des sens issus de ce réseau. Chaque sens est associé à un ensemble de synonymes que nous appelons un vecteur de synonymes[5]. Comme nous l'avons mentionné dans le chapitre 1 (*cf.* sous-section 1.2.1) et à la fin du chapitre 3 (*cf.* section 3.4), BABELNET a été construit d'une manière automatique en reliant WORDNET avec plusieurs ressources lexicales et encyclopédiques telles que le WIKTIONNAIRE[6] et WIKIPÉDIA[7], et comprend l'ajout de traductions automatiques entre plusieurs langues. Face à cette masse d'information provenant de BABELNET, nous étions confrontés à deux problèmes majeurs :

1. Le bruit, à savoir la présence de mots techniques et de mots provenant d'une langue étrangère.
2. La granularité de sens qui est trop fine.

Le tableau 6.1 liste le nombre de sens décrits dans BABELNET pour le français et pour chacune des catégories grammaticales ouvertes (noms, adjectifs, adverbes et verbes). Dans ce tableau, les traductions automatiques provenant des ressources utilisées pour construire BABELNET ne sont pas prises en compte. On observe que la classe des noms de BABELNET est largement majoritaire ($\approx 97\%$). Le tableau 6.2 décrit le nombre de mots monosémiques (monos) et

4. Nous avons utilisé la version 2.5.1 (http://babelnet.org/download).
5. Nous signalons que l'objectif du projet RESYF est d'avoir à disposition des représentations de sens-synonymes sans tenir compte de la présence des entités nommées. Nous considérons un sens comme étant un concept.
6. https://www.wiktionary.org
7. https://www.wikipedia.org

polysémiques (polys) dans BabelNet. La classe des noms reste toujours majoritaire que le mot soit ambigu ou non ($\approx 84\%$ des mots polysémiques sont des noms).

POS	BabelNet
Noms	622 132
Adjectifs	7 576
Adverbes	1 634
Verbes	8 050
Total	639 392

Table 6.1. – Nombre de sens décrits dans BabelNet pour le français sans prise en compte des traductions automatiques

POS	Mots monos	Mots polys
Noms	551 365	30 167
Adjectifs	3 954	2 272
Adverbes	893	690
Verbes	2 280	2 878
Total	558 492	36 007

Table 6.2. – Données de BabelNet pour le français sans tenir compte des entités nommées et des traductions automatiques

Afin de réduire la liste de mots-synonymes proposés par BabelNet, d'une part, nous n'avons pas tenu compte des traductions automatiques, et d'autre part, nous avons fait le choix d'utiliser un filtrage sur la base des lemmes présents dans JeuxDeMots. Ce dernier est un jeu associatif (les mots sont proposés directement par des humains).

Pour réduire le nombre de sens par entrée lexicale polysémique, nous avons opté pour l'utilisation de NASARI[8] (Camacho-Collados *et al.*, 2015) (*cf.* chapitre 1, sous-section 1.3.1), afin de ne garder que des sens bien distincts, c'est-à dire, dont la similarité entre sens est faible. NASARI décrit des représentations vectorielles pour les sens de BabelNet.

L'approche que nous avons proposée produit une ressource lexicale de mots-synonymes regroupés en plusieurs sens dont le vocabulaire provient de JeuxDeMots et l'organisation des sens provient de BabelNet.

NASARI ne propose des vecteurs sémantiques que pour les noms. Même si la classe des noms dans BabelNet reste largement majoritaire par rapport aux autres classes ouvertes, la polysémie n'est traitée ici que pour les noms communs. Nous avons utilisé NASARI avec le type de représentation à base de mots pour le calcul de la similarité sémantique entre sens. La similarité sur laquelle nous nous sommes basés pour la comparaison des vecteurs est Weighted Overlap (WO) (Pilehvar *et al.*, 2013) (*cf.* chapitre 2, sous-section 2.2.1). Nous avons préféré d'utiliser la mesure WO au lieu du Cosinus en raison du petit nombre de dimensions dont tiennent compte les vecteurs. En effet, la me-

8. http://lcl.uniroma1.it/nasari

sure COSINUS a tendance à retourner des scores relativement faibles lorsque les dimensions des deux vecteurs sont petites, contrairement à la mesure WO qui n'est pas affectée par le nombre de dimensions.

Filtrage des sens

Nous avons d'abord fait un tri des sens du plus fort vers le plus faible. Le sens le plus fort est celui qui contient le plus grand nombre de connexions sémantiques dans le réseau. Une comparaison entre une paire de sens a été effectuée. Si une similarité forte entre les deux sens existe, le plus fort est gardé et le plus faible est supprimé. Le seuil au-delà duquel une similarité est considérée comme forte est 0.5. La comparaison a été effectuée par la suite sur une autre paire de sens et ainsi de suite jusqu'à l'obtention d'un ensemble de sens distincts.

Nous n'avons pas pris la piste de regroupement de sens parce qu'en général les mots les plus techniques se trouvent dans les niveaux les plus profonds (sens possédant une faible connexion sémantique). Par exemple, pour le nom '*avocat*', BABELNET fournit trois sens pour le français qui décrivent un même raffinement sémantique. Ci-dessous, la liste de synonymes de chaque sens :

1. *avocat, homme de loi*.
2. *avocat, solliciteur*.
3. *avocat, attorney at law*.

Le premier sens possède $2\,208$ connexions sémantiques avec les autres sens du réseau BABELNET contrairement à 272 connexions sémantiques pour le deuxième et 49 pour le troisième. Le '*solliciteur*' est un type d'avocat qui pratique, par exemple, le conseil et la rédaction d'actes. Le nom '*attorney at law*' est obtenu par une traduction automatique. Le tableau 6.3 décrit le nombre d'entrées lexicales par utilisation de BABELNET et de la méthode de filtrage décrite ci-dessus (un filtrage de sens par utilisation de NASARI et un filtrage de mots par utilisation d'un vocabulaire provenant de JEUXDEMOTS).

POS	Mots monos	Mots polys	Total
Noms	17 017	4 309	21 326
Adjectifs	1 377	–	1 377
Adverbes	395	–	395
Verbes	870	–	870
Total	19 659	4 309	23 968

Table 6.3. – Description des données obtenues à partir de BabelNet par filtrage de sens et un vocabulaire provenant de JeuxDeMots

Contrairement aux données décrites dans le tableau 6.2, le nombre de données du tableau 6.3 est nettement inférieur. Sans surprise, la méthode de filtrage nous a permis d'obtenir un jeu de données plus souhaitable pour la ressource ReSyf. Cependant, avec cette méthode, nous avons réussi à réaliser un filtrage de sens seulement pour les noms et il est plus intéressant d'avoir des raffinements sémantiques pour tous les mots à classe ouverte. Dans les sous-sections suivantes, nous proposons une solution à ce problème en passant directement à l'utilisation des sens (raffinements sémantiques) provenant de JeuxDeMots.

6.2.2. Utilisation des synonymes provenant de JeuxDeMots

La méthode, que nous proposons ici, repose sur l'utilisation de JeuxDeMots et tient compte des raffinements sémantiques, s'ils existent, présents dans ce réseau lexical. Comme ce dernier est en constante évolution et, qu'à l'heure actuelle, il propose des synonymes pour les raffinements sémantiques, nous faisons une extraction directe des sens-synonymes. L'avantage de JeuxDeMots est qu'il permet d'avoir une représentation des différents sens d'un mot donné sous la forme d'un arbre (Lafourcade et Joubert, 2009), ce qui n'est pas le cas pour BabelNet. Cela nous permet ainsi d'identifier directement les sens les plus importants, situés au premier niveau de l'arbre. La figure 6.1 décrit la structure hiérarchique des raffinements sémantiques du nom '*phare*'.

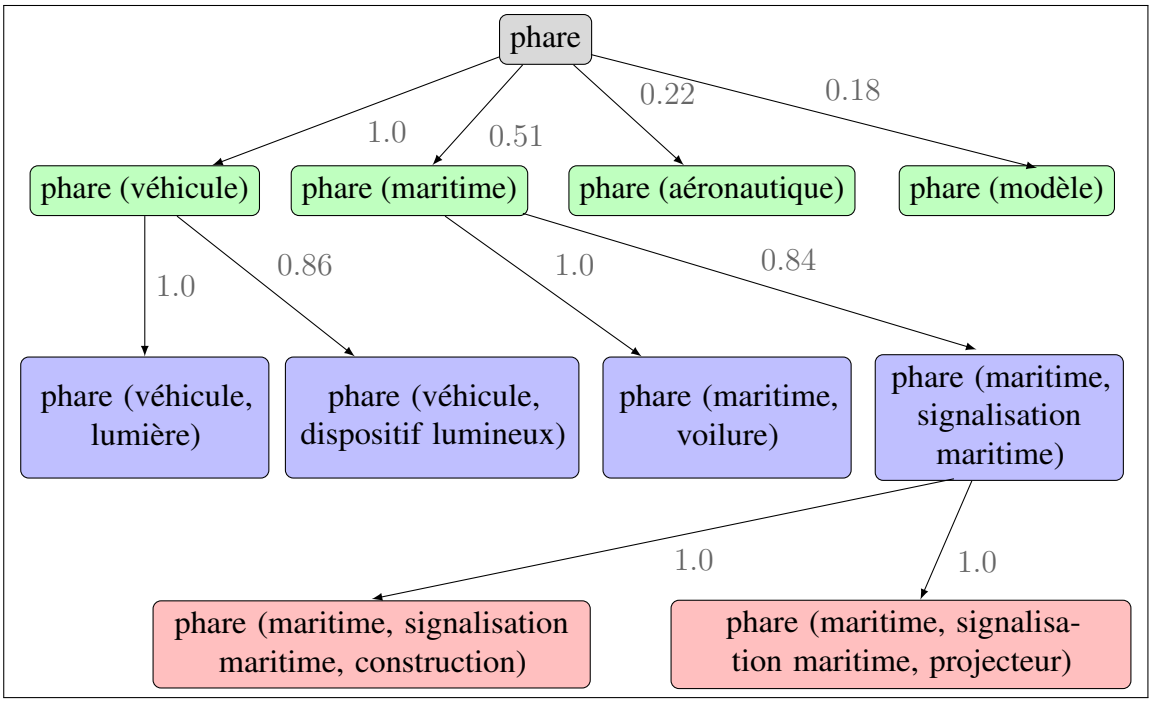


Figure 6.1. – Description de la structure hiérarchique des raffinements sémantiques, à partir de JeuxDeMots, du nom '*phare*'

Le nom '*phare*' possède 4 raffinements sémantiques du premier niveau : (1) '*phare (véhicule)*' ; (2) '*phare (maritime)*' ; (3) '*phare (aéronautique)*' ; et (4) '*phare (modèle)*' ordonnés de gauche à droite selon l'importance de leur poids sémantique. Ce dernier est le poids de la relation *Raffinement sémantique*. Il est normalisé avec la norme infinie qui est définie par la fonction : $\|\vec{X}\|_\infty = max(|x_1|, |x_2|, \ldots, |x_n|)$ avec x_i le ième raffinement sémantique de l'entrée lexicale donnée.

Le raffinement sémantique '*phare (véhicule)*' possède lui-même deux raffinements sémantiques (un deuxième niveau de raffinement) ; quant au raffinement sémantique '*phare (maritime)*', il possède lui-même quatre raffinements sémantiques : deux de niveau 2 et deux de niveau 3. Pour notre ressource, nous tenons compte seulement du premier niveau des raffinements sémantiques lors de l'extraction des synonymes.

Compte tenu de l'aspect associatif du réseau lexical JeuxDeMots, en enrichissement constant à ce jour, la relation de synonymie ne couvre pas tous les raffinements sémantiques. Plus précisément, nous avons remarqué que le réseau de synonymes de JeuxDeMots est très riche par rapport à BabelNet mais pour les mots polysémiques, une grande majorité de synonymes est liée directement à ces mots ambigus mais en aucun cas à leurs raffinements sémantiques. Par exemple, pour le nom '*phare*', JeuxDeMots propose 20 synonymes. Le tableau 6.4 décrit ces synonymes avec leur poids normalisé. Nous les avons récupérés depuis les signatures sémantiques à base de *Synonyme* [9] que nous avons créées et validées dans le chapitre 4.

feu : 1.0	*balise (repérage) :* 0.54	*gloire :* 0.54	*lumière (éclairage) :* 0.54
sémaphore : 1.0	*code :* 0.54	*illustration :* 0.54	*modèle :* 0.54
fanal : 0.64	*code (feux de route) :* 0.54	*lanterne :* 0.54	*projecteur :* 0.54
fleuron : 0.6	*feu (signal) :* 0.54	*lumière :* 0.54	*visionnaire :* 0.54
balise : 0.54	*flambeau :* 0.54	*lumière (clarté) :* 0.54	*guide :* 0.27

Table 6.4. – Liste de synonymes du nom '*phare*' provenant du réseau lexical JeuxDeMots

Parmi les quatre raffinements sémantiques du premier niveau (*cf.* figure 6.1), seulement le raffinement sémantique '*phare (modèle)*' propose des synonymes, à savoir : *modèle* et *guide*. Tous les autres synonymes ne sont associés à aucun raffinement sémantique du nom '*phare*'. Dans la sous-section suivante, nous proposons une méthode de regroupement (*clustering*) permettant de regrouper automatiquement les synonymes non désambiguïsés manuellement dans les raffinements sémantiques. En d'autres termes, il s'agit d'une méthode d'enrichissement automatique des listes de sens-synonymes.

9. Nous avons utilisé les données collectées de la base lexicale du réseau JeuxDeMots datant de Janvier 2018.

Dans la ressource ReSyf, nous ne nous intéressons pas à garder des synonymes avec une étiquette de raffinement sémantique. Par exemple, dans le tableau 6.4, trois synonymes représentent un même mot, à savoir : *lumière*, *lumière (clarté)* et *lumière (éclairage)*. Pour ReSyf, l'objectif par rapport à cet exemple est de garder seulement *lumière*. S'il y avait seulement les deux raffinements sémantiques ('*lumière (clarté)*' et '*lumière (éclairage)*') comme synonymes, nous retirons chaque étiquette de raffinement et nous obtiendrons dans ce cas-là uniquement *lumière* comme synonyme.

Aussi, et avant d'appliquer la méthode de regroupement automatique de sens-synonymes, nous avons vérifié si l'étiquette de chaque raffinement sémantique représentait un synonyme potentiel. Par exemple, *modèle* et *phare* sont synonymes et *modèle* est une étiquette d'un raffinement sémantique du nom '*phare*'. À partir des données de la base lexicale de JeuxDeMots datant de Janvier 2018, nous avons réalisé une extraction de tous les noms de domaine ayant un poids positif non nul en utilisant la relation *Domaine*. L'idée consiste à vérifier si chaque étiquette d'un raffinement sémantique ne présente pas un nom de domaine :

Si une étiquette d'un raffinement sémantique d'une entrée lexicale donnée $Entrée_i$ *n'est pas un nom de domaine alors l'étiquette est ajoutée comme un synonyme désambiguïsé directement au raffinement sémantique.*

Par exemple, *modèle* n'est pas un nom de domaine par contre *aéronautique* l'est. De ce fait, *aéronautique* n'est pas un synonyme pour le raffinement sémantique '*phare (aéronautique)*'.

6.2.3. Enrichissement des listes de sens-synonymes

Nous nous intéressons ici à enrichir la liste de synonymes, associée à chaque raffinement sémantique du premier niveau d'un terme donné, avec des synonymes non désambiguïsés manuellement par les internautes qui jouent à JeuxDeMots. Afin d'inclure, si possible, les synonymes non désambiguïsés dans les listes de sens-synonymes, nous avons utilisé une méthode de désambiguïsation basée sur les signatures sémantiques de mots et de sens (Billami *et al.*, 2018). Nous avons déjà proposé dans le chapitre 4 la méthodologie de création et validation de ces signatures sémantiques. Nous avons fait le choix d'utiliser des signatures à base de la relation *Idée associée* car il s'agit de la relation contenant le plus grand nombre d'instances puisqu'elle englobe tous les termes faisant penser à une entrée lexicale donnée. Aussi, il s'agit de la relation qui grandit le plus rapidement. Le nombre d'instances de cette relation est de $\mathbf{105\ 133\ 268}$ selon les données de la base lexicale datant de Janvier 2018.

Pour un mot-cible polysémique à traiter, nous avons à disposition un ensemble de raffinements sémantiques du premier niveau qui lui est associé et une liste de synonymes non désambiguïsés manuellement. Nous considérons les raffine-

ments sémantiques comme des sens candidats. Le principe consiste à trouver pour chaque synonyme le raffinement sémantique du mot-cible le plus proche afin qu'ils soient associés.

Nous avons proposé deux algorithmes de désambiguïsation sémantique :

1. Le premier algorithme consiste à comparer directement la signature sémantique de chaque raffinement sémantique candidat à la signature sémantique d'un synonyme donné syn_a (*cf.* voir l'algorithme 4 ci-dessous).
2. Le deuxième algorithme utilise l'hypothèse proposée par BUDANITSKY et HIRST (2006) : « *la similarité entre deux mots est celle de leurs sens les plus proches* » (*cf.* voir l'algorithme 5 ci-dessous).

La similarité sémantique que nous utilisons pour comparer deux signatures est une fonction d'activation qui prend en compte l'existence de la relation d'une *idée associée* entre les deux éléments mis en comparaison. Cette fonction consiste à vérifier si l'un des deux éléments à comparer représente une dimension dans la signature sémantique de l'autre. Si c'est vrai, la fonction renvoie une similarité parfaite (score de similarité $= 1$) sinon la similarité COSINUS est estimée en utilisant la signature sémantique des deux éléments. Dans les deux chapitres précédents, nous avons fait référence à cette fonction par Act_1.

Algorithme 4 : Comparaison de chaque sens du mot-cible avec chaque synonyme syn_a

Entrées :
$mot - cible$: mot à traiter
$raff\_sem\ (mot - cible)$: ensemble de sens du premier niveau pour le mot-cible
syn_a : synonyme du mot-cible
ε : seuil de validation d'une similarité
Résultat :
$\mathbf{\hat{Sens}_{mot-cible}}$: sens du mot-cible ayant le meilleur score
Données :
$S_{idée\_a}$: ensemble de signatures dont les dimensions sont des idées associées

Initialisation :
$Score_{raffs\_C} = \emptyset$ // Score des sens du mot-cible
pour chaque $sens_i \in raff\_sem\ (mot - cible)$ **faire**
    $Score = \begin{cases} 1\ si\ (*) \\ Cosinus\ (S_{idée\_a}(sens_i), S_{idée\_a}(syn_a))\ sinon \end{cases}$
    $(*) : sens_i \in S_{idée\_a}(syn_a) \vee syn_a \in S_{idée\_a}(sens_i)$;
    si $(Score \geqslant \varepsilon)$ **alors**
        $Score_{raffs\_C} \leftarrow Score_{raffs\_C} \bigcup (sens_i,\ Score)$;
$\mathbf{\hat{Sens}_{mot-cible}} \leftarrow \mathbf{Best}\ (Score_{raffs\_C})$

Algorithme 5 : Comparaison de chaque sens du mot-cible avec chaque sens d'un synonyme syn_a (comparaison directe avec le synonyme s'il n'a pas de raffinements sémantiques)

Entrées :
$mot - cible$: mot à traiter
$raff\_sem\ (mot - cible)$: ensemble de sens du premier niveau pour le mot-cible
syn_a : synonyme du mot-cible
$raff\_sem\ (syn_a)$: ensemble de sens du premier niveau pour le synonyme syn_a
ε : seuil de validation d'une similarité
Résultat :
$\hat{\mathbf{Sens}}_{\mathbf{mot-cible}}$: sens du mot-cible ayant le meilleur score
Données :
$S_{idée\_a}$: ensemble de signatures dont les dimensions sont des idées associées

Initialisation :
$Score_{raffs\_C} = \emptyset$ // Score des sens du mot-cible
si $raff\_sem\ (syn_a) \neq \emptyset$ **alors**
  pour chaque $sens_i \in raff\_sem\ (mot - cible)$ **faire**
    $Score = 0;\ maxScore = 0;$
    pour chaque $sens_j \in raff\_sem\ (syn_a)$ **faire**
      $Score = \begin{cases} 1\ si\ (*) \\ Cosinus\ (S_{idée\_a}(sens_i), S_{idée\_a}(sens_j))\ sinon \end{cases}$
      $(*) : sens_i \in S_{idée\_a}(sens_j)\ \vee\ sens_j \in S_{idée\_a}(sens_i)$
      si $(Score > maxScore)$ **alors**
        $maxScore = Score;$
    si $(maxScore \geqslant \varepsilon)$ **alors**
      $Score_{raffs\_C} \leftarrow Score_{raffs\_C} \bigcup (sens_i,\ maxScore)$
sinon
  pour chaque $sens_i \in raff\_sem\ (mot - cible)$ **faire**
    $Score = \begin{cases} 1\ si\ (*) \\ Cosinus\ (S_{idée\_a}(sens_i), S_{idée\_a}(syn_a))\ sinon \end{cases}$
    $(*) : sens_i \in S_{idée\_a}(syn_a)\ \vee\ syn_a \in S_{idée\_a}(sens_i);$
    si $(Score \geqslant \varepsilon)$ **alors**
      $Score_{raffs\_C} \leftarrow Score_{raffs\_C} \bigcup (sens_i,\ Score);$
$\hat{\mathbf{Sens}}_{\mathbf{mot-cible}} \leftarrow \mathbf{Best}\ (Score_{raffs\_C})$

L'algorithme 4 permet de désambiguïser un synonyme syn_a en choisissant le sens le plus proche du mot-cible. Pour chaque paire (syn_a, $sens_i$), avec $i \in 1 \ldots n$ (n : nombre de sens candidats du mot-cible), l'algorithme 4 vérifie d'abord si l'un des éléments de la paire représente une dimension dans la signature

sémantique de l'autre élément. Si c'est vrai, le synonyme est intégré directement dans la liste de synonymes du $sens_i$ sinon la similarité COSINUS est estimée. Dans ce dernier cas, les sens les plus proches ayant la meilleure similarité sont sélectionnés. Nous utilisons un seuil $\varepsilon = 0.01$.

L'algorithme 5 permet de désambiguïser un synonyme syn_a en comparant chaque sens candidat à chaque sens de syn_a. Si le synonyme syn_a est polysémique et possède des raffinements sémantiques dans JEUXDEMOTS alors ses sens mis en comparaison sont ses raffinements sémantiques du premier niveau. Dans le cas contraire, si syn_a est monosémique ou n'a pas de raffinements sémantiques dans JEUXDEMOTS, l'algorithme 4 est utilisé ici pour le désambiguïser. La validation des algorithmes 4 et 5 est décrite dans la section 6.4.

Reprenons l'exemple du nom '*phare*' (*cf.* sous-section 6.2.2), si nous appliquons l'algorithme 4, alors le résultat que nous obtenons est montré dans la figure 6.2.

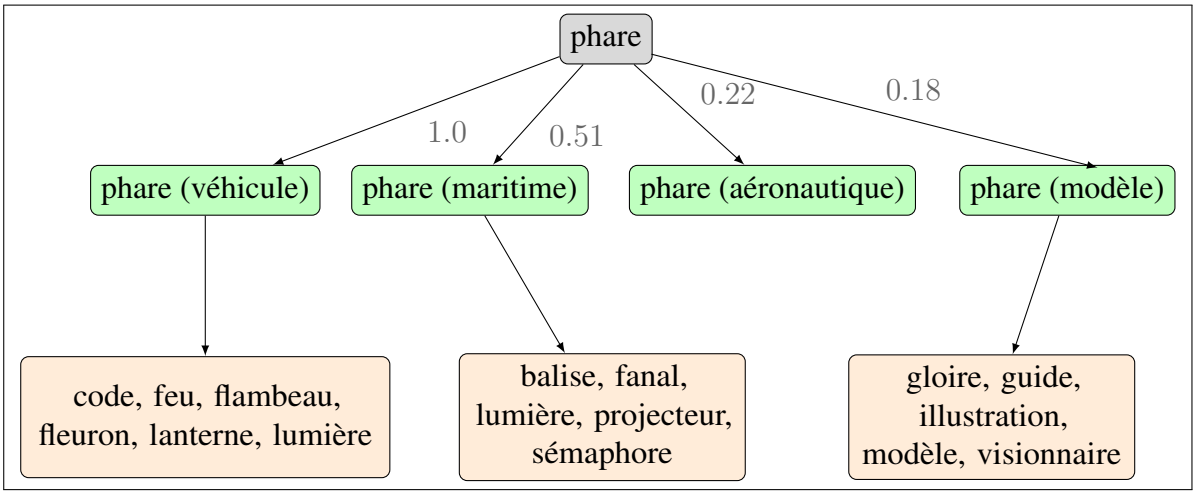


Figure 6.2. – Liste de raffinements sémantiques du nom '*phare*' avec des synonymes désambiguïsés à partir de JeuxDeMots

L'algorithme permet ici de désambiguïser automatiquement tous les synonymes non désambiguïsés manuellement. Nous rappelons que les noms *guide* et *modèle* sont déjà proposés comme synonymes associés au raffinement sémantique '*phare (modèle)*'. Sur l'ensemble, 13 synonymes sont ajoutés automatiquement. Le nom '*lumière*' est ajouté à la fois comme synonyme associé pour le sens '*phare (véhicule)*' et '*phare (maritime)*'. Il se trouve que dans JEUXDEMOTS, *lumière* est une *idée associée* aux deux sens.

La section 6.5 décrit les données quantitatives obtenues par utilisation des sens-synonymes provenant de JEUXDEMOTS. Les données de la dernière version de la ressource lexicale RESYF sont les données obtenues par utilisation des raffinements sémantiques de JEUXDEMOTS.

6.3. Méthode d'ordonnancement de synonymes

François *et al.* (2016) ont proposé un modèle statistique capable de trier les éléments d'un vecteur de synonymes du plus simple au plus complexe en se basant sur un ensemble de variables linguistiques et psycholinguistiques. Ce modèle permet d'attribuer dynamiquement des rangs de 1 à n, en fonction du nombre de synonymes dans le vecteur, au lieu de donner des graduations statiques. Ce type de modèle est régulièrement utilisé en recherche d'information pour trier les résultats d'une requête par ordre de pertinence (Li, 2015).

L'approche utilisée par François *et al.* (2016) est supervisée, de type *pairewise* et repose sur l'application de l'algorithme SVMRank (Herbrich *et al.*, 2000). L'apprentissage repose sur l'utilisation des représentations vectorielles de mots associées à un niveau de difficulté. Les $19\ 038$ lemmes utilisés pour l'entraînement proviennent de Manulex (Lété *et al.*, 2004) et appartiennent à des classes ouvertes (noms, adjectifs, adverbes et verbes). Les dimensions des représentations vectorielles de mots sont des valeurs de variables linguistiques et psycholinguistiques : 69 variables ont été proposées (François *et al.*, 2016). Sur l'ensemble, il peut s'agir de :

1. Critères orthographiques comme le nombre de lettres, phonèmes ou syllabes par mot.
2. Critères sémantiques comme une variable binaire indiquant la présence ou non de l'ambiguïté du mot à traiter selon les informations fournies par le réseau lexical JeuxDeMots.
3. Critères fréquentiels comme le logarithme de la fréquence du mot obtenue dans Lexique3 (New *et al.*, 2007).
4. Variables morphologiques comme le nombre de morphèmes ou une variable binaire indiquant la présence ou non de préfixes et de suffixes.

Des paires d'entraînement ont été créées par sélection de deux mots de difficultés différentes. Ensuite, de nouveaux vecteurs ont été générés en fusionnant les vecteurs des deux mots de chaque paire d'entraînement. La fusion a été effectuée par soustraction des deux vecteurs. Cette méthode a été soutenue par Tanaka-Ishii *et al.* (2010) : les auteurs ont montré qu'elle produisait de meilleurs résultats pour la tâche d'ordonnancement de lisibilité de textes. Chaque vecteur issu de la fusion lui a été attribué un niveau : si le niveau du premier mot de la paire est considéré comme supérieur à celui du second mot dans Manulex alors le niveau de la paire vaut 1, tandis que c'est la valeur -1 qui a été attribuée dans le cas inverse. Comme nous l'avons mentionné au début de ce chapitre, Manulex n'attribue pas un niveau de difficulté unique pour les mots. François *et al.* (2016) ont proposé l'hypothèse suivante : « *le niveau unique de difficulté d'un mot selon* Manulex *est le premier des trois niveaux pour lequel la fréquence du mot n'est pas nulle* ».

Le tableau 6.5 décrit les fréquences estimées d'usage par un million de mots provenant de MANULEX pour le nom '*phare*' et certains de ses synonymes.

Données de MANULEX	CP	CE1	CE2–CM2
phare	86.78	50.58	26.40
code	16.18	28.76	45.96
guide	11.87	5.17	39.14
lumière	**95.86**	**174.51**	**212.51**
modèle	–	6.63	5.29
projecteur	3.96	3.96	12.28

Table 6.5. – Données de Manulex présentées en fréquence estimée d'usage par un million de mots pour le nom '*phare*' et ses synonymes

Avec les $19\,038$ lemmes, il a été possible de créer des centaines de millions de combinaisons de paires d'entraînement. FRANÇOIS *et al.* (2016) ont mis en œuvre un échantillonnage au hasard des paires en retenant uniquement 20 paires par mot. Cela a donné un total de $238\,728$ paires de mots. Une évaluation intrinsèque de la qualité des vecteurs de combinaisons obtenus a été réalisée. Il a été montré que l'utilisation d'un sous-ensemble de 21 meilleures variables, après calcul de la corrélation de Spearman (ρ) entre chaque variable et le niveau de difficulté des paires de mots, est efficace. Le modèle qui a été utilisé pour trier les vecteurs de synonymes pour RESYF tient compte seulement des 21 variables les mieux corrélées.

6.4. Évaluation de la qualité de la ressource RESYF

Nous présentons dans cette section l'évaluation de la qualité des données de RESYF. Tout d'abord, nous évaluons dans la sous-section 6.4.1 les algorithmes de désambiguïsation sémantique que nous avons développés afin de rajouter les synonymes non désambiguïsés manuellement (BILLAMI *et al.*, 2018). Ensuite, nous présentons brièvement dans la sous-section 6.4.2 l'évaluation du modèle d'ordonnancement sur des vecteurs de synonymes (FRANÇOIS *et al.*, 2016).

6.4.1. Évaluation des algorithmes d'enrichissement des listes de sens-synonymes

Afin de valider nos algorithmes de désambiguïsation sémantique pour l'enrichissement des listes de sens-synonymes, nous avons utilisé comme jeu de

test la liste de synonymes désambiguïsés manuellement dans JeuxDeMots [10]. Pour cela, nous avons réalisé une extraction de tous les synonymes associés aux raffinements sémantiques du premier niveau.

Nous avons à disposition un ensemble de 33 039 paires de 'raffinement sémantique – synonyme'. Le tableau 6.6 décrit les résultats de précision de notre évaluation en appliquant les deux algorithmes de *clustering*.

	Annotations correctes	Ensemble total des annotations	%
Algorithme 4	32 802	33 039	**99.28**
Algorithme 5	25 307	33 039	76.6

Table 6.6. – Résultats d'évaluation des deux algorithmes de regroupement sens-synonymes sur l'ensemble des instances de la relation sens-synonyme annotées manuellement dans le réseau lexical JeuxDeMots

En désambiguïsation sémantique, la précision est le rapport entre le nombre d'affectations correctes (bonnes réponses fournies) et le nombre d'affectations réalisées (total de réponses fournies), tandis que le rappel est le rapport entre le nombre d'affectations correctes et le nombre total d'affectations à réaliser (total de réponses à fournir) (Navigli, 2009) (*cf.* chapitre 1, sous-section 1.4.2). Nos signatures sémantiques à base d'idées associées couvrent tous les synonymes et les raffinements sémantiques du jeu de test. Par conséquent, nous avons la même valeur pour la précision et le rappel.

Sans surprise, les résultats montrent qu'il y a une meilleure performance lorsque l'algorithme 4 est utilisé. En effet, la signature sémantique d'un synonyme est plus informative que la signature sémantique de chacun de ses sens.

Nous avons donc appliqué l'algorithme 4 pour enrichir les listes de sens-synonymes utilisées comme données de la version finale de notre ressource ReSyf. Dans la section 6.5, nous présentons en détail les données de la ressource.

6.4.2. Évaluation du modèle d'ordonnancement des synonymes

Les performances du modèle d'ordonnancement ont été évaluées à l'aide d'un jeu de données de référence obtenu grâce à une campagne d'annotation met-

10. Au moment où nous avons obtenu la liste de synonymes désambiguïsés manuellement dans JeuxDeMots, nous nous sommes référé aux données de la base datant de Décembre 2017 ; quant à l'obtention des signatures sémantiques, nous avons utilisé comme dans les chapitres 4 et 5 les données datant de Janvier 2018.

tant des jugements humains à la relative difficulté de synonymes. Cette campagne d'annotation a fait appel à quarante annotateurs. Ces derniers ont été invités à trier manuellement, du plus simple au plus complexe, des synonymes se trouvant dans une liste de quarante sens (vecteurs de synonymes). Ces sens contiennent 2 à 5 synonymes (une moyenne de 3.5 synonymes par vecteur), avec un total de 150 mots à classe ouverte (53% noms, 23% adjectifs, 1% adverbes et 23% verbes). Les synonymes ont été proposés aléatoirement (en termes de difficulté) et n'étaient pas contextualisés. La liste de synonymes utilisés dans cette campagne figure dans l'annexe B.

Le tri a été effectué manuellement par 28 francophones et 12 non-francophones ayant un niveau C1/C2 selon l'échelle du CECR [11]. Ces non-francophones vivent en France depuis au moins cinq ans et ont une autre langue romane comme langue maternelle. Tous étaient des adultes dans le domaine académique : étudiants préparant un master ou un doctorat, professeurs assistants et chercheurs, avec une moyenne d'age de 28.23 ans (écart-type de 10.07).

Le jeu de données de référence final qui a été obtenu après les annotations contient 134 mots et 36 sens. Quatre sens correspondant à 16 synonymes ont été supprimés de la liste originale parce qu'il y avait soit une égalité des annotations (deux synonymes avec un même niveau de difficulté), soit une présence d'un terme non pertinent ou inconnu dans le vecteur (jugé comme tel par plus d'un tiers des annotateurs).

Pour chaque vecteur, le coefficient alpha de Krippendorff (α) [12] a été calculé. L'accord global obtenu est de 0.4, il varie légèrement lorsqu'il est calculé spécifiquement pour des vecteurs à 3 ou 5 synonymes. Il a été remarqué dans cette évaluation que moins il y a de synonymes à annoter, plus l'accord inter-annotateur est élevé.

Le modèle d'ordonnancement permet de donner un niveau de difficulté pour les mots ainsi que pour les expressions polylexicales. Pour ces dernières, le modèle se base actuellement sur une moyenne des mots pleins composants, ce qui reste une approximation de la réalité.

L'évaluation du modèle d'ordonnancement a montré que 83.33% des vecteurs sont triés exactement comme les annotateurs humains les ont proposé, ou avec une légère différence d'un rang. Seulement 16.67% des vecteurs montrent un couple de synonymes classés avec plus de deux rangs de différence.

11. Cadre Européen Commun de Référence pour les langues : apprendre, enseigner et évaluer (CECR).

12. Il s'agit d'une mesure statistique de l'accord inter-annotateurs obtenu lors du codage d'un ensemble d'unités d'analyse en fonction des valeurs d'une variable (ici, le niveau de difficulté des synonymes). Le coefficient alpha de Krippendorff (α) est régulièrement utilisé par les chercheurs dans le domaine de l'analyse de contenu.

L'évaluation du modèle d'ordonnancement a montré que, dans 91% des synonymes, les rangs dans le jeu de données de référence correspondent aux rangs donnés par les annotateurs humains. La ressource ReSyf peut être utilisée pour l'aide à la lecture et peut également être intégrée dans un modèle qui permet de proposer semi-automatiquement la simplification lexicale d'un texte donné.

6.5. Données de la ressource lexicale ReSyf

Dans cette section, nous décrivons les données disponibles dans le lexique ReSyf. Ce lexique fournit des vecteurs de termes équivalents classés selon leur niveau de difficulté de lecture et compréhension [13]. Si nous reprenons l'exemple du nom '*phare*' (*cf.* figure 6.2), avec l'application de l'algorithme d'ordonnancement décrit dans la section 6.3, nous obtenons les vecteurs de synonymes suivants :

1. *phare (véhicule)* : [(1) *feu* ; (2) *code* ; (3) *lumière* ; (4) *phare* ; (5) *fleuron* ; (6) *lanterne* ; (7) *flambeau*].
2. *phare (maritime)* : [(1) *lumière* ; (2) *phare* ; (3) *balise* ; (4) *fanal* ; (5) *projecteur* ; (6) *sémaphore*].
3. *phare (modèle)* : [(1) *modèle* ; (2) *phare* ; (3) *guide* ; (4) *gloire* ; (5) *visionnaire* ; (6) *illustration*].

Afin de distinguer les termes selon les quatre classes ouvertes (noms communs, adjectifs, adverbes et verbes), nous avons filtré tous les mots singuliers dans leur forme lemmatisée avec la ressource de référence pour le français : Lexique3 (New *et al.*, 2007). Pour les expressions polylexicales – *Multiword Expressions* (*MWE*), nous avons utilisé l'analyseur syntaxique Talismane [14] (Urieli, 2013) en tant qu'outil pour une annotation en parties du discours. Nous avons considéré que la catégorie grammaticale d'une expression polylexicale est la même catégorie assignée au premier mot à classe ouverte qui apparaît dans l'expression polylexicale.

Le tableau 6.7 décrit la distribution des entrées lexicales dans ReSyf (nombre total d'entrées : 57 589 dont 10 333 sont polysémiques et 47 256 sont monosémiques [15].

13. La ressource est librement disponible pour téléchargement dans un format LMF (Lexical Markup Framework), utilisé principalement pour coder les données dans un fichier XML (eXtensible Markup Language). Aussi, elle est consultable en ligne (http://cental.uclouvain.be/resyf/) et prochainement sur ORTOLANG (*Open Resources and Tools for Language*).

14. http://redac.univ-tlse2.fr/applications/talismane.html

15. Il est à noter que pour ces 47 256 entrées lexicales, il y a des termes qui ne sont pas ambigus comme il y a aussi des termes ambigus dont le réseau lexical JeuxDeMots ne propose pas actuellement de raffinements sémantiques. Par exemple, le nom *année* a déjà trois raffinements sémantiques : (1) '*année (astronomie)*' ; (2) '*année (calendrier)*' ; et (3) '*année (décennie)*' ou le verbe *refaire* avec deux raffinements sémantiques : (1) '*refaire (reconstruire autrement)*' ; et (2) '*refaire (répéter avec la même façon)*'.

	Noms	Adjectifs	Adverbes	Verbes	Total
#Entrées polysémiques (*EP*)	**6 737**	1 691	126	1 779	**10 333**
#Entrées monosémiques (*EM*)	**30 869**	6 606	1 393	8 388	**47 256**
Entrées singulières	**21 495**	7 635	1 105	5 065	**35 300**
Entrées d'expressions polylexicales	**16 111**	662	414	5 102	**22 289**
Nombre moyen de synonymes par *EP*	12.95	16.93	6.16	**17.97**	14.39
Nombre moyen de synonymes par *EM*	4.19	**9.27**	4.71	6.86	5.39
Nombre moyen de sens par *EP*	2.95	2.65	2.25	**3.03**	2.9
Nombre moyen de synonymes par sens d'une *EP*	4.39	**6.39**	2.73	5.92	4.95

Table 6.7. – Distribution des entrées lexicales dans ReSyf

Le nombre de noms communs est supérieur à celui des termes appartenant aux autres classes ouvertes, que ce soit pour les entrées polysémiques ou monosémiques. La base lexicale du réseau JeuxDeMots est en constante évolution et, par conséquent, plus de raffinements sémantiques seront disponibles dans l'avenir (plus d'entrées polysémiques et plus de synonymes à désambiguïser automatiquement).

Le tableau 6.7 montre que le nombre moyen de synonymes par sens d'une entrée polysémique est de 4.95, ce qui est supérieur à ce que nous pouvons obtenir en utilisant d'autres ressources telles que BabelNet.

Le tableau 6.8 décrit les statistiques relatives à la distribution des annotations de la synonymie dans ReSyf pour les entrées polysémiques. Comme il est montré dans ce tableau 6.8, JeuxDeMots propose 27 466 synonymes associés manuellement aux raffinements sémantiques sur l'ensemble des catégories grammaticales. En appliquant notre algorithme de *clustering* (*cf.* algorithme 4), nous sommes en mesure de rajouter automatiquement 121 182 synonymes désambiguïsés. L'annotation automatique représente un bénéfice de 4.41 de plus de ce que JeuxDeMots propose.

	Noms	Adjectifs	Adverbes	Verbes	Total
#Annotations automatiques	**69 323**	24 851	629	26 379	**121 182**
#Annotations manuelles	**17 954**	3 781	147	5 584	**27 466**

Table 6.8. – Distribution des annotations de la synonymie pour les entrées polysémiques dans ReSyf

Dans les tableaux ci-dessous, nous décrivons plus en détail, d'une part, la distribution des entrées lexicales dans ReSyf pour les mots singuliers (*cf.* voir tableau 6.9) et les expressions polylexicales (*cf.* voir tableau 6.10) et, d'autre part, la distribution des annotations de la synonymie pour les entrées polysé-

miques : mots singuliers (*cf.* voir tableau 6.11) et expressions polylexicales (*cf.* voir tableau 6.12).

	Noms	**Adjectifs**	**Adverbes**	**Verbes**	**Total**
#Entrées polysémiques (*EP*)	**6 241**	1 654	106	1 302	**9 303**
#Entrées monosémiques (*EM*)	**15 254**	5 981	999	3 763	**25 997**
Nombre moyen de synonymes par *EP*	13.79	17.27	6.8	**22.99**	15.61
Nombre moyen de synonymes par *EM*	6.73	10.07	5.79	**11.28**	8.12
Nombre moyen de sens par *EP*	3.06	2.68	2.38	**3.44**	3.04
Nombre moyen de synonymes par sens d'une *EP*	4.51	6.45	2.86	**6.68**	5.14

Table 6.9. – Distribution des mots singuliers dans ReSyf

Pour les mots singuliers, la classe des noms est plus grande en terme de couverture des entrées lexicales. Pour les verbes polysémiques, le nombre moyen de synonymes est de 23. Pour les verbes n'ayant pas de raffinements sémantiques, nous avons remarqué que ce nombre moyen est inférieur (11 synonymes en moyenne). Sur l'ensemble des classes grammaticales et pour les entrées lexicales représentant des mots singuliers, nous avons au moins 5 synonymes en moyenne. Nous avons aussi remarqué que le nombre moyen de synonymes par sens pour les entrées polysémiques est inférieur au nombre moyen de synonymes pour les entrées proposant qu'un seul sens (c'est-à-dire, n'ayant pas de raffinements sémantiques). Cela est tout à fait logique puisque les synonymes sont répartis sur les différents sens des mots ambigus.

	Noms	**Adjectifs**	**Adverbes**	**Verbes**	**Total**
#Entrées polysémiques (*EP*)	**496**	37	20	477	**1 030**
#Entrées monosémiques (*EM*)	**15 615**	625	394	4 625	**21 259**
Nombre moyen de synonymes par *EP*	2.5	1.65	2.75	**4.26**	3.29
Nombre moyen de synonymes par *EM*	1.7	1.63	1.97	**3.27**	2.05
Nombre moyen de sens par *EP*	1.55	1.30	1.6	**1.92**	1.71
Nombre moyen de synonymes par sens d'une *EP*	1.62	1.27	1.72	**2.22**	1.92

Table 6.10. – Distribution des expressions polylexicales dans ReSyf

Nous avons remarqué que le même phénomène se reproduit avec les entrées lexicales représentant des expressions polylexicales (voir tableau 6.10) : (1) la classe des noms est majoritaire ; et (2) le nombre moyen de synonymes est plus grand pour les entrées polysémiques.

Pour les entrées polysémiques singulières, les synonymes rajoutés automatiquement représentent un nombre plus grand que les synonymes proposés manuellement. En effet, l'algorithme 4 de désambiguisation sémantique, que nous

	Noms	Adjectifs	Adverbes	Verbes	Total
#Annotations automatiques	**68 884**	24 822	598	25 388	**119 692**
#Annotations manuelles	**17 151**	3 749	123	4 542	**25 565**

Table 6.11. – Distribution des annotations de la synonymie pour les entrées singulières polysémiques dans ReSyf

	Noms	Adjectifs	Adverbes	Verbes	Total
#Annotations automatiques	439	29	31	**991**	**1 490**
#Annotations manuelles	803	32	24	**1 042**	**1 901**

Table 6.12. – Distribution des annotations de la synonymie pour les entrées d'expressions polylexicales polysémiques dans ReSyf

avons présenté dans la sous-section 6.2.3, nous a permis d'avoir un bénéfice de 4.68 de plus de ce que JeuxDeMots propose comme synonymes associés aux raffinements sémantiques. Pour les entrées lexicales représentant des expressions polylexicales, le bénéfice est de 0.78 : il s'agit de $1\,490$ synonymes rajoutés automatiquement.

6.6. Conclusion

Dans ce chapitre, nous avons présenté ReSyf, une ressource lexicale pour le français proposant des synonymes désambiguïsés et gradués en fonction de leur niveau de difficulté. Notre contribution dans la construction de cette ressource était principalement dans la sélection des synonymes pour chaque sens d'une entrée lexicale donnée. Nous avons testé deux ressources, à savoir : BabelNet et JeuxDeMots. Nous avons remarqué que l'utilisation de JeuxDeMots est bien meilleure. D'abord, parce que les raffinements sémantiques et les synonymes sont proposés par des annotateurs humains. Ensuite, JeuxDeMots offre une structure hiérarchique des raffinements sémantiques qui nous a permis de prendre pour ReSyf uniquement les raffinements sémantiques appartenant au premier niveau (granularité plus optimale par rapport à la granularité trop fine de BabelNet où tous les sens sont représentés sur un même niveau).

D'autre part, nous avons proposé des algorithmes de regroupement de sens-synonymes afin d'enrichir les vecteurs de synonymes de chaque raffinement sémantique d'une entrée lexicale donnée. Ces algorithmes reposent sur l'utilisation des signatures sémantiques de mots et de sens à base de la relation *Idée associée*, la relation qui contient le plus grand nombre d'instances dans JeuxDeMots. L'évaluation de ces algorithmes a été effectuée sur un ensemble de $33\,039$ paires 'raffinement sémantique – synonyme', annotées manuellement

dans le réseau. Dans la plupart des cas, l'algorithme 4, que nous avons proposé, regroupe correctement les synonymes dans le bon sens ($32\ 802$ synonymes sur $33\ 039$).

Les résultats obtenus par l'algorithme d'ordonnancement de synonymes, proposé par François *et al.* (2016), ont été comparés aux annotations humaines et dans 91% des cas, les synonymes sont automatiquement classés du plus simple au plus complexe. ReSyf est un lexique qui peut être utilisé en ligne pour une assistance et un entraînement à la lecture comme il peut être intégré dans un outil de simplification lexicale afin de réduire la complexité lexicale des textes.

Dans le chapitre suivant, nous décrivons une application Android qui a été utilisée sur tablette et que nous avons développée pour des tests de lecture. Ce développement a été mené dans le cadre de l'étude des bénéfices de la simplification lexicale de textes pour faciliter la lecture des enfants dyslexiques et faibles lecteurs.

Simplification lexicale : application pour des tests de lecture

7.1. Introduction

Bien que le support de lecture papier reste encore le plus utilisé lors de la scolarité, l'utilisation du numérique dans les apprentissages scolaires, et concrètement dans l'apprentissage de la lecture, est en augmentation [1]. Par exemple, Ecalle *et al.* (2016) ont proposé des applications sur tablettes tactiles pour aider les enfants en difficulté dans l'apprentissage de la lecture. Ils se sont intéressés à stimuler l'apprentissage du code alphabétique et du décodage (stimulation des habiletés phonologiques, apprentissage des lettres et traitement syllabique).

Dans ce chapitre, nous présentons une application Android, appelée « *Lecture de textes* », que nous avons développée et qui a été utilisée sur tablette. Cette application a permis de proposer un support numérique de lecture pour des tests de lecture s'inscrivant dans le cadre du projet ALECTOR [2]. Notre application a été utilisée par Nandiegou et Reboul (2018) qui ont étudié le bénéfice de la simplification lexicale de textes auprès d'enfants présentant des troubles de la lecture (enfants dyslexiques et faibles lecteurs). Les auteurs se sont basées sur l'hypothèse suivante : « *la lecture de textes simplifiés sur le plan lexical peut améliorer la qualité et la vitesse de la lecture ainsi que la compréhension des textes lus chez des enfants présentant des difficultés de lecture* ». Les tests de lecture menés ont permis de valider cette hypothèse de l'utilité de la simplification lexicale qui utilise essentiellement une désambiguïsation sémantique. Lever l'ambiguïté des mots complexes avant de les simplifier s'avère ainsi une étape importante pour un système de simplification automatique de textes.

Dans ce chapitre, nous présentons tout d'abord dans la section 7.2 les différents corpus de données qui ont été intégrés dans notre application. Ces corpus ont été simplifiés manuellement et ont été proposés avec des questionnaires de compréhension. Nous nous intéressons ensuite aux différentes fonctionnalités que propose l'application « *Lecture de textes* » (*cf.* section 7.3) avant de décrire

1. http://eduscol.education.fr/numerique/dossier/apprendre/tablette-tactile/applications-utiles-enseignement
2. https://alectorsite.wordpress.com

les éléments observés suite à l'expérience effectuée en utilisant notre application (*cf.* section 7.4). Enfin, dans la conclusion, nous résumons les points les plus importants de ce chapitre (*cf.* section 7.5).

7.2. Corpus de données

Dans cette section, nous présentons d'abord les textes qui ont été utilisés pour des tests de lecture ainsi que la méthodologie mise en place par les simplifications manuelles (*cf.* voir sous-section 7.2.1). Ensuite, nous décrivons brièvement les questionnaires de compréhension qui ont été utilisés pour l'évaluation de la compréhension des textes lus par les enfants (*cf.* voir sous-section 7.2.2) avant de terminer par présenter un exemple de texte de lecture et son format XML [3] (*cf.* voir sous-section 7.2.3).

7.2.1. Textes pour des tests de lecture

Les textes ont été choisis parmi des textes scolaires (entre CE1 et CM1). Deux types de textes ont été sélectionnés : (1) textes narratifs issus d'œuvres littéraires déjà adaptées ; et (2) textes documentaires comportant des informations plus scientifiques.

— 5 textes littéraires ont été sélectionnés à partir du manuel de lecture de niveau cours moyen *A Loisir* aux éditions *Hachette* [4], contenant une version abrégée de plusieurs œuvres de littérature jeunesse.

— 5 textes documentaires ont été sélectionnés, ils sont utilisés principalement pour des tests de vitesse de lecture [5] (IReST, *International Reading Speed Texts*) et ont été développés par le IReST Study Group. Ils sont de même longueur égalisés en difficulté et en complexité linguistique.

Les 10 textes ont été simplifiés au niveau lexical. Les outils qui ont été employés dans le cadre de la simplification sont : (1) Manulex [6] (Lété *et al.*, 2004) ; et (2) un dictionnaire de synonymes en ligne [7], utilisé comme base de synonymes.

Les caractéristiques globales de l'ensemble de textes sont présentées dans le tableau 7.1. La troisième et la cinquième colonne du tableau, à savoir : nombre de mots et nombre moyen de mots par phrase, présentent des informations quantitatives sur les textes dans les deux versions : *originale/simplifiée*.

Dans le cadre des tests de lecture, un enfant doit lire tous les textes (5 en version originale et 5 en version simplifiée, une seule version pour chaque texte).

3. eXtensible Markup Language.
4. https://www.hachette.fr
5. http://www.vision-research.eu/index.php?id=641
6. http://www.manulex.org
7. http://www.synonymes.com

Type de texte	Texte	Nombre de mots	Nombre de phrases	Nombre moyen de mots par phrase
Littéraire	**1**	279/273	26	10.73/10.5
	2	290/290	25	11.60/11.60
	3	**321/317**	**34**	9.44/9.32
	4	316/298	28	11.29/10.64
	5	273/273	28	9.75/9.75
	Moyenne	295.80/290.20	28.20	10.56/10.36
Documentaire	**6**	136/130	9	15.11/14.44
	7	127/117	8	15.88/14.63
	8	133/123	9	14.78/13.67
	9	126/125	11	11.45/11.36
	10	128/124	8	**16.00/15.50**
	Moyenne	130.00/123.80	9.00	14.64/13.92
	Moyenne	**212.90/207.00**	**18.60**	**12.60/12.14**

Table 7.1. – Caractéristiques globales des textes utilisés par l'application « *Lecture de textes* »

Le résultat de la simplification est décrit comme suit :

— 60.73% de substitutions par un synonyme de même nature, cela constitue la majorité de la simplification.

— 14.17% de modifications morphologiques, en l'absence de synonyme plus simple. Il s'agit d'un remplacement d'un mot par un autre de la même famille morphologique mais de nature grammaticale différente (par exemple, un nom par un infinitif : *construction* remplacé par *construire* où d'une part, *construction* n'a pas de synonyme plus court, plus fréquent et plus régulier, et d'autre part, *construire* est plus fréquent que *construction* et la terminaison irrégulière est évitée).

— 19.43% de suppressions de mots, en l'absence de synonyme plus simple. Aussi, le mot n'est supprimé ici que seulement et seulement si l'information n'est pas indispensable à la compréhension du texte.

— 5.67% de remplacements de la transcription orthographique des nombres par leur transcription arabe (par exemple, *vingt* par 20).

— Des modifications morpho-syntaxiques secondaires ont été appliquées suite aux substitutions effectuées. Par exemple, *ces murs immenses* deviennent *ces grands murs* et *une période* devient *un moment*.

7.2.2. Questionnaires de compréhension

Pour mesurer l'effet de la simplification lexicale sur la compréhension des textes, un questionnaire à choix multiple pour chaque texte a été proposé. Chaque questionnaire comprend cinq questions, chaque question est proposée avec un choix unique de quatre réponses : une seule réponse juste et trois réponses fausses.

La position de la bonne réponse a été randomisée automatiquement au moyen d'un générateur aléatoire de 1 à 4. Le questionnaire proposé est le même, que l'enfant ait lu la version originale ou simplifiée du texte. Le choix a été fait de ne pas porter les questions sur les unités lexicales simplifiées pour la raison suivante : « *l'enfant qui avait lu le texte en version originale pouvait ne pas reconnaître l'unité lexicale simplifiée* » (*cf.* figure 7.2).

7.2.3. Exemple de texte de lecture et son format XML

Nous présentons dans cette sous-section un texte parmi les 10 textes choisis pour l'expérience. Comme nous l'avons mentionné dans la sous-section 7.2.1, chaque texte a été simplifié manuellement sur le niveau lexical. Nous avons créé un fichier XML pour chaque texte suivant la norme TEI [8] (TEXT ENCODING INITIATIVE). Un fichier XML propose le contenu d'un texte dans ses deux versions : (1) version originale ; et (2) version simplifiée. La figure 7.1 présente le fichier XML contenant les deux versions du sixième texte sélectionné pour les tests de lecture (premier texte documentaire scientifique). La figure 7.2, quant à elle, présente le format utilisé pour le questionnaire de compréhension.

Nous avons fait le choix d'ajouter des noms en français pour certaines balises. Par exemple, <tests_comprehension>...</tests_comprehension> pour présenter un questionnaire de compréhension ou <reponse>...</reponse> pour présenter une réponse possible d'une question donnée.

8. http://www.tei-c.org

```
<?xml version="1.0" encoding="UTF-8" ?>
<TEI xmlns="http://www.tei-c.org/ns/1.0"
        xmlns:xs="http://www.w3.org/2001/XMLSchema"
        xmlns:lis="http://www.lis-lab.fr/"
        schemaLocation="http://www.tei-c.org/ns/1.0
                                http://www.lodel.org/ns/tei.openedition.1.0.xsd">
<teiHeader>
  <fileDesc>
    <titleStmt>
      <title>IREST_6_os</title>
    </titleStmt>
  </fileDesc>
</teiHeader>
<text type="texte_original"><body>
Le castor est un excellent nageur. Dans l'eau, il peut nager à une vitesse atteignant
dix kilomètres heure. Il est protégé du froid grâce à sa fourrure faite de milliers de
poils et à une épaisse couche de graisse. Ses poumons volumineux lui permettent
de rester sous l'eau pendant facilement vingt minutes. Le castor peut non seulement
abattre adroitement des arbres, mais il est aussi un expert pour la construction de
barrages. Quand le castor abat un arbre, il ronge une entaille dans le tronc, de sorte
que les parties supérieure et inférieure ne sont plus reliées que par une surface très
fine. Quand la connexion est étroite, le vent accomplit le reste. Les petites branches
sont coupées par le castor et empilées comme réserve. Les grosses branches sont
séparées et utilisées comme bois pour la construction de barrages.
</body></text>
<text type="texte_simplifie"><body>
Le castor est un très bon nageur. Dans l'eau, il peut nager à une vitesse de 10
kilomètres heure. Il est protégé du froid par sa fourrure faite de milliers de poils et
par une grosse couche de graisse. Ses gros poumons lui permettent de rester sous
l'eau pendant 20 minutes. Le castor peut couper des arbres, mais il est aussi habile
pour construire des barrages. Quand le castor coupe un arbre, il fait une découpe dans
le tronc, ainsi les parties du haut et du bas ne sont plus liées que par une partie très
fine. Quand le lien est fin, le vent fait le reste. Les petites branches sont coupées par
le castor et rangées comme réserve. Les grosses branches sont séparées et utilisées
comme bois pour construire des barrages.
</body></text>
<tests_comprehension>
   <!- - Questionnaire de compréhension - ->
</tests_comprehension>
</TEI>
```

Figure 7.1. – Contenu d'un fichier XML décrivant un texte dans sa version originale et simplifiée

```
<tests_comprehension>
  <question id="1">1. Qu'apprend-on sur le castor ? <reponses>
    <reponse id="11">a. C'est un bon coureur.</reponse>
    <reponse id="12">b. C'est un bon chasseur.</reponse>
    <reponse id="13">c. C'est un bon nageur.</reponse>
    <reponse id="14">d. C'est un bon sauteur.</reponse> </reponses>
  </question>
  <question id="2">2. Qu'est-ce qui protège le castor du froid ? <reponses>
    <reponse id="21">a. Ses pattes et sa queue.</reponse>
    <reponse id="22">b. Ses poumons.</reponse>
    <reponse id="23">c. L'eau chaude.</reponse>
    <reponse id="24">d. Sa fourrure et sa graisse.</reponse> </reponses>
  </question>
  <question id="3">3. Comment le castor peut-il rester sous l'eau 20 minutes ? <reponses>
    <reponse id="31">a. Grâce à son gros cœur.</reponse>
    <reponse id="32">b. Grâce à la même respiration que les poissons.</reponse>
    <reponse id="33">c. Grâce à ses gros poumons.</reponse>
    <reponse id="34">d. Grâce à sa graisse.</reponse> </reponses>
  </question>
  <question id="4">4. Que fait le castor avec les petites branches ? <reponses>
    <reponse id="41">a. Il les jette dans l'eau.</reponse>
    <reponse id="42">b. Il les utilise pour construire des barrages.</reponse>
    <reponse id="43">c. Il les utilise pour dormir dessus.</reponse>
    <reponse id="44">d. Il les coupe et en fait une réserve.</reponse> </reponses>
  </question>
  <question id="5">5. Que fait le castor avec les grosses branches ? <reponses>
    <reponse id="51">a. Il construit des barrages.</reponse>
    <reponse id="52">b. Il construit une maison.</reponse>
    <reponse id="53">c. Il construit une réserve.</reponse>
    <reponse id="54">d. Il construit un radeau.</reponse></reponses>
  </question>
</tests_comprehension>
```

Figure 7.2. – Contenu d'un fichier XML décrivant le questionnaire de compréhension par rapport au premier texte documentaire « *Le castor* »

Au moment où nous avons proposé ce format XML pour l'intégration des dix textes avec les deux versions dans l'application « *Lecture de textes* », que nous décrivons ci-après, nous avons aussi construit un autre fichier en format XML contenant l'ensemble de ces textes mais avec une lemmatisation et une annotation en parties du discours à l'aide de l'outil TALISMANE (URIELI, 2013) ainsi qu'une identification de termes ayant été substitués. L'annexe C présente le

même texte décrit dans la figure 7.1 (*cf.* « *Le castor* ») avec un nouveau format qui est similaire à celui utilisé dans la campagne d'évaluation SemEval–2013 (Navigli *et al.*, 2013) pour la tâche de désambiguïsation sémantique. Nous pouvons considérer, avec l'utilisation de ce nouveau fichier XML, les textes originaux comme des textes appartenant à un corpus d'évaluation pour la tâche de substitution lexicale [9].

7.3. Description de l'application Android « Lecture de textes »

L'application « *Lecture de textes* » propose une lecture phrase par phrase précisément chronométrée mais également enregistrée via le micro de l'appareil utilisé. Cette application a été développée en utilisant le langage de programmation JAVA pour Android. L'expérience effectuée par Nandiegou et Reboul (2018), et que nous présentons brièvement dans la section 7.4, a fait appel à l'utilisation de tablettes qui fonctionnent sur le système d'exploitation mobile Android. Aussi, l'application « *Lecture de textes* » intègre non seulement la lecture de phrases mais également le questionnaire de compréhension qui s'affiche juste après la lecture d'un texte donné.

L'application « *Lecture de textes* », que nous avons développée, propose les fonctionnalités suivantes [10] :

1. Formulaire proposant des champs pour saisir des informations sur le nom, prénom et école scolaire de l'enfant.
2. Randomisation dans l'ordre des textes à lire pour chaque enfant.
3. Randomisation de la version du texte à lire : originale ou simplifiée. Une altération automatique est mise en œuvre : si le texte lu à l'instant t par l'enfant $e1$ est de version originale alors l'enfant $e1$ lira à l'instant $t + 1$ un texte différent avec une version simplifiée.
4. Présentation du texte phrase par phrase : l'enfant appuie sur l'icône 'livre' pour passer à la phrase suivante (*cf.* voir figure 7.3).
5. Enregistrement automatique du son de la lecture à haute voix via le micro de l'appareil utilisé. Chaque lecture de phrase est enregistrée dans un fichier son unique ayant un format de codage audio MP3 [11].
6. Présentation du questionnaire de compréhension après lecture de chaque texte : l'enfant sélectionne une seule réponse parmi les quatre proposées

9. Une perspective de ce travail est d'appliquer nos systèmes de désambiguïsation sémantique pour l'aide à la substitution lexicale. Notre ressource ReSyf pourra être utilisée pour choisir les substituts les plus simples.

10. Cette application n'est pas publiée sur Google Play Store et le recueil des données est effectué avec l'autorisation du tuteur légal de chaque enfant ayant participé à l'expérience.

11. *Third audio format of the* MPEG-1 *standard* (*Moving Picture Experts Group Phase* 1).

pour chacune des cinq questions (*cf.* voir figure 7.4). Pour le questionnaire, l'enfant est aidé avec une lecture d'un examinateur auprès de lui.

7. Recueil des données textuelles de manière automatique. Pour chaque utilisateur, l'application permet de récupérer : le temps de lecture en secondes de chaque phrase, le temps de lecture global d'un texte (en secondes) et les réponses des questionnaires de compréhension.

Figure 7.3. – Présentation de phrase (phrase 1 du texte original 6) sur l'application « *Lecture de textes* »

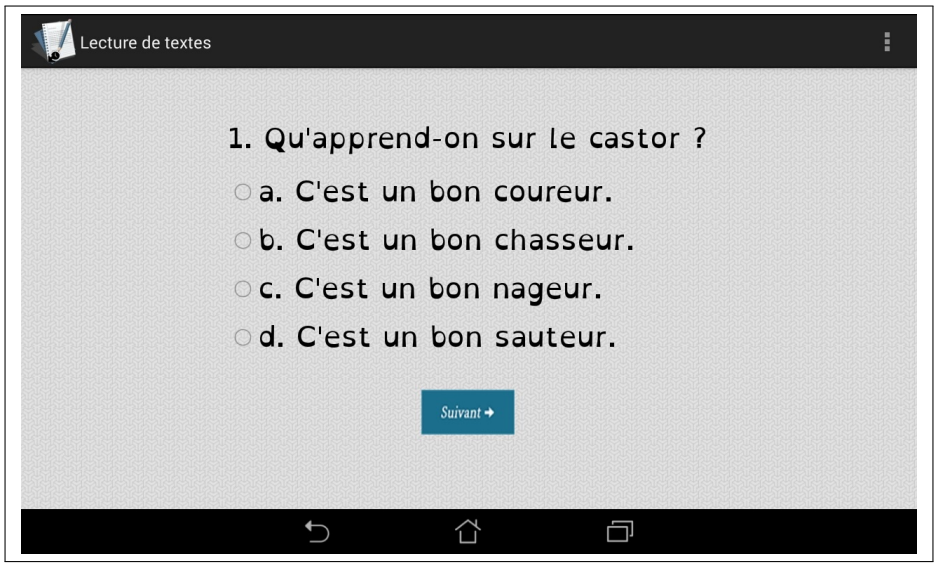


Figure 7.4. – Présentation de question (question 1 du texte 6) sur l'application « *Lecture de textes* »

Étant donné que les enfants ciblés pour les tests de lecture étaient des dyslexiques et faibles lecteurs, nous avons fait le choix d'utiliser la police d'écriture *Open Dyslexic Regular*[12]. Nous avons utilisé un fichier de police de caractères de type TTF (TRUE TYPE FONTS).

La première fois où l'application est utilisée, deux fichiers CSV (COMMA-SEPARATED VALUES) sont créés. Pour de nouvelles utilisations, les fichiers créés sont enrichis avec de nouvelles données :

1. Un fichier « *Résultat_global.csv* » permettant de récupérer les éléments suivants : *Nom, Prénom(s), École primaire, Document, Version du document, id, Phrase* et *Temps de lecture*. Ces éléments sont récupérés pour chaque enfant et pour chaque lecture de phrase. Le point a été considéré comme le délimiteur de phrases.
2. Un fichier « *Synthèse.csv* » permettant de récupérer les éléments suivants : *Nom, Prénom(s), École primaire, Document, Version du document, Temps de lecture* et *Question*$_i$*/réponse* avec $i \in 1 \dots 5$.

La colonne *Document* fournit le nom du texte sélectionné (par exemple, IREST_6_os). La *Version du document* a deux valeurs possibles : *Texte original* ou *Texte simplifié*. L'identifiant *id* est le numéro de la phrase. La colonne *Phrase* fournit le texte de la phrase.

Pour le fichier « *Résultat_global.csv* », le *Temps de lecture* représente le temps ayant été nécessaire pour la lecture d'une phrase donnée alors que pour le fichier « *Synthèse.csv* » , ce temps représente le temps total de lecture d'un texte donné. L'annexe D présente les différentes étapes qui peuvent avoir lieu en utilisant l'application « *Lecture de textes* ».

7.4. Expérimentation avec l'utilisation de l'application « Lecture de textes »

L'application « *Lecture de textes* » a été utilisée sur des tablettes afin de faire lire des textes aux enfants. Deux tablettes ont été employées : le modèle de tablette qui a été utilisé est LENOVO TAB3 7 ESSENTIAL (LENOVO TB3 — 710F). Chaque tablette possédait un écran de 7 pouces ainsi qu'une résolution de 1024 x 600 pixels. Le système d'exploitation mobile utilisé était la version 5.0 d'ANDROID.

L'expérience effectuée a eu lieu de Novembre 2017 à Mars 2018, dans le cabinet d'orthophonistes. Un ensemble de 21 enfants a été présent pour l'expérience (âgés de 9 ans 11 mois à 12 ans 7 mois) et pris en charge en orthophonie.

12. https://www.opendyslexic.org

Les résultats observés suite à l'expérience avec les enfants ayant utilisée l'application « *Lecture de textes* » sont les suivants :

- Les enfants lisent plus rapidement les textes simplifiés que les textes originaux. Cet effet a été remarqué avec une importance majeure pour les textes documentaires scientifiques.
- Le pourcentage moyen de mots mal lus à la lecture des textes simplifiés est significativement inférieur au pourcentage moyen de mots mal lus à la lecture des textes originaux.
- Le pourcentage moyen de mots simplifiés mal lus dans les textes simplifiés est très significativement inférieur au pourcentage moyen de mots d'origine mal lus dans les textes originaux.
- Les réponses à chaque questionnaire de compréhension pour l'ensemble des enfants sont très peu liées à la version du texte lu.
- Les questionnaires de compréhension des textes littéraires n'ont généré que 6.29% de mauvaises réponses contre 23.43% de mauvaises réponses pour les questionnaires des textes documentaires scientifiques.

7.5. Conclusion

Dans ce dernier chapitre, nous avons présenté l'application ANDROID « *Lecture de textes* ». Cette application a été développée en utilisant le langage de programmation JAVA pour ANDROID. Elle présente une lecture de textes, phrase par phrase précisément chronométrée mais également enregistrée via le micro de l'appareil mobile utilisé, avec des tests de compréhension. Elle a été utilisée par deux orthophonistes dans le cadre de l'étude du bénéfice de la simplification lexicale de textes auprès d'enfants présentant des troubles de la lecture.

L'expérience de la lecture, dont l'utilisation de l'application « *Lecture de textes* » a permis de proposer un support numérique, a montré que la simplification lexicale est bénéfique pour améliorer la vitesse de la lecture et réduire le nombre d'erreurs (mots mal lus) pour chaque texte donné. La simplification lexicale constitue ainsi un support pour faciliter la lecture des enfants dyslexiques et faibles lecteurs. Les substituts simplifiés sont une source de moins d'erreurs que les mots d'origine. Dans l'ensemble, il a été remarqué, suite à cette expérience, que les mots complexes provoquent souvent des erreurs de lecture.

Toutefois, la simplification lexicale est généralement accompagnée avec les autres types de simplification comme la simplification syntaxique et discursive. Brunel et Combes (2015), dans une étude similaire précédente, ont étudié le bénéfice de la simplification de textes sur tous les niveaux pour les enfants dyslexiques et ont également observé des gains significatifs sur la vitesse et la qualité de la lecture ainsi que sur la compréhension de textes écrits.

La simplification lexicale de textes représente l'une des étapes les plus importantes de la simplification et consiste essentiellement à remplacer les mots complexes par des substituts équivalents généralement plus courts, plus fréquents et plus réguliers, **tout en conservant le même sens en contexte**. La désambiguïsation sémantique est donc essentielle pour le module de simplification lexicale de textes, il est possible d'intégrer nos systèmes de désambiguïsation sémantique dans la perspective d'une simplification semi-automatique du lexique.

Conclusions et travail à venir

Bilan de la thèse

Dans ce travail, nous avons présenté nos contributions dans le cadre du projet ALECTOR [13] (***A**ide à la **LECT**ure pour amélio**R**er l'accès aux documents pour enfants dyslexiques et faibles lecteurs*), qui aborde le défi de la simplification automatique de textes. Nous avons proposé différents systèmes de désambiguïsation sémantique pour l'aide à la simplification au niveau lexical. Ce niveau constitue l'un des challenges les plus importants de la simplification automatique de textes. Cependant, les autres niveaux de simplification restent aussi importants afin d'agir sur les erreurs visuo-attentionnelles ou les formulations syntaxiques complexes, par exemple.

Comme nous avons vu dans les deux premiers chapitres, le processus de simplification lexicale se heurte à un problème de taille en TAL, celui de la désambiguïsation sémantique (essentielle à d'autres applications comme la traduction automatique et la recherche d'information). Malgré de nombreux travaux dans le domaine, de nombreux aspects restent encore problématiques. En effet, une source de difficulté réside dans les ressources lexico-sémantiques qu'un système de désambiguïsation utilise. Dans certains cas, les connaissances provenant de ces ressources ne sont pas suffisantes ou bien la qualité des descriptions des mots ou des sens n'est pas assez précise, alors que dans d'autres cas cette description peut être présente mais trop fine pour être utilisée directement.

Dans le chapitre 3, nous avons vu que les systèmes de désambiguïsation sémantique utilisant la ressource BabelNet comme inventaire de sens possèdent plusieurs limites, notamment la granularité très fine de sens et l'absence d'une représentation structurée des sens d'un mot donné, ce qui rend la tâche de désambiguïsation sémantique encore plus difficile. Toutefois, la plupart des systèmes de désambiguïsation que nous avons décrits proposent une optimisation combinatoire qui consiste à choisir les voisins les plus proches autour de chaque mot à désambiguïser. Le principe est de comparer chaque sens candidat uniquement à chaque sens des voisins les plus proches au lieu de faire la comparaison avec chaque sens de tous les mots du contexte. Nous avons vu que l'utilisation des voisins distributionnels (sélectionnés après analyse distributionnelle

13. https://alectorsite.wordpress.com

d'un corpus de travail de grande taille) permet d'avoir de meilleurs résultats de désambiguïsation que l'utilisation des voisins linéaires. Aussi, BabelNet comme WordNet, permettent de fournir une description des sens de mots. Cependant, il n'existe pas dans ces deux réseaux sémantiques une description unique du mot lui-même, ce qui n'est pas le cas avec JeuxDeMots.

Dans le chapitre 4, nous avons proposé une méthode de création de représentations sémantiques à la fois pour les mots et pour les sens, ce que nous appelons "signatures sémantiques". Ces signatures sémantiques sont créées à partir du réseau lexical JeuxDeMots. Nous avons réalisé une double évaluation de la qualité de ces signatures : (1) évaluation intrinsèque sur les mesures de similarité sémantique ; et (2) évaluation extrinsèque sur la substitution lexicale. Nous avons vu que la combinaison de relations lexico-sémantiques permet l'obtention d'une meilleure qualité des signatures sémantiques.

Dans le chapitre 5, nous avons proposé de nouveaux systèmes de désambiguïsation sémantique utilisant les signatures sémantiques. Ces nouveaux systèmes répondent aux limites rencontrées lors de l'utilisation de BabelNet. JeuxDeMots est utilisé ici comme inventaire de sens. Ce réseau lexical décrit les sens comme des raffinements sémantiques et permet de fournir une structure hiérarchique des sens, cela permet d'avoir une granularité plus optimale de sens si nous tenons compte uniquement du premier niveau de la structure. Contrairement aux systèmes présentés au chapitre 3, la sélection des voisins est faite ici en se basant sur une relation associative définie dans JeuxDeMots, à savoir : la relation *Inhibition*. Toutefois, la sélection est faite pour chaque sens candidat. Plus concrètement, un voisin est sélectionné seulement et seulement s'il n'existe pas une inhibition entre lui et le sens candidat du mot à désambiguïser. Cette manière de procéder à la sélection des mots du contexte donne l'avantage aux sens qui excluent moins de mots d'avoir un score plus important. Aussi, la comparaison s'effectue non pas entre sens (sens candidats et sens des voisins) mais plutôt directement entre sens candidats et voisins (représentations vectorielles des voisins). Nous avons proposé une variante utilisant les *word embeddings* et permettant de comparer chaque représentation de sens candidat à la représentation du contexte. Cela permet d'avoir un temps d'exécution plus réduit. Les systèmes de désambiguïsation décrits dans le chapitre 5 utilisent essentiellement les données provenant de JeuxDeMots. Ce réseau lexical est en constante évolution et par conséquent les résultats de désambiguïsation seront meilleurs dans l'avenir, au fur et à mesure que le réseau s'enrichit.

Dans le chapitre 6, nous avons présenté nos contributions dans le processus de constitution de ReSyf, une ressource lexicale pour le français proposant des synonymes désambiguïsés et gradués en fonction de leur niveau de difficulté de lecture et compréhension. Le vocabulaire de ReSyf provient de JeuxDeMots et il en est de même pour l'organisation des sens. Compte tenu de l'aspect associatif de JeuxDeMots, la relation de synonymie ne couvre pas tous les raffinements sémantiques. Nous avons proposé une méthode d'enrichissement automatique

des listes de sens-synonymes. Il s'agit de regrouper automatiquement, pour une entrée lexicale donnée, des synonymes non désambiguïsés manuellement (par les internautes qui jouent à JeuxDeMots) avec les sens de l'entrée lexicale. Ensuite, les synonymes de chaque liste ont été classés du plus simple au plus complexe par utilisation d'un modèle d'ordonnancement en se basant sur un ensemble de variables linguistiques et psycholinguistiques.

Enfin, nous avons présenté notre dernière contribution dans le chapitre 7. Il s'agit du développement de l'application Android « *Lecture de textes* » créée dans le cadre de l'étude du bénéfice de la simplification lexicale de textes auprès d'enfants dyslexiques et faibles lecteurs. Notre application a été utilisée sur des tablettes tactiles et a permis de proposer un support numérique pour des tests de lecture à « voix haute», lus phrase par phrase, précisément chronométrée mais également enregistrée via le micro des tablettes. Ces tests de lecture s'inscrivent dans le cadre du projet ALECTOR. Les résultats obtenus après cette étude ont montré que la lecture de textes simplifiés sur le plan lexical améliore la qualité et la vitesse de la lecture ainsi que la compréhension des textes.

Pour conclure, cette thèse nous a permis de mettre en œuvre différents modèles de désambiguïsation sémantique dans le cadre du développement d'un système de simplification automatique de textes. Nous avons vu que la simplification manuelle au niveau lexical était bénéfique pour la lecture auprès d'enfants dyslexiques et faibles lecteurs, il en reste à faire la même expérience avec des textes simplifiés automatiquement.

Perspectives

Des perspectives s'ouvrent à nous à l'issue de ce travail. Ci-dessous, nous en développons quelques-unes.

Nous avons effectué seulement une évaluation intrinsèque de nos systèmes de désambiguïsation sémantique. Le premier travail qui reste à faire, et dans le but de mieux valider ces systèmes, est d'effectuer une évaluation extrinsèque pour la tâche de substitution lexicale. Il s'agit d'une démarche nécessaire qui n'a pas pu être menée faute de temps. Concrètement, deux corpus d'évaluation peuvent être utilisés :

1. Le corpus SemDis que nous avons décrit et utilisé pour la validation de la qualité des signatures sémantiques de mots (*cf.* chapitre 4). Après avoir réalisé une désambiguïsation sémantique pour chacun des mots-cibles, nous pouvons utiliser les synonymes provenant de ReSyf. À ce stade, nous pouvons soit prendre la liste telle qu'elle se présente dans ReSyf (synonymes triés en fonction de leur niveau de difficulté, du plus simple au plus complexe), soit calculer un score de similarité sémantique pour chaque synonyme désambiguïsé (substitut candidat) en fonction du contexte dans lequel les mots-cibles apparaissent.

2. Le corpus de simplification lexicale (*cf.* voir annexe C pour un exemple de texte). Après désambiguïsation sémantique, les synonymes provenant de ReSyf peuvent être utilisés pour réaliser une substitution lexicale. À ce stade, nous prenons la liste de synonymes telle qu'elle se présente dans notre ressource (l'ordre à prendre est celui du tri en fonction du niveau de difficulté).

Par ailleurs, JeuxDeMots propose un type de relation portant sur la prédiction de ce que peut faire un sujet ou ce qui peut être fait avec un objet : il s'agit des relations *Agent* et *Patient*. Par exemple, pour la relation *Patient*, il existe une instance dans JeuxDeMots entre '*souris (rongeur)*' et '*manger*', alors qu'il n'y a pas une instance entre '*souris (informatique)*' et '*manger*'. Ces instances peuvent être étudiées en tant que traits syntaxiques, ce qui nous ramène à enrichir nos systèmes de désambiguïsation à base de connaissances provenant de JeuxDeMots. Cependant, avant de désambiguïser, il faut tenir compte de l'analyse syntaxique de la phrase dans laquelle les mots-cibles apparaissent. Par exemple, les deux phrases « *le chat mange la souris* » et « *la souris est mangée par le chat* » racontent la même chose (c'est bien la *souris* qui se fait manger à chaque fois) mais le sujet des deux phrases n'est pas le même. La première phrase est à la forme (ou voix) active car c'est le sujet qui effectue l'action. La deuxième phrase est à la forme passive car c'est le sujet qui subit l'action. Nous pensons que l'utilisation des traits syntaxiques récupérés depuis JeuxDeMots peut améliorer les résultats obtenus par nos systèmes de désambiguïsation sémantique. Concrètement, les traits syntaxiques peuvent être utilisés comme suit :

— Premièrement, on pourrait renforcer le score de similarité des sens candidats possédant des traits syntaxiques liés à la phrase dans laquelle le mot à désambiguïser apparaît. Le score à renforcer peut être local comme il peut être global, c'est-à-dire, il peut s'agir soit du score de similarité entre un sens candidat et un mot précis du contexte, soit du score de similarité entre un sens candidat et l'ensemble des mots du contexte. Le renforcement peut être, par exemple, avec une multiplication du score par un coefficient donné.

— Ensuite, on pourrait utiliser les traits syntaxiques dans le cas où deux sens candidats ou plus ont le meilleur score de similarité (c'est-à-dire, les scores des sens sont à égalité). En effet, nous pouvons utiliser les traits syntaxiques pour prendre une décision sur le sens le plus adéquat : le sens choisi pourra être celui qui possède le plus de traits syntaxiques par rapport à la phrase dans laquelle le mot à désambiguïser apparaît. Nous rappelons que pour un tel cas, nous avons déjà appliqué une première heuristique qui consistait à retourner le sens ayant le poids le plus important dans JeuxDe-

MOTS. L'utilisation des traits syntaxiques représente donc pour ce cas une deuxième heuristique pour choisir le sens le plus adéquat en cas d'égalité. Nous pouvons même utiliser à la fois les deux heuristiques en cas d'égalité du nombre de traits syntaxiques ou du poids entre les sens candidats.

Avec l'évolution constante de la base lexicale du réseau JEUXDEMOTS et la possibilité d'enrichir nos systèmes de désambiguïsation avec les traits syntaxiques, nous pensons pouvoir annoter sémantiquement en sens et d'une manière efficace un corpus de grande taille. Cela nous permettra ainsi d'avoir à disposition un corpus d'apprentissage réunissant des instances désambiguïsées de mots. En effet, les systèmes de désambiguïsation sémantique les plus performants sont basés sur un apprentissage supervisé et utilisent généralement un corpus annoté manuellement en sens. Cependant, aucun corpus de de ce genre n'existe pour le français. L'annotation manuelle en sens pour un corpus de grande taille est une tâche très coûteuse et demande beaucoup de temps. Par conséquent et plus généralement, la construction automatique d'un corpus annoté en sens pour le français a un intérêt majeur pour le développement et l'évaluation des systèmes de désambiguïsation basés sur un apprentissage supervisé.

Enfin, avoir à disposition un corpus annoté sémantiquement en sens nous permettra d'entraîner des *sense embeddings*. Nous avons proposé dans le chapitre 5 des systèmes de désambiguïsation utilisant des vecteurs de sens construits à partir des *word embeddings*. Ces vecteurs de sens représentaient le vecteur centroïde défini à partir des vecteurs de tous les mots singuliers appartenant aux signatures sémantiques des sens. Nos systèmes de désambiguïsation à base d'*embeddings* peuvent être améliorés en utilisant directement, d'une part, des *word embeddings* et, d'autre part, des *sense embeddings* entraînés sur un corpus de grande taille annoté en sens. En général, une désambiguïsation d'une meilleure qualité permet de fournir des substituts pertinents et est essentielle dans le cadre de la simplification lexicale. L'enjeu de la désambiguïsation sémantique est donc capital pour le bon déroulement de la substitution lexicale.

Bibliographie

ACHANANUPARP, Palakorn, HU, Xiaohua et SHEN, Xiajiong (2008). « The Evaluation of Sentence Similarity Measures ». In : *Proceedings of the 10th International Conference on Data Warehousing and Knowledge Discovery*. DAWAK '10. Turin, Italie : Springer-Verlag, p. 305–316 (cf. p. 52).

AGIRRE, Eneko et EDMONDS, Philip (2007). *Word Sense Disambiguation : Algorithms and Applications*. Springer Publishing Company, Incorporated (cf. p. 44).

AGIRRE, Eneko et MARTINEZ, David (2001). « Knowledge Sources for Word Sense Disambiguation ». In : *Text, Speech and Dialogue : 4th International Conference,* TSD 2011. Sous la dir. de MATOUŠEK VÁCLAV, MAUTNER PAVEL, MOUCEK ROMAN ET TAUŠER KAREL. T. 2166. Lecture Notes in Computer Science. Berlin, Allemagne : Springer, p. 1–10 (cf. p. 26).

AGIRRE, Eneko, MARTÍNEZ, David, DE-LACALLE, Oier López et SOROA, Aitor (2006). « Two Graph-based Algorithms for State-of-the-art WSD ». In : *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*. Sydney, Australie, p. 585–593 (cf. p. 38).

AGIRRE, Eneko et STEVENSON, Mark (2007). « Knowledge Sources for WSD ». In : *Word Sense Disambiguation : Algorithms and Applications*. Sous la dir. d'AGIRRE ENEKO ET EDMONDS PHILIP. T. 33. Text, Speech, and Language Technology. Springer. Chap. 8, p. 217–251 (cf. p. 26).

APIDIANAKI, Marianna et SAGOT, Benoît (2012). « Applying cross-lingual WSD to wordnet development ». In : *Proceedings of the Eighth International Conference on Language Resources and Evaluation*. Istanbul, Turquie, p. 833–840 (cf. p. 28).

ATKINS, Sue (1992). « Tools for Computer-aided Corpus Lexicography : the Hector Project ». In : *Papers in Computational Lexicography*. COMPLEX '92 41. Sous la dir. de FERENC KIEFER, GÁBOR KISS ET JÚLIA PAJZS, p. 1–59 (cf. p. 46).

AUDIBERT, Laurent (2003). « Outils d'exploration de corpus et désambiguïsation lexicale automatique ». Thèse de doct. Université de Provence – Aix-Marseille I (cf. p. 25).

AUDIBERT, Laurent (2007). « Désambiguïsation lexicale automatique : sélection automatique d'indices ». In : *Traitement Automatique des Langues Naturelles (*TALN*)*. Toulouse, France : IRIT Press, p. 13–22 (cf. p. 40).

BAEZA-YATES, Ricardo, RELLO, Luz et DEMBOWSKI, Julia (2015). « CASSA : a context-aware synonym simplification algorithm ». In : *Proceedings of the 2015* NAACL :HLT *Conference*, p. 1380–1385 (cf. p. 123).

BAKER, Simon, REICHART, Roi et KORHONEN, Anna (2014). « An Unsupervised Model for Instance Level Subcategorization Acquisition ». In : *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (*EMNLP*)*. Doha, Qatar, p. 278–289 (cf. p. 63).

BAKX, Gerard Escudero (2006). « Machine Learning Techniques for Word Sense Disambiguation ». Thèse de doct. École polytechnique de Catalogne, Barcelone : Département LSI (cf. p. 31).

BALDWIN, Timothy et KIM, Su Nam (2009). « Multiword Expressions ». In : *Handbook of Natural Language Processing (*NLP*)*. Boca Raton, Floride, USA : Chapman and HALL/CRC, p. 267–292 (cf. p. 112).

BANERJEE, Satanjeev et PEDERSEN, Ted (2002). « An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet ». In : *Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing*. CICLING '02. Londres, UK : Springer-Verlag, p. 136–145 (cf. p. 71).

BANERJEE, Satanjeev et PEDERSEN, Ted (2003). « Extended Gloss Overlaps As a Measure of Semantic Relatedness ». In : *Proceedings of the 18th International Joint Conference on Artificial Intelligence*. IJCAI '03. Acapulco, Mexique : Morgan Kaufmann Publishers Inc., p. 805–810 (cf. p. 45, 59, 60).

BARONI, Marco, BERNARDINI, Silvia, FERRARESI, Adriano et ZANCHETTA, Eros (2009). « The WaCky wide web : a collection of very large linguistically processed web-crawled corpora ». In : *Language Resources and Evaluation* 43.3, p. 209–226 (cf. p. 31, 101, 111, 122).

BARONI, Marco, DINU, Georgiana et KRUSZEWSKI, Germán (2014). « Don't count, predict ! A systematic comparison of context-counting vs. context-predicting semantic vectors ». In : *52nd Annual Meeting of the Association for Computational Linguistics,* ACL – *Proceedings of the Conference* 1, p. 238–247 (cf. p. 32, 35).

BARONI, Marco et LENCI, Alessandro (2010). « Distributional Memory : A General Framework for Corpus-based Semantics ». In : *Computational Linguistics* 36.4. MIT Press, p. 673–721 (cf. p. 34).

BENGIO, Yoshua, DUCHARME, Réjean, VINCENT, Pascal et JANVIN, Christian (2003). « A Neural Probabilistic Language Model ». In : *The Journal of Machine Learning Research* 3. JMLR.org, p. 1137–1155 (cf. p. 34).

BENGIO, Yoshua, SCHWENK, Holger, SENÉCAL, Jean-Sébastien, MORIN, Fréderic et GAUVAIN, Jean-Luc (2006). « Neural probabilistic language models ». In : *Innovations in Machine Learning*, p. 137–186 (cf. p. 33).

BHINGARDIVE, Sudha, SINGH, Dhirendra, MURTHY V, Rudra, REDKAR, Hanumant et BHATTACHARYYA, Pushpak (2015). « Unsupervised Most Frequent Sense Detection using Word Embeddings ». In : HLT–NAACL. Sous la dir. de MIHALCEA RADA, CHAI JOYCE YUE ET SARKAR ANOOP, p. 1238–1243 (cf. p. 44, 45, 116).

BILLAMI, Mokhtar Boumedyen (2015). « Désambiguïsation lexicale à base de connaissances par sélection distributionnelle et traits sémantiques ». In : *Actes de la 22ème Conférence sur le Traitement Automatique des Langues Naturelles et 17ème Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues,* ***Prix du meilleur article RÉCITAL***. Caen, France, p. 13–24 (cf. p. 72).

BILLAMI, Mokhtar Boumedyen et GALA, Núria (2016). « Approches d'analyse distributionnelle pour améliorer la désambiguïsation sémantique ». In : *13ème Journées internationales d'Analyse statistique des Données Textuelles (*JADT*)* (cf. p. 72).

BILLAMI, Mokhtar Boumedyen et GALA, Núria (2017). « Création et validation de signatures sémantiques : application à la mesure de similarité sémantique et à la substitution lexicale ». In : *Actes de la 24ème Conférence sur le Traitement Automatique des Langues Naturelles et 19ème Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues*. Orléans, France (cf. p. 94).

BILLAMI, Mokhtar Boumedyen, GALA, Núria et FRANÇOIS, Thomas (2018). « RESYF : a French lexicon with ranked synonyms ». In : *The 27th International Conference on Computational Linguistics (*COLING *2018)*. Santa Fe, Nouveau-Mexique, USA (cf. p. 130, 135).

BIRAN, Or, BRODY, Samuel et ELHADAD, Noémie (2011). « Putting It Simply : A Context-aware Approach to Lexical Simplification ». In : *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics : Human Language Technologies : Short Papers*. T. 2. HLT '11. Portland, Oregon, p. 496–501 (cf. p. 49).

BLEI, David M., NG, Andrew Y. et JORDAN, Michael I. (2003). « Latent Dirichlet Allocation ». In : *Journal of Machine Learning Research* 3. JMLR.org, p. 993–1022 (cf. p. 34, 53).

BOHNET, Bernd et NIVRE, Joakim (2012). « A Transition-based System for Joint Part-of-speech Tagging and Labeled Non-projective Dependency Parsing ». In : *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. EMNLP–CONLL '12. L'île de Jeju, Corée du Sud, p. 1455–1465 (cf. p. 74).

BRODY, Samuel et LAPATA, Mirella (2009). « Bayesian Word Sense Induction ». In : *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*. EACL '09. Athènes, Grèce, p. 103–111 (cf. p. 38).

BROUWERS, Laetitia, BERNHARD, Delphine, LIGOZAT, Anne-Laure et FRANÇOIS, Thomas (2014). « Syntactic sentence simplification for French ». In : *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations (*PITR*) @EACL 2014*. Göteborg, Suède, p. 47–56 (cf. p. 17).

BRUNEL, Aurore et COMBES, Mathilde (2015). « Simplification de textes pour faciliter leur lisibilité et leur compréhension ». Mém.de mast. Faculté de médecine de Marseille, France : Mémoire en vue de l'obtention du certificat de capacité en orthophonie (cf. p. 153).

BRUNI, Elia, TRAN, Nam Khanh et BARONI, Marco (2014). « Multimodal distributional semantics ». In : *Journal of Artificial Intelligence Research* 49, p. 1–47 (cf. p. 63).

BRUNSWICK, Nicola (2010). « Unimpaired reading development and dyslexia across different languages ». In : *Learning to read and spell in different orthographies*. Sous la dir. de BRUNSWICK NICOLA, MCDOUGALL SINE ET DE MORNAY DAVIES, PAUL. Hove : Psychology Press, p. 131–154 (cf. p. 16).

BUDANITSKY, Alexander et HIRST, Graeme (2006). « Evaluating WordNet-based Measures of Lexical Semantic Relatedness ». In : *Computational Linguistics* 32.1. MIT Press, p. 13–47 (cf. p. 50, 95, 131).

BURNARD Lou, ed. (2007). « Reference Guide for the British National Corpus (XML Edition) ». In : *British National Corpus Consortium* (cf. p. 31).

BUSCALDI, Davide, LE ROUX, Joseph, FLORES, Jorge J. García et POPESCU, Adrian (2013). « LIPN-CORE : Semantic Text Similarity using n-grams, WordNet, Syntactic Analysis, ESA and Information Retrieval based Features ». In : *Proceedings of the Second Joint Conference on Lexical and Computational Semantics,* *SEM *2013*. Atlanta, Géorgie, USA, p. 162–168 (cf. p. 53).

CAI, Jun Fu, LEE, Wee Sun et TEH, Yee Whye (2007). « NUS-ML : Improving Word Sense Disambiguation Using Topic Features ». In : *Proceedings of the 4th International Workshop on Semantic Evaluations*. SEMEVAL '07. Prague, République tchèque, p. 249–252 (cf. p. 37).

CALZOLARI, Nicoletta et CORAZZARI, Ornella (2000). « SENSEVAL/ROMANSEVAL : The Framework for Italian ». In : *Computers and the Humanities* 34. Kluwer Academic Publishers, p. 61–78 (cf. p. 45).

CAMACHO-COLLADOS, José, PILEHVAR, Mohammad Taher et NAVIGLI, Roberto (2015). « NASARI : a Novel Approach to a Semantically-Aware Representation of Items ». In : *Proceedings of the 2015* NAACL :HLT *Conference*. Denver, Colorado, p. 567–577 (cf. p. 32, 94, 95, 126).

CAMACHO-COLLADOS, José, PILEHVAR, Mohammad Taher et NAVIGLI, Roberto (2016). « NASARI : Integrating explicit knowledge and corpus statistics for a multilingual representation of concepts and entities ». In : *Artificial Intelligence* 240, p. 36–64 (cf. p. 33).

CARPUAT, Marine et WU, Dekai (2007). « Improving Statistical Machine Translation using Word Sense Disambiguation ». In : *In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, p. 61–72 (cf. p. 19, 25).

CARROLL, John, MINNEN, Guido, CANNING, Yvonne, DEVLIN, Siobhan et TAIT, John (1998). « Practical Simplification of English Newspaper Text to Assist

Aphasic Readers ». In : *Workshop on Integrating Artificial Intelligence and Assistive Technology*, p. 7–10 (cf. p. 15).

CHEN, Xinxiong, LIU, Zhiyuan et SUN, Maosong (2014). « A Unified Model for Word Sense Representation and Disambiguation ». In : *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP, A meeting of SIGDAT, a Special Interest Group of the* ACL. Doha, Qatar, p. 1025–1035 (cf. p. 34, 40, 112).

COLLINS-THOMPSON, Kevyn (2014). « Computational assessment of text readability : A survey of current and future research ». In : *International Journal of Applied Linguistics* 165.2. John Benjamins, p. 97–135 (cf. p. 15).

COLLOBERT, Ronan et WESTON, Jason (2008). « A Unified Architecture for Natural Language Processing : Deep Neural Networks with Multitask Learning ». In : *Proceedings of the 25th International Conference on Machine Learning*. New York, NY, USA, p. 160–167 (cf. p. 34).

COLLOBERT, Ronan, WESTON, Jason, BOTTOU, Léon, KARLEN, Michael, KAVUKCUOGLU, Koray et KUKSA, Pavel (2011). « Natural Language Processing (Almost) from Scratch ». In : *Journal of Machine Learning Research* 12. JMLR.org, p. 2493–2537 (cf. p. 33).

CONSEIL DE L'EUROPE (2001). *Cadre européen commun de référence pour les langues : apprendre, enseigner, évaluer*. Paris, France : Hatier (cf. p. 123).

CORDEIRO, Silvio, RAMISCH, Carlos et VILLAVICENCIO, Aline (2016). « Nominal Compound Compositionality : A Multilingual Lexicon and Predictive Model ». In : *Proceedings of the 7th PARSEME General Meeting*. Dubrovnik, Croatie (cf. p. 112).

CORLEY, Courtney et MIHALCEA, Rada (2005). « Measuring the Semantic Similarity of Texts ». In : *Proceedings of the* ACL *Workshop on Empirical Modeling of Semantic Equivalence and Entailment*. EMSEE '05. Ann Arbor, Michigan, p. 13–18 (cf. p. 50, 53).

COWIE, Jim, GUTHRIE, Joe et GUTHRIE, Louise (1992). « Lexical Disambiguation Using Simulated Annealing ». In : *Proceedings of the 14th Conference on Computational Linguistics*. COLING '92. Nantes, France, p. 359–365 (cf. p. 40).

CUADROS, Montse et RIGAU, German (2006). « Quality Assessment of Large Scale Knowledge Resources ». In : *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*. EMNLP '06. Stroudsburg, PA, USA, p. 534–541 (cf. p. 39).

CUNNINGHAM, Anne E. et STANOVICH, Keith E. (1998). « What reading does for the mind ». In : *Am Educator* 22, p. 8–15 (cf. p. 15).

CURRAN, James R. (2003). « From distributional to semantic similarity ». Thèse de doct. Université d'Édimbourg (cf. p. 53).

DAHER, Hani, BESANÇON, Romaric, FERRET, Olivier, LE BORGNE, Hervé, DAQUO, Anne–Laure et TAMAAZOUSTI, Youssef (2017). « Désambiguïsation d'entités nommées par apprentissage de modèles d'entités à large échelle ». In : *Conférence en Recherche d'Information et Applications (*CORIA*)* (cf. p. 41).

DEERWESTER, Scott C., DUMAIS, Susan T., LANDAUER, Thomas K., FURNAS, George W. et HARSHMAN, Richard A. (1990). « Indexing by Latent Semantic Analysis ». In : *JASIS* 41.6, p. 391–407 (cf. p. 33).

DENDIEN, Jacques et PIERREL, Jean-Marie (2003). « Le Trésor de la Langue Française Informatisé : un exemple d'informatisation d'un dictionnaire de langue de référence ». In : *Traitement Automatique des Langues*. Sous la dir. d'HERMÈS (cf. p. 27).

DICE, Lee Raymond (1945). « Measures of the Amount of Ecologic Association Between Species ». In : *Ecology*. T. 26. 3, p. 297–302 (cf. p. 54).

ECALLE, Jean, NAVARRO, Marion, LABAT, Hélène, GOMES, Christophe, CROS, Laurent et MAGNAN, Annie (2016). « Concevoir des applications sur tablettes tactiles pour stimuler l'apprentissage de la lecture : avec quelles hypothèses scientifiques ? » In : *Sciences et Technologies de l'Information et de la Communication pour l'Éducation et la Formation (*STICEF*)* 23.2, p. 33–56 (cf. p. 143).

EDMONDS, Philip (2000). « Designing a task for SENSEVAL–2 ». In : *Technical Note*. Université de Brighton, U.K. (cf. p. 36).

EDMONDS, Philip (2002). « SENSEVAL : The Evaluation of Word Sense Disambiguation Systems ». In : *Bulletin d'*ELRA 7.3 (cf. p. 45).

EDMONDS, Philip et KILGARRIFF, Adam (2002). « Introduction to the special issue on evaluating word sense disambiguation systems ». In : *Natural Language Engineering* 8.4, p. 279–291 (cf. p. 42, 46, 125).

EVERT, Stefan (2005). « The statistics of word cooccurrences : word pairs and collocations ». Thèse de doct. Université de Stuttgart, Allemagne (cf. p. 33).

FABRE, Cécile, HATHOUT, Nabil, HO-DAC, Lydia-Mai, MORLANE-HONDÈRE, François, MULLER, Philippe, SAJOUS, Franck, TANGUY, Ludovic et VAN DE CRUYS, Tim (2014). « Présentation de l'atelier SemDis 2014 : sémantique distributionnelle pour la substitution lexicale et l'exploration de corpus spécialisés ». In : *Actes de l'atelier SemDis 2014, 21e Conférence sur le Traitement Automatique des Langues Naturelles*. Marseille, France, p. 196–205 (cf. p. 20, 49, 101, 102, 104).

FAUCONNIER, Jean-Philippe (2015). *French Word Embeddings*. URL : http://fauconnier.github.io (cf. p. 111).

FELLBAUM, Christiane (1998). *WordNet : an electronic lexical database*. Cambridge, MA, USA : MIT Press (cf. p. 27, 28, 30, 68, 77, 122).

FERRAND, Ludovic (1999). « 640 homophones et leurs caractéristiques ». In : *L'Année psychologique*, p. 687–708. URL : https://www.persee.fr/doc/psy_0003-5033_1999_num_99_4_28503 (cf. p. 20).

FERRET, Olivier (2014a). « Compounds and Distributional Thesauri ». In : *Proceedings of the Ninth International Conference on Language Resources and Evaluation*. LREC '14. Reykjavik, Islande : European Language Resources Association (ELRA), p. 2979–2984 (cf. p. 27).

FERRET, Olivier (2014b). « Utiliser un modèle neuronal générique pour la substitution lexicale ». In : *Actes de l'atelier SemDis 2014, 21e Conférence sur le*

Traitement Automatique des Langues Naturelles. Marseille, France, p. 218–227 (cf. p. 101, 102, 104).

FINKELSTEIN, Lev, GABRILOVICH, Evgeniy, MATIAS, Yossi, RIVLIN, Ehud, SOLAN, Zach, WOLFMAN, Gadi et RUPPIN, Eytan (2001). « Placing Search in Context : The Concept Revisited ». In : *Proceedings of the 10th International Conference on World Wide Web*. WWW '01. Hong Kong, Chine, p. 406–414 (cf. p. 62–65).

FIRTH, J. R. (1957). « A synopsis of linguistic theory 1930-55 ». In : *Studies in Linguistic Analysis (special volume of the Philological Society)*. T. 1952–59. Oxford University Press, p. 1–32 (cf. p. 37).

FRANÇOIS, Thomas (2015). « When readability meets computational linguistics : a new paradigm in readability ». In : *Revue française de linguistique appliquée* 20.2. Pub. linguistiques, p. 79–97 (cf. p. 15).

FRANÇOIS, Thomas, BILLAMI, Mokhtar Boumedyen, GALA, Núria et BERNHARD, Delphine (2016). « Bleu, contusion, ecchymose : tri automatique de synonymes en fonction de leur difficulté de lecture et compréhension ». In : *Actes de la 23ème Conférence sur le Traitement Automatique des Langues Naturelles, 31ème Journées d'Études sur la Parole et 18ème Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (*JEP–TALN–RÉCITAL*)*. T. 2. Paris, France, p. 15–28 (cf. p. 124, 134, 135, 142).

FRANÇOIS, Thomas, GALA, Núria, WATRIN, Patrick et FAIRON, Cédrick (2014). « FLELex : a graded lexical resource for French foreign learners ». In : *International conference on Language Resources and Evaluation (*LREC*)*. LREC '14. Reykjavik, Islande (cf. p. 123).

GALA, Núria (2015). « Approches multidisciplinaires pour l'étude du lexique et la création de ressources lexicales nouvelles ». Mémoire d'Habilitation à Diriger les Recherches. Aix-Marseille Université (cf. p. 122).

GALA, Núria, BILLAMI, Mokhtar Boumedyen, FRANÇOIS, Thomas et BERNHARD, Delphine (2015). « Graded lexicons : new resources for educational purposes and much more ». In : *22nd Computer Assisted Language Learning Conference (*EUROCALL–2015*)*. Padoue, Italie, p. 204–209 (cf. p. 124).

GALA, Núria, FRANÇOIS, Thomas, BERNHARD, Delphine et FAIRON, Cédrick (2014). « Un modèle pour prédire la complexité lexicale et graduer les mots ». In : *Actes de* TALN *2014*. Marseille, France (cf. p. 124).

GALA, Núria, FRANÇOIS, Thomas et FAIRON, Cédrick (2013). « Towards a French lexicon with difficulty measures : NLP helping to bridge the gap between traditional dictionaries and specialized lexicons ». In : *E-lexicography in the 21st century : thinking outside the paper*. Tallinn, Estonie (cf. p. 124).

GALA, Núria, FRANÇOIS, Thomas, JAVOUREY-DREVET, Ludivine et ZIEGLER, Johannes (2018). « La simplification de textes, une aide à l'apprentissage de la lecture ». In : *Langue française « L'apprentissage de la lecture en français langue maternelle et seconde »*. Armand Colin (cf. p. 17).

GALE, William A., CHURCH, Kenneth W. et YAROWSKY, David (1992). « One Sense Per Discourse ». In : *Proceedings of the* DARPA *Speech and Natural Language Workshop*. HLT '91. Harriman, New York, USA, p. 233–237 (cf. p. 118, 121).
GELBUKH, Alexander, SIDOROV, Grigori et HAN, San-yong (2003). « Evolutionary Approach to Natural Language Word Sense Disambiguation through Global Coherence Optimization ». In : *World Scientific and Engineering Academy and Society (*WSEAS*) Transactions on Communications* 1.2, p. 11–19 (cf. p. 40).
GONZALEZ-DIOS, Itziar, DIAZ DE ILARRAZA, Arantza et IRUSKIETA, Mikel (2017). « Framework for the Analysis of Simplified Texts Taking Discourse into Account : the Basque Causal Relations as Case Study ». In : *Proceedings of the 6th Workshop Recent Advances in* RST *and Related Formalisms*. Saint-Jacques-de-Compostelle, Espagne, p. 48–57 (cf. p. 17).
GUINAND, Frédéric et LAFOURCADE, Mathieu (2010). « Artificial ants for Natural Language Processing ». In : *Artificial Ants. From Collective Intelligence to Real-life Optimization and Beyond*. Sous la dir. de MONMARCHÉ NICOLAS, GUINAND FRÉDÉRIC ET SIARRY PATRICK. Chap. 20, p. 455–492 (cf. p. 40).
HAJIČ, Jan, CIARAMITA, Massimiliano, JOHANSSON, Richard, KAWAHARA, Daisuke, MARTÍ, Maria Antònia, MÀRQUEZ, Lluís, MEYERS, Adam, NIVRE, Joakim, PADÓ, Sebastian, ŠTĚPÁNEK, Jan, STRAŇÁK, Pavel, SURDEANU, Mihai, XUE, Nianwen et ZHANG, Yi (2009). « The CoNLL-2009 Shared Task : Syntactic and Semantic Dependencies in Multiple Languages ». In : *Proceedings of the Thirteenth Conference on Computational Natural Language Learning : Shared Task*. CoNLL '09. Boulder, Colorado, USA, p. 1–18 (cf. p. 75).
HANOKA, Valérie et SAGOT, Benoît (2012). « Wordnet creation and extension made simple : A multilingual lexicon-based approach using wiki resources ». In : *LREC 2012 : 8th international conference on Language Resources and Evaluation*. Istanbul, Turquie, p. 3473–3478 (cf. p. 28).
HARISPE, Sébastien, RANWEZ, Sylvie, JANAQI, Stefan et MONTMAIN, Jacky (2015). *Semantic Similarity from Natural Language and Ontology Analysis*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers (cf. p. 53, 61).
HARRIS, Zellig (1954). « Distributional structure ». In : *Word* 10.23, p. 146–162 (cf. p. 33, 51).
HEARST, Marti A. (1991). « Noun homograph disambiguation using local context in large text corpora ». In : *Proceedings of the 7th Annual Conference of the University of Waterloo Centre for the New OED and Text Research*. Oxford, Royaume-Uni, p. 1–15 (cf. p. 38).
HERBRICH, Ralf, GRAEPEL, Thore et OBERMAYER, Klaus (2000). « Large margin rank boundaries for ordinal regression ». In : *Advances in neural information processing systems*. Cambridge, MA, USA : MIT Press. Chap. 7, p. 115–132 (cf. p. 134).

HILL, Felix, REICHART, Roi et KORHONEN, Anna (2014). « SimLex–999 : Evaluating Semantic Models with (Genuine) Similarity Estimation ». In : *Computing Research Repository* abs/1408.3456 (cf. p. 63).

HOVY, Eduard, MARCUS, Mitchell, PALMER, Martha, RAMSHAW, Lance et WEISCHEDEL, Ralph (2006). « OntoNotes : The 90% Solution ». In : *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume : Short Papers*. NAACL–Short '06. Stroudsburg, PA, USA, p. 57–60 (cf. p. 46).

HUANG, Eric H., SOCHER, Richard, MANNING, Christopher D. et NG, Andrew Y. (2012). « Improving Word Representations via Global Context and Multiple Word Prototypes ». In : *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*. T. 1. ACL '12. Stroudsburg, PA, USA, p. 873–882 (cf. p. 38, 63).

HUENERFAUTH, Matt, FENG, Lijun et ELHADAD, Noémie (2009). « Comparing Evaluation Techniques for Text Readability Software for Adults with Intellectual Disabilities ». In : *Proceedings of the 11th International* ACM SIGACCESS *Conference on Computers and Accessibility*. ASSETS '09. Pittsburgh, Pennsylvanie, USA, p. 3–10 (cf. p. 15).

IACOBACCI, Ignacio, PILEHVAR, Mohammad Taher et NAVIGLI, Roberto (2016). « Embeddings for Word Sense Disambiguation : An Evaluation Study ». In : *Proceedings of the 54th International Joint Conference on Artificial Intelligence (*ACL*)*. Berlin, Allemagne, p. 897–907 (cf. p. 37).

IDE, Nancy et SUDERMAN, Keith (2004). « The American National Corpus First Release ». In : *Proceedings of the Fourth International Conference on Language Resources and Evaluation*. Sous la dir. de LINO MARIA TERESA, XAVIER MARIA FRANCISCA, FERREIRA FÁTIMA, COSTA RUTE ET RAQUEL SILVA. LREC '04, p. 1681–1684 (cf. p. 31).

IDE, Nancy et VÉRONIS, Jean (1998). « Word Sense Disambiguation : The State of the Art ». In : *Computational Linguistics* 24.1, p. 1–41 (cf. p. 19, 25, 26, 42, 67).

INSALL, Matt, ROWLAND, Todd et WEISSTEIN, Eric W. (2018). « *Embedding* ». In : *From MathWorld–A Wolfram Web Resource*. URL : http://mathworld.wolfram.com/Embedding.html (cf. p. 34).

JACCARD, Paul (1901). « Distribution de la flore alpine dans le bassin des Dranses et dans quelques régions voisines ». In : *Bulletin de la Société Vaudoise des Sciences Naturelles*. T. 37. 140, p. 241–272 (cf. p. 54).

JACQUIN, Christine, DESMONTILS, Emmanuel et MONCEAUX, Laura (2007). « French EUROWORDNET Lexical Database Improvements ». In : *Proceedings of the Eighth International Conference on Computational Linguistics and Intelligent Text Processing*. T. 4394. CICLING '08. Mexico, Mexique, p. 12–22 (cf. p. 28).

JONES, Karen Spärck (1972). « A statistical interpretation of term specificity and its application in retrieval ». In : *Journal of Documentation* 28, p. 11–21 (cf. p. 33).

JOUBARNE, Colette et INKPEN, Diana (2011). « Comparison of Semantic Similarity for Different Languages Using the Google n-gram Corpus and Second-Order Co-occurrence Measures ». In : *Advances in Artificial Intelligence - 24th Canadian Conference on Artificial Intelligence, Canadian AI*, p. 216–221 (cf. p. 50, 62–64, 95, 96, 100).

KAMP, Hans, LENCI, Alessandro et PUSTEJOVSKY, James (2014). « Computational Models of Language Meaning in Context (Dagstuhl Seminar 13462) ». In : *Dagstuhl Reports* 3.11, p. 79–116 (cf. p. 53).

KILGARRIFF, Adam (1997). « What is word sense disambiguation good for ? » In : *The fourth Natural Language Processing Pacific Rim Symposium (*NLPRS–1997*)* cmp-lg/9712008, p. 209–214 (cf. p. 19).

KILGARRIFF, Adam et ROSENZWEIG, J. (2000). « Framework and Results for English SENSEVAL ». In : *Computers and the Humanities* 34. Kluwer Academic Publishers, p. 15–48 (cf. p. 45, 59, 112).

KOEHN, Philipp (2005). « Europarl : A Parallel Corpus for Statistical Machine Translation ». In : *Conference Proceedings : the tenth Machine Translation Summit*. Phuket, Thaïlande : AAMT, p. 79–86 (cf. p. 31, 74, 122).

LAFON, Pierre (1980). « Sur la variabilité de la fréquence des formes dans un corpus ». In : t. 1. Mots, p. 127–165 (cf. p. 33).

LAFOURCADE, Mathieu (2007). « Making people play for Lexical Acquisition with the JeuxDeMots prototype ». In : *SNLP '07 : 7th International Symposium on NLP*. Pattaya, Chonburi, Thaïlande (cf. p. 29, 30, 88, 89, 106, 124).

LAFOURCADE, Mathieu (2011). « Lexique et analyse sémantique de textes - structures, acquisitions, calculs, et jeux de mots ». Mémoire d'Habilitation à Diriger les Recherches. Université Montpellier 2, LIRMM, p. 297 (cf. p. 31, 56, 61, 92, 94).

LAFOURCADE, Mathieu et JOUBERT, Alain (2009). « Similitude entre les sens d'usage d'un terme dans un réseau lexical ». In : *Traitement Automatique des Langues* 50.1. ATALA, p. 179–200 (cf. p. 128).

LANDAUER, Thomas K. et DUMAIS, Susan T. (1997). « A Solution to Plato's Problem : The Latent Semantic Analysis Theory of Acquisition, Induction, and Representation of Knowledge ». In : *Psychological Review* 104.2, p. 211–240 (cf. p. 34, 63).

LAVIE, Alon et DENKOWSKI, Michael J. (2009). « The METEOR Metric for Automatic Evaluation of Machine Translation ». In : *Machine Translation* 23.2–3. Kluwer Academic Publishers, p. 105–115 (cf. p. 50).

LEE, Yoong Keok et NG, Hwee Tou (2002). « An Empirical Evaluation of Knowledge Sources and Learning Algorithms for Word Sense Disambiguation ». In : *Proceedings of the Empirical Methods in Natural Language Processing*. T. 10. EMNLP '02. Philadelphie, Pennsylvanie, USA, p. 41–48 (cf. p. 36).

LESK, Michael (1986). « Automatic Sense Disambiguation Using Machine Readable Dictionaries : How to Tell a Pine Cone from an Ice Cream Cone ». In :

Proceedings of the 5th Annual International Conference on Systems Documentation. SIGDOC '86. Toronto, Ontario, Canada, p. 24–26 (cf. p. 57, 58, 71).

LÉTÉ, Bernard, SPRENGER-CHAROLLES, Liliane et COLÉ, Pascale (2004). « Manulex : A grade-level lexical database from French elementary-school readers ». In : *Behavior Research Methods, Instruments and Computers* 36, p. 156–166 (cf. p. 122, 134, 144).

LI, Hang (2015). *Learning to Rank for Information Retrieval and Natural Language Processing*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers (cf. p. 134).

LIN, Dekang (1998a). « An Information-Theoretic Definition of Similarity ». In : *Proceedings of the Fifteenth International Conference on Machine Learning*. ICML '98. San Francisco, CA, USA : Morgan Kaufmann Publishers Inc., p. 296–304 (cf. p. 40, 54, 55, 62, 69, 70, 75).

LIN, Dekang (1998b). « Automatic Retrieval and Clustering of Similar Words ». In : *Proceedings of the 17th International Conference on Computational Linguistics*. T. 2. COLING '98. Montréal, Québec, Canada, p. 768–774 (cf. p. 33, 54).

LING, Xiao, SINGH, Sameer et WELD, Daniel S. (2015). « Design Challenges for Entity Linking ». In : *Transactions of the Association for Computational Linguistics (*TACL*)* 3, p. 315–328 (cf. p. 41).

LINTEAN, Mihai, MOLDOVAN, Cristian, RUS, Vasile et MCNAMARA, Danielle (2010). « The Role of Local and Global Weighting in Assessing the Semantic Similarity of Texts Using Latent Semantic Analysis ». In : *Proceedinf of the 23rd International Conference of the Florida Artificial Intelligence Research Society*. FLAIRS '2010, p. 235–240 (cf. p. 53).

LUONG, Minh-Thang, SOCHER, Richard et MANNING, Christopher D. (2013). « Better word representations with recursive neural networks for morphology ». In : *Proceedings of the Thirteenth Annual Conference on Natural Language Learning*. CoNLL '13 (cf. p. 63, 64).

MANANDHAR, Suresh, KLAPAFTIS, Ioannis P., DLIGACH, Dmitriy et PRADHAN, Sameer S. (2010). « SemEval-2010 Task 14 : Word Sense Induction & Disambiguation ». In : *Proceedings of the 5th International Workshop on Semantic Evaluation*. SEMEVAL '10. Uppsala, Suède, p. 63–68 (cf. p. 38).

MANNING, Christopher D., RAGHAVAN, Prabhakar et SCHÜTZE, Hinrich (2008). *Introduction to Information Retrieval*. New York, NY, USA : Presses universitaires de Cambridge (cf. p. 34).

MARCO, Antonio Di et NAVIGLI, Roberto (2013). « Clustering and Diversifying Web Search Results with Graph-Based Word Sense Induction ». In : *Computational Linguistics* 39.3, p. 709–754 (cf. p. 38).

MATUSCHEK, Michael et GUREVYCH, Iryna (2013). « Dijkstra-WSA : A Graph-Based Approach to Word Sense Alignment ». In : *Transactions of the Association for Computational Linguistics* 1, p. 151–164 (cf. p. 50).

MCCARTHY, Diana, KOELING, Rob, WEEDS, Julie et CARROLL, John (2004). « Finding Predominant Word Senses in Untagged Text ». In : *Proceedings of the 42Nd*

Annual Meeting on Association for Computational Linguistics. ACL '04. Barcelone, Espagne (cf. p. 40, 69).

McCarthy, Diana, Koeling, Rob, Weeds, Julie et Carroll, John (2007). « Unsupervised Acquisition of Predominant Word Senses ». In : *Computational Linguistics* 33.4, p. 553–590 (cf. p. 45).

McCarthy, Diana et Navigli, Roberto (2009). « The English Lexical Substitution Task ». In : *Language Resources and Evaluation* 43.2, p. 139–159 (cf. p. 20, 25, 49, 100).

Mihalcea, Rada (2006). « Using Wikipedia for Automatic Word Sense Disambiguation ». In : *Proceedings of the Main Conference* NAACL–HLT. T. 2007. Synthesis Lectures on Human Language Technologies, p. 196–203 (cf. p. 27).

Mihalcea, Rada et Chklovski, Timothy (2003). « Open Mind Word Expert : Creating Large Annotated Data Collections with Web Users' Help ». In : *Proceedings of the Fourth International Workshop on Linguistically Interpreted Corpora (*LINC–03*)*. Sous la dir. d'Abeillé Anne, Hansen-Schirra Silvia et Uszkoreit Hans, p. 53–60 (cf. p. 31).

Mihalcea, Rada, Corley, Courtney et Strapparava, Carlo (2006). « Corpus-based and Knowledge-based Measures of Text Semantic Similarity ». In : *Proceedings of the 21st National Conference on Artificial Intelligence*. T. 1. AAAI '06. Boston, Massachusetts, USA : The Association for the Advancement of Artificial Intelligence and The MIT Press, p. 775–780 (cf. p. 52).

Mihalcea, Rada et Edmonds, Philip (2004). *Proceedings of the 3rd International Workshop on the Evaluation of Systems for the Semantic Analysis of Text (*Senseval–3*)*. Barcelone, Espagne (cf. p. 45).

Mihalcea, Rada et Faruque, Ehsanul (2004). « Senselearner : Minimally Supervised Word Sense Disambiguation for All Words in Open Text ». In : *Proceedings of* ACL/SIGLEX Senseval–3. T. 3. Barcelone, Espagne, p. 155–158 (cf. p. 37).

Mikolov, Tomáš (2012). « Statistical Language Models Based on Neural Networks ». Thèse de doct. Université de technologie de Brno (cf. p. 33).

Mikolov, Tomáš, Chen, Kai, Corrado, Greg et Dean, Jeffrey (2013a). « Efficient Estimation of Word Representations in Vector Space ». In : *Proceedings of the International Conference on Learning Representations*, p. 1–12 (cf. p. 32, 34, 54, 70).

Mikolov, Tomáš, Sutskever, Ilya, Chen, Kai, Corrado, Greg S. et Dean, Jeff (2013b). « Distributed Representations of Words and Phrases and their Compositionality ». In : *Advances in Neural Information Processing Systems 26*. Sous la dir. de Burges C. J. C., Bottou L., Welling M., Ghahramani Z. et Weinberger K. Q. Curran Associates, Inc., p. 3111–3119 (cf. p. 35).

Mikolov, Tomáš, V. Le, Quoc et Sutskever, Ilya (2013c). « Exploiting Similarities among Languages for Machine Translation ». In : *Computing Research Repository* (cf. p. 34).

MILLER, George A., BECKWITH, Richard, FELLBAUM, Christiane, GROSS, Derek et MILLER, Katherine (1990). « WordNet : An on-line lexical database ». In : *International Journal of Lexicography* 3, p. 235–244 (cf. p. 44).

MILLER, George A. et CHARLES, Walter G. (1991). « Contextual correlates of semantic similarity ». In : *Language & Cognitive Processes*. T. 6. 1. Psychology Press, p. 1–28 (cf. p. 63, 65).

MILLER, George A., LEACOCK, Claudia, TENGI, Randee et BUNKER, Ross T. (1993). « A Semantic Concordance ». In : *Proceedings of the Workshop on Human Language Technology*. HLT '93. Princeton, New Jersey, p. 303–308 (cf. p. 31, 36, 44, 72).

MNIH, Andriy et HINTON, Geoffrey E (2009). « A Scalable Hierarchical Distributed Language Model ». In : *Advances in Neural Information Processing Systems 21*. Sous la dir. de KOLLER D., SCHUURMANS D., BENGIO Y. ET BOTTOU L. Curran Associates, Inc., p. 1081–1088 (cf. p. 33).

MOHAMMAD, Saif, GUREVYCH, Iryna, HIRST, Graeme et ZESCH, Torsten (2007). « Cross-lingual Distributional Profiles of Concepts for Measuring Semantic Distance ». In : *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. EMNLP–CoNLL '7, p. 571–580 (cf. p. 62, 63).

MOHAMMAD, Saif et HIRST, Graeme (2012). « Distributional Measures as Proxies for Semantic Relatedness ». In : *Computing Research Repository* abs/1203.1889 (cf. p. 51, 53).

MORO, Andrea et NAVIGLI, Roberto (2015). « SEMEVAL–2015 Task 13 : Multilingual All-Words Sense Disambiguation and Entity Linking ». In : *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), in the 2015 Conference of the North American Chapter of the Association for Computational Linguistics (*NAACL *2015)*. Denver, Colorado, p. 288–297 (cf. p. 45).

MORO, Andrea, RAGANATO, Alessandro et NAVIGLI, Roberto (2014). « Entity Linking meets Word Sense Disambiguation : a Unified Approach ». In : *Transactions of the Association for Computational Linguistics (*TACL*)*. T. 2, p. 231–244 (cf. p. 41, 72).

NANDIEGOU, Marie et REBOUL, Stella (2018). « La simplification lexicale comme outil pour faciliter la lecture des enfants dyslexiques et faibles lecteurs ». Mém.de mast. Faculté de médecine de Marseille, France : Mémoire en vue de l'obtention du certificat de capacité en orthophonie (cf. p. 17, 24, 143, 149).

NASR, Alexis, BÉCHET, Frédéric, REY, Jean-François, FAVRE, Benoît et LE ROUX, Joseph (2011). « MACAON : An NLP Tool Suite for Processing Word Lattices ». In : *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics : Human Language Technologies : Systems Demonstrations*. HLT '11. Portland, Oregon, p. 86–91 (cf. p. 73).

NAVARRO, Emmanuel, SAJOUS, Franck, GAUME, Bruno, PRÉVOT, Laurent, HSIEH, ShuKai, KUO, Ivy, MAGISTRY, Pierre et HUANG, Chu-Ren (2009). « Wiktionary and NLP : Improving synonymy networks ». In : *Proceedings of the 2009 ACL-*

IJCNLP Workshop on The People's Web Meets NLP : Collaboratively Constructed Semantic Resources. Suntec, Singapour, p. 19–27 (cf. p. 27).

NAVIGLI, Roberto (2009). « Word Sense Disambiguation : A Survey ». In : *ACM Computing Surveys* 41.2, p. 1–69 (cf. p. 19, 25, 26, 31, 50, 60, 67, 70, 106, 125, 136).

NAVIGLI, Roberto, JURGENS, David et VANNELLA, Daniele (2013). « SEMEVAL–2013 Task 12 : Multilingual Word Sense Disambiguation ». In : *Proceedings of the 7^{th} International Workshop on Semantic Evaluation (SemEval 2013), in conjunction with the Second Joint Conference on Lexical and Computational Semantics (*SEM 2013)*. Atlanta, USA, p. 222–231 (cf. p. 45, 107, 113, 114, 149).

NAVIGLI, Roberto, LITKOWSKI, Kenneth C. et HARGRAVES, Orin (2007). « SEMEVAL–2007 Task 07 : Coarse-Grained English All-Words Task ». In : *Proceedings of the 4^{th} International Workshop on Semantic Evaluations (SemEval-2007)*. Prague, République tchèque, p. 30–35 (cf. p. 45).

NAVIGLI, Roberto et PONZETTO, Simone Paolo (2012). « BabelNet : The Automatic Construction, Evaluation and Application of a Wide-coverage Multilingual Semantic Network ». In : t. 193. Essex, UK : Elsevier Science Publishers Ltd., p. 217–250 (cf. p. 28, 30, 68, 77, 106, 122).

NEW, Boris, BRYSBAERT, Marc, VÉRONIS, Jean et PALLIER, Christophe (2007). « The use of film subtitles to estimate word frequencies ». In : *Applied Psycholinguistics* 28.04, p. 661–677 (cf. p. 134, 138).

NG, Hwee Tou (1997). « Getting Serious about Word Sense Disambiguation ». In : *Proceedings of the* ACL SIGLEX *Workshop on Tagging Text with Lexical Semantics : Why, What, and How ? (Washington D.C.)* P. 1–7 (cf. p. 36).

NG, Hwee Tou et LEE, Hian Beng (1996). « Integrating Multiple Knowledge Sources to Disambiguate Word Sense : An Exemplar-based Approach ». In : *Proceedings of the 34th Annual Meeting on Association for Computational Linguistics*. ACL '96. Santa Cruz, Californie, p. 40–47 (cf. p. 36, 38).

OTEGI, Arantxa, ARREGI, Xabier, ANSA, Olatz et AGIRRE, Eneko (2015). « Using knowledge-based relatedness for information retrieval ». In : *Knowledge and Information Systems* 44.3, p. 689–718 (cf. p. 50).

PAETZOLD, Gustavo H. et SPECIA, Lucia (2017). « A survey on lexical simplification ». In : *Journal of Artificial Intelligence Research* 60, p. 549–593 (cf. p. 17).

PALMER, Martha, NG, Hwee Tou et DANG, Hoa Trang (2007). « Evaluation of WSD Systems ». In : *Word Sense Disambiguation : Algorithms and Applications*. Sous la dir. d'AGIRRE ENEKO ET EDMONDS PHILIP. T. 33. Text, Speech, and Language Technology. Springer. Chap. 4, p. 75–106 (cf. p. 42, 43).

PANCHENKO, Alexander (2013). « Similarity Measures for Semantic Relation Extraction ». Thèse de doct. Université catholique de Louvain (cf. p. 52, 53).

PANCHENKO, Alexander et MOROZOVA, Olga (2012). « A Study of Hybrid Similarity Measures for Semantic Relation Extraction ». In : *Proceedings of the Work-*

shop on Innovative Hybrid Approaches to the Processing of Textual Data, p. 10–18 (cf. p. 61, 62).

PASINI, Tommaso et CAMACHO-COLLADOS, José (2018). « A Short Survey on Sense-Annotated Corpora for Diverse Languages and Resources ». In : *https://arxiv.org/abs/1802.04744* (cf. p. 36).

PEDERSEN, Ted, BANERJEE, Satanjeev et PATWARDHAN, Siddharth (2003). « Maximizing Semantic Relatedness to Perform Word Sense Disambiguation ». In : Université du Minnesota, USA (cf. p. 39).

PEDERSEN, Ted, PAKHOMOV, Serguei V. S., PATWARDHAN, Siddharth et CHUTE, Christopher G. (2007). « Measures of semantic similarity and relatedness in the biomedical domain ». In : *Biomedical informatics* 40.3, p. 288–299 (cf. p. 63, 64).

PENNINGTON, Jeffrey, SOCHER, Richard et MANNING, Christopher D. (2014). « Glove : Global Vectors for Word Representation ». In : *Proceedings of the 2014 EMNLP*. Doha, Qatar, p. 1532–1543 (cf. p. 32, 35, 95).

PIERREL, Jean-Marie (2000). *Ingénierie des Langues*. Traité IC2 (Information, communication et commande). Hermes (cf. p. 25).

PILEHVAR, Mohammad Taher, JURGENS, David et NAVIGLI, Roberto (2013). « Align, Disambiguate and Walk : A Unified Approach for Measuring Semantic Similarity ». In : *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics,* ACL. T. 1. Sofia, Bulgarie, p. 1341–1351 (cf. p. 54, 55, 93, 126).

PILEHVAR, Mohammad Taher et NAVIGLI, Roberto (2014). « A Large-scale Pseudoword based Evaluation Framework for State-of-the-Art Word Sense Disambiguation ». In : *Computational Linguistics* 40.4. MIT Press, p. 837–881 (cf. p. 36).

PLOUX, Sabine et VICTORRI, Bernard (1998). « Construction d'espaces sémantiques à l'aide de dictionnaires de synonymes ». In : *Traitement Automatique des Langues* 39.1, p. 161–182 (cf. p. 104).

POLGUÈRE, Alain (2002). *Notions de base en lexicologie*. Montréal, Québec, Canada : Observatoire de Linguistique Sens-Texte, Université de Montréal (cf. p. 22, 123).

PONZETTO, Simone Paolo et NAVIGLI, Roberto (2009). « Large-Scale Taxonomy Mapping for Restructuring and Integrating Wikipedia ». In : *Proceedings of the* 21^{st} *International Joint Conference on Artificial Intelligence*. Pasadena, Californie, p. 2083–2088 (cf. p. 39).

PONZETTO, Simone Paolo et NAVIGLI, Roberto (2010). « Knowledge-rich Word Sense Disambiguation Rivaling Supervised System ». In : *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Uppsala, Suède, p. 1522–1531 (cf. p. 39, 60).

RAMAGE, Daniel, RAFFERTY, Anna N. et MANNING, Christopher D. (2009). « Random Walks for Text Semantic Similarity ». In : *Proceedings of the 2009 Work-*

shop on Graph-based Methods for Natural Language Processing. TextGraphs '4. Suntec, Singapour, p. 23–31 (cf. p. 53).

REDDY, Siva, MCCARTHY, Diana et MANANDHAR, Suresh (2011). « An Empirical Study on Compositionality in Compound Nouns ». In : *Proceedings of the 5th International Joint Conference on Natural Language Processing*. IJCNLP–11. Chiang Mai, Thailand (cf. p. 112).

REISINGER, Joseph et MOONEY, Raymond J. (2010). « Multi-prototype Vector-space Models of Word Meaning ». In : *Human Language Technologies : The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. HLT '10. Stroudsburg, PA, USA, p. 109–117 (cf. p. 38).

RELLO, Luz, BAEZA-YATES, Ricardo, DEMPERE-MARCO, Laura et SAGGION, Horacio (2013). « Frequent Words Improve Readability and Short Words Improve Understandability for People with Dyslexia ». In : *Human-Computer Interaction – INTERACT*. T. 8120. Lecture Notes in Computer Science. Le Cap, Afrique du Sud : Springer, p. 203–219 (cf. p. 16).

RELLO, Luz, SAGGION, Horacio et BAEZA-YATES, Ricardo (2014). « Keyword Highlighting Improves Comprehension for People with Dyslexia ». In : *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations (*PITR*)*. Göteborg, Suède, p. 30–37 (cf. p. 15).

RESNIK, Philip (1995). « Using Information Content to Evaluate Semantic Similarity in a Taxonomy ». In : *Proceedings of the 14th International Joint Conference on Artificial Intelligence*. T. 1. IJCAI '95. Montréal, Québec, Canada : Morgan Kaufmann Publishers Inc., p. 448–453 (cf. p. 61).

RICHARDSON, Stephen D., DOLAN, William B. et VANDERWENDE, Lucy (1998). « MINDNET : acquiring and structuring semantic information from text ». In : *Proceedings of the 17th International Conference on Computational Linguistics*. COLING '17. Montréal, Canada, p. 1098–1102 (cf. p. 20).

ROTHE, Sascha et SCHÜTZE, Hinrich (2015). « AutoExtend : Extending Word Embeddings to Embeddings for Synsets and Lexemes ». In : *In Proceedings of the 53rd* ACL. T. 1. Pékin, Chine, p. 1793–1803 (cf. p. 37).

RUBENSTEIN, Herbert et GOODENOUGH, John B. (1965). « Contextual Correlates of Synonymy ». In : *Communications of the ACM* 8.10, p. 627–633 (cf. p. 62–64, 95).

RUMELHART, David E., HINTON, Geoffrey E. et WILLIAMS, Ronald J. (1988). « Neurocomputing : Foundations of Research ». In : Cambridge, MA, USA : MIT Press. Chap. Learning Representations by Back-propagating Errors, p. 696–699 (cf. p. 33).

SAGOT, Benoît et FIŠER, Darja (2008). « Building a free French wordnet from multilingual resources ». In : *Ontolex 2008*. Marrakech, Maroc (cf. p. 28, 30, 122).

SAHLGREN, Magnus (2008). « The distributional hypothesis ». In : *Italian Journal of Linguistics* 20.1, p. 33–54 (cf. p. 34, 53).

SALEHI, Bahar, COOK, Paul et BALDWIN, Timothy (2015). « A Word Embedding Approach to Predicting the Compositionality of Multiword Expressions ». In : HLT–NAACL. Sous la dir. de MIHALCEA RADA, CHAI JOYCE YUE ET SARKAR ANOOP, p. 977–983 (cf. p. 112).
SALTON, Gerard, WONG, Andrew et YANG, Chung Shu (1975). « A Vector Space Model for Automatic Indexing ». In : *Communications of the ACM* 18.11, p. 613–620 (cf. p. 33).
SCHNABEL, Tobias, LABUTOV, Igor, MIMNO, David M. et JOACHIMS, Thorsten (2015). « Evaluation methods for unsupervised word embeddings ». In : *EMNLP*. Sous la dir. de MÀRQUEZ LLUÍS, CALLISON-BURCH CHRIS, SU JIAN, PIGHIN DANIELE ET MARTON YUVAL, p. 298–307 (cf. p. 35).
SCHÜTZE, Hinrich et PEDERSEN, Jan O. (1995). « Information Retrieval Based on Word Senses ». In : *Proceedings of the Fourth Annual Symposium on Document Analysis and Information Retrieval*. SDAIR '04, p. 161–175 (cf. p. 19, 25).
SCHWAB, Didier, GOULIAN, Jérôme et GUILLAUME, Nathan (2011). « Désambiguïsation lexicale par propagation de mesures sémantiques locales par algorithmes à colonies de fourmis ». In : *Traitement Automatique des Langues Naturelles (*TALN*)*. Montpellier, France (cf. p. 40).
SCHWARTZ, Hansen Andrew et GOMEZ, Fernando (2011). « Evaluating Semantic Metrics on Tasks of Concept Similarity ». In : *the Twenty–Fourth International Florida Artificial Intelligence Research Society Conference*, p. 299–304 (cf. p. 63, 64).
SEBASTIANI, Fabrizio (2002). « Machine Learning in Automated Text Categorization ». In : *ACM Computing Surveys* 34.1, p. 1–47 (cf. p. 34).
SEGOND, Frédérique (2000). « Framework and Results for French ». In : *Computers and the Humanities* 34.1. Kluwer Academic Publishers, p. 49–60 (cf. p. 45, 114).
SERETAN, Violeta (2012). « Acquisition of Syntactic Simplification Rules for French ». In : *Proceedings of the Eight International Conference on Language Resources and Evaluation*. LREC '12. Istanbul, Turquie : European Language Resources Association (ELRA) (cf. p. 17).
SHARDLOW, Matthew (2014). « A Survey of Automated Text Simplification ». In : *International Journal of Advanced Computer Science and Applications (*IJACSA*), Special Issue on Natural Language Processing* 4.1, p. 58–70 (cf. p. 17, 21, 22).
SHEN, Wei, WANG, Jianyong et HAN, Jiawei (2015). « Entity Linking with a Knowledge Base : Issues, Techniques, and Solutions ». In : *Transactions on Knowledge & Data Engineering* 27.2, p. 443–460 (cf. p. 41).
SIDDHARTHAN, Advaith (2014). « A survey of research on text simplification ». In : ITL–*International Journal of Applied Linguistics* 165.2. John Benjamins Publishing Company, p. 259–298 (cf. p. 17).
SITBON, Laurianne et BELLOT, Patrice (2008). « How to cope with questions typed by dyslexic users ». In : *ACM Press, 2nd* ACM SIGIR *workshop on Analy-*

*tics for noisy unstructured text data (*SIGIR *2008)*. T. 303. ACM International Conference Proceeding Series, p. 1–8 (cf. p. 16).

SITBON, Laurianne, BELLOT, Patrice et BLACHE, Philippe (2010). « Vers une recherche d'information adaptée aux utilisateurs dyslexiques ». In : *Document Numérique* 13.1, p. 161–185 (cf. p. 15).

SNYDER, Benjamin et PALMER, Martha (2004). « The English All-Words Task ». In : SENSEVAL–3 *: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*. Sous la dir. de MIHALCEA RADA ET EDMONDS PHILIP. Barcelone, Espagne, p. 41–43 (cf. p. 46).

SOCHER, Richard, PERELYGIN, Alex, WU, Jean, CHUANG, Jason, MANNING, Christopher D., NG, Andrew Y. et POTTS, Christopher (2013). « Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank ». In : *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Stroudsburg, PA, p. 1631–1642 (cf. p. 34).

STETINA, Jiri, KUROHASHI, Sadao et NAGAO, Makoto (1998). « General Word Sense Disambiguation Method Based on a Full Sentential Context ». In : *Usage of WordNet in Natural Language Processing, Proceedings of* COLING–ACL *Workshop*. Montréal, Québec, Canada (cf. p. 37).

SUCHANEK, Fabian M., KASNECI, Gjergji et WEIKUM, Gerhard (2008). « YAGO : A Large Ontology from Wikipedia and WordNet ». In : *Web Semantics : Science, Services and Agents on the World Wide Web* 6.3, p. 203–217 (cf. p. 39).

TAGHIPOUR, Kaveh et NG, Hwee Tou (2015). « Semi-Supervised Word Sense Disambiguation Using Word Embeddings in General and Specific Domains ». In : *Proceedings of the 2015 Annual Conference of the* NAACL. Denver, Colorado, p. 314–323 (cf. p. 37).

TANAKA-ISHII, Kumiko, TEZUKA, Satoshi et TERADA, Hiroshi (2010). « Sorting Texts by Readability ». In : *Computational Linguistics* 36.2, p. 203–227 (cf. p. 134).

TCHECHMEDJIEV, Andon (2012). « État de l'Art : Mesures de Similarité Sémantique Locales et Algorithmes Globaux pour la Désambiguïsation Lexicale à Base de Connaissances ». In : *Proceedings of the Joint Conference* JEP–TALN–RÉCITAL. T. 3. Grenoble, France, p. 295–308 (cf. p. 31, 40).

TURIAN, Joseph, RATINOV, Lev et BENGIO, Yoshua (2010). « Word Representations : A Simple and General Method for Semi-supervised Learning ». In : *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. ACL '10. Stroudsburg, PA, USA, p. 384–394 (cf. p. 33).

TURNEY, Peter D. (2001). « Mining the Web for Synonyms : PMI–IR versus LSA on TOEFL ». In : *Proceedings of the Twelfth European Conference on Machine Learning (*ECML *2001)*. Fribourg-en-Brisgau, Allemagne : Springer Berlin Heidelberg, p. 491–502 (cf. p. 63).

TURNEY, Peter D. et PANTEL, Patrick (2010). « From Frequency to Meaning : Vector Space Models of Semantics ». In : *Journal of Artificial Intelligence Research* 37.1. AI Access Foundation, p. 141–188 (cf. p. 32, 33, 49, 89).

URIELI, Assaf (2013). « Robust French syntax analysis : reconciling statistical methods and linguistic knowledge in the Talismane toolkit ». Thèse de doct. Université de Toulouse II le Mirail, France (cf. p. 102, 138, 148).

VAIVRE-DOURET, Laurence et TURSZ, Anne (1999). « Les troubles de l'apprentissage chez l'enfant : un problème de santé publique ? » In : ADSP : *actualité et dossier en santé publique* 26, p. 23–66. URL : https://www.hcsp.fr/Explore.cgi/Telecharger?NomFichier=ad262366.pdf (cf. p. 15).

VAN DE CRUYS, Tim et APIDIANAKI, Marianna (2011). « Latent Semantic Word Sense Induction and Disambiguation ». In : *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics : Human Language Technologies*. T. 1. HLT '11. Portland, Oregon, p. 1476–1485 (cf. p. 37, 38).

VASILESCU, Florentina, LANGLAIS, Philippe et LAPALME, Guy (2004). « Evaluating Variants of the Lesk Approach for Disambiguating Words ». In : *Proceedings of LREC 2004, the 4th International Conference On Language Resources And Evaluation*. Lisbonne, Portugal, p. 633–636 (cf. p. 39).

VICKREY, David, BIEWALD, Luke, TEYSSIER, Marc et KOLLER, Daphne (2005). « Word-sense Disambiguation for Machine Translation ». In : *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*. HLT '05. Vancouver, Colombie-Britannique, Canada, p. 771–778 (cf. p. 19, 25).

VOORHEES, Ellen M. (1994). « Query Expansion Using Lexical-semantic Relations ». In : *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '94. Dublin, Irlande, p. 61–69 (cf. p. 49).

VOSSEN, Piek (1998). EUROWORDNET : *A Multilingual Database with Lexical Semantic Networks*. Norwell, MA, USA : Kluwer Academic Publishers (cf. p. 28, 30).

WEAVER, Warren (1949). « Translation ». In : *Machine Translation of Languages*. Sous la dir. de LOCKE WILLIAM N. ET BOOTHE A. DONALD. Reprinted from a memorandum written by Weaver in 1949. New York, USA : MIT Press, p. 15–23 (cf. p. 19, 25).

WILKS, Yorick et STEVENSON, Mark (1996). « The Grammar of Sense : Is word-sense tagging much more than part-of-speech tagging ? » In : Université de Sheffield, Royaume-Uni (cf. p. 19, 25).

WU, Zhaohui et GILES, C. Lee (2015). « Sense-aware Semantic Analysis : A Multi-prototype Word Representation Model Using Wikipedia ». In : *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*. AAAI '15. Austin, Texas : The Association for the Advancement of Artificial Intelligence and The MIT Press, p. 2188–2194 (cf. p. 32).

YAROWSKY, David (1993). « One Sense Per Collocation ». In : *Proceedings of the Workshop on Human Language Technology*. HLT '93. Princeton, New Jersey, p. 266–271 (cf. p. 40).

ZESCH, Torsten, MULLER, Christof et GUREVYCH, Iryna (2008). « Using Wiktionary for Computing Semantic Relatedness ». In : *Proceedings of the 23rd National Conference on Artificial Intelligence*. T. 2. AAAI '08. Chicago, Illinois : The Association for the Advancement of Artificial Intelligence and The MIT Press, p. 861–866 (cf. p. 32).

ZHONG, Zhi et NG, Hwee Tou (2010). « It Makes Sense : A Wide-coverage Word Sense Disambiguation System for Free Text ». In : *Proceedings of the* ACL *2010 System Demonstrations*. Uppsala, Sweden, p. 78–83 (cf. p. 37).

ZHONG, Zhi et NG, Hwee Tou (2012). « Word Sense Disambiguation Improves Information Retrieval ». In : *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*. T. 1. ACL '12. L'île de Jeju, Corée du Sud, p. 273–282 (cf. p. 19, 25).

ZIEGLER, Johannes, PERRY, Conrad, MA-WYATT, Anna, LADNER, Diana et SCHULTE-KÖRNE, Gerd (2003). « Developmental dyslexia in different languages : Language-specific or universal ? » In : *Journal of experimental child psychology* 86.3, p. 169–193 (cf. p. 15).

ZIEGLER, Johannes, PERRY, Conrad et ZORZI, Marco (2014). « Modeling reading development through phonological decoding and self-teaching : Implications for dyslexia ». In : *Philosophical Transactions of the Royal Society B* (cf. p. 15).

ZIPF, George K. (1949). *Human Behaviour and the Principle of Least Effort*. Addison-Wesley (Reading MA) (cf. p. 60).

ZORZI, Marco, BARBIERO, Chiara, FACOETTI, Andrea, LONCIARI, Isabella, CARROZZI, Marco, MONTICO, Marcella, BRAVAR, Laura, GEORGE, Florence, PECH-GEORGEL, Catherine et ZIEGLER, Johannes (2012). « Extra-large letter spacing improves reading in dyslexia ». In : *Proceedings of the National Academy of Sciences of the United States of America (*PNAS*)* (cf. p. 15).

ANNEXES

A. Liste de référence RG–65 pour le français

Mot$_A$	**Mot**$_B$	**Sc**$_H$	**Sc**$_{r\_syn}$(**r=0.89**)	**Sc**$_{r\_idée\_associée}$(**r=0.82**)	**Sc**$_{r\_traits\_avec\_coeff}$(**r= 0.81**)	**Sc**$_{r\_traits\_égaux}$(**r= 0.85**)
autographe	*rivage*	0.0	0.0	0.0	0.0	0.0
automobile	*sorcier*	0.0	0.0	0.0022	0.0	$9.10E-4$
corde	*sourire*	0.0	0.0	0.0	0.0	0.0
grimace	*instrument*	0.0	0.0	0.0	0.0	0.0
midi	*ficelle*	0.0	0.0	0.0	0.0	0.0
refuge	*fruit*	0.0	0.03	0.0020	0.0098	0.0069
automobile	*coussin*	0.06	0.0	0.01	0.03	0.02
coq	*périple*	0.06	0.0	0.0026	$4.38E-4$	0.0010
monticule	*four*	0.06	0.0	0.0	0.0	0.0
oiseau	*bois*	0.06	0.0	0.02	0.0032	0.0077
verre	*magicien*	0.06	0.0	$9.55E-4$	0.0011	0.0044
cimetière	*bois*	0.11	0.0	0.0020	0.01	0.0058
fruit	*fournaise*	0.11	0.0	0.0	0.0	0.0
grimace	*gars*	0.11	0.0	0.03	$7.84E-4$	0.0048
coussin	*bijou*	0.17	0.0	0.0020	$2.04E-4$	$9.21E-4$
forêt	*cimetière*	0.17	0.0	0.0010	0.04	0.01
moine	*esclave*	0.17	0.0	0.02	0.0077	0.01
monticule	*rivage*	0.17	0.0	0.0	0.0	0.0
cimetière	*asylum*	0.22	–	–	–	–
cimetière	*monticule*	0.22	0.0	0.0	0.0	0.0
côte	*forêt*	0.22	0.0	0.02	0.04	0.03
refuge	*moine*	0.22	0.0	0.0	0.0	0.0
grue	*coq*	0.28	0.0	0.35	0.69	0.62
rivage	*trip*	0.28	0.0	0.0	0.0	0.0
garçon	*sage*	0.29	0.0	0.04	0.08	0.06
auto	*voyage*	0.33	0.0	0.04	0.03	0.03
rivage	*bois*	0.33	0.0	0.0076	0.01	0.0097
moine	*oracle*	0.39	0.0	0.01	0.01	0.01
colline	*bois*	0.44	0.0	0.01	0.01	0.01
garçon	*coq*	0.44	0.03	0.13	0.19	0.18
gars	*sorcier*	0.44	0.0	0.07	0.10	0.08
refuge	*cimetière*	0.5	0.0	0.01	0.16	0.079
fournaise	*instrument*	0.56	0.0	0.0	0.0	0.0
magicien	*oracle*	0.56	0.10	0.1	0.03	0.06
verre	*bijou*	0.56	0.0	0.01	0.0015	0.0058
nourriture	*coq*	0.61	0.0	0.03	0.03	0.03
sage	*sorcier*	0.83	0.05	0.08	0.07	0.08
grue	*instrument*	0.94	0.0	0.0	0.0012	$6.91E-5$
oracle	*sage*	1.28	0.0	0.0079	0.02	0.01
grimace	*sourire*	1.5	0.03	0.12	0.08	0.09
oiseau	*grue*	1.65	0.0	1.0	1.0	1.0
serf	*esclave*	1.89	0.76	0.62	0.77	0.75
frère	*gars*	2.0	0.0	0.07	0.21	0.16
côte	*colline*	2.17	0.61	0.16	0.21	0.24
midi	*dîner*	2.17	0.0	0.40	0.15	0.32
oiseau	*coq*	2.41	0.40	0.58	1.0	1.0
côte	*rivage*	2.5	1.0	0.96	0.96	0.97
voyage	*périple*	2.59	1.0	0.99	1.0	1.0
magicien	*sorcier*	2.67	0.98	0.59	0.61	0.67
fournaise	*four*	2.78	0.69	0.59	0.57	0.69
nourriture	*fruit*	2.78	0.0	0.19	0.50	0.36
frère	*moine*	2.89	1.0	0.29	0.58	0.55
colline	*monticule*	2.94	1.0	0.36	0.54	0.54
coussin	*oreiller*	3.0	0.89	1.0	1.0	1.0
instrument	*outil*	3.0	0.95	0.51	1.0	1.0
joyau	*bijou*	3.22	1.0	0.30	0.28	0.39
refuge	*asile*	3.28	1.0	0.65	1.0	1.0
corde	*ficelle*	3.33	1.0	0.95	0.37	0.69
verre	*goblet*	3.39	–	–	–	–
autographe	*signature*	3.56	1.0	1.0	1.0	1.0
forêt	*bois*	3.72	0.96	1.0	1.0	1.0
garçon	*gars*	3.83	0.98	0.37	0.46	0.50
automobile	*auto*	3.94	1.0	1.0	1.0	1.0

Table .2. – Paires de mots de la liste de référence RG–65 pour le français avec les scores d'évaluation

Scores d'évaluation

– Application de la troisième fonction d'activation (Act_3) pour le calcul des scores de similarité sémantique entre deux mots de chaque paire de la liste RG–65 (*cf.* tableau .2). Utilisation de la base lexicale de JeuxDeMots datant de Janvier 2017.

– **Sc**$_H$: score des évaluateurs humains.

– **Sc**$_R$: score du système par utilisation des signatures sémantiques de mots à base des relations appartenant à l'ensemble R.

– **r_syn :** relation de *Synonyme*.

– **r_idée_associée :** relation d'*Idée associée*.

– **r_traits_avec_coeff :** groupe de relations « *traits avec coefficient* ».

– **r_traits_égaux :** groupe de relations « *traits égaux* ».

Combinaison des relations

– **r_traits_avec_coeff** $\rightarrow$ $R =$ {"*Domaine*" : 6, "*Acception*" : 5, "*Hyperonyme*" : 4, "*Hyponyme*" : 4, "*Synonyme*" : 3, "*Agent*" : 2, "*Patient*" : 2, "*Idée associée*" : 1}.

– **r_traits_égaux** $\rightarrow$ $R =$ {"*Domaine*" : 1, "*Acception*" : 1, "*Hyperonyme*" : 1, "*Hyponyme*" : 1, "*Synonyme*" : 1, "*Agent*" : 1, "*Patient*" : 1, "*Idée associée*" : 1}.

Corrélation de Pearson (r)

– **Sc**$_{r\_syn} \rightarrow r = 0.89$.

– **Sc**$_{r\_traits\_égaux} \rightarrow r = 0.85$.

– **Sc**$_{\textbf{Sc}_{r\_idée\_associée}} \rightarrow r = 0.82$.

– **Sc**$_{r\_traits\_avec\_coeff} \rightarrow r = 0.81$.

B. Liste de synonymes évalués sur le niveau de difficulté par des jugements humains

associer	combiner	assimiler	entrêmeler	amalgamer
bleu	azur	céruléen		
bleu	fromage			
bleu	contusion	ecchymose		
bleu	bizut	débutant	béjaune	
brûler	cramer	incendier	cautériser	incinérer
intellectuel	cérébral			
chic	élégant	huppé	aristocrate	
agent	gendarme	connétable	agent de police	
conte	fable	allégorie	apologue	histoire
conte	narration			
proximité	voisinage	contiguïté		
noble	généreux	galant	héroïque	chevaleresque
dépouiller	apercevoir	constater	déceler	analyser
voler	piquer	dépouiller	dérober	
inventer	forger	formuler		
murmure	bruissement	gazouillis	gazouillement	
pardon	grâce	amnistie	droit de grâce	grâce présidentielle
injure	affront	insulte		
mine	galerie	gissement	excavation	creusement
mine	puits	charbonnage	huillère	
mine	plomb	mine de crayon		
mine	gueule	galibot		
mine	mine antichar	mine antipersonnel		
air	mine	manière	présence	comportement
parfois	tantôt	quelquefois	occasionnellement	
mémoire	rappel	réminiscence		
rappel	descente en rappel			
rougir	empourprer	cramoisir		
fin	spirituel	mental		
merveilleux	fantastique	fabuleux	formidable	splendide
maigre	osseux	squelettique		
sévère	rigoureux	strict	austère	
gémir	rugir	vagir		
crier	hurler	brailler	beugler	vociférer
rugir	ronfler	bourdonner	vrombir	

Table .3. – Liste de synonymes utilisés dans la campagne d'annotation mettant des jugements humains à la relative difficulté de synonymes

C. Corpus de simplification lexicale

Ci-dessous, la grammaire DTD [14] utilisée pour la validation du document XML représentant le corpus de simplification lexicale.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!ELEMENT corpus (text+)>
<!ATTLIST corpus lang CDATA #REQUIRED>

<!ELEMENT text (sentence+)>
<!ATTLIST text id ID #REQUIRED>
<!ATTLIST text source CDATA #REQUIRED>

<!ELEMENT sentence (#PCDATA | instance | wf)*>
<!ATTLIST sentence id ID #REQUIRED>

<!ELEMENT wf (#PCDATA)>
<!ATTLIST wf lemma CDATA #REQUIRED>
<!ATTLIST wf pos CDATA #REQUIRED>

<!ELEMENT instance (#PCDATA)>
<!ATTLIST instance id ID #REQUIRED>
<!ATTLIST instance lemma CDATA #REQUIRED>
<!ATTLIST instance pos CDATA #REQUIRED>
```

Ci-dessous, un exemple du document IREST_6 avec ses deux versions : originale et simplifiée. Le document possède 24 instances (termes ayant été remplacés). Le tableau .4 décrit l'ensemble des instances et leur substitut (une correspondance vers les raffinements sémantiques proposés par JEUXDEMOTS, pour les termes polysémiques, est présentée).

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE corpus SYSTEM "Corpus_SimpLex.dtd">
<corpus lang="fr">
<!-- Exemple de texte original -->
<text id="d006O" source="irest_orig_6.txt">
  <sentence id="d006.s001">
    <wf lemma="le" pos="DET">Le</wf>
    <wf lemma="castor" pos="NC">castor</wf>
    <wf lemma="être" pos="V">est</wf>
    <wf lemma="un" pos="DET">un</wf>
    <instance id="d006.s001.t003" lemma="excellent" pos="ADJ">
       excellent</instance>
    <wf lemma="nageur" pos="NC">nageur</wf>
    <wf lemma="." pos="PONCT">.</wf>
  </sentence>
  <sentence id="d006.s002">
    <wf lemma="dans" pos="P">Dans</wf>
```

14. **D**ocument **T**ype **D**efinition : une grammaire pour vérifier la conformité d'un document XML (fichier de la grammaire : "Corpus_SimpLex.dtd").

ID Instance	*Source* Token	*Source* POS	*Source* Lemme	*Cible* Token	*Cible* POS	*Cible* Lemme	Sens (JeuxDeMots)
d006.s001.t003	*excellent*	ADJ	*excellent*	*très bon*	ADJ	*très bon*	*excellent (formidable)*
d006.s002.t006	*dix*	DET	*dix*	10	DET	10	_
d006.s003.t004	*grâce à*	P	*grâce à*	*par*	P	*par*	_
d006.s003.t009	*à*	P	*à*	*par*	P	*par*	_
d006.s003.t010	*épaisse*	ADJ	*épais*	*grosse*	ADJ	*gros*	*épais (dimension)*
d006.s004.t002	*volumineux*	ADJ	*volumineux*	*gros*	ADJ	*gros*	_
d006.s004.t007	*vingt*	DET	*vingt*	20	DET	20	_
d006.s005.t004	*abattre*	VINF	*abattre*	*couper*	VINF	*couper*	*abattre (couper un arbre)*
d006.s005.t009	*expert*	NC	*expert*	*habile*	ADJ	*habile*	*expert (expérimenté)*
d006.s005.t010	*construction*	NC	*construction*	*construire*	VINF	*construire*	*construction (construire)*
d006.s005.t011	*de*	P	*de*	*des*	DET	*de*	_
d006.s006.t002	*abat*	V	*abattre*	*coupe*	V	*couper*	*abattre (couper un arbre)*
d006.s006.t004	*ronge*	V	*ronger*	*fait*	V	*faire*	*ronger (grignoter)*
d006.s006.t005	*entaille*	NC	*entaille*	*découpe*	NC	*découpe*	_
d006.s006.t007	*de sorte que*	CS	*de la sorte*	*ainsi*	ADV	*ainsi*	_
d006.s006.t009	*supérieure*	ADJ	*supérieur*	*haut*	NC	*haut*	*supérieur (position)*
d006.s006.t010	*inférieure*	ADJ	*inférieur*	*bas*	NC	*bas*	*inférieur (plus bas)*
d006.s006.t014	*reliées*	VPP	*relier*	*liées*	VPP	*lier*	*relier (lier)*
d006.s006.t015	*surface*	NC	*surface*	*partie*	NC	*partie*	*surface (partie extérieure)*
d006.s007.t001	*la*	DET	*le*	*le*	DET	*le*	*le (Déterminant)*
d006.s007.t002	*connexion*	NC	*connexion*	*lien*	NC	*lien*	*connexion (rapport)*
d006.s007.t004	*étroite*	ADJ	*étroit*	*fin*	ADJ	*fin*	*étroit (restreint)*
d006.s007.t006	*accomplit*	V	*accomplir*	*fait*	V	*faire*	_
d006.s008.t006	*empilées*	VPP	*empiler*	*rangées*	VPP	*ranger*	*empiler (amasser)*

Table .4. – Instances de termes pour le texte IReST_6 « *Le castor* »

```
17    <wf lemma="le" pos="DET">l'</wf>
18    <wf lemma="eau" pos="NC">eau</wf>
19    <wf lemma="," pos="PONCT">,</wf>
20    <wf lemma="il" pos="CLS">il</wf>
21    <wf lemma="pouvoir" pos="V">peut</wf>
22    <wf lemma="nager" pos="VINF">nager</wf>
23    <wf lemma="à" pos="P">à</wf>
24    <wf lemma="un" pos="DET">une</wf>
25    <wf lemma="vitesse" pos="NC">vitesse</wf>
26    <wf lemma="atteindre" pos="VPR">atteignant</wf>
27    <instance id="d006.s002.t006" lemma="dix" pos="DET">dix</instance>
28    <wf lemma="kilomètre" pos="NC">kilomètres</wf>
29    <wf lemma="heure" pos="NC">heure</wf>
30    <wf lemma="." pos="PONCT">.</wf>
31  </sentence>
32  <sentence id="d006.s003">
33    <wf lemma="il" pos="CLS">Il</wf>
34    <wf lemma="être" pos="V">est</wf>
35    <wf lemma="protéger" pos="VPP">protégé</wf>
36    <wf lemma="de" pos="P+D">du</wf>
```

```
    <wf lemma="froid" pos="NC">froid</wf>
    <instance id="d006.s003.t004" lemma="grâce_à" pos="P">grâce_à</
        instance>
    <wf lemma="sa" pos="DET">sa</wf>
    <wf lemma="fourrure" pos="NC">fourrure</wf>
    <wf lemma="faire" pos="VPP">faite</wf>
    <wf lemma="de" pos="P">de</wf>
    <wf lemma="millier" pos="NC">milliers</wf>
    <wf lemma="de" pos="P">de</wf>
    <wf lemma="poil" pos="NC">poils</wf>
    <wf lemma="et" pos="CC">et</wf>
    <instance id="d006.s003.t009" lemma="à" pos="P">à</instance>
    <wf lemma="un" pos="DET">une</wf>
    <instance id="d006.s003.t010" lemma="épais" pos="ADJ">épaisse</
        instance>
    <wf lemma="couche" pos="NC">couche</wf>
    <wf lemma="de" pos="P">de</wf>
    <wf lemma="graisse" pos="NC">graisse</wf>
    <wf lemma="." pos="PONCT">.</wf>
  </sentence>
  <sentence id="d006.s004">
    <wf lemma="ses" pos="DET">Ses</wf>
    <wf lemma="poumon" pos="NC">poumons</wf>
    <instance id="d006.s004.t002" lemma="volumineux" pos="ADJ">
        volumineux</instance>
    <wf lemma="lui" pos="CLO">lui</wf>
    <wf lemma="permettre" pos="V">permettent</wf>
    <wf lemma="de" pos="P">de</wf>
    <wf lemma="rester" pos="VINF">rester</wf>
    <wf lemma="sous" pos="P">sous</wf>
    <wf lemma="le" pos="DET">l'</wf>
    <wf lemma="eau" pos="NC">eau</wf>
    <wf lemma="pendant" pos="P">pendant</wf>
    <wf lemma="facilement" pos="ADV">facilement</wf>
    <instance id="d006.s004.t007" lemma="vingt" pos="DET">vingt</
        instance>
    <wf lemma="minute" pos="NC">minutes</wf>
    <wf lemma="." pos="PONCT">.</wf>
  </sentence>
  <sentence id="d006.s005">
    <wf lemma="le" pos="DET">Le</wf>
    <wf lemma="castor" pos="NC">castor</wf>
    <wf lemma="pouvoir" pos="V">peut</wf>
    <wf lemma="non_seulement" pos="ADV">non_seulement</wf>
    <instance id="d006.s005.t004" lemma="abattre" pos="VINF">abattre</
        instance>
    <wf lemma="adroitement" pos="ADV">adroitement</wf>
    <wf lemma="de" pos="DET">des</wf>
    <wf lemma="arbre" pos="NC">arbres</wf>
    <wf lemma="," pos="PONCT">,</wf>
    <wf lemma="mais" pos="CC">mais</wf>
    <wf lemma="il" pos="CLS">il</wf>
```

```
84    <wf lemma="être" pos="V">est</wf>
85    <wf lemma="aussi" pos="ADV">aussi</wf>
86    <wf lemma="un" pos="DET">un</wf>
87    <instance id="d006.s005.t009" lemma="expert" pos="NC">expert</
          instance>
88    <wf lemma="pour" pos="P">pour</wf>
89    <wf lemma="le" pos="DET">la</wf>
90    <instance id="d006.s005.t010" lemma="construction" pos="NC">
          construction</instance>
91    <instance id="d006.s005.t011" lemma="de" pos="P">de</instance>
92    <wf lemma="barrage" pos="NC">barrages</wf>
93    <wf lemma="." pos="PONCT">.</wf>
94  </sentence>
95  <sentence id="d006.s006">
96    <wf lemma="quand" pos="CS">Quand</wf>
97    <wf lemma="le" pos="DET">le</wf>
98    <wf lemma="castor" pos="NC">castor</wf>
99    <instance id="d006.s006.t002" lemma="abattre" pos="V">abat</
          instance>
100   <wf lemma="un" pos="DET">un</wf>
101   <wf lemma="arbre" pos="NC">arbre</wf>
102   <wf lemma="," pos="PONCT">,</wf>
103   <wf lemma="il" pos="CLS">il</wf>
104   <instance id="d006.s006.t004" lemma="ronger" pos="V">ronge</
          instance>
105   <wf lemma="un" pos="DET">une</wf>
106   <instance id="d006.s006.t005" lemma="entaille" pos="NC">entaille</
          instance>
107   <wf lemma="dans" pos="P">dans</wf>
108   <wf lemma="le" pos="DET">le</wf>
109   <wf lemma="tronc" pos="NC">tronc</wf>
110   <wf lemma="," pos="PONCT">,</wf>
111   <instance id="d006.s006.t007" lemma="de_la_sorte" pos="CS">
          de_sorte_que</instance>
112   <wf lemma="le" pos="DET">les</wf>
113   <wf lemma="partie" pos="NC">parties</wf>
114   <instance id="d006.s006.t009" lemma="supérieur" pos="ADJ">supé
          rieure</instance>
115   <wf lemma="et" pos="CC">et</wf>
116   <instance id="d006.s006.t010" lemma="inférieur" pos="ADJ">infé
          rieure</instance>
117   <wf lemma="ne" pos="ADV">ne</wf>
118   <wf lemma="être" pos="V">sont</wf>
119   <wf lemma="plus" pos="ADV">plus</wf>
120   <instance id="d006.s006.t014" lemma="relier" pos="VPP">reliées</
          instance>
121   <wf lemma="que" pos="CS">que</wf>
122   <wf lemma="par" pos="P">par</wf>
123   <wf lemma="un" pos="DET">une</wf>
124   <instance id="d006.s006.t015" lemma="surface" pos="NC">surface</
          instance>
125   <wf lemma="très" pos="ADV">très</wf>
```

```
126     <wf lemma="fin" pos="ADJ">fine</wf>
127     <wf lemma="." pos="PONCT">.</wf>
128   </sentence>
129   <sentence id="d006.s007">
130     <wf lemma="quand" pos="CS">Quand</wf>
131     <instance id="d006.s007.t001" lemma="le" pos="DET">la</instance>
132     <instance id="d006.s007.t002" lemma="connexion" pos="NC">connexion
          </instance>
133     <wf lemma="être" pos="V">est</wf>
134     <instance id="d006.s007.t004" lemma="étroit" pos="ADJ">étroite</
          instance>
135     <wf lemma="," pos="PONCT">,</wf>
136     <wf lemma="le" pos="DET">le</wf>
137     <wf lemma="vent" pos="NC">vent</wf>
138     <instance id="d006.s007.t006" lemma="accomplir" pos="V">accomplit<
          /instance>
139     <wf lemma="le" pos="DET">le</wf>
140     <wf lemma="reste" pos="NC">reste</wf>
141     <wf lemma="." pos="PONCT">.</wf>
142   </sentence>
143   <sentence id="d006.s008">
144     <wf lemma="le" pos="DET">Les</wf>
145     <wf lemma="petit" pos="ADJ">petites</wf>
146     <wf lemma="branche" pos="NC">branches</wf>
147     <wf lemma="être" pos="V">sont</wf>
148     <wf lemma="couper" pos="VPP">coupées</wf>
149     <wf lemma="par" pos="P">par</wf>
150     <wf lemma="le" pos="DET">le</wf>
151     <wf lemma="castor" pos="NC">castor</wf>
152     <wf lemma="et" pos="CC">et</wf>
153     <instance id="d006.s008.t006" lemma="empiler" pos="VPP">empilées</
          instance>
154     <wf lemma="comme" pos="P">comme</wf>
155     <wf lemma="réserve" pos="NC">réserve</wf>
156     <wf lemma="." pos="PONCT">.</wf>
157   </sentence>
158   <sentence id="d006.s009">
159     <wf lemma="le" pos="DET">Les</wf>
160     <wf lemma="gros" pos="ADJ">grosses</wf>
161     <wf lemma="branche" pos="NC">branches</wf>
162     <wf lemma="être" pos="V">sont</wf>
163     <wf lemma="séparer" pos="VPP">séparées</wf>
164     <wf lemma="et" pos="CC">et</wf>
165     <wf lemma="utiliser" pos="VPP">utilisées</wf>
166     <wf lemma="comme" pos="P">comme</wf>
167     <wf lemma="bois" pos="NC">bois</wf>
168     <wf lemma="pour" pos="P">pour</wf>
169     <wf lemma="le" pos="DET">la</wf>
170     <wf lemma="construction" pos="NC">construction</wf>
171     <wf lemma="de" pos="P">de</wf>
172     <wf lemma="barrage" pos="NC">barrages</wf>
173     <wf lemma="." pos="PONCT">.</wf>
```

```
  </sentence>
</text>
<!-- Exemple de texte simplifié -->
<text id="d006S" source="irest_simp_6.txt">
  <sentence id="d006.s001">
    <wf lemma="le" pos="DET">Le</wf>
    <wf lemma="castor" pos="NC">castor</wf>
    <wf lemma="être" pos="V">est</wf>
    <wf lemma="un" pos="DET">un</wf>
    <instance id="d006.s001.t003" lemma="très_bon" pos="ADJ">très_bon<
        /instance>
    <wf pos="NC">nageur</wf>
    <wf lemma="." pos="PONCT">.</wf>
  </sentence>
  <sentence id="d006.s002">
    <wf lemma="dans" pos="P">Dans</wf>
    <wf lemma="le" pos="DET">l'</wf>
    <wf lemma="eau" pos="NC">eau</wf>
    <wf lemma="," pos="PONCT">,</wf>
    <wf lemma="il" pos="CLS">il</wf>
    <wf lemma="pouvoir" pos="V">peut</wf>
    <wf lemma="nager" pos="VINF">nager</wf>
    <wf lemma="à" pos="P">à</wf>
    <wf lemma="une" pos="DET">une</wf>
    <wf lemma="vitesse" pos="NC">vitesse</wf>
    <wf lemma="de" pos="P">de</wf>
    <instance id="d006.s002.t006" lemma="10" pos="DET">10</instance>
    <wf lemma="kilomètre" pos="NC">kilomètres</wf>
    <wf lemma="heure" pos="NC">heure</wf>
    <wf lemma="." pos="PONCT">.</wf>
  </sentence>
  <sentence id="d006.s003">
    <wf lemma="il" pos="CLS">Il</wf>
    <wf lemma="être" pos="V">est</wf>
    <wf lemma="protéger" pos="VPP">protégé</wf>
    <wf lemma="de" pos="P+D">du</wf>
    <wf lemma="froid" pos="NC">froid</wf>
    <instance id="d006.s003.t004" lemma="par" pos="P">par</instance>
    <wf lemma="sa" pos="DET">sa</wf>
    <wf lemma="fourrure" pos="NC">fourrure</wf>
    <wf lemma="faire" pos="VPP">faite</wf>
    <wf lemma="de" pos="P">de</wf>
    <wf lemma="millier" pos="NC">milliers</wf>
    <wf lemma="de" pos="P">de</wf>
    <wf lemma="poil" pos="NC">poils</wf>
    <wf lemma="et" pos="CC">et</wf>
    <instance id="d006.s003.t009" lemma="par" pos="P">par</instance>
    <wf lemma="une" pos="DET">une</wf>
    <instance id="d006.s003.t010" lemma="gros" pos="ADJ">grosse</
        instance>
    <wf lemma="couche" pos="NC">couche</wf>
    <wf lemma="de" pos="P">de</wf>
```

```
224     <wf lemma="graisse" pos="NC">graisse</wf>
225     <wf lemma="." pos="PONCT">.</wf>
226   </sentence>
227   <sentence id="d006.s004">
228     <wf lemma="ses" pos="DET">Ses</wf>
229     <instance id="d006.s004.t002" lemma="gros" pos="ADJ">gros</
          instance>
230     <wf pos="NC">poumons</wf>
231     <wf lemma="lui" pos="CLO">lui</wf>
232     <wf lemma="permettre" pos="V">permettent</wf>
233     <wf lemma="de" pos="P">de</wf>
234     <wf lemma="rester" pos="VINF">rester</wf>
235     <wf lemma="sous" pos="P">sous</wf>
236     <wf lemma="le" pos="DET">l'</wf>
237     <wf lemma="eau" pos="NC">eau</wf>
238     <wf lemma="pendant" pos="P">pendant</wf>
239     <instance id="d006.s004.t007" lemma="20" pos="DET">20</instance>
240     <wf lemma="minute" pos="NC">minutes</wf>
241     <wf lemma="." pos="PONCT">.</wf>
242   </sentence>
243   <sentence id="d006.s005">
244     <wf lemma="le" pos="DET">Le</wf>
245     <wf lemma="castor" pos="NC">castor</wf>
246     <wf lemma="pouvoir" pos="V">peut</wf>
247     <instance id="d006.s005.t004" lemma="couper" pos="VINF">couper</
          instance>
248     <wf lemma="des" pos="DET">des</wf>
249     <wf lemma="arbre" pos="NC">arbres</wf>
250     <wf lemma="," pos="PONCT">,</wf>
251     <wf lemma="mais" pos="CC">mais</wf>
252     <wf lemma="il" pos="CLS">il</wf>
253     <wf lemma="être" pos="V">est</wf>
254     <wf lemma="aussi" pos="ADV">aussi</wf>
255     <instance id="d006.s005.t009" lemma="habile" pos="ADJ">habile</
          instance>
256     <wf lemma="pour" pos="P">pour</wf>
257     <instance id="d006.s005.t010" lemma="construire" pos="VINF">
          construire</instance>
258     <instance id="d006.s005.t011" lemma="de" pos="DET">des</instance>
259     <wf lemma="barrage" pos="NC">barrages</wf>
260     <wf lemma="." pos="PONCT">.</wf>
261   </sentence>
262   <sentence id="d006.s006">
263     <wf lemma="quand" pos="CS">Quand</wf>
264     <wf lemma="le" pos="DET">le</wf>
265     <wf lemma="castor" pos="NC">castor</wf>
266     <instance id="d006.s006.t002" lemma="couper" pos="V">coupe</
          instance>
267     <wf lemma="un" pos="DET">un</wf>
268     <wf lemma="arbre" pos="NC">arbre</wf>
269     <wf lemma="," pos="PONCT">,</wf>
270     <wf lemma="il" pos="CLS">il</wf>
```

```
271     <instance id="d006.s006.t004" lemma="faire" pos="V">fait</instance
            >
272     <wf lemma="une" pos="DET">une</wf>
273     <instance id="d006.s006.t005" lemma="découpe" pos="NC">découpe</
            instance>
274     <wf lemma="dans" pos="P">dans</wf>
275     <wf lemma="le" pos="DET">le</wf>
276     <wf lemma="tronc" pos="NC">tronc</wf>
277     <wf lemma="," pos="PONCT">,</wf>
278     <instance id="d006.s006.t007" lemma="ainsi" pos="ADV">ainsi</
            instance>
279     <wf lemma="les" pos="DET">les</wf>
280     <wf lemma="partie" pos="NC">parties</wf>
281     <wf lemma="de" pos="P+D">du</wf>
282     <instance id="d006.s006.t009" lemma="haut" pos="NC">haut</instance
            >
283     <wf lemma="et" pos="CC">et</wf>
284     <wf lemma="de" pos="P+D">du</wf>
285     <instance id="d006.s006.t010" lemma="bas" pos="NC">bas</instance>
286     <wf lemma="ne" pos="ADV">ne</wf>
287     <wf lemma="être" pos="V">sont</wf>
288     <wf lemma="plus" pos="ADV">plus</wf>
289     <instance id="d006.s006.t014" lemma="lier" pos="VPP">liées</
            instance>
290     <wf lemma="que" pos="CS">que</wf>
291     <wf lemma="par" pos="P">par</wf>
292     <wf lemma="une" pos="DET">une</wf>
293     <instance id="d006.s006.t015" lemma="partie" pos="NC">partie</
            instance>
294     <wf lemma="très" pos="ADV">très</wf>
295     <wf lemma="fin" pos="ADJ">fine</wf>
296     <wf lemma="." pos="PONCT">.</wf>
297   </sentence>
298   <sentence id="d006.s007">
299     <wf lemma="quand" pos="CS">Quand</wf>
300     <instance id="d006.s007.t001" lemma="le" pos="DET">le</instance>
301     <instance id="d006.s007.t002" lemma="lien" pos="NC">lien</instance
            >
302     <wf lemma="être" pos="V">est</wf>
303     <instance id="d006.s007.t004" lemma="fin" pos="ADJ">fin</instance>
304     <wf lemma="," pos="PONCT">,</wf>
305     <wf lemma="le" pos="DET">le</wf>
306     <wf lemma="vent" pos="NC">vent</wf>
307     <instance id="d006.s007.t006" lemma="faire" pos="V">fait</instance
            >
308     <wf lemma="le" pos="DET">le</wf>
309     <wf lemma="reste" pos="NC">reste</wf>
310     <wf lemma="." pos="PONCT">.</wf>
311   </sentence>
312   <sentence id="d006.s008">
313     <wf lemma="les" pos="DET">Les</wf>
314     <wf lemma="petit" pos="ADJ">petites</wf>
```

```
315     <wf lemma="branche" pos="NC">branches</wf>
316     <wf lemma="être" pos="V">sont</wf>
317     <wf lemma="couper" pos="VPP">coupées</wf>
318     <wf lemma="par" pos="P">par</wf>
319     <wf lemma="le" pos="DET">le</wf>
320     <wf lemma="castor" pos="NC">castor</wf>
321     <wf lemma="et" pos="CC">et</wf>
322     <instance id="d006.s008.t006" lemma="ranger" pos="VPP">rangées</
          instance>
323     <wf lemma="comme" pos="P">comme</wf>
324     <wf lemma="réserve" pos="NC">réserve</wf>
325     <wf lemma="." pos="PONCT">.</wf>
326   </sentence>
327   <sentence id="d006.s009">
328     <wf lemma="les" pos="DET">Les</wf>
329     <wf id="d006.s009.t001" lemma="gros" pos="ADJ">grosses</wf>
330     <wf id="d006.s009.t002" lemma="branche" pos="NC">branches</wf>
331     <wf id="d006.s009.t003" lemma="être" pos="V">sont</wf>
332     <wf id="d006.s009.t004" lemma="séparer" pos="VPP">séparées</wf>
333     <wf lemma="et" pos="CC">et</wf>
334     <wf id="d006.s009.t005" lemma="utiliser" pos="VPP">utilisées</wf>
335     <wf lemma="comme" pos="P">comme</wf>
336     <wf id="d006.s009.t006" lemma="bois" pos="NC">bois</wf>
337     <wf lemma="pour" pos="P">pour</wf>
338     <wf id="d006.s009.t007" lemma="construire" pos="VINF">construire</
          wf>
339     <wf lemma="des" pos="DET">des</wf>
340     <wf id="d006.s009.t008" lemma="barrage" pos="NC">barrages</wf>
341     <wf lemma="." pos="PONCT">.</wf>
342   </sentence>
343 </text><!-- ... Suite des autres textes ... -->
344 </corpus>
```

Corpus/Simp_Lex.xml

D. Application Android : « *Lecture de textes* »

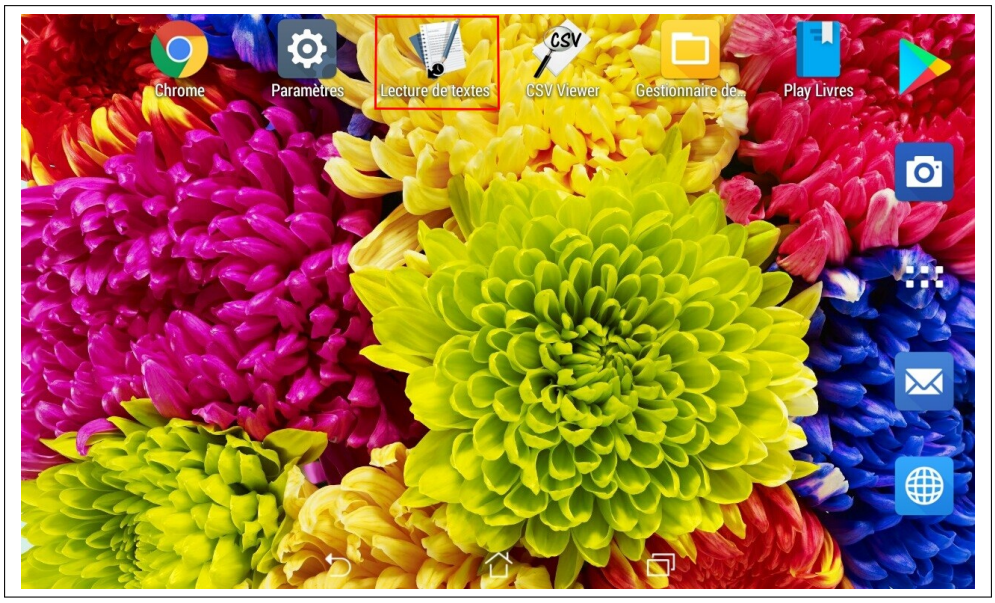


Figure .5. – L'icône de l'exécutable de l'application « *Lecture de textes* »

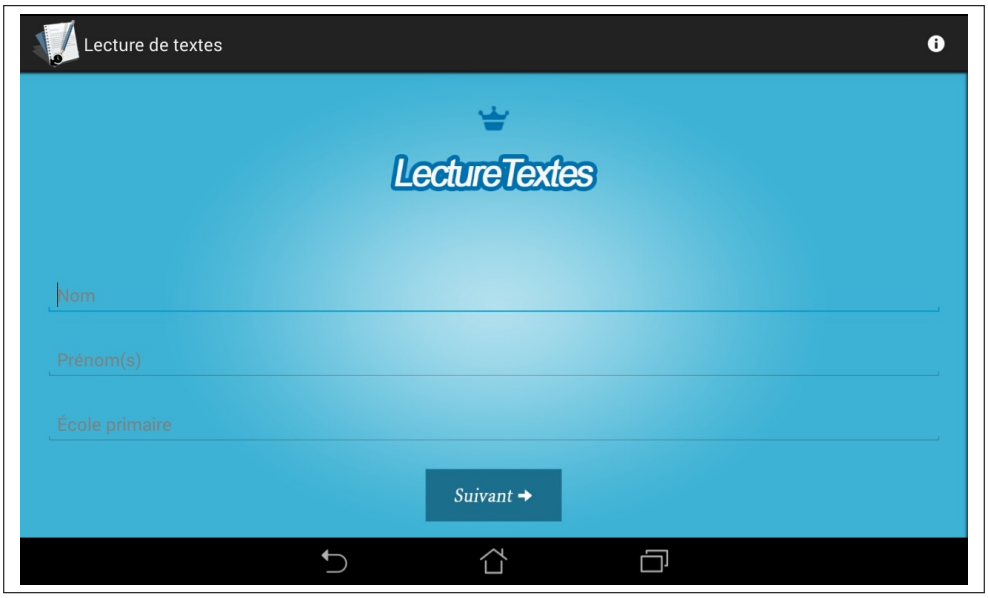


Figure .6. – Première interface de l'application « *Lecture de textes* » – Formulaire que l'enfant utilisateur doit remplir avant de commencer son expérience

Figure .7. – Une petite fenêtre « *À propos* » pour décrire l'application « *Lecture de textes* »

Figure .8. – Consignes à prendre en compte avant de commencer l'expérience « *Lecture de textes* »

Figure .9. – Exemple de deux phrases d'un texte tiré aléatoirement (phrases 1 et 5 du texte) – Lecture phrase par phrase de l'enfant utilisateur

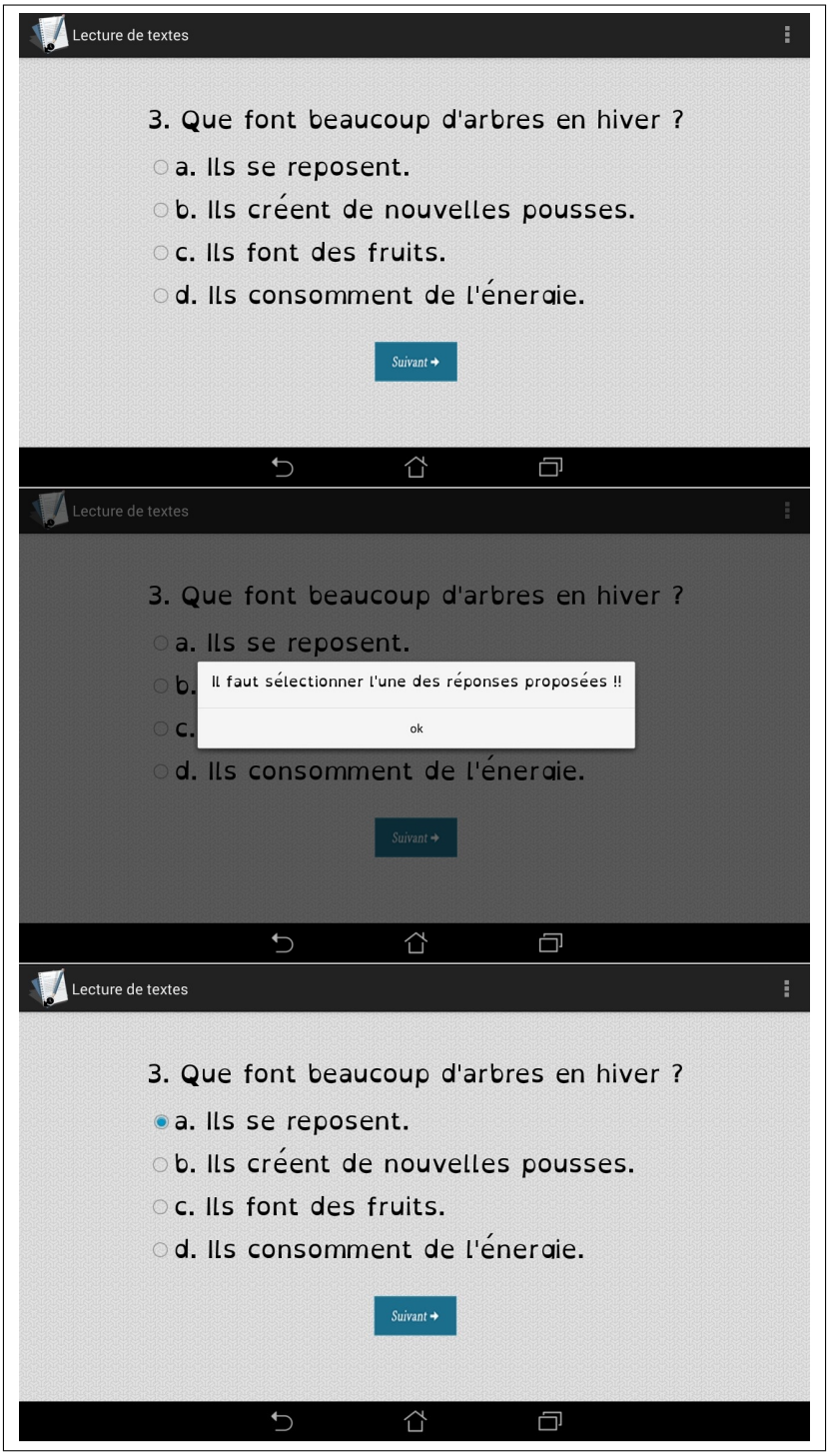


Figure .10. – Exemple d'un test de compréhension – Test sous forme de QCM : une seule réponse est à sélectionner avant de passer à l'étape suivante

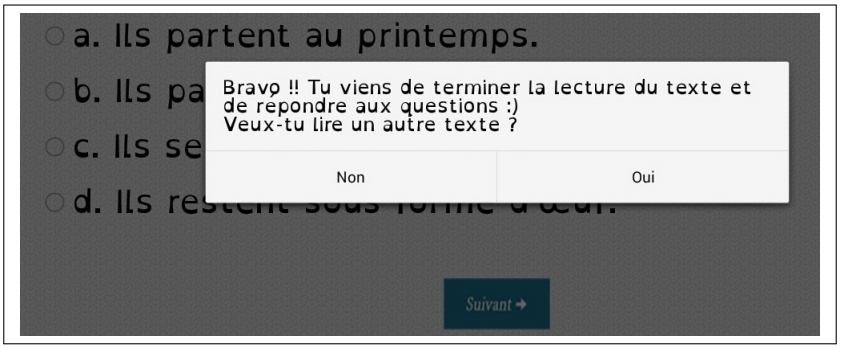


Figure .11. – Proposition de l'application « *Lecture de textes* » : choisir de lire ou non un autre texte

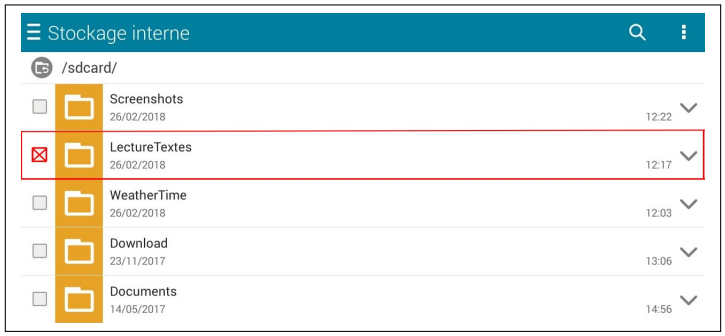


Figure .12. – Sauvegarde des données collectées dans le dossier « *LectureTextes* » créé dans la mémoire interne des tablettes utilisées pour les expériences

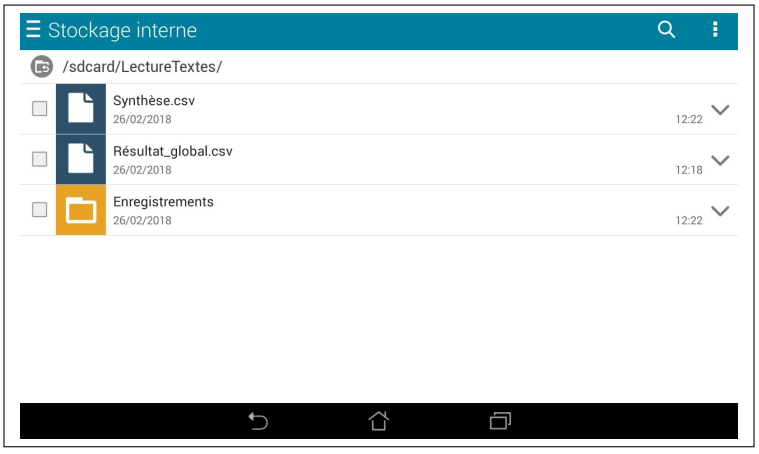
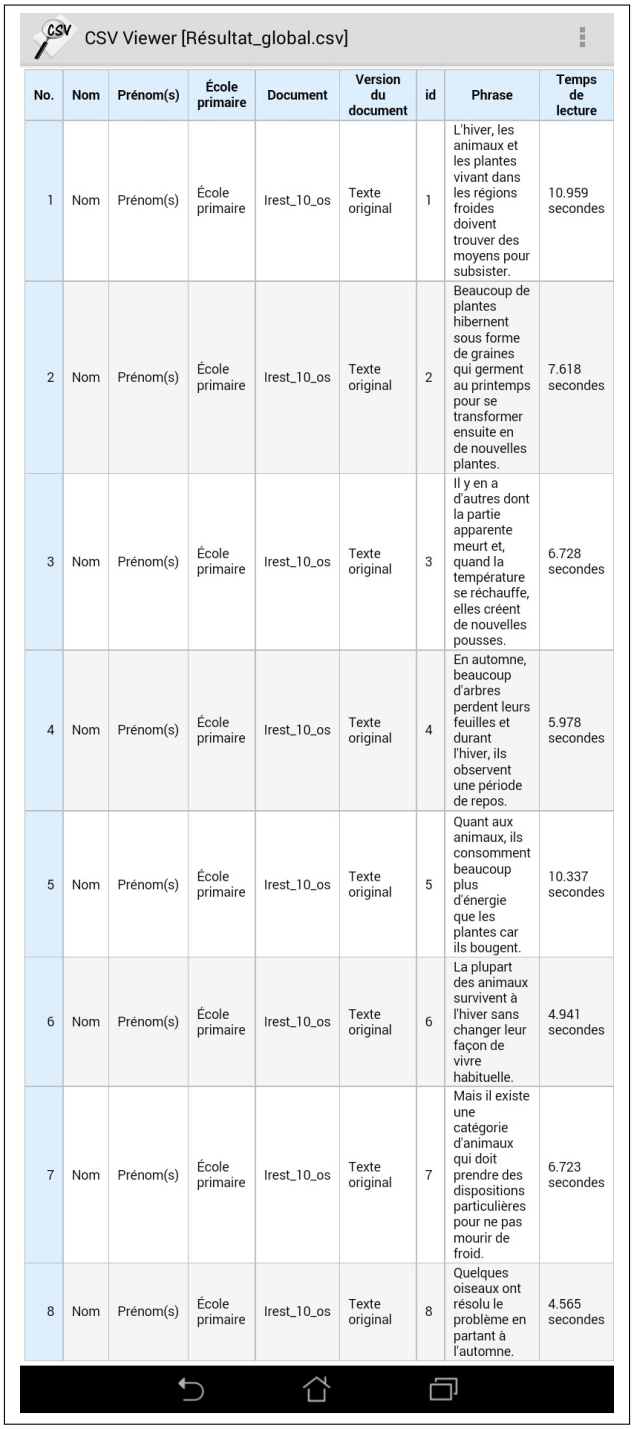


Figure .13. – Contenu du dossier « *LectureTextes* » : fichiers textuels en format CSV et enregistrements vocaux



CSV Viewer [Résultat_global.csv]

No.	Nom	Prénom(s)	École primaire	Document	Version du document	id	Phrase	Temps de lecture
1	Nom	Prénom(s)	École primaire	Irest_10_os	Texte original	1	L'hiver, les animaux et les plantes vivant dans les régions froides doivent trouver des moyens pour subsister.	10.959 secondes
2	Nom	Prénom(s)	École primaire	Irest_10_os	Texte original	2	Beaucoup de plantes hibernent sous forme de graines qui germent au printemps pour se transformer ensuite en de nouvelles plantes.	7.618 secondes
3	Nom	Prénom(s)	École primaire	Irest_10_os	Texte original	3	Il y en a d'autres dont la partie apparente meurt et, quand la température se réchauffe, elles créent de nouvelles pousses.	6.728 secondes
4	Nom	Prénom(s)	École primaire	Irest_10_os	Texte original	4	En automne, beaucoup d'arbres perdent leurs feuilles et durant l'hiver, ils observent une période de repos.	5.978 secondes
5	Nom	Prénom(s)	École primaire	Irest_10_os	Texte original	5	Quant aux animaux, ils consomment beaucoup plus d'énergie que les plantes car ils bougent.	10.337 secondes
6	Nom	Prénom(s)	École primaire	Irest_10_os	Texte original	6	La plupart des animaux survivent à l'hiver sans changer leur façon de vivre habituelle.	4.941 secondes
7	Nom	Prénom(s)	École primaire	Irest_10_os	Texte original	7	Mais il existe une catégorie d'animaux qui doit prendre des dispositions particulières pour ne pas mourir de froid.	6.723 secondes
8	Nom	Prénom(s)	École primaire	Irest_10_os	Texte original	8	Quelques oiseaux ont résolu le problème en partant à l'automne.	4.565 secondes

Figure .14. – Création du fichier CSV « *Résultat_global.csv* » contenant le temps de lecture de chaque phrase provenant de chaque texte lu par chaque enfant utilisateur

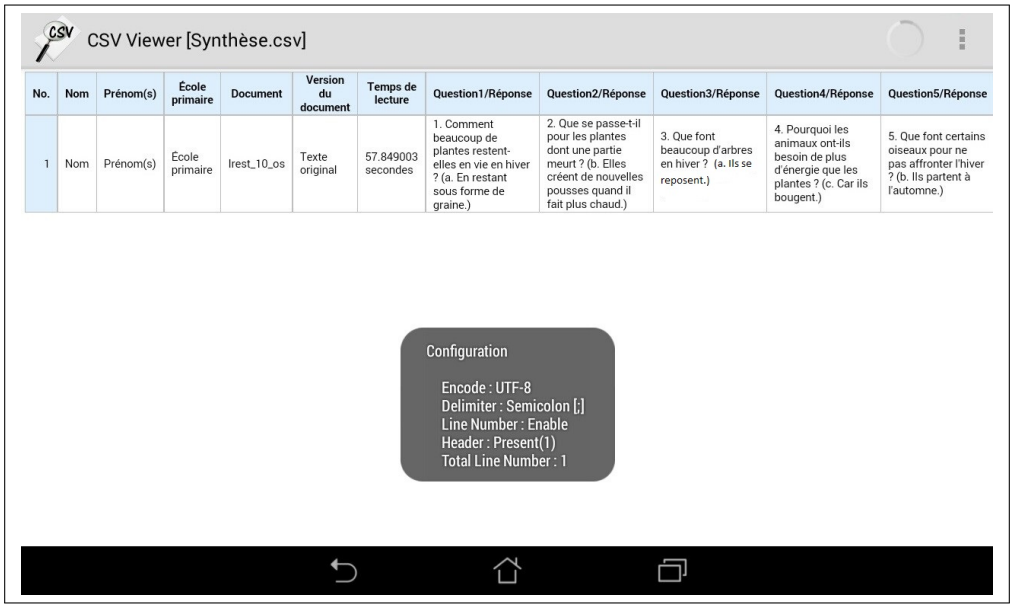


Figure .15. – Création du fichier CSV « *Synthèse.csv* » contenant le temps global de lecture de chaque texte ainsi que les réponses des tests de compréhension par chaque enfant utilisateur

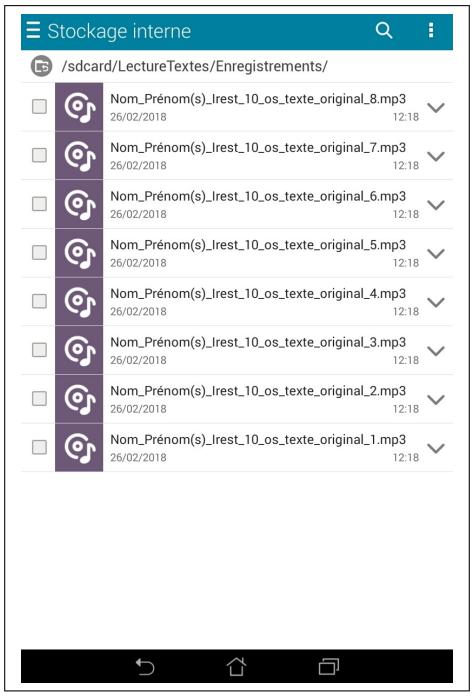


Figure .16. – Enregistrements vocaux suite à la lecture à voix haute des phrases par chaque enfant utilisateur

Mes publications

2018 **Mokhtar Boumedyen BILLAMI**, Thomas FRANÇOIS et Núria GALA

ReSyf : a French lexicon with ranked synonyms.

The 27th International Conference on Computational Linguistics (COLING 2018).

Santa Fe, Nouveau-Mexique, USA.

https://hal.archives-ouvertes.fr/hal-01861652/document

2017 **Mokhtar Boumedyen BILLAMI** et Núria GALA

Création et validation de signatures sémantiques : application à la mesure de similarité sémantique et à la substitution lexicale.

*Actes de la 24ème Conférence sur le Traitement Automatique des Langues Naturelles et 19ème Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (*TALN–RÉCITAL*).*

Orléans, France.

https://hal.archives-ouvertes.fr/hal-01528117/document

2016 Thomas FRANÇOIS, **Mokhtar Boumedyen BILLAMI**, Núria GALA et Delphine BERNHARD

Bleu, contusion, ecchymose : tri automatique de synonymes en fonction de leur difficulté de lecture et compréhension.

*Actes de la 23ème Conférence sur le Traitement Automatique des Langues Naturelles, 31ème Journées d'Études sur la Parole et 18ème Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (*JEP–TALN–RÉCITAL*).*

Paris, France.

https://hal.archives-ouvertes.fr/hal-01346538/document

2016 **Mokhtar Boumedyen BILLAMI** et Núria GALA

Approches d'analyse distributionnelle pour améliorer la désambiguïsation sémantique.

*13ème Journées internationales d'Analyse statistique des Données Textuelles (*JADT*)*.

Nice, France.

https://hal.archives-ouvertes.fr/hal-01477502/document

2015 Núria GALA, **Mokhtar Boumedyen BILLAMI**, Thomas FRANÇOIS et Delphine BERNHARD

Graded lexicons : new resources for educational purposes and much more.

*22nd Computer Assisted Language Learning Conference (*EUROCALL*).*

Padoue, Italie.

2015 **Mokhtar Boumedyen BILLAMI**

Désambiguïsation lexicale à base de connaissances par sélection distributionnelle et traits sémantiques.

*Actes de la 22ème Conférence sur le Traitement Automatique des Langues Naturelles et 17ème Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (*TALN–RÉCITAL*).*

Prix du meilleur article RÉCITAL.

Caen, France.

https://hal.archives-ouvertes.fr/hal-01477463/document