2m 11.3274.10

Université de Montréal

Réduction de dimension pour modèles graphiques probabilistes appliqués à la désambiguïsation sémantique

par
Maryse Boisvert

Département d'informatique et de recherche opérationnelle
Faculté des arts et sciences

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

Faculté des Études Supérieures
Grade octroyé
à compter du
13 JAN 2005
Université de Montréal

août, 2004

© Maryse Boisvert, 2004

QA
76
U54
2005
V. 009

Université de Montréal

Direction des bibliothèques

AVIS

L'auteur a autorisé l'Université de Montréal à reproduire et diffuser, en totalité ou en partie, par quelque moyen que ce soit et sur quelque support que ce soit, et exclusivement à des fins non lucratives d'enseignement et de recherche, des copies de ce mémoire ou de cette thèse.

L'auteur et les coauteurs le cas échéant conservent la propriété du droit d'auteur et des droits moraux qui protègent ce document. Ni la thèse ou le mémoire, ni des extraits substantiels de ce document, ne doivent être imprimés ou autrement reproduits sans l'autorisation de l'auteur.

Afin de se conformer à la Loi canadienne sur la protection des renseignements personnels, quelques formulaires secondaires, coordonnées ou signatures intégrées au texte ont pu être enlevés de ce document. Bien que cela ait pu affecter la pagination, il n'y a aucun contenu manquant.

NOTICE

The author of this thesis or dissertation has granted a nonexclusive license allowing Université de Montréal to reproduce and publish the document, in part or in whole, and in any format, solely for noncommercial educational and research purposes.

The author and co-authors if applicable retain copyright ownership and moral rights in this document. Neither the whole thesis or dissertation, nor substantial extracts from it, may be printed or otherwise reproduced without the author's permission.

In compliance with the Canadian Privacy Act some supporting forms, contact information or signatures may have been removed from the document. While this may affect the document page count, it does not represent any loss of content from the document.

Université de Montréal
Faculté des études supérieures

Ce mémoire intitulé:

Réduction de dimension pour modèles graphiques probabilistes appliqués à la désambiguïsation sémantique

présenté par:

Maryse Boisvert

a été évalué par un jury composé des personnes suivantes:

Jian-Yun Nie
président-rapporteur

Yoshua Bengio
directeur de recherche

Balàzs Kégl
co-directeur

Philippe Langlais
membre du jury

Mémoire accepté le 7 décembre 2004

Sommaire

Un modèle de désambiguïsation sémantique tente d'assigner à une occurrence d'un mot dans un texte l'acception à laquelle il réfère dans ce contexte particulier. Afin de contrer la sous-représentation, ces modèles sont habituellement lissés à l'aide de méthodes de redistribution de la masse de probabilité faisant appel à diverses heuristiques. Nous proposons d'appliquer un algorithme de réduction de la dimension aux matrices de co-occurrence des mots et des sens afin d'extraire un espace thématique implicite aux données. Ceci a l'avantage de permettre la représentation compacte des données dans un espace à la dimensionnalité réduite. Cette idée prend la forme de modèles graphiques dans lesquels des variables cachées représentent des thèmes de plus haut niveau que les mots et les sens. Puisqu'il n'est pas possible d'obtenir des observations pour ces variables, les paramètres du modèle sont appris à l'aide de l'algorithme EM. Une approximation de rang inférieur des matrices de co-occurrence obtenues par la décomposition en valeurs singulières constitue l'estimation initiale des paramètres.

Mots-clés : **désambiguïsation sémantique, décomposition en valeurs singulières, algorithme EM, WordNet, modèles graphiques**.

Abstract

A word sense disambiguation model tries to guess the meaning of a particular word instance given its context. To fight data sparseness, such models are often smoothed with probabibility redistribution schemes which rely on various heuristics. We propose to apply a dimensionality reduction algorithm to word and sense co-occurrence matrices to extract a thematic space which is implied by the data. This yields the advantage of allowing a compact data representation in a space which dimensionality is reduced. This idea is implemented by graphical models where hidden variables correspond to themes of a higher level of abstraction than those of words and senses. Since it is not possible to obtain observations for those variables, the model parameters are learned with the EM algorithm. A lower rank approximation of co-occurrence matrices obtained by singular value decomposition is used to initialize the parameters.

Keywords : **word sense disambiguation**, **singular value decomposition**, **EM algorithm**, **WordNet**, **graphical models**.

Table des matières

Sommaire iii

Abstract iv

Table des matières v

Liste des figures vii

Liste des tableaux vii

1 Introduction 1
- 1.1 Polysémie et désambiguïsation du sens 1
- 1.2 Pourquoi faire de la WSD ? 3
- 1.3 Aperçu du mémoire 4

2 La désambiguïsation sémantique 5
- 2.1 Qu'est-ce que le contexte d'un mot ? 5
- 2.2 Qu'est-ce que le sens d'un mot ? 6
- 2.3 WordNet 7
- 2.4 Désambiguïsation à l'aide de dictionnaires 10
- 2.5 Désambiguïsation par apprentissage statistique 12
  - 2.5.1 Apprentissage supervisé 12
  - 2.5.2 Apprentissage non-supervisé 16
  - 2.5.3 Apprentissage semi-supervisé 18
- 2.6 Comparaison 19

3 Réduction de dimension pour la désambiguïsation sémantique 21
- 3.1 Introduction 21
- 3.2 Modèles graphiques 23

3.3 Réduction de la dimension 25
3.3.1 Décomposition en valeurs singulières 25
3.4 Modèles proposés 27
3.4.1 Modèle Contexte-Concept-Sens (CCS) 29
3.4.2 Modèle Contexte-Sens-Concept-Sens (CSCS) 30
3.5 Estimation des paramètres du modèle 32
3.5.1 L'algorithme EM 33
3.5.2 Lissage 37

4 Résultats **40**
4.1 Corpus et métriques d'évaluation 40
4.2 Résultats 42
4.2.1 Modèles de référence 42
4.2.2 Modèles de désambiguïsation avec concepts implicites . . 44
4.2.3 Test 49
4.3 Analyse des résultats 50
4.3.1 La réduction de la dimension en WSD 50
4.3.2 La classification et la vraisemblance 51
4.3.3 Le manque de ressources 52
4.3.4 Le contexte 53

5 Conclusion **54**
5.1 Synthèse des contributions 55

6 Annexe **57**
6.1 Modèle Naive Bayes 57
6.2 Modèle CCS 60
6.3 Modèle CSCS 63

Références **66**

Liste des figures

3.1 Le modèle Naive Bayes 24
3.2 La variable concept dans le modèle CCS 29
3.3 La variable concept dans le modèle CSCS 31

4.1 Valeurs singulières de la matrice des mots de contexte et des sens . 45
4.2 Valeurs singulières de la matrice des sens des mots de contexte et des sens 48

Liste des tableaux

2.1 La polysémie dans WordNet version 1.7 8
2.2 Une liste de décision pour le mot "bass" 13

4.1 Précision des modèles de référence 43
4.2 Précision du modèle Naive Bayes 43
4.3 Précision du modèle Naive Bayes utilisant la pondération géométrique des mots de contexte 44
4.4 Précision du modèle CCS selon le type d'initialisation 46
4.5 Log-probabilité moyenne du modèle CCS 46
4.6 Précision du modèle CSCS selon le type d'initialisation 47
4.7 Log-probabilité moyenne du modèle CSCS 49
4.8 Comparaison de la précision des modèles 49

6.1 Précision du modèle Naive Bayes 58
6.2 Précision du modèle Naive Bayes utilisant la pondération géométrique des mots de contexte 59
6.3 Précision du modèle CCS avec initialisation aléatoire des paramètres 60
6.4 Précision du modèle CCS avec initialisation des paramètres par la SVD et la translation 60
6.5 Précision du modèle CCS avec initialisation des paramètres par la SVD et la softmax 61
6.6 Log-probabilité moyenne sans la factorisation de la matrice de co-occurrence des mots de contexte et des sens 61
6.7 Log-probabilité moyenne du modèle CCS avec initialisation aléatoire des paramètres 61
6.8 Log-probabilité moyenne du modèle CCS avec initialisation des paramètres par la SVD et la translation 62
6.9 Log-probabilité moyenne du modèle CCS avec initialisation des paramètres par la SVD et la softmax 62

6.10 Précision du modèle CSCS avec initialisation aléatoire des paramètres 63
6.11 Précision du modèle CSCS avec initialisation des paramètres par la SVD et la translation 63
6.12 Précision du modèle CSCS avec initialisation des paramètres par la SVD et la softmax 64
6.13 Log-probabilité moyenne sans la factorisation de la matrice de co-occurrence des sens 64
6.14 Log-probabilité moyenne du modèle CSCS avec initialisation aléatoire des paramètres 64
6.15 Log-probabilité moyenne du modèle CSCS avec initialisation des paramètres par la SVD et la translation 65
6.16 Log-probabilité moyenne du modèle CSCS avec initialisation des paramètres par la SVD et la softmax 65

À mes parents

CHAPITRE 1

INTRODUCTION

L'analyse et la compréhension de la langue naturelle est un des plus grands défis de l'intelligence artificielle et de l'apprentissage automatique. Le moindre progrès dans ce domaine est susceptible de mener à d'importantes améliorations de la performance de certaines applications telles que la reconnaissance de la parole, la traduction automatique et la recherche d'information. En particulier, un problème fondamental auquel font face ces applications est celui de l'apprentissage de la signification des mots et de leur utilisation.

1.1 Polysémie et désambiguïsation du sens

La plupart des mots dans une langue comme l'anglais ou le français peuvent référer à plusieurs idées différentes. Un mot est dit polysémique s'il possède plusieurs acceptions. Son interprétation est donc fonction de son utilisation dans un contexte particulier. Par exemple, le mot féminin "livre" possède deux sens courants ; il peut dénoter l'unité monétaire (livre sterling) ou encore la mesure de masse. Il est important de noter qu'un mot polysémique employé seul ne réfère à aucun concept ; c'est seulement dans un contexte qu'il est possible de lui donner une signification.

Le sens auquel réfère un mot dans une phrase donnée est la plupart du temps évident pour un être humain. Par exemple, si on dit "Ceci coûte dix livres.", on comprend immédiatement que le mot "livre" réfère au concept de monnaie, alors qu'avec "Ceci pèse dix livres.", il se rapporte clairement à l'interprétation massique. Ce processus d'association d'un sens au mot "livre", ne requérant aucun effort conscient de la part d'un être humain, est pourtant parmi les tâches les plus difficiles à automatiser. On remarque que, dans chacune des phrases ci-dessus, un seul mot est pertinent dans le choix d'un sens, soit les verbes "coûter" et "peser", ce qui peut donner l'illusion que des règles simples sont à l'origine du processus de désambiguïsation. La raison pour laquelle le mot "livre" dans la phrase "J'ai besoin de dix livres." se rapporte probablement au concept de monnaie est cependant moins évidente que dans les exemples précédents. En réalité, même si l'on dispose de très peu de connaissances sur le problème de la désambiguïsation, il est évident que des processus complexes et subtils en sont à l'origine.

La désambiguïsation sémantique (généralement dénotée par WSD[1]) est la tâche consistant à assigner à un mot dans un texte l'acception à laquelle il réfère dans ce contexte. Ce problème, simple en apparence, laisse perplexe la communauté de chercheurs depuis les débuts de l'intelligence artificielle et du traitement automatique de la langue naturelle. La première formulation du problème de la désambiguïsation sémantique a eu lieu lors des premières tentatives de traduction automatique dans les années cinquante. À cette époque, la WSD était typiquement impliquée dans la résolution de tâches de plus haut niveau. Le développement de nombreuses bases de données sémantiques et lexicales dans les années quatre-vingt a permis aux chercheurs de s'attaquer directement au problème de la WSD, qui est maintenant un intérêt de recherche de premier plan.

[1] *Word Sense Disambiguation*

1.2 Pourquoi faire de la WSD ?

La performance de plusieurs applications de traitement automatique du langage (NLP[2]) serait plus grande si le sens de chaque mot dans un texte était connu. Par exemple, la connaissance du sens exact d'un mot ambigü dans une langue pourrait certainement réduire le nombre de traductions possibles dans une autre langue. De la même façon, les applications de recherche d'information font face à des problèmes qui relèvent de la WSD, soit la polysémie et la synonymie. En effet, il est souvent possible d'utiliser plus d'un mot pour exprimer la même idée, ce qui fait en sorte que les termes utilisés dans une requête d'un utilisateur peuvent ne pas correspondre aux termes utilisés dans un document pertinent. De plus, étant donné le caractère polysémique des mots, des documents non pertinents peuvent être inclus dans les résultats d'une requête. Si les sens des mots composant une requête étaient connus, un moteur de recherche pourrait plus facilement sélectionner les documents pertinents en réduisant l'espace de recherche.

Il est cependant très difficile d'évaluer présentement l'apport qu'un bon modèle de désambiguïsation aurait sur la performance d'une application de recherche d'information ou de traduction automatique car les systèmes actuels sont loin d'être parfaits. De plus, la communauté semble divisée sur la question de la pertinence de la WSD dans le domaine de la recherche d'information. Sanderson (SANDERSON 1997) affirme que l'amgibuïté sémantique affecte très peu la performance d'un moteur de recherche et qu'un modèle de désambiguïsation ayant un taux de précision inférieur à 90% a même un effet néfaste sur cette dernière. Cette étude contredit des travaux plus récents qui constatent une amélioration de la recherche de l'information dans des textes en incluant une composante de désambiguïsation sémantique (GONZALO, VERDEJO, CHUGUR et CIGARRAN 1998; STOKOE, OAKES et TAIT 2003). Il est intéressant de noter que même la performance humaine n'est pas aussi grande dans le domaine de la WSD que sur d'autres problèmes de classification et que la variabilité des jugements est très grande. Malgré tout, il est clair qu'un bon modèle de désambiguïsation peut mener au raffinement de nombreuses applications de traitement automatique du langage qui en dépendent.

[2]*Natural Language Processing*

1.3 Aperçu du mémoire

Le second chapitre traite plus en profondeur des différentes facettes de la désambiguïsation du sens. Nous y discutons d'abord de notions fondamentales reliées à la WSD, soit celles de contexte et de sens d'un mot. Nous y présentons également la base de données lexicale WordNet, une référence importante dans le développement d'algorithmes de traitement automatique du langage. Un survol des différentes approches en désambiguïsation sémantique est également présenté.

Le troisième chapitre propose de nouveaux modèles de WSD. La conception et l'estimation de ces modèles sont exposées en détail. Nous y discutons également les modèles graphiques, dont le formalisme est utilisé afin d'illustrer les idées, et des algorithmes de réduction de la dimension, en particulier la décomposition en valeurs singulières, sur lesquels reposent les idées que nous développons.

Le quatrième chapitre constitue une étude comparative des modèles proposés. Les corpus et métriques employés afin d'évaluer leur performance y sont présentés. Ce chapitre présente également un résumé des résultats des expériences qui ont été faites dans le cadre de cette étude. Les résultats de toutes les expériences sont rapportées en annexe. Nous comparons finalement la performance de nos modèles à celle de modèles de référence et proposons une analyse critique des résultats.

Le cinquième chapitre conclut en résumant les contributions de notre étude, et en proposant quelques pistes de recherche supplémentaires.

CHAPITRE 2

LA DÉSAMBIGUÏSATION SÉMANTIQUE

Ce chapitre constitue un approfondissement du problème de la désambiguïsation sémantique. Nous y détaillons les notions de contexte et de sens des mots et y traitons également de WordNet, une base de données lexicale utilisée comme référence dans cette étude. Les sections subséquentes présentent un survol des méthodes proposées parmi les plus significatives. Le chapitre se termine avec une brève comparaison de ces méthodes.

2.1 Qu'est-ce que le contexte d'un mot ?

Il est possible d'assigner un sens à un mot sans considérer le contexte l'accompagnant et d'obtenir néanmoins le sens véritable dans beaucoup de cas. Une procédure triviale de désambiguïsation consiste en effet à toujours étiqueter un mot avec son sens le plus fréquent. Si l'on désire une meilleure performance, il est cependant souhaitable de tenter de désambiguïser un mot *en contexte*. Le contexte d'un mot est composé de tout ce qui peut influencer son occurrence dans un texte

ou un discours et regroupe ainsi un ensemble de phénomèmes linguistiques très vaste et difficile à caractériser.

Par exemple, le type de document et le domaine auquel il appartient (un texte scientifique, un article de journal, etc.) induisent une sélection de mots et conditionnent d'une certaine façon leur utilisation ; le vocabulaire et les significations que peuvent prendre les mots sont restreints. D'une manière moins générale, les documents eux-mêmes sont une source d'information contextuelle qui est plus accessible que le domaine. Les mots composant les documents constituent une façon de caractériser le contexte d'un mot ambigü (on parle de "mots de contexte"). Les mots de contexte peuvent être utilisés en fonction de leur relation avec le mot ambigü. Cette relation peut être de différentes natures : les relations syntaxiques, les collocations et les relations de distance en sont quelques exemples. Les mots de contexte peuvent également être pris comme un groupe sans égard pour le rôle qu'ils jouent dans la phrase et pour leur position dans le contexte ; c'est l'hypothèse communément appelée "sac de mots"[1].

En somme, l'information véhiculée par le contexte influence certainement le sens d'un mot polysémique dans un texte, mais son rôle précis dans la désambiguïsation n'est pas encore bien modélisé et reste un problème ouvert.

2.2 Qu'est-ce que le sens d'un mot ?

Un des objectifs des travaux sur la désambiguïsation est de reconnaître les différentes acceptions d'un mot associées à chacune de ses occurrences. Le but n'est donc pas précisément l'étude des sens des mots de façon générale, mais plutôt celle du rattachement de leurs occurences à des étiquettes (WILKS et STEVENSON 1997). En WSD, on considère généralement que le sens d'un mot est une de ses utilisations possibles dans un discours. La différence de signification des mots est donc étroitement liée à la différence des contextes dans lesquels ils sont utilisés.

La démarche de la désambiguïsation sémantique consiste à regrouper les occurrences en fonction de leur contexte. Ces regroupements peuvent ensuite être

[1] *bag of words*

associés à un sens prédéfini. De ce point de vue, les dictionnaires et thésaurus constituent de véritables "répertoires" de sens. On y trouve un catalogue des différentes façons d'employer un mot accompagnées d'explications sur le contexte qui en permet l'emploi. Les dictionnaires et thésauri procurent une caractérisation du sens qui est indépendante des applications en faisant usage. Ceci a l'avantage de permettre la comparaison de modèles utilisant les mêmes références. Cependant, tous les dictionnaires ne recensent pas les mêmes acceptions pour les mots et cette absence d'uniformité est problématique.

L'association d'une étiquette sémantique peut également être effectuée en dehors d'une référence précise comme un dictionnaire. Un sens est alors un regroupement distinct parmi toutes les "manières" d'utiliser un mot dans un texte et ces manières sont extraites à l'aide des textes eux-mêmes. Les sens sont plutôt induits par le contexte et ne sont pas associés à un concept ou une idée tiré d'un ensemble prédéfini que l'utilisation d'un dictionnaire rend possible. Cette caractérisation des sens est aussi valable que celle proposée par des dictionnaires, mais leur emploi dépend entièrement des textes utilisés afin de les extraire.

Une polémique reliée est celle de la *granularité* des sens. Les dictionnaires présentent souvent des distinctions très subtiles entre les sens d'un mot alors qu'en pratique ces nuances sont parfois indistinguables. Il est probable également que des applications différentes exigent des distinctions sémantiques à des niveaux différents. Tant la caractérisation des sens que leur niveau de généralité relèvent des applications qui en font usage.

2.3 WordNet

WordNet (FELLBAUM 1998) est une base de données lexicale pour la langue anglaise développée par des linguistes de l'Université Princeton. Disponible gratuitement et électroniquement, elle est accompagnée d'interfaces pour divers langages de programmation qui en facilitent l'utilisation. Cette base de données est en constante évolution, et des versions pour d'autres langues sont en développement. WordNet est à ce jour parmi les ressources lexicales les plus utilisées dans

Tableau 2.1 – La polysémie dans WordNet version 1.7

	Nombre de formes uniques	Nombre de sens
Noms	94474	116317
Verbes	10319	22066
Adverbes	20170	29881
Adjectifs	4546	5677

les applications de NLP, particulièrement par les modèles dans le domaine de la désambiguïsation sémantique. La base de données combine les fonctions des ressources lexicales classiques. On y retrouve les différentes interprétations d'un mot et une définition pour chacune d'elles, tout comme dans un dictionnaire. Les mots eux-mêmes sont reliés par des *synsets* qui sont un ensemble de synonymes représentant un concept lexical unique, comme dans un thésaurus. WordNet définit des relations entre les mots, comme la synonymie et l'antonymie, ainsi que des relations entre les synsets, comme l'hyponymie et l'hypernymie. Ces notions seront clarifiées plus loin.

WordNet est organisé en quatre catégories de très haut niveau : les noms, les verbes, les adjectifs et les adverbes. Les unités lexicales correspondent à une forme orthographique unique. Pour chacune d'entre elles, on a répertorié un ensemble de sens possibles. Le Tableau 2.1 recense le nombre de formes lexicales uniques et le nombre de sens pour chacune des catégories de Wordnet. Le degré de polysémie est très variable, et on remarque que les verbes sont plus polysémiques que les noms.

Le synset est la composante fondamentale de WordNet. Un synset est un ensemble de mots partageant un même sens et est uniquement identifié par le *gloss*, qui est composé de l'ensemble des mots appartenant à ce synset ainsi que d'une description. Un mot polysémique appartient à plusieurs synsets. Par exemple, on définit le nom "bass" par les synsets suivants :

```
1. bass -- (the lowest part of the musical range)
2. bass, bass part -- (the lowest part in polyphonic music)
3. bass, basso -- (an adult male singer with the lowest voice)
4. sea bass, bass -- (flesh of lean-fleshed saltwater fish of
   the family Serranidae)
```

```
5. freshwater bass, bass -- (any of various North American
   lean-fleshed freshwater fishes especially of the genus
   Micropterus)
6. bass, bass voice, basso -- (the lowest adult male singing
   voice)
7. bass -- (the member with the lowest range of a family of
   musical instruments)
8. bass -- (nontechnical name for any of numerous edible marine
   and freshwater spiny-finned fishes)
```

Chacun de ces synsets se rapporte à une acception particulière du mot "bass" et désigne un concept lexical précis. Ce concept peut être exprimé par n'importe quel mot appartenant à ce synset. Ainsi, dans le contexte où un synset est désigné, les mots le composant deviennent interchangeables. On pourrait, par exemple, remplacer "bass" dans la phrase "the bass sings very well" par "basso" sans modifier la signification de l'énoncé.

Les synsets peuvent également représenter des concepts de plus haut niveau que les mots et leurs sens. Joints à une relation d'ordre sémantique, les synsets définissent une ontologie, une organisation permettant de classifier les éléments d'un univers linguistique. Ces ontologies prennent la forme d'un graphe acyclique non-dirigé comportant des sous-graphes correspondant chacun à une des quatre catégories principales. WordNet définit ainsi une multitude de relations sémantiques permettant d'organiser les mots et leurs sens de façon cohérente. Par exemple, la relation d'hypernymie permet d'associer à un concept les concepts plus généraux dont il est une expression spécifique. Ainsi, le sens d'un hypernyme d'un synset particulier contient entièrement le sens de ce synset. En construisant une ontologie à l'aide de cette relation et des synsets, nous obtenons pour le septième sens du mot "bass" l'arbre de concepts progressivement plus généraux suivant :

```
7. bass
  => musical instrument
    => instrument
      => device
        => instrumentality, instrumentation
          => artifact, artefact
            => object, physical object
```

```
=> entity, something
```

Bien que WordNet soit une ressource complexe et polyvalente pour les mots et leurs sens, il lui manque cependant certaines notions relatives à leur utilisation. WordNet donne peu d'information quant à la fréquence d'un événement. Par exemple, certains sens du mot "bass" sont très rares alors que d'autres, très communs. Bien que WordNet propose un ordonnancement des acceptions les plus courantes d'un mot en fonction de leur fréquence, l'utilisation d'un corpus est préconisée afin d'extraire cette information. Malgré tout, WordNet constitue une source d'information très précieuse sur les mots et leurs significations qui est grandement profitable aux modèles de WSD et à leurs applications.

2.4 Désambiguïsation à l'aide de dictionnaires

Les premiers algorithmes de désambiguïsation sémantique, dans les années soixante, mettaient à profit des dictionnaires afin d'extraire des concepts primitifs qui servaient ensuite de noeuds dans des réseaux sémantiques et connectionnistes. Dans les années quatre-vingt, un effort considérable a été fourni afin de développer des outils tels que des bases de données lexicales et sémantiques disponibles électroniquement, ce qui a permis une utilisation plus aisée de ce type de ressource. Alors que les dictionnaires (unilingues et multilingues) et thésaurus sont maintenant couramment utilisés comme référence dans des applications de WSD, il est néanmoins intéressant de noter que des modèles de désambiguïsation relativement performants ont été conçus à l'aide de la seule information contenue dans ces ressources.

Les définitions incluses dans un dictionnaire ou un thésaurus peuvent être directement utilisées afin de déterminer le sens d'un mot ambigü. Une famille d'algorithmes reposant sur le modèle proposé par Lesk (LESK 1986) parviennent à déterminer le sens d'un mot polysémique uniquement à l'aide des définitions des sens de ce mot et des mots de son voisinage. L'idée exploitée par ces modèles est que les sens les plus plausibles pour des mots co-occurrants sont ceux qui maximisent le degré de corrélation entre ces sens. Un sens est représenté par une liste

formée des mots apparaissant dans sa définition et le degré de corrélation est défini par l'enchevêtrement, soit la quantité de mots qu'ont en commun deux listes. Ainsi, le sens le plus probable pour un mot ambigü est celui pour lequel l'enchevêtrement avec les sens des mots de contexte est le plus grand. Par exemple, afin de désambiguïser l'occurrence de "bass" dans le contexte "bass player" à l'aide de WordNet, il suffit de comparer chacun des sens de "bass" aux définitions du mot "player" :

```
1. player, participant -- (a person who participates
   in or is skilled at some game)
2. musician, instrumentalist, player -- (someone who
   plays a musical instrument (as a profession))
3. actor, histrion, player, thespian, role player --
   (a theatrical performer)
```

La désambiguïsation est effectuée en choisissant le sens pour lequel il y a le plus de mots en commun dans les descriptions, soit le septième sens du mot "bass".

Des travaux beaucoup plus récents (WILKS, SLATOR et GUTHRIE 1996) utilisent des variantes sophistiquées de cette idée. Les modèles de désambiguïsation à base de dictionnaires se distinguent en ceci qu'ils ne nécessitent ni l'analyse humaine, ni des données d'entraînement. L'effort est plutôt mis sur l'extraction de l'information contenue dans des dictionnaires afin de produire des bases de connaissances complexes. La performance de ces approches est cependant très sensible à la constitution des ressources utilisées. En effet, de petites variations dans les définitions peuvent mener à des étiquetages différents.

On peut également mettre à profit les dictionnaires bilingues lors de la désambiguïsation en exploitant les différences de polysémie dans deux langues. En effet, les mots polysémiques dans une langue se traduisent souvent par des termes distincts dans une autre langue. Par exemple, le mot anglais"bass" se traduit en français par "bar" ou "basse" selon le concept qu'il désigne. Il faut cependant disposer de corpus bilingues alignés. Ces derniers peuvent être obtenus par différentes méthodes statistiques mais leur précision n'est pas toujours satisfaisante.

2.5 Désambiguïsation par apprentissage statistique

Le développement de corpus annotés dans les années quatre-vingts a légitimé l'application de méthodes d'apprentissage statistique à la WSD. Ces méthodes permettent d'apprendre les sens des mots dans leur contexte directement à partir d'un corpus sans avoir recours à une coûteuse analyse humaine. La tâche consiste à apprendre un classificateur qui assigne un sens à un mot dans son contexte, et ce directement à partir d'exemples, qui peuvent être des mots étiquetés par leur sens (apprentissage supervisé) ou simplement des mots (apprentissage non-supervisé).

2.5.1 Apprentissage supervisé

L'apprentissage supervisé est effectué à l'aide d'exemples étiquetés tirés de la même distribution que l'ensemble de test. Parmi les approches supervisées, on compte les listes de décision, l'apprentissage à base d'exemples (ou *exemplar-based*), le classificateur bayésien ainsi que le boosting[2]. Le classificateur bayésien ainsi que le classificateur exemplar-based sont des algorithmes de base et leur performance a longtemps été difficile à surpasser. Le boosting n'a été appliqué à la WSD que beaucoup plus récemment et obtient de meilleurs résultats que les deux premières approches.

Listes de décision

Une liste de décision (RIVEST 1987) est une séquence ordonnée de paires "attribut-valeur" décrivant les objets que l'on désire classifier. Les attributs sont appris des caractéristiques des données d'entraînement et les valeurs sont les classes possibles. On peut interpréter les listes de décisions comme une séquence ordonnée de règles du type "si ... alors". La classification est effectuée en testant séquentiellement chaque règle. La classe sélectionnée est celle associée à la première règle de la liste s'appliquant à cet exemple. Lorsque aucune règle ne s'applique, une valeur par défaut est retournée. Une liste de décision est donc une manière de définir un

[2]Ce survol des techniques d'apprentissage supervisé pour la WSD est inspiré de (ESCUDERO, MÀRQUEZ et RIGAU 2000).

Tableau 2.2 – Une liste de décision pour le mot "bass"

Règle	Étiquette
"bass player"	sense7
"tenor" dans le voisinage	sense3

un objet par un patron général accompagné d'exceptions.

Dans les travaux de Yarowsky sur la désambiguïsation sémantique (YAROWSKY 1994; YAROWSKY 2000), les attributs sont des collocations incluant le mot ambigü, c'est-à-dire des paires "mot de contexte-mot ambigü". Les listes de décision sont entraînées en générant et en ordonnant les règles basées sur les caractéristiques des données d'entraînement. Par exemple, pour un mot ambigü donné, on peut considérer que toute paire "collocation-sens" se retrouvant dans l'ensemble d'entraînement est une règle possible. Ces règles sont ordonnées par ratio de log-vraisemblance. Pour chaque sens s_j du mot ambigü et chaque collocation c_i tirée de l'ensemble d'entraînement, on calcule

$$\left|\log\left(\frac{P(s_j|c_i)}{P(\neg s_j|c_i)}\right)\right|$$

avec les probabilités dérivées des fréquences relatives empiriques des événements dans l'ensemble d'entraînement. Les collocations les plus fortement reliées à un sens obtiennent les ratios les plus élevés. La liste est créée en ordonnant les paires sur la base de la log-vraisemblance. Si aucune règle ne s'applique, le sens le plus fréquent est assigné. Le Tableau 2.2 présente quelques exemples de règles qui pourraient être contenues dans une liste de décision pour le mots "bass".

Apprentissage à base d'exemples

En apprentissage à base d'exemples (COVER et HART 1967; AHA, KIBLER et ALBERT 1991), toutes les données d'entraînement sont conservées en mémoire et il n'y a pas de généralisation sur ces dernières. La classification d'un nouvel exemple est basée sur les exemples d'entraînement les plus "similaires". L'algorithme des k plus proches voisins est utilisé pour choisir une étiquette en fonction des exemples

d'entraînement les plus "proches" de l'occurrence à classifiifier. Avec $k > 1$, le sens attribué à un nouvel exemple est celui obtenu par vote de majorité des k plus proches voisins. Si plus d'un sens obtiennent le même nombre de votes, le sens attribué sera alors celui parmi ces sens qui a la meilleure probabilité *a priori*.

Dans les travaux de Ng (NG et LEE 1996), les exemples d'entraînement sont décrits par un ensemble d'attributs formés des mots accompagnant l'occurence du mot ambigü, leurs parties du discours et un ensemble de mots-clés entourant fréquemment le mot ambigü. Les exemples sont conservés en mémoire sous la forme d'un vecteur de valeurs qui correpondent à chacun des attributs. La distance entre deux vecteurs u et v est définie comme la somme des distances entre les valeurs particulières de ces exemples sur tous les attributs :

$$d((u_1, \ldots, u_M), (v_1, \ldots, v_M)) = \sum_{i=1}^{M} d(u_i, v_i)$$

La distance entre deux valeurs u_i et v_i pour un mot et un attribut donnés est :

$$d(u_i, v_i) = \sum_{j=1}^{n} |P(s_j|u_i) - P(s_j|v_i)|$$

où la somme est effectuée sur tous les sens du mot ambigü. Les probabilités $P(s_j|v_i)$ sont estimées par le ratio entre le nombre d'exemples d'entraînement ayant la valeur v_i et étant étiqueté avec le sens s_j et le nombre total d'exemples d'entraînement ayant la valeur v_i pour cet attribut.

Classificateur bayésien

La tâche de désambiguïsation dans une perspective probabiliste peut être énoncée dans les termes suivants : trouver le sens s^* le plus probable étant donné le mot cible w et son voisinage $h_1, \ldots, h_{2n}$:

$$s^* = \arg\max_s P(s|w, h_1, \ldots, h_{2n})$$

En appliquant la règle de Bayes et en faisant l'hypothèse que les attributs w et $h_1, \ldots, h_{2n}$ sont indépendants, on obtient :

$$s^* = \arg\max_s \prod_{i=1}^{2n} P(h_i|s)P(w|s)P(s)$$

Cette technique est connue sous le nom de Naive Bayes (DUDA et HART 1973) et la formulation décrite ici est due à (MANNING et SCHÜTZE 1999). Ce type de classificateur est largement utilisé dans les applications de traitement du langage. En désambiguïsation sémantique, Gale, Church et Yarowsky (GALE, CHURCH et YAROWSKY 1992b; YAROWSKY 1992) ont utilisé avec succès le classificateur Naive Bayes avec plusieurs variantes et leurs travaux représentent à ce jour une référence incontournable en WSD.

On peut évaluer les densités impliquées dans le modèle Naive Bayes à l'aide de l'estimateur par maximum de vraisemblance (MLE[3]) sur un corpus étiqueté. Cet estimateur maximise la log-vraisemblance des données d'entraînement sous les paramètres du modèle. La solution obtenue par maximum de vraisemblance ici correspond à la simple fréquence relative des événements dans le corpus d'entraînement. Pour une distribution de probabilité conditionnelle, on retrouve donc l'estimateur suivant :

$$P_{MLE}(i|j) = \frac{C(i,j)}{\sum_k C(k,j)} = \frac{C(i,j)}{C(j)}$$

La distribution empirique d'un mot ambigü selon ses sens est assez dense pour être utilisée directement par un algorithme tel que le classificateur bayésien ; seul un petit nombre de sens sont répertoriés pour chaque mot par rapport à un ensemble d'une centaine de milliers de sens au total dans une ontologie telle que celle dérivée de WordNet. Il est raisonnable de croire que le corpus d'entraînement est ainsi susceptible de représenter la polysémie d'un mot. Si un sens n'a jamais été vu pour un mot, on peut faire l'hypothèse que ce sens est très rare et que la performance finale du modèle de désambiguïsation n'est pas influencée significativement par ces derniers.

Une quantité colossale de données étiquetées serait cependant nécessaire pour assurer une représentation fidèle des mots de contexte pertinents à la sélection d'un

[3] *Maximum Likelihood Estimation*

des sens d'un mot polysémique. Les distributions obtenues à l'aide d'un corpus étiqueté, de taille très limitée, sont très peu denses et peu représentatives. À peu près aucune information n'est disponible afin de restreindre l'ensemble des mots de contexte pertinents à la discrimination du sens. L'emploi des techniques de lissage est critique avec l'utilisation d'un classificateur bayésien puisqu'elles permettent d'attribuer une probabilité non nulle aux événements rares ou non observés. Ces notions seront approfondies dans le chapitre 3.

Boosting

Le boosting est un algorithme d'apprentissage permettant de produire un classificateur arbitrairement fiable sur l'ensemble d'apprentissage en combinant plusieurs classificateurs moyennement fiables, appelés *hypothèses faibles*. Les hypothèses faibles sont entraînées séquentiellement sur les mêmes données. Chacune d'elles se concentre sur les exemples les plus difficiles à classifier par les précédentes hypothèses. Le boosting a été appliqué avec succès à des tâches pour lesquelles la dimension de l'espace des attributs est très grande et les données peu denses, comme c'est bien souvent le cas en traitement automatique du langage.

L'algorithme maintient une distribution de probabilité sur l'ensemble des exemples étiquetés. À chaque itération, un algorithme d'apprentissage génère une hypothèse avec un taux d'erreur relativement faible par rapport à cette distribution. La distribution est ensuite mise à jour de telle sorte que les exemples mal classifiés aient un poids plus grand que les exemples bien classifiés, ce qui permet aux prochaines hypothèses de se concentrer sur les points les plus difficiles à classifier. La technique de boosting a été appliquée à la désambiguïsation (ESCUDERO, MÀRQUEZ et RIGAU 2000) avec succès ; ils rapportent un taux d'erreur plus petit que ceux de Naive Bayes et exemplar-based.

2.5.2 Apprentissage non-supervisé

Les méthodes de désambiguïsation par apprentissage non-supervisé ne permettent pas d'étiqueter un mot avec un sens selon une référence précise puisque ceci nécessite une caractérisation des sens qui est absente si les données ne sont

pas étiquetées. Cependant, la discrimination du sens d'un mot peut être accomplie de façon non supervisée.

Techniques de classification non-supervisée

Il est possible de diviser les occurrences d'un mot ambigü en classes distinctes à partir de textes à l'aide de techniques de classification non-supervisée. Ces classes peuvent être différenciées sans avoir à leur assigner un sens précis ; les éléments d'une même classe doivent simplement être "proches" et les différentes classes être "éloignées" les unes des autres. Ceci implique la définition d'une notion appropriée de similarité entre les contextes. Les éléments formant une classe sont des vecteurs de caractéristiques qui correspondent à une occurence particulière d'un mot ambigü et dont les caractéristiques sont issues des données d'entraînement. Les mots de contexte servent à dériver ces caractéristiques.

Par exemple, on peut former un vecteur de co-occurrence du mot ambigü avec les mots de son voisinage. Chaque exemple du mot ambigü correspond ainsi à un vecteur d'entrées binaires indiquant la présence ou l'absence de chaque mot du vocabulaire dans le voisinage (SCHUTZE 1998). Les vecteurs semblables sont ensuite regroupés en classes à l'aide d'un algorithme de classification non-supervisée. Chaque groupe correspond ainsi à une des acceptions du mot ambigü. Un nouvel exemple du mot ambigü est classifié en choisissant le sens (c'est-à-dire le groupe) dont le vecteur de contexte est le plus "proche".

Un espace d'attributs et une mesure de distance sont cependant nécessaires à l'emploi des techniques de classification non-supervisée. Typiquement, les vecteurs sont en très haute dimension (la taille du vocabulaire) ce qui n'en facilite pas l'analyse et soulève une des questions les plus fondamentales de l'apprentissage statistique.

Algorithme EM

L'algorithme EM[4] (DEMPSTER, LAIRD et RUBIN 1977) est un algorithme itératif de maximum de vraisemblance lorsque les données sont incomplètes. Il permet

[4] *Expectation Maximisation*

entre autres l'apprentissage de distributions de probabilité impliquant des variables pour lesquelles il n'y a pas toujours de données supervisées. Les paramètres d'un modèle de désambiguïsation, par exemple un classificateur bayésien, peuvent ainsi être appris sur un corpus non-étiqueté avec EM. La technique itérative, qui sera détaillée dans la chapitre 3, repose sur le calcul de l'espérance des variables cachées étant donné les paramètres du modèle courant puis sur la recherche des nouveaux paramètres tels que la vraisemblance jointe des données et des variables cachées est maximisée.

Pedersen et Bruce (PEDERSEN et BRUCE 1998) déterminent un ensemble de paramètres $P(w_i|s_j)$ et $P(s_j)$ qui expliquent le mieux les données d'entraînement. Les paramètres sont inilialisés aléatoirement. À chaque itération, ils déterminent $C(w_i, s_j) = P(s_j|w_i)C(w_i)$ pour tous les mots et tous les sens. Les paramètres sont ensuite mis à jour de la façon suivante :

$$P(w_i|s_j) = \frac{C(w_i, s_j)}{\sum_k C(w_k, s_j)}$$

Les paramètres qui ne sont pas appris par EM, soit le nombre total de sens pour expliquer les données, peuvent être déterminés par validation croisée. Ceci a pour avantage de permettre une désambiguïsation très spécifique (beaucoup de sens) ou très générale (peu de sens).

2.5.3 Apprentissage semi-supervisé

Un grand ensemble d'exemples étiquetés n'est pas nécessaire à l'emploi des techniques d'aprentissage semi-supervisé. Il suffit qu'une petite portion de l'ensemble d'entraînement soit étiquetée et des performances comparables à celle d'algorithmes entièrement supervisés peuvent être atteintes. L'idée exploitée par ces techniques est d'entraîner séquentiellement des classificateurs qui génèrent eux-mêmes de nouvelles étiquettes.

Les exemples étiquetés sont appelés germes et contribuent à entraîner un classificateur de base. Ce classificateur est ensuite appliqué à assigner une classe aux données non-étiquetées. Les exemples pour lesquels le classificateur était le plus "sûr" (par exemple ceux pour lesquels la probabilité était au-dessus d'un seuil pré-

déterminé) sont ajoutés à l'ensemble des germes. Ce nouvel ensemble est utilisé afin d'entraîner un deuxième classificateur. Le processus est ainsi répété jusqu'à ce que l'ensemble d'entraînement supervisé soit stable. Il en résulte une série de classificateurs dont la performance s'améliore généralement. Yarowsky (YAROWSKY 1995) décrit un algorithme semi-supervisé basé sur des listes de décisions (voir section 2.5.1).

2.6 Comparaison

L'approche de Lesk a longtemps été une référence en désambiguïsation sémantique. Les algorithmes d'apprentissage statistique sont maintenant plus performants que les algorithmes de désambiguïsation à l'aide de dictionnaires. En particulier, les algorithmes d'apprentissage supervisés obtiennent de meilleurs résultats que les algorithmes non-supervisés. Le sens le plus fréquent est maintenant considéré comme le niveau de performance de base pour la comparaison des modèles de désambiguïsation sémantique (GALE, CHURCH et YAROWSKY 1992a). La plupart des algorithmes d'apprentissage statistique présentés dans ce chapitre obtiennent plus de succès que cette heuristique très simple. Les auteurs rapportent des résultats sur des tâches très variées et la plupart du temps très spécifiques. Il est donc difficile de produire une analyse comparative de ces algorithmes.

Nous pouvons néanmoins nous référer au travail de Mooney (MOONEY 1996) qui propose une étude comparative de certains algorithmes d'apprentissage appliqués à la WSD, dont les algorithmes supervisés décrits plus tôt. La tâche de désambiguïsation porte sur six des sens du mot "line". Parmi les algorithmes évalués, le classificateur Naive Bayes est celui qui obtient les meilleurs résultats. De façon générale, la technique Naive Bayes a une bonne performance sur les tâches de classification. Ce constat positif est observé malgré l'hypothèse d'indépendance conditionnelle qui peut être injustifiée dans les domaines étudiés, en particulier en traitement automatique du langage. L'apprentissage à base d'exemples se classe également en tête. Ces deux techniques constituent l'état de l'art en WSD. Cependant, la performance des meilleures techniques dépasse de peu celle de modèles

utilisant des règles très simples, par exemple la classification basée sur le sens le plus fréquent.

CHAPITRE 3

RÉDUCTION DE DIMENSION POUR LA DÉSAMBIGUÏSATION SÉMANTIQUE

Le présent chapitre propose de nouvelles approches afin de modéliser l'influence des mots de contexte sur le sens d'un mot dans une application de désambiguïsation. Les premières sections sont une introduction aux idées exploitées ainsi qu'au formalisme utilisé afin d'illustrer les modèles de désambiguïsation, les modèles graphiques. Les techniques de réduction de la dimension, en particulier la décomposition en valeurs singulières, sont ensuite brièvement présentées. Les sections suivantes décrivent en détail les modèles suggérés ainsi que les algorithmes impliqués dans leur conception et leur apprentissage.

3.1 Introduction

Les modèles de désambiguïsation présentés dans les prochains paragraphes sont inspirés du classificateur Naive Bayes. L'objectif est de "factoriser" la distribution

des mots de contexte et des sens ($P(h|s)$). Dans un classificateur Naive Bayes, $P(h|s)$ obtenue par MLE n'est pas fiable puisque très peu de données sont disponibles pour son estimation. Les modèles que nous proposons permettent de lisser cette distribution implicitement en utilisant une décomposition algébrique de la matrice de co-occurrence des mots de contexte et des sens. Cette idée prend la forme de variables cachées que nous interprétons comme des thématiques générales dans les documents. L'introduction de ces variables dans le modèle résulte en une réduction de la dimension de l'espace des paramètres. Les sens et les mots sont ainsi représentés dans un espace implicite aux données d'entraînement. Cet espace est beaucoup plus compact que l'espace dans lequel les données sont représentées à l'origine et permet de lisser la distribution des mots de contexte et des sens.

Des technique similaires ont été appliquées avec succès dans d'autres problèmes de traitement automatique du langage. Dans le domaine de la recherche d'information (RI), les mots et les documents sont représentés dans un espace avec peu de dimensions à l'aide de l'"analyse sémantique latente" (LSA[1]) (DEERWESTER, DUMAIS, FURNAS, LANDAUER et HARSHMAN 1990) des mots et des documents. La LSA est une méthode statistique pour l'analyse de données de co-occurrence. L'idée de base est de projeter des vecteurs de comptes qui sont en très haute dimension, tels ceux présents dans des espaces dérivés de documents, vers une représentation en plus petite dimension, appelée "espace sémantique latent". Les mots présents dans les documents sont traités non pas comme des unités représentatives de la réalité mais plutôt comme des exemples tirés d'un grand ensemble de mots qui auraient pu être observés. Cet espace à faible dimension est dérivé en approximant à l'aide de l'algèbre linéaire une matrice de co-occurrence par une matrice de rang inférieur en utilisant ses composantes principales.

Dans le cadre de la WSD, les matrices de co-occurrence des mots et des sens sont très similaires aux matrices d'occurrence des mots dans les documents qui sont utilisées en recherche d'information. La dimension de l'espace, correspondant à la taille du vocabulaire, est très grande dans les deux problèmes et la densité des matrices, très petite. Le succès obtenu par les algorithmes de réduction de dimension en RI a motivé l'application de techniques semblables en désambiguïsation séman-

[1] *Latent Semantic Analysis*

tique. L'objectif des modèles décrits dans ce chapitre est le même que celui visé par l'analyse sémantique latente : représenter les mots et les sens dans un espace sémantique "caché" dans les données d'entraînement.

En somme, cette étude propose de nouvelles méthodes de désambiguïsation sémantique exploitant l'idée de la décomposition algébrique des matrices de co-occurrence des mots et des sens et de la projection des données dans un espace thématique ayant peu de dimensions. Les modèles sont décrits en détail dans la section 3.4. Nous discutons d'abord des modèles graphiques et les techniques de réduction de la dimension.

3.2 Modèles graphiques

Les modèles graphiques (JORDAN 1998) sont un formalisme issu à la fois de la théorie des probabilités et de l'apprentissage statistique. Ils constituent un cadre formel pour la conception et l'analyse de nombreux algorithmes d'apprentissage statistique. L'idée de base est de diminuer la complexité de distributions de probabilité jointe en introduisant l'indépendance conditionnelle entre des variables. Les modèles graphiques permettent de structurer les variables d'un modèle et de visualiser l'ensemble des dépendances entre ces dernières. Une multitude de problèmes étudiés en théorie de l'information et en apprentissage statistique sont des cas particuliers de modèles graphiques, par exemple les chaînes de Markov et les modèles de Markov cachés (BENGIO 1999).

Un modèle graphique est un graphe dont les noeuds représentent des variables aléatoires décrivant les états possibles du système et les arcs, des dépendances conditionnelles entre ces variables. Dans un modèle graphique dirigé, l'absence d'arc entre A et B implique que A et B sont conditionnellement indépendants aux variables séparant A et B dans le graphe. L'ensemble des variables comportant un arc vers A sont appelés les parents de A dans le graphe. Les modèles graphiques permettent de simplifier la représentation d'un ensemble de variables aléatoires. En particulier, la distribution de probabilité jointe d'un ensemble de variables aléatoires $V = A_1, \ldots, A_n$ peut toujours être exprimée par :

$$P(A_1, \ldots, A_n) = P(A_1) \prod_{i=2}^{n} P(A_i | A_1^{i-1}))$$

Les hypothèses d'indépendance conditionnelle dans un modèle graphique dirigé permettent d'exprimer cette probabilité jointe par :

$$P(A_1, \ldots, A_n) = \prod_{i=1}^{n} P(A_i | parents(A_i))$$

où $parents(A_i) = \{B \in V | \text{il y a un arc de } B \text{ vers } A_i\}$.

Dans le contexte de la désambiguïsation sémantique, nous utilisons des modèles grahiques dirigés et probabilistes. Les noeuds représentent des variables aléatoires discrètes comme les mots et les sens, alors que les arcs sont associés à des distributions de probabilité conditionnelle. Ce type de modèle graphique est aussi appelé réseau bayésien. Par exemple, un modèle graphique pour le classificateur Naive Bayes décrit dans la section 2.5.1 est donné dans la Figure 3.1. Le sens S d'un mot W, dans ce modèle, est influencé par ce mot ainsi que son voisinage H_1^{2n}. Les mots de contexte et le mot ambigü sont indépendants étant donné le sens du mot ambigü.

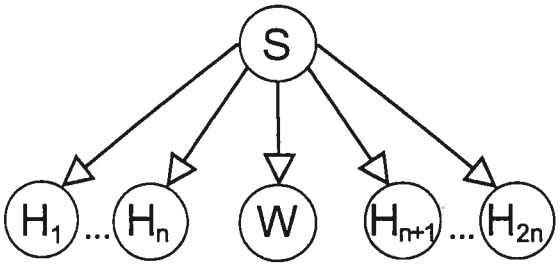

Figure 3.1 – Modèle graphique correspondant au modèle Naive Bayes.

Les problèmes d'apprentissage d'un modèle graphique probabiliste sont ceux de la détermination de sa topologie particulière et de l'optimisation des paramètres des distributions de probabilité décrivant les relations entre les variables du modèle. Cette dernière tâche peut être effectuée à l'aide d'algorithmes d'optimisation comme l'algorithme EM. Dans le cadre de cette étude, nous proposons de nouvelles

architectures pour des modèles graphiques probabilistes, impliquant une réduction de la dimension. Ces architectures permettent de modéliser les dépendances entre les mots et les sens au sein d'une application de désambiguïsation sémantique. Avant de présenter ces modèles ainsi que leur apprentissage sur un corpus, nous introduisons les techniques de réduction de la dimension, en particulier la décomposition en valeurs singulières.

3.3 Réduction de la dimension

Les techniques de réduction de la dimension sont des méthodes permettant de représenter dans un espace en faible dimension des objets décrits au départ dans un espace en haute dimension. Les méthodes statistiques classiques sont l'analyse en composantes principales (PCA[2]) (JOLLIFFE 1986) et l'analyse factorielle (BARTHOLOMEW 1987). Ces méthodes sont basées sur des combinaisons linéaires des attributs des données d'entraînement. Ces attributs peuvent être ceux qui présentent le moins de variance dans les données d'entraînement ou ceux qui sont les plus corrélés entre eux. Les algorithmes de réduction de la dimension permettent la visualisation, la catégorisation et la simplification de grands ensembles d'entraînement décrits avec beaucoup de paramètres et sont appliqués dans une variété de problèmes.

3.3.1 Décomposition en valeurs singulières

La décomposition en valeurs singulières (SVD[3]) (GOLUB et LOAN 1991) est un algorithme de réduction de la dimensionalité exploitant les directions de plus grande variation dans les données, tout comme la PCA et l'analyse factorielle. La SVD est un algorithme très proche de l'analyse en composantes principales. La PCA est une méthode de projection des objets sur les axes correspondant aux plus grandes variations dans les données. Ces axes sont ceux dont la direction est ob-

[2] *Principal Component Analysis*
[3] *Singular Value Decomposition*

tenue par les vecteurs propres de la matrice de covariance des données d'entraînement. Une valeur propre de la matrice A est une racine de l'équation caractéristique $|A - \lambda I| = 0$. Le nombre de valeurs propres non nulles de A est égal au rang de A.

Définition 3.1 *Soit A une matrice carrée $n \times n$, v un vecteur non nul et λ un scalaire tels que $Av = \lambda v$. Alors, v est appelé vecteur propre de A et λ, valeur propre de A.*

On a que $A = \sum_{i=1}^{n} \lambda_i v_i v_i^T$. On peut approximer la matrice A par

$$A = \sum_{i=1}^{d<n} \lambda_i v_i v_i^T$$

où $|\lambda_i| \geq |\lambda_{i+1}|$. La réduction de la dimension est effectuée en négligeant les directions exprimant le moins de variance. L'analyse en composantes principales ne s'applique qu'aux matrices carrées, alors que la décomposition en valeurs singulières peut être effectuée sur des matrices quelconques.

Proposition 3.1 *Soit A une matrice de dimension $n \times m$ et de rang $r < min(n, m)$. Alors, il existe U, V et σ tels que*

$$A = U \Sigma V^T$$

La matrice U, de dimension $n \times r$, contient les vecteurs singuliers gauches et est formée des vecteurs propres de $A^T A$; V, de dimension $r \times m$, contient les vecteurs singuliers droits et est formée des vecteurs propres de AA^T ; $\Sigma = diag(\alpha_i)$ contient les valeurs singulières de A sur la diagonale, qui sont les racines des valeurs propres de $A^T A$.

Une matrice A peut être approximée par une matrice $\hat{A}$ de rang inférieur telle que la distance euclidienne entre A et $\hat{A}$ est minimisée à l'aide de la décomposition en valeurs singulières. La meilleure approximation de rang k en norme euclidienne de A est obtenue en conservant les k plus grandes valeurs singulières dans Σ et les vecteurs singuliers qui y sont associés dans U et V. $\hat{A}$ fournit une description canonique de la matrice A en tant que somme de k matrices de rang 1 d'importance

décroissante. Cette importance est mesurée par les valeurs singulières. La SVD est accomplie de telle sorte que $\hat{A}$ minimise la norme de Frobenius de l'erreur, c'est-à-dire que parmi toutes les matrices A' de rang k,

$$\begin{aligned}\hat{A} &= \arg\min_{A'} ||A - A'||_F \\ &= \arg\min_{A'} \sqrt{\Sigma_{i,j}(A_{ij} - A'_{ij})^2}\end{aligned}$$

L'approximation d'une matrice A par une autre de rang k signifie que les rangées et les colonnes de A sont maintenant exprimées dans un espace à k dimensions. En choisissant k plus petit que le rang r de la matrice A, la SVD permet de réduire la dimension tout en conservant l'information la plus pertinente. L'approximation d'une matrice par la SVD permet en outre de réduire le bruit présent dans les données, puisque les directions qui correspondent aux plus petites valeurs singulières sont souvent celles qui sont dues au bruit. Le rang de l'approximation est donc souvent beaucoup plus petit que le rang de la matrice originale. La SVD est très utile en traitement d'images car elle permet de les compresser et de les filtrer. En recherche d'information, la SVD est à la base de l'analyse sémantique latente. L'espace dans lequel sont projetées les données à l'aide de l'approximation par la SVD est appelé espace sémantique latent.

3.4 Modèles proposés

Les modèles de désambiguïsation présentés dans les prochains paragraphes exploitent l'idée de la LSA sur des matrices de co-occurrence de mots et de sens. Étant donné que la variable de concept propre à nos modèles n'est pas observée dans les données, les distributions de probabilité qui l'impliquent sont apprises à l'aide de l'algorithme EM. La décomposition en valeurs singulières de la distribution de co-occurrence des mots de contexte et des sens est appliquée afin d'initialiser les paramètres du modèle.

La notation qui suit sera utilisée tout au long du chapitre. Des lettres majuscules sont utilisées pour dénoter les variables aléatoires, et des lettres minuscules pour leurs valeurs. Le problème est caractérisé par les variables aléatoires suivantes :

- $W \in \mathcal{W} = \{w_1, \ldots, w_N\}$ est le mot ambigü
- $H_i \in \mathcal{H} = \{h_1, \ldots, h_M\}$ est le i^e mot de contexte
- $S \in \mathcal{S} = \{s_1, \ldots, s_L\}$ est le sens du mot ambigü
- $R_i \in \mathcal{S}$ est le sens du i^e mot de contexte
- $C \in \mathcal{C} = \{c_1, \ldots, c_K\}$ est un concept

Pour un mot ambigü donné w, on note le sous-ensemble de mots dans le contexte de w et des sens répertoriés dans WordNet pour w par les lettres correspondant aux ensembles de mots de contexte et de sens indicés par w.

- $\mathcal{H}_w = \{h_1, \ldots, h_{2n}\} \subset \mathcal{H}$ désigne l'ensemble des mots dans le contexte de w
- $\mathcal{S}_w = \{s_1, \ldots, s_l\} \subset \mathcal{S}$ désigne l'ensemble des sens possibles de w

Le corpus d'entraînement $\mathcal{T}$ est composé d'exemples étiquetés sémantiquement. Le t^e exemple est constitué d'un mot ambigü w_t ainsi que des $2n$ mots de contexte, notés par $h_{t,1} \ldots h_{t,2n}$. Le sens de w_t dans le corpus est s_t alors que les sens des mots de contexte sont notés par $r_{t,1} \ldots r_{t,2n}$ Nous dénotons ainsi $\mathcal{T}$ par l'ensemble des paires de mots et de leurs étiquettes sémantiques :

$$\mathcal{T} = \{(h_{1,1}, r_{1,1}, \ldots, h_{1,2n}, r_{1,2n}, w_1, s_1), \ldots, (h_{T,1}, r_{T,1}, \ldots, h_{T,2n}, r_{T,2n}, w_T, s_T)\}$$

Finalement, les distributions de probabilité sont représentées par des matrices. La notation M_X désigne la matrice décrivant la distribution de la variable X. Pour une distribution de probabilité conditionnelle $P(X = x|Y = y)$, nous utilisons le symbole M_{XY}. Ainsi, la probabilité $P(X = x_i)$ est donnée par la i^e ligne du vecteur M_X. Afin d'alléger les équations, $P(X = x)$ sera noté $P(x)$.

3.4.1 Modèle Contexte-Concept-Sens (CCS)

Le modèle de désambiguïsation CCS introduit une variable de concept qui lie un mot de contexte et le sens du mot cible. La variable de concept peut être interprétée comme un thème plus général que les mots et les sens et dont la valeur conditionne le sens du mot ambigü. Un modèle graphique correspondant à cette idée est donné avec la Figure 3.2. L'objectif du modèle CCS est la factorisation de la matrice creuse de co-occurrence des mots de contexte et des sens. Cette factorisation permet de lisser implicitement la probabilité $P(h|s)$ que l'on retrouve dans le modèle Naive Bayes.

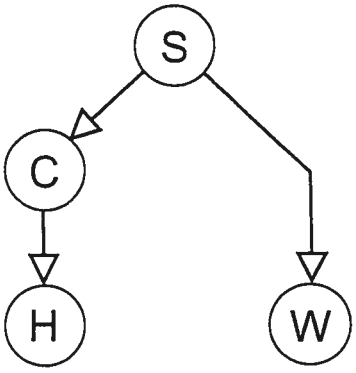

Figure 3.2 – Modélisation graphique de la factorisation à l'aide de la variable de concept C de la probabilité d'un mot de contexte H étant donné le sens S du mot ambigü W.

La probabilité d'un mot de contexte h étant donné le sens s d'un mot ambigü est obtenue en sommant sur toutes les valeurs prises par la variable de concept :

$$\begin{aligned} P(h|s) &= \sum_{c \in \mathcal{C}} P(h, c|s) \\ &= \sum_{c \in \mathcal{C}} P(h|c, s) P(c|s) \end{aligned}$$

Étant donné que le graphe suppose l'indépendance conditionnelle des mots de contexte et du sens du mot cible, nous avons que :

$$P(h|s) = \sum_{c \in \mathcal{C}} P(h|c)P(c|s)$$

Dans le modèle CCS, cette estimation de la probabilité d'un mot de contexte étant donné le sens du mot ambigü remplace dans un Naive Bayes la simple estimation de $P(h|s)$ obtenue par MLE. Les mots sont intégrés dans le modèle sans considération pour leur position par rapport au mot ambigü, c'est-à-dire $P(h|c)$ ne dépend que de l'identité du mot h et pas de sa position dans le contexte. Ceci implique un partage des paramètres dans le modèle.

La matrice de probabilité des mots de contexte selon les sens M_{HS} est factorisée en le produit de deux matrices $M_{HC} \times M_{CS}$. Nous choisissons la taille de l'ensemble C $K \ll M, N$. Étant donné que la cardinalité de l'ensemble des valeurs prises par la variable C est très petite par rapport à celles de W et S, les matrices M_{HC} et M_{CS} comportent beaucoup moins de paramètres que la matrice M_{HS}, mais elles sont par contre plus denses. Cette factorisation permet donc de réduire le nombre de paramètres à estimer. En reprenant la notation introduite dans la section 2.5.1, le classificateur CCS est décrit par

$$s^* = \arg\max_{s \in \mathcal{S}_w} P(w|s)P(s) \prod_{h \in \mathcal{H}_w} \sum_{c \in \mathcal{C}} P(h|c)P(c|s)$$

3.4.2 Modèle Contexte-Sens-Concept-Sens (CSCS)

Le modèle CSCS intègre les sens des mots de contexte au modèle précédent. Les variables cachées symbolisant les thématiques relient maintenant le sens d'un mot de contexte au sens du mot ambigü. Les thèmes peuvent être interprétés comme des synsets de haut niveau dans WordNet conditionnant les sens des mots voisins dans un texte. Une interprétation graphique de cette idée est donnée avec la Figure 3.3.

La relation entre un mot de contexte et le sens du mot cible est conditionnée par les sens du mot de contexte, qui sont eux-mêmes conditionnés par la présence de la variable de concept. La probabilité d'un mot de contexte h conditionnelle à un sens

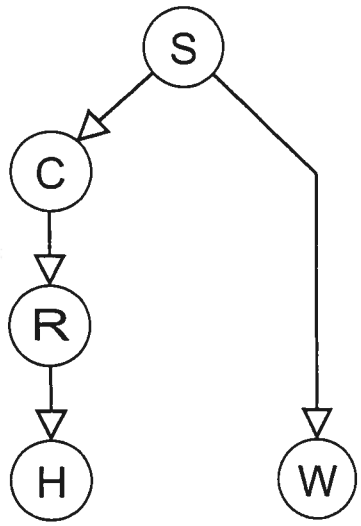

Figure 3.3 – Modélisation graphique de la factorisation à l'aide de la variable de concept C de la probabilité du sens R d'un mot de contexte H étant donné le sens S du mot ambigü W.

s est obtenue en sommant sur tous les sens de h puis sur toutes les valeurs prises par la variable concept. Nous faisons l'hypothèse que les mots de contexte sont indépendants du concept étant donné leur sens et que les sens sont indépendants étant donné la connaissance de la valeur de la variable de concept, ce qui permet le passage de la deuxième à la troisième égalité ci-bas :

$$\begin{aligned} P(h|s) &= \sum_{r \in \mathcal{S}_h} \sum_{c \in \mathcal{C}} P(h, r, c|s) \\ &= \sum_{r \in \mathcal{S}_h} \sum_{c \in \mathcal{C}} P(h|r, c) P(r|c, s) P(c|s) \\ &= \sum_{r \in \mathcal{S}_h} P(h|r) \sum_{c \in \mathcal{C}} P(r|c) P(c|s) \end{aligned}$$

Nous supposons également que $P(h|r)$ et $P(r|c)$ ne dépendent pas de la position

du mot de contexte h et de son sens r dans le contexte.

La matrice de co-occurrence des sens des mots voisins M_{RS} est factorisée en $M_{RC} \times M_{CS}$ par le biais de la variable concept. À nouveau, la présence des concepts dans le modèle permet de réduire le nombre de paramètres à estimer, puisque $K \ll M$. La distribution des mots et de leur sens est, comme nous l'avons mentionné précédemment, plus dense que la distribution des mots de contexte et des sens. Le sens assigné à un mot w par le classificateur CSCS est donné par

$$s^* = \arg\max_{s \in \mathcal{S}_w} P(w|s)P(s) \prod_{h \in \mathcal{H}_w} \sum_{r \in \mathcal{S}_h} P(h|r) \sum_{c \in \mathcal{C}} P(r|c)P(c|s)$$

Les détails concernant l'estimation des paramètres du modèle sur un corpus d'entraînement à l'aide de l'algorithme EM sont présentés dans la section suivante.

3.5 Estimation des paramètres du modèle

Comme dans le cas du modèle Naive Bayes, les distributions de probabilité régissant les dépendances entre les sens et les mots sont estimées par maximum de vraisemblance sur un corpus de mots étiquetés avec leur sens. Les estimateurs MLE des paramètres des distributions $P(S)$, $P(W|S)$, $P(H|S)$ et $P(R|S)$ consistent simplement en les fréquences relatives de ces événements sur le corpus d'entraînement :

$$\begin{aligned}
P(s) &= \frac{\sum_{t=1}^{T} \delta_{s_t=s}}{T} \\
P(w|s) &= \frac{\sum_{t=1}^{T} \delta_{s_t=s, w_t=w}}{\sum_{t=1}^{T} \delta_{s_t=s}} \\
P(h|s) &= \frac{\sum_{t=1}^{T} \sum_{i=1}^{2n} \delta_{s_t=s, h_{t,i}=h}}{\sum_{t=1}^{T} \sum_{i=1}^{2n} \delta_{s_t=s}} \\
P(r|s) &= \frac{\sum_{t=1}^{T} \sum_{i=1}^{2n} \delta_{s_t=s, s_{t,i}=r}}{\sum_{t=1}^{T} \sum_{i=1}^{2n} \delta_{s_t=s}}
\end{aligned}$$

Cependant, la présence dans le modèle de la variable de concept, pour laquelle aucune donnée n'est disponible, donne lieu à une phase d'entraînement à l'aide de l'algorithme EM, qui permet l'optimisation de la vraisemblance des données d'entraînement sous les paramètres du modèle. Il est à noter que cette technique d'optimisation ne garantit pas de trouver un maximum global, seulement d'approcher un maximum local. Des techniques de lissage sont finalement appliquées afin de permettre l'attribution d'une probabilité non-nulle à un événement non-observé. Les prochains paragraphes décrivent en détail l'algorithme EM et le lissage ainsi que leur application à notre problème.

3.5.1 L'algorithme EM

L'algorithme EM est une technique d'optimisation basée sur le principe de maximum de vraisemblance. Il est employé lorsque les données sont incomplètes, c'est-à-dire lorsque certaines variables d'un modèle ne sont pas représentées dans les données. Les paramètres du modèle sont mis à jour d'une manière garantissant l'amélioration itérative de la vraisemblance des données d'entraînement. EM permet d'apprendre des distributions de probabilité impliquant des variables cachées mais qui influencent néanmoins les variables observées.

L'algorithme EM est itératif et converge vers un maximum local de la fonction de vraisemblance. Ainsi, le choix des valeurs initiales des paramètres peut influencer le résultat de l'algorithme. À chaque itération, on peut choisir les nouveaux paramètres de telle sorte que la vraisemblance des données d'entraînement ne diminue pas. La technique repose sur deux phases appliquées successivement jusqu'à la convergence du modèle : le calcul de l'espérance des variables cachées étant donné le modèle courant et les données (E-step) et la recherche des nouveaux paramètres du modèle tels que la vraisemblance des données est maximisée (M-step).

Dans le cadre du modèle CCS, l'espérance de la variable concept étant donné les données d'entraînement et le modèle courant est donnée par :

$$
\begin{aligned}
P(c|h,s) &= \frac{P(h|c,s)P(c|s)}{P(h|s)} \\
&= \frac{P(h|c)P(c|s)}{P(h|s)}
\end{aligned}
$$

Les paramètres du modèle CCS sont mis à jour de la façon suivante :

$$
\begin{aligned}
P(h|c) &= \frac{\sum_{t=1}^{T}\sum_{i=1}^{2n} P(c_t = c|s,h)\delta_{h_{t,i}=h}}{\sum_{t=1}^{T}\sum_{i=1}^{2n} P(c_t = c|s_t,h_{t,i})} \\
P(c|s) &= \frac{\sum_{t=1}^{T}\sum_{i=1}^{2n} P(c_t = c|s_t,h_{t,i})\delta_{s_t=s}}{\sum_{t=1}^{T}\sum_{i=1}^{2n} \delta_{s_t=s}}
\end{aligned}
$$

De façon similaire, on obtient pour le modèle CSCS les probabilités postérieures et les formules de ré-estimation suivantes :

$$
\begin{aligned}
P(c|r,s) &= \frac{P(r|c)P(c|s)}{P(r|s)} \\
P(r|c) &= \frac{\sum_{t=1}^{T}\sum_{i=1}^{2n} P(c_t = c|s,r)\delta_{s_{t,i}=r}}{\sum_{t=1}^{T}\sum_{i=1}^{2n} P(c_t = c|s_t,r_{t,i})} \\
P(c|s) &= \frac{\sum_{t=1}^{T}\sum_{i=1}^{2n} P(c_t = c|s_t,r_{t,i})\delta_{s_t=s}}{\sum_{t=1}^{T}\sum_{i=1}^{2n} \delta_{s_t=s}}
\end{aligned}
$$

En résumé, à chaque itération, nous procédons 1) au calcul des probabilités postérieures $P(c|w,s)$ (modèle CCS) ou de $P(c|r,s)$ (modèle CSCS) et 2) à la mise à jour des probabilités $P(w|c)$ et $P(c|s)$ (modèle CCS) ou de $P(r|c)$ et $P(c|s)$ $\forall c \in \mathcal{C}, h \in \mathcal{H}, s, r \in \mathcal{S}$.

L'algorithme EM nécessite une approximation initiale des paramètres. Étant donné que EM converge vers un minimum local, il est possible que les valeurs initiales des paramètres affectent la performance finale du système. Deux des nom-

breuses manières d'intialiser les paramètres sont décrites dans les prochains paragraphes.

Initialisation aléatoire des paramètres

Une première approche consiste en l'initialisation aléatoire des paramètres. Chaque paramètre des distributions $P(H|C)$ et $P(C|S)$ se voit d'abord attribué un compte tiré aléatoirement d'une distribution uniforme. Les distributions sont ensuite normalisées.

$$P(i|j) = \frac{c(i,j)}{\sum_i c(i,j)}$$

$c(i,j)$ est un nombre réel positif obtenu à l'aide d'un générateur de nombre aléatoires. Bien que l'initialisation aléatoire constitue une manière raisonnable d'approximer les paramètres initiaux, une solution plus élaborée a été envisagée. L'idée est de mettre à profit un algorithme de compression des données, la décomposition en valeurs singulières.

Initialisation des paramètres à l'aide de la SVD

Nous suggérons d'initialiser les paramètres du modèle en utilisant directement la décomposition en valeurs singulières des matrices de co-occurrence des mots de contexte et des sens dans le cas du modèle CCS et des matrices de co-occurrence des sens dans le cas du modèle CSCS. Les vecteurs singuliers associés aux K plus grandes valeurs singulières sont utilisés afin de dériver l'approximation initiale des vecteurs de probabilité conditionnelle impliquant la variable concept. Ainsi, dans le modèle CCS, la décomposition en valeurs singulières $M_{HS} = U\Sigma V^T$ nous permet l'interprétation des matrices U et ΣV^T comme des matrices contenant des poids pouvant être utilisées afin d'initialiser les paramètres des distributions M_{HC} et M_{CS}. De façon similaire, les paramètres de M_{SC} et M_{CS} sont initialisés à partir de l'approximation par la SVD de la matrice M_{SS}.

Pour une matrice positive, les valeurs singulières et vecteurs singuliers sont réels. Cependant, il n'est pas garanti que les éléments des vecteurs singuliers seront positifs. Dans le cadre des modèles CCS et CSCS, il est nécessaire d'avoir

une interprétation probabiliste des vecteurs singuliers obtenus à l'aide de la SVD puisqu'ils représentent des vecteurs de probabilités conditionnelles utilisées par les modèles. On doit donc transformer les vecteurs singuliers en vecteurs de probabilité avant de ré-estimer les paramètres qu'ils contiennent avec l'algorithme EM. Quelques unes des transformations possibles sont décrites dans les paragraphes suivants.

Une façon très simple d'obtenir un vecteur de probabilité à partir d'un vecteur quelconque est d'effectuer une translation de tous ses éléments et ensuite de normaliser afin que la somme des éléments soit 1. La translation déplace le vecteur jusqu'au quadrant positif sans changer son orientation. En additionnant une constante assez grande, tous les éléments d'un vecteur deviennent positifs. On peut ainsi normaliser le vecteur et obtenir une approximation initiale des probabilités recherchées. Les équations suivantes décrivent la translation normalisée :

$$
\begin{aligned}
P(w_i|c_j) &= \frac{U_{n\times k}(i,j) + \min_l U_{n\times k}(l,j) + \epsilon}{\sum_l U_{n\times k}(l,j) + \min_l U_{n\times k}(l,j) + \epsilon} \\
P(c_i|s_j) &= \frac{V_{k\times m}(i,j) + \min_l V_{k\times m}(l,j) + \epsilon}{\sum_l V_{k\times m}(l,j) + \min_l V_{k\times m}(l,j) + \epsilon}
\end{aligned}
$$

On peut vérifier que P est bien une distribution de probabilité conditionnelle, c'est-à-dire que $P(i|j) \geq 0 \; \forall \, i, j$ et $\sum_i P(i|j) = 1$.

On peut également convertir des poids quelconques en probabilités à l'aide de la fonction softmax. La fonction softmax est une fonction exponentielle normalisée qui est souvent utilisée par exemple pour dériver une approximation de la probabilité conditionnelle d'appartenance à une classe dans des réseaux de neurones. La fonction softmax permet d'encoder les contraintes probabilistes (éléments entre 0 et 1 et somme des éléments égale à 1) de manière simple et fournit une interprétation commode des poids des éléments d'une population comme des ratios positifs. L'utilisation de cette fonction repose sur la distance à la moyenne et sur l'écart-type de la population initiale. Les valeurs atypiques sont donc transformées en valeurs proches de 0 ou de 1.

Les matrices composées des vecteurs singuliers peuvent donc être transformées

en des matrices de probabilité de transition en appliquant la fonction softmax puis en normalisant.

$$
\begin{aligned}
P(h_i|c_j) &= \frac{\exp(U_{n\times k}(i,j))}{\sum_l \exp(U_{n\times k}(l,j)} \\
P(c_i|s_j) &= \frac{\exp(V_{k\times m}(i,j)}{\sum_l \exp(V_{k\times m}(l,j)}
\end{aligned}
$$

Il n'est cependant pas garantit que la translation normalisée ou la fonction softmax préserve toute l'information véhiculée par les vecteurs singuliers. Nous utiliserons néanmoins ces techniques afin d'initialiser les paramètres des modèles présentés dans ce chapitre.

L'implantation de la SVD pour les matrices creuses a été réalisée grâce à SVDPACKC (BERRY, DO, KRISHNA et VARADHAN 1993), un programme C fournissant les r plus grandes valeurs singulières ainsi que les vecteurs singuliers de grandes matrices ayant une faible densité. SVDPACKC a été développé à l'Université du Tenessee afin de calculer des approximations de rang élevé de matrices termes-documents dans des applications de recherche d'information et est disponible gratuitement.

3.5.2 Lissage

Un des problèmes inhérents à l'apprentissage statistique est la sous-représentation des données. Lorsque l'on recherche une distribution d'événements qui sont représentés dans des espaces en haute dimension, les données sont souvent insuffisantes pour estimer les paramètres recherchés. En particulier, il existe très peu de données supervisées dans le domaine de la désambiguïsation sémantique. Les corpus de mots étiquetés avec leur sens disponibles présentement sont de petite taille. En réalité, le nombre d'événements possibles n'est pas restreint, et ce malgré les contraintes grammaticales, lexicales, sémantiques et pragmatiques qui limitent néanmoins les contextes liés aux acceptions d'un mot. Les corpus restent incomplets afin de capter toutes les variations linguistiques. Les techniques de lissage

permettent d'estimer de manière grossière la probabilité d'un événement rare en le reliant à des événements plus simples et plus fréquents.

Lorsque l'on estime les paramètres d'un classificateur bayésien par maximum de vraisemblance, la plupart des mots obtiennent une probabilité nulle d'apparaître dans le contexte d'un mot ambigü pour un sens donné. Si un mot h a été vu dans le contexte pour certains sens mais qu'il a été vu très peu souvent au total, le choix basé sur la probabilité obtenue par MLE ne sera pas fiable. Les distributions obtenues par maximum de vraisemblance doivent donc être lissées afin que le modèle puisse discerner, parmi les sens possibles d'un mot, celui qui obtient une meilleure probabilité auprès des mots de contexte, même si certains des événements n'ont jamais été observés. L'objectif des techniques de lissage (CHEN et GOODMAN 1996) est donc de répartir la masse de probabilité sur tous les événements possibles à partir des événements qui ont été observés lors de l'entraînement.

Une façon très simple d'y parvenir est de faire l'hypothèse que chaque événement a été vu au moins une fois, indépendamment des fréquences réelles des événements dans le corpus d'entraînement. Cette technique, appelée *add-one smoothing* (LAPLACE 1814), consiste donc à ajouter 1 à la fréquence de chaque événement, observé ou non. Bien que cette technique soit très simple à implanter en pratique, elle a pour effet d'octroyer trop de masse de probabilité aux événements non observés lorsque la cardinalité de l'ensemble des événements possibles est grande. Une variante du lissage de Laplace consiste à additionner $0 < \lambda < 1$ au compte de chaque événement. Ces techniques sont appelées *discounting methods* puisqu'une partie du poids, qui était d'abord alloué aux événements observés, est "enlevée" pour être redistribuée à l'ensemble de tous les événements. Le lissage additif peut être vu comme l'interpolation entre la distribution des fréquences relatives dans le corpus d'entraînement et la distribution uniforme sur tous les événements. L'utilisation d'une distribution uniforme afin de répartir la masse de probabilité a cependant pour effet d'allouer autant de poids à des événements non plausibles qu'à des événements plausibles.

On peut également répartir la masse de probabilité selon une distribution sur des événements plus simples. Dans un modèle à repli (KATZ 1987), différents modèles sont consultés tour à tour. Les modèles sont ordonnés en fonction de leur niveau

de généralité sur les événements. L'idée est de consulter le modèle le plus précis possible. Si ce modèle ne peut donner une information suffisante à propos d'un événement, des modèles progressivement plus simples sont consultés. Ces techniques de lissage sont équivalentes à la combinaison de divers estimateurs statistiques en un seul modèle global. Elles sont généralement préférables lorsque les données sont peu denses.

Par exemple, si l'on cherche la probabilité d'un mot étant donné un sens $P(W|S)$, on peut lisser en utilisant la distribution des mots $P(W)$ dans le corpus d'entraînement. Ainsi, parmi les mots non observés avec un sens particulier dans les données d'entraînement, un mot plus fréquent recevra une probabilité plus grande qu'un mot moins fréquent dans le corpus. Les modèles CCS et CSCS ont été lissés avec un modèle à repli. Ainsi, la distribution conditionnelle des mots ambigüs pour un sens est lissée en "décomptant" une partie du poids sur les mots vus avec ce sens lors de l'entraînement et en répartissant ce poids sur la distribution des mots ambigüs dans le corpus. La probabilité d'un mot w étant donné un sens s est donc donnée par :

$$P(w|s) = \begin{cases} \frac{C(w|s)-D}{\sum_{w':C(w'|s)\neq 0} C(w'|s)} + \frac{\sum_{w':C(w'|s)\neq 0} DP(w)}{\sum_{w':C(w'|s)\neq 0} C(w'|s)} & \text{si } C(w|s) \neq 0 \\ \frac{\sum_{w':C(w'|s)\neq 0} DP(w)}{\sum_{w':C(w'|s)\neq 0} C(w'|s)} & \text{sinon} \end{cases}$$

$C(w|s)$ dénote la fréquence relative du mot w pour le sens s et D est une constante. La distribution des mots de contexte en fontion d'un sens est lissée de façon similaire. La distribution des sens dans le corpus est lissée en redistribuant le poids sur l'uniforme.

CHAPITRE 4

RÉSULTATS

Ce chapitre décrit l'évaluation des modèles proposés dans cette étude. Les corpus et métriques utilisés à cette fin sont d'abord présentés. Les résultats pour nos modèles ainsi que pour les modèles de référence sont par la suite rapportés. Le chapitre se termine par une analyse critique des résultats.

4.1 Corpus et métriques d'évaluation

Nous utilisons deux sources importantes de données annotées sémantiquement. L'acquisition des modèles de désambiguïsation est accomplie à l'aide du corpus Semcor, alors que leur évaluation et la comparaison avec d'autres modèles sont basées sur l'outil Senseval. Les prochains paragraphes sont dédiés à la description de ces ressources, ainsi qu'aux métriques d'évaluation de la performance d'un modèle de désambiguïsation, la précision et le rappel.

Le corpus Semcor (MILLER, CHODOROW, LANDES, LEACOCK et THOMAS 1994) est un sous-ensemble du corpus Brown consitué de 235 000 occurrences de mots annotés sémantiquement. Les sens utilisés correspondent aux synsets de WordNet. Environ 24 000 mots différents et 33 000 sens différents sont représentés,

ce qui est peu comparativement aux 90 000 mots de vocabulaire et 100 000 synsets de WordNet. Environ 135 000 occurences sont utilisées pour l'entraînement ; l'ensemble de validation est de taille 50 000. Le modèle est testé sur les 25 000 mots restant.

Senseval (KILGARRIFF 1998) est un outil d'évaluation de systèmes de désambiguïsation sémantique et a été développé dans le but de forcer un concensus sur les aspects critiques de la WSD. L'existence d'un tel outil a également permis à la communauté de la WSD de se regrouper et d'unir les efforts pour le développement de ressources communes. Bien que les récentes versions de Senseval incluent des systèmes d'évaluation pour plusieurs langues, seules les tâches pour la langue anglaise ont été utilisées dans le cadre de cette étude. On distingue dans Senseval deux niveaux de désambiguïsation : un premier situé sur l'ensemble des 1082 mots à désambiguïser ("all words") et un deuxième sur un échantillon de 92 mots ("lexical sample").

La performance du modèle est évaluée par la précision et le rappel. La précision mesure la proportion de sens corrects retournés par le modèle par rapport au nombre de mots désambiguïsés par ce dernier.

$$\text{Précision} = \frac{\#\text{sens corrects}}{\#\text{sens retournés}}$$

Le rappel mesure la quantité d'information pertinente extraite par le modèle et est en quelque sorte un indice de la portée de ce dernier. Il est donné par la proportion de sens corrects retournés par le modèle sur le nombre total de mots à désambiguïser dans la tâche.

$$\text{Rappel} = \frac{\#\text{sens corrects}}{\#\text{sens à retourner}}$$

Étant donné que, dans nos expériences, le modèle retourne un sens pour chaque mot à désambiguïser, la précision et le rappel sont équivalents. Ainsi, nous rapporterons la performance des modèles de désambiguïsation par la précision seulement.

4.2 Résultats

Les prochains paragraphes décrivent les résultats obtenus par les modèles de référence et les modèles proposés dans cette étude sur les corpus Semcor et Senseval.

4.2.1 Modèles de référence

Une heuristique très simple et ne nécessitant pas d'entraînement consiste à utiliser uniquement l'information contenue dans l'ontologie dérivée de WordNet. Pour un mot donné, le premier sens répertorié dans la base de données est, parmi ses sens, celui qui désigne la signification la plus courante de ce mot. Nous avons également estimé cette même distribution des sens et des mots sur le corpus de Semcor. Le sens le plus fréquent d'un mot est ainsi donné par :

$$s^* = \arg\max_s P(s|w)$$

Le tableau 4.1 résume la performance de ces deux estimations du sens le plus fréquent d'un mot sur le corpus de test de Semcor et celui de Senseval. La précision est donnée sous forme de pourcentage. Nous constatons que l'estimation obtenue à l'aide du corpus Semcor obtient de meilleurs résultats que l'estimation donnée par WordNet.

Puisque ces deux modèles n'utilisent pas les mots de contexte, nous comparerons plutôt nos algorithmes au modèle Naive Bayes. Le sens attribué à un mot par le classificateur bayésien est donné par :

$$s^* = \arg\max_{s \in \mathcal{S}_w} P(w|s)P(s) \prod_{h \in \mathcal{H}_w} P(h|s)$$

Nous avons étudié l'influence de la taille de la fenêtre de mots de contexte, de la technique de lissage employée ainsi que l'effet de la pondération géométrique des mots de contexte sur la précision du classificateur. Des expériences pour des fenêtres de taille 1 à 20 pour le modèle Naive Bayes lissé par Laplace et par repli ont été menées. Les résultats se retrouvant en annexe, nous ne présentons ici que les points saillants. Le tableau 4.2 résume la précision du modèle Naive Bayes sur l'ensemble de validation en fonction de la taille de la fenêtre de mots de contexte

Tableau 4.1 – Précision sur les ensembles de test de Semcor et de Senseval (all-words) des modèles sans utilisation des mots de contexte.

Modèle	Semcor	Senseval (all-words)
Premier sens de WordNet	52.71	49.5
Sens le plus fréquent	71.53	61.1

Tableau 4.2 – Précision du modèle Naive Bayes sur l'ensemble de validation de Semcor en fonction de la taille de la fenêtre (n) et de la technique de lissage utilisée (Laplace et à repli).

n	Laplace	Repli
1	72.18	70.78
5	71.93	66.81
10	71.65	66.75
15	71.34	66.87
20	71.2	67.25

et du lissage.

Nous avons également étudié la possibilité d'utiliser la pondération géométrique des probabilités des mots de contexte. Puisque l'hypothèse d'indépendance conditionnelle entre les mots de contexte n'est pas raisonnable, le simple produit des probabilités alloue trop de poids au contexte par rapport au mot ambigü. Ainsi, nous proposons de pondérer la probabilité de chaque mot de contexte dans le produit en effectuant une moyenne géométrique des contributions des poids :

$$s^* = \arg\max_{s \in \mathcal{S}_w} P(w|s)P(s) \prod_{h \in \mathcal{H}_w} (P(h|s))^{1/2n}$$

Le tableau 4.3 étudie l'effet de la pondération géométrique des mots de contexte. La précision a été mesurée sur l'ensemble de validation de Semcor. Le lissage à repli obtient de meilleurs résultats que le lissage de Laplace. On remarque que plus la taille de la fenêtre des mots de contexte est grande, plus la précision se stabilise. De plus, on constate que la pondération est préférable et qu'elle nous permet d'exploiter une plus grande fenêtre de mots de contexte.

Tableau 4.3 – Précision sur l'ensemble de validation de Semcor du modèle Naive Bayes en employant la pondération géométrique des probabilités des mots de contexte. La précision est rapportée en fonction de la taille de la fenêtre (n) et de la technique de lissage utilisée (Laplace et à repli).

n	Laplace	Repli
1	71.89	72.09
5	71.75	73.02
10	71.87	73.27
15	71.91	73.28
20	71.88	73.26

4.2.2 Modèles de désambiguïsation avec concepts implicites

Les algorithmes CCS et CSCS ont été entraînés et testés sur les mêmes corpus que Naive Bayes, c'est-à-dire Semcor et Senseval. Les modèles ont été lissés par un modèle à repli. La pondération géométrique des mots de contexte a également été appliquée.

CCS

Nous rappelons d'abord que le modèle CCS (dont la contribution des mots de contexte est pondérée géométriquement) assigne à une occurence d'un mot le sens le plus probable s^* donné par l'équation ci-bas :

$$s^* = \arg\max_{s \in \mathcal{S}_w} P(w|s)P(s) \prod_{h \in \mathcal{H}_w} (\sum_{c \in C} P(t|c)P(c|s))^{1/2n}$$

La figure 4.1 démontre le spectre des valeurs singulières de la matrice de probabilité conditionnelle des mots de contexte étant donné un sens (modèle CCS). On voit que les quarante premières valeurs singulières sont grandes, alors que les suivantes sont presques constantes et petites. Ces dernières sont considérées comme du bruit dans les données et donc une approximation raisonnable à la matrice initiale ne comprendrait que les quarante ou quatre-vingts premières valeurs singulières et vecteurs singuliers.

La précision du modè CCS selon le rang de l'approximation par l'algorithme

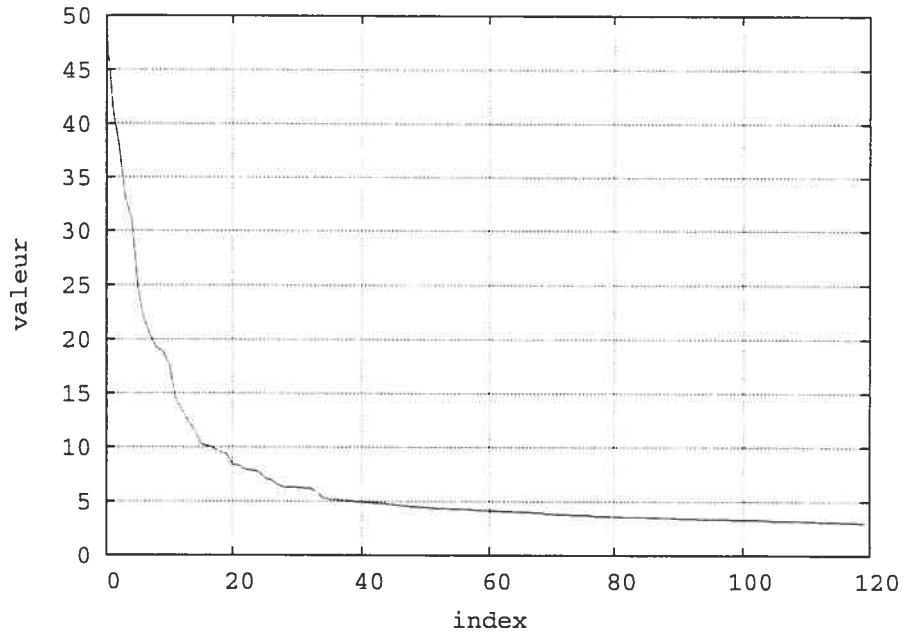

Figure 4.1 – Valeurs singulières de la matrice des mots de contexte et des sens. Les valeurs sont indexées en ordre décroissant.

SVD est rapportée dans le tableau 4.4. La taille de la fenêtre des mots de contexte était de 5. Les résultats pour les autres tailles sont en annexe. Les résultats obtenus sont semblables à ceux du modèle Naive Bayes sur la même tâche, mais la performance est toujours un peu plus basse.

Nous avons également calculé la log-probabilité moyenne des modèles avec la variable de concept et du même modèle sans cette variable. La log-probabilité moyenne pour le modèle sans la factorisation de la matrice de co-occurrence des mots de contexte et des sens est donnée par :

$$\frac{\sum_{t=1}^{T}\sum_{i=1}^{2n}\log P(h_{t,i}|s_t)}{2*T*n}$$

Tableau 4.4 – Précision sur l'ensemble de validation de Semcor du modèle CCS en fonction du nombre de concepts et de la méthode d'initialisation des paramètres (aléatoire, par décomposition en valeurs singulières avec la translation et avec la softmax). La taille de la fenêtre des mots de contexte est de 5.

	Nombre de concepts						
	2	5	10	20	40	80	100
Aléatoire	71.2	71.19	71.20	71.20	71.19	71.26	71.33
Translation	71.29	71.26	71.21	71.22	71.21	71.22	71.23
Softmax	71.26	71.24	71.25	71.24	71.24	71.24	71.22

Tableau 4.5 – Log-probabilité moyenne sur l'ensemble de validation de Semcor du modèle CCS en fonction de la méthode d'initialisation des paramètres (aléatoire, par décomposition en valeurs singulières avec la translation et avec la softmax). La log-probabilité pour le modèle sans la factorisation est également rapportée. La taille de la fenêtre des mots de contexte est de 5 et le nombre de concepts est 5.

Modèle	Log-probabilité
Sans factorisation	-9.88
Aléatoire	-8.77
Translation	-8.73
Softmax	-8.74

alors que la log-probabilité moyenne du modèle CCS avec la factorisation est donnée par :

$$\frac{\sum_{t=1}^{T}\sum_{i=1}^{2n}\log\sum_{c\in C}P(h_{t,i}|c)P(c|s_t)}{2*T*n}$$

La log-probabilité moyenne sur l'ensemble de validation est donnée dans le tableau 4.5. La taille de la fenêtre des mots de contexte était de 5. Les résultats pour les autres tailles sont en annexe. Nous constatons que la vraisemblance est plus grande avec les paramètres des modèles CCS et CSCS qu'avec le modèle Naive Bayes. Les probabilités des mots de contexte conditionnelles au sens du mot ambigü $P(h|s)$ obtenues par MLE sont plus petites que celles estimées par le modèle CCS, c'est-à-dire $\sum_{c\in C}P(h|c)P(c|s)$. Ainsi, la factorisation et l'estimation des pa-

Tableau 4.6 – Précision sur l'ensemble de validation de Semcor du modèle CSCS en fonction du nombre de concepts et de la méthode d'initialisation des paramètres (aléatoire, par décomposition en valeurs singulières avec la translation et avec la softmax). La taille de la fenêtre des mots de contexte est de 5.

	Nombre de concepts				
	2	5	10	20	40
Aléatoire	71.21	71.19	71.24	71.29	71.23
Translation	71.25	71.19	71.23	71.24	71.23
Softmax	71.23	71.21	71.23	71.22	71.22

ramètres par l'algorithme EM permettent de donner une probabilité plus grande aux bonnes associations de mots de contexte et de sens.

CSCS

Le sens assigné à un mot ambigü par le modèle CSCS, dont la contribution des mots de contexte est pondérée géométriquement, est donné par l'équation suivante :

$$s^* = \arg\max_{s \in \mathcal{S}_w} P(w|s)P(s) \prod_{h \in \mathcal{H}_w} (\sum_{r \in \mathcal{S}_h} P(h|r) \sum_{c \in \mathcal{C}} P(r|c)P(c|s))^{1/2n}$$

La figure 4.2 démontre le spectre des valeurs singulières de la matrice de probabilité conditionnelle des sens des mots de contexte étant donné un sens (modèle CSCS).

La précision du modèle CSCS selon le rang de l'approximation par l'algorithme SVD est rapportée dans le tableau 4.6. La taille de la fenêtre des mots de contexte était de 5. Les résultats pour les autres tailles sont en annexe. À nouveau, on remarque que la précision est plus basse que celle de Naive Bayes.

La log-probabilité moyenne pour le modèle sans la factorisation de la matrice de co-occurrence des sens est donnée par :

$$\frac{\sum_{t=1}^{T} \sum_{i=1}^{2n} \log P(r_{t,i}|s_t)}{2 * T * n}$$

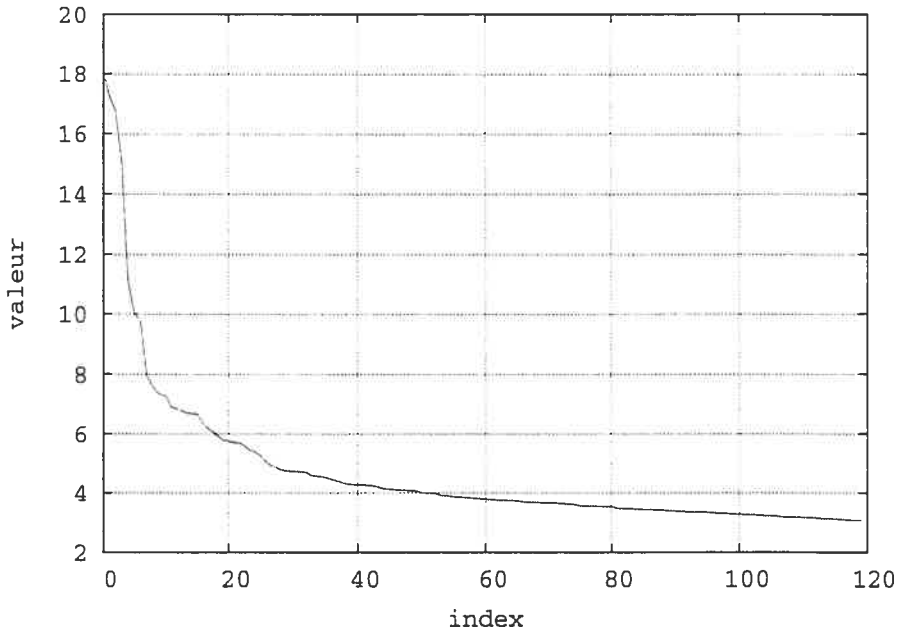

Figure 4.2 – Valeurs singulières de la matrice des sens des mots de contexte et des sens. Les valeurs sont indexées en ordre décroissant.

alors que la log-probabilité moyenne du modèle CSCS avec la factorisation est donnée par :

$$\frac{\sum_{t=1}^{T}\sum_{i=1}^{2n}\log\sum_{c\in C}P(r_{t,i}|c)P(c|s_t)}{2*T*n}$$

La log-probabilité moyenne sur l'ensemble de validation est donnée dans le tableau 4.7. La taille de la fenêtre des mots de contexte était de 5. Les résultats pour les autres tailles sont en annexe. À nouveau, nous constatons que la log-probabilité est en moyenne plus grande avec les modèles utilisant la factorisation de la matrice de co-occurrence des sens.

Tableau 4.7 – Log-probabilité moyenne sur l'ensemble de validation de Semcor du modèle CSCS en fonction de la méthode d'initialisation des paramètres (aléatoire, par décomposition en valeurs singulières avec la translation et avec la softmax). La log-probabilité pour le modèle sans la factorisation de la matrice de co-occurrence des sens est également rapportée. La taille de la fenêtre des mots de contexte est de 5 et le nombre de concepts est 5.

Modèle	Log-probabilité
Sans factorisation	-10.41
Aléatoire	-9.05
Translation	-9.05
Softmax	-9.02

Tableau 4.8 – Précision des modèles Naive Bayes, CCS et CSCS sur les ensembles de test de Semcor et de Senseval (all-words). Le modèle Naive Bayes utilise une fenêtre de 15 mots de contexte alors que les modèles CCS et CSCS ont été testés avec une fenêtre de mots de contexte de taille 5. Les paramètres des modèles CCS et CSCS ont été initialiés par la décomposition en valeurs singulières suivie d'une translation. Les modèles utilisent 2 concepts.

Modèle	Semcor	Senseval all-words
Naive Bayes	73.73	62.40
CCS	71.48	60.90
CSCS	71.42	61.20

4.2.3 Test

La précision du modèle Naive Bayes avec 15 mots de contexte sur les ensembles de test de Semcor et de Senseval est résumée dans le tableau 4.8. Les résultats sur ces mêmes ensembles des modèles CCS et CSCS avec 2 concepts et 5 mots de contexte s'y trouvent également. Nous utilisons le lissage par repli et la pondération géométrique des mots de contexte pour tous les modèles.

4.3 Analyse des résultats

Les résultats obtenus par les modèles avec concepts implicites sont comparables à ceux des modèles de référence dans la plupart des expériences effectuées. Dans nos modèles, la probabilité donnée par un mot de contexte au sens du mot cible est plus grande que celle obtenue sans réduction de la dimension. Cependant, nous ne constatons pas d'amélioration de la précision par rapport au modèle Naive Bayes.

Les prochaines sections tentent d'expliquer ces faibles résultats. Quelques critiques sont dirigées à l'endroit de certaines composantes fondamentales de nos modèles ainsi que vers les ressources qui sont utilisées par ces derniers. En particulier, la quantité de données étiquetées disponible est probablement insuffisante.

4.3.1 La réduction de la dimension en WSD

Les résultats obtenus par les modèles suggèrent que la SVD parvient à extraire des aspects thématiques des données d'entraînement. Cependant, le succès obtenu par nos modèles de désambiguïsation utilisant une réduction de la dimension n'est pas comparable au succès de leurs équivalents en recherche d'information, la LSA et la PLSA. La PLSA (HOFMANN 1999) se base sur l'algorithme EM afin de dériver les paramètres de l'espace sémantique latent pour représenter les mots et les sens. Cette technique améliore la pertinence des documents sélectionnés pour une requête. Les résultats obtenus sont meilleurs qu'avec l'unique emploi de la décomposition en valeurs singulières.

Les corpus de mots étiquetés avec leur sens selon le contexte ne semblent pas dégager plus d'information que celle déjà saisie par des modèles simples comme le classificateur Naive Bayes. De plus, il semble que l'ajout de variables cachées ne permet pas de capter des relations plus abstraites entre les mots et les sens. Il apparaît que le problème de la désambiguïsation sémantique se situe à un niveau plus fin que celui de la recherche de l'information dans des textes, où la dimension de l'espace des paramètres est plus petite. Quelques milliers de documents seulement constituent habituellement l'ensemble des documents de référence, alors que le nombre de sens dans WordNet est près d'une centaine de milliers.

Une objection à l'application de la décomposition en valeurs singulières à des

problèmes de traitement automatique du langage est que la SVD, comme toutes les méthodes de moindres carrés, est conçue pour des données normalement distribuées. La distribution normale n'est pas adaptée pour modéliser des données de co-occurrence comme celle des mots et des sens dans des textes. De plus, étant donné que la construction de la matrice de rang inférieur à partir de la matrice de comptes est basée sur une distribution normale, la matrice obtenue par l'algorithme SVD peut contenir des valeurs négatives, qui sont clairement des approximations inappropriées pour des fréquences relatives. Les transformations appliquées afin d'obtenir une interprétation probabiliste des poids résultent probablement en une perte de l'information véhiculée par les vecteurs singuliers.

4.3.2 La classification et la vraisemblance

Le taux de précision obtenu par les modèles CCS et CSCS sur Semcor et Senseval sont comparables à ceux obtenus par le modèle Naive Bayes. La performance de nos modèles en termes de classification est légèrement en-deça de celle du classificateur bayésien mais meilleure que celle du sens le plus fréquent. La vraisemblance des données d'entraînement est cependant plus élevée avec les modèles CCS et CSCS qu'avec Naive Bayes. La courbe d'apprentissage de EM démontre que l'algorithme parvient effectivement à augmenter la probabilité des bonnes étiquettes sémantiques des mots dans le corpus d'entraînement. La classification d'un nouvel exemple n'en est cependant pas meilleure. L'emploi de l'algorithme EM dans nos modèle permet l'optimisation de la vraisemblance des données d'entraînement mais ne mène pas à une meilleure discrimination des acceptions d'un mot ambigü dans son contexte.

Ceci amène une importante remarque : les modèles génératifs comme le classificateur Naive Bayes ne parviennent pas à discriminer la bonne classe parmi les autres classes. Ces modèles estiment la probabilité conditionnelle d'appartenance à une classe C, $P(Exemple|C)$, et sont entraînés à expliquer le mieux possible des données séquentielles. Les modèles discriminants, pour leur part, estiment directement la probabilité postérieure $P(C|Exemple)$. Ces derniers sont peut-être plus appropriés pour des tâches de classification comme la désambiguïsation sémantique.

4.3.3 Le manque de ressources

Il semble que la désambiguïsation sémantique est beaucoup plus complexe que ce que les ressources disponibles nous permettent d'en saisir. Le problème est décrit par des centaines de milliers de paramètres et les corpus Semcor et Senseval ne suffisent pas à l'apprentissage de toutes les variations linguistiques. Les corpus annotés sémantiquement sont très limités et contiennent une quantité dérisoire de données pour entraîner un modèle statistique. De ce point de vue, le lissage est certainement une composante essentielle dans nos modèle. Le poids est réparti selon des distributions elles-mêmes dérivées des données d'entraînement, mais aucune méthode de lissage ne permet à ce jour de répartir le poids selon le degré de corrélation entre un mot de contexte et un sens.

La formulation de nos modèles inclut en outre la caractérisation des sens de WordNet, dont la granularité est excessivement fine. Il est souvent très difficile de saisir les nuances sémantiques décrites dans la base de données. Certains sens sont pratiquement indifférenciables et les distinctions sont trop subtiles. Les sens étant plus "fins", ils sont également plus nombreux. Semcor comporte environ deux fois plus d'exemples au total que le nombre de sens répertoriés dans WordNet. Le nombre d'exemples de mots de contexte reliés à un sens donné est donc très limité dans Semcor. Ces exemples, trop peu nombreux, sont clairement peu représentatifs de l'ensemble de tous les mots reliés à une acception d'un mot particulière se retrouvant dans le contexte.

Le fait de reposer entièrement sur WordNet est également discutable. La caractérisation des sens dans WordNet en est une parmi toutes les caractérisations possibles. La variabilité des jugements sur les contextes d'acception d'un mot est très grande, et les sens de WordNet sont trop précis et fins pour être utiles dans des applications concrètes de recherche d'information ou de traduction automatique. De plus, WordNet ne fournit pas d'information quant aux restrictions sémantiques et pragmatiques à l'oeuvre pour chacun des sens d'un mot. Des déséquilibres sont également présents, empêchant une définition uniforme des synsets et de leur degré de généralité. On remarque ainsi que la hiérarchie de la branche nominale est beaucoup plus développée que celle des verbes.

4.3.4 Le contexte

Un autre problème auquel font face nos modèles est celui du rôle du contexte en désambiguïsation sémantique. Comme nous l'avons brièvement indiqué plus tôt, les classificateurs bayésiens ne tiennent pas compte de la position des mots de contexte par rapport au mot ambigü. La seule information de contexte qui est considérée par les modèles que nous avons présentés est celle de la présence ou de l'absence d'un mot dans une fenêtre d'une certaine taille entourant le mot cible. L'hypothèse "sac de mots" qui est bien souvent posée en WSD limite l'utilisation de l'information du contexte et n'est probablement pas adaptée à la complexité du problème. Le rôle des mots de contexte dans la discrimination du sens n'est pas encore bien modélisé et reste un problème ouvert.

CHAPITRE 5

CONCLUSION

Les études portant sur la désambiguïsation sémantique ont une histoire aussi longue que celle de l'intelligence artificielle et du traitement automatique du langage en général, mais il semble que peu de progrès ait été réalisé depuis les débuts de la recherche. Certains algorithmes sophistiqués obtiennent des résultats prometteurs mais ne parviennent à désambiguïser qu'un ensemble très restreint de mots et ce bien souvent en encourant de grands coûts en espace et en temps de calcul. Il appert qu'une désambiguïsation efficace peut être obtenue dans le cadre d'applications portant sur des tâches très précises, mais qu'un modèle de désambiguïsation ayant une large portée semble difficile à obtenir. Les approches de base comme le classificateur bayésien sont encore parmi les plus performantes et la plupart des tentatives pour l'améliorer se sont avérées infructueuses. Les conclusions de cette étude s'inscrivent dans la même direction puisque nos efforts afin de modéliser les dépendances entre les mots et les sens en ajoutant des thématiques d'un niveau plus abstrait s'avèrent moins profitables à la désambiguïsation que les mêmes modèles dénudés de ces caractéristiques.

5.1 Synthèse des contributions

Cette étude nous aura permis l'exploration d'un cadre pour la modélisation des liens entre les mots et les sens où la capacité de représentation est augmentée grâce à l'ajout de variables cachées. Ces variables supplémentaires offraient la possibilité d'extraire des caractéristiques sémantiques de haut niveau à partir des données d'entraînement et permettaient la représentation des mots et des sens de façon compacte. L'emploi de la décomposition en valeurs singulières afin d'extraire cette représentation thématique à partir du corpus présentait l'avantage de lisser implicitement les distributions de probabilité des mots et des sens et d'éliminer en partie le bruit présent dans les données d'entraînement. La contribution de cette étude est également d'avoir appliqué des techniques de réduction de la dimension en désambiguïsation sémantique. Ces mêmes méthodes ont connu un certain succès dans des problèmes similaires, notamment en recherche d'information. Cette étude peut donc inspirer des recherches similaires futures et ouvrir la voie à l'application d'algorithmes de réduction de la dimension dans d'autres problèmes de traitement automatique du langage.

Un constat intéressant se dégageant de cette étude porte sur le problème de l'utilisation de modèles génératifs dans un problème de classification comme la WSD. Nous avons démontré que la vraisemblance des données était plus grande sous nos modèles que sous un simple Naive Bayes, mais que ce dernier obtenait une meilleure performance en termes de classification. Cette observation suggère que des modèles discriminants, comme le boosting, seraient plus appropriés dans des tâches de classification sémantique. Ce travail aura également permis d'observer que l'utilisation de dictionnaires comme WordNet amène souvent une caractérisation trop précise des sens et que cette finesse cause des problèmes combinatoires importants. De plus, l'utilisation des dictionnaires comme WordNet ne semble appropriée à des problèmes concrets puisque les dictionnaires ne sont pas conçus pour représenter les acceptions des mots selon les contextes dans lesquels ils sont utilisés en pratique (IDE et VÉRONIS 1998).

Le rôle du contexte, la manière de combiner différents types d'information, l'utilisation des dictionnaires et l'introduction de propriétés discriminantes dans le cadre de la désambiguïsation sémantique sont autant de problèmes ouverts. Les

critiques émises à l'endroit des composantes fondamentales des modèles de désambiguïsation, comme l'utilisation qui est faite des ressources et du contexte, peuvent aider à mieux saisir la complexité du problème de la désambiguïsation sémantique. Les recherches futures bénéficieraient certainement d'une meilleure compréhension des travaux dans le domaine de la linguistique computationelle et de la sémantique.

CHAPITRE 6

ANNEXE

Cette annexe présente les résultats des expériences qui ont été effectuées dans le cadre de cette étude. Les premiers tableaux étudient l'effet de la pondération géométrique des mots de contexte et de la technique de lissage dans le modèle Naive Bayes. La précision et la log-probabilité des modèles CCS et CSCS est ensuite présentée en fontion de l'initialisation des paramètres, soit aléatoire, par la décomposition en valeurs singulières suivie d'une translation et par la décomposition en valeurs singulières suivie d'une softmax. Dans les expériences, les modèles utilisent la pondération géométrique des probabilités des mots de contexte et sont lissés par repli.

6.1 Modèle Naive Bayes

Le tableau 6.1 relate la précision du modèle Naive Bayes sans pondération des mots de contexte sur l'ensemble de validation de Semcor. La précision est rapportée en fonction de la taille de la fenêtre et de la technique de lissage utilisée. Le tableau 6.2 présente la précision avec la pondération géométrique dans les mêmes conditions.

Tableau 6.1 – Précision du modèle Naive Bayes sur l'ensemble de validation de Semcor en fonction de la taille de la fenêtre (n) et de la technique de lissage utilisée (Laplace et à repli).

n	Laplace	Repli
1	72.18	70.78
2	72.33	68.31
3	72.23	66.85
4	72.03	66.78
5	71.93	66.81
6	71.88	66.84
7	71.89	66.76
8	71.86	66.74
9	71.77	66.85
10	71.65	66.75
11	71.52	66.54
12	71.54	66.65
13	71.53	66.81
14	71.45	66.81
15	71.34	66.87
16	71.36	67.01
17	71.31	67.22
18	71.32	67.2
19	71.27	67.2
20	71.2	67.25

Tableau 6.2 – Précision sur l'ensemble de validation de Semcor du modèle Naive Bayes en employant la pondération géométrique des probabilités des mots de contexte. La précision est rapportée en fonction de la taille de la fenêtre (n) et de la technique de lissage utilisée (Laplace et à repli).

n	Laplace	Repli
1	71.89	72.09
2	71.85	72.67
3	71.86	72.86
4	71.81	73.00
5	71.75	73.02
6	71.78	73.04
7	71.84	73.20
8	71.88	73.24
9	71.87	73.22
10	71.87	73.27
11	71.88	73.23
12	71.93	73.23
13	71.90	73.29
14	71.90	73.30
15	71.91	73.28
16	71.87	73.29
17	71.90	73.33
18	71.87	73.29
19	71.90	73.30
20	71.88	73.26

Tableau 6.3 – Précision sur l'ensemble de validation de Semcor du modèle CCS avec initialisation aléatoire des paramètres en fonction du nombre de concepts et de la taille de la fenêtre des mots de contexte (n).

	Nombre de concepts						
n	2	5	10	20	40	80	100
1	71.13	71.07	71.13	71.02	71.19	71.22	71.32
5	71.2	71.19	71.2	71.2	71.19	71.26	71.33
10	71.29	71.22	71.29	71.21	71.29	71.18	71.33

Tableau 6.4 – Précision sur l'ensemble de validation de Semcor du modèle CCS avec initialisation des paramètres par la décomposition en valeurs singulières et la translation. La précision est rapportée en fonction du nombre de concepts et de la taille de la fenêtre des mots de contexte (n).

	Nombre de concepts						
n	2	5	10	20	40	80	100
1	71.21	71.29	71.28	71.27	71.31	71.25	71.25
5	71.29	71.26	71.21	71.22	71.21	71.22	71.23
10	71.23	71.24	71.24	71.23	71.25	71.21	71.22

6.2 Modèle CCS

Les tableaux 6.3, 6.4 et 6.5 rapportent la précision du modèle CCS pour les méthodes d'initialisation aléatoire et par la décomposition en valeurs singulières. Les taux de précision sont donnés en fonction du nombre de concepts dans le modèle et la taille de la fenêtre de mots de contexte. Les modèles sont lissés par repli et utilisent la pondération géométrique des mots de contexte.

Les tableaux 6.7, 6.8 et 6.9 rapportent la log-probabilité moyenne sur l'ensemble de validation du modèle CCS pour les méthodes d'initialisation aléatoire et par la décomposition en valeurs singulières. Le tableau 6.6 donne la log-probabilité moyenne du modèle sans la variable de concept, c'est-à-dire $P(h|s)$. Les modèles sont lissés par repli et utilisent la pondération géométrique des mots de contexte.

Tableau 6.5 – Précision sur l'ensemble de validation de Semcor du modèle CCS avec initialisation des paramètres par la décomposition en valeurs singulières et la softmax. La précision est rapportée en fonction du nombre de concepts et de la taille de la fenêtre des mots de contexte (n).

	Nombre de concepts						
n	2	5	10	20	40	80	100
1	71.22	71.23	71.22	71.22	71.23	71.2	71.19
5	71.26	71.24	71.25	71.24	71.24	71.24	71.22
10	71.23	71.25	71.24	71.25	71.23	71.24	71.23

Tableau 6.6 – Log-probabilité moyenne sur l'ensemble de validation de Semcor du modèle n'utilisant pas la factorisation de la matrice de co-occurrence des mots de contexte et des sens. La log-probabilité est rapportée en fonction de la taille de la fenêtre (n).

n	Log-probabilité
1	-10.12
5	-9.88
10	-9.74

Tableau 6.7 – Log-probabilité moyenne sur l'ensemble de validation de Semcor du modèle CCS avec initialisation aléatoire des paramètres. La log-probabilité est rapportée en fonction du nombre de concepts et de la taille de la fenêtre des mots de contexte (n).

	Nombre de concepts						
n	2	5	10	20	40	80	100
1	-8.99	-8.99	-8.99	-8.97	-8.95	-8.92	-8.91
5	-8.77	-8.77	-8.76	-8.75	-8.73	-8.71	-8.71
10	-8.69	-8.68	-8.68	-8.67	-8.66	-8.51	-8.63

Tableau 6.8 – Log-probabilité moyenne sur l'ensemble de validation de Semcor du modèle CCS avec initialisation des paramètres par la décomposition en valeurs singulières et la translation. La log-probabilité est rapportée en fonction du nombre de concepts et de la taille de la fenêtre des mots de contexte (n).

	Nombre de concepts						
n	2	5	10	20	40	80	100
1	-8.94	-8.88	-8.86	-8.85	-8.85	-8.89	-8.88
5	-8.75	-8.73	-8.72	-8.7	-8.69	-8.69	-8.69
10	-8.68	-8.65	-8.64	-8.63	-8.62	-8.61	-8.61

Tableau 6.9 – Log-probabilité moyenne sur l'ensemble de validation de Semcor du modèle CCS avec initialisation des paramètres par la décomposition en valeurs singulières et la softmax. La log-probabilité est rapportée en fonction du nombre de concepts et de la taille de la fenêtre des mots de contexte (n).

	Nombre de concepts						
n	2	5	10	20	40	80	100
1	-8.97	-8.94	-8.93	-8.92	-8.92	-8.93	-8.93
5	-8.76	-8.74	-8.72	-8.71	-8.7	-8.69	-8.69
10	-8.68	-8.66	-8.65	-8.63	-8.62	-8.61	-8.61

Tableau 6.10 – Précision sur l'ensemble de validation de Semcor du modèle CCS avec initialisation aléatoire des paramètres en fonction du nombre de concepts et de la taille de la fenêtre des mots de contexte (n).

	Nombre de concepts				
n	2	5	10	20	40
1	71.24	71.2	71.12	71.22	71.23
5	71.21	71.19	71.24	71.29	71.23
10	71.25	71.29	71.24	71.27	71.23

Tableau 6.11 – Précision sur l'ensemble de validation de Semcor du modèle CCS avec initialisation des paramètres par la décomposition en valeurs singulières et la translation. La précision est rapportée en fonction du nombre de concepts et de la taille de la fenêtre des mots de contexte (n).

	Nombre de concepts				
n	2	5	10	20	40
1	71.19	71.17	71.28	71.29	71.26
5	71.25	71.19	71.23	71.24	71.23
10	71.21	71.26	71.27	71.29	71.28

6.3 Modèle CSCS

Les tableaux 6.10, 6.11 et 6.12 rapportent la précision du modèle CSCS pour les méthodes d'initialisation aléatoire et par la décomposition en valeurs singulières. Les taux de précision sont donnés en fonction du nombre de concepts dans le modèle et la taille de la fenêtre de mots de contexte. Les modèles sont lissés par repli et utilisent la pondération géométrique des mots de contexte.

Les tableaux 6.14, 6.15 et 6.16 rapportent la log-probabilité moyenne sur l'ensemble de validation du modèle CCS pour les méthodes d'initialisation aléatoire et par la décomposition en valeurs singulières. Le tableau 6.13 donne la log-probabilité moyenne du modèle sans concepts, c'est-à-dire $P(r|s)$. Les modèles sont lissés par repli et utilisent la pondération géométrique des mots de contexte.

Tableau 6.12 – Précision sur l'ensemble de validation de Semcor du modèle CCS avec initialisation des paramètres par la décomposition en valeurs singulières et la softmax. La précision est rapportée en fonction du nombre de concepts et de la taille de la fenêtre des mots de contexte (n).

	Nombre de concepts				
n	2	5	10	20	40
1	71.22	71.23	71.21	71.2	71.16
5	71.23	71.21	71.23	71.22	71.22
10	71.24	71.26	71.26	71.25	71.24

Tableau 6.13 – Log-probabilité moyenne sur l'ensemble de validation de Semcor du modèle n'utilisant pas la factorisation de la matrice de co-occurrence des sens des mots de contexte et des sens du mot ambigü. La log-probabilité est rapportée en fonction de la taille de la fenêtre (n).

n	Log-probabilité
1	-10.42
5	-10.41
10	-10.33

Tableau 6.14 – Log-probabilité moyenne sur l'ensemble de validation de Semcor du modèle CSCS avec initialisation aléatoire des paramètres. La log-probabilité est rapportée en fonction du nombre de concepts et de la taille de la fenêtre des mots de contexte (n).

	Nombre de concepts				
n	2	5	10	20	40
1	-9.05	-9.05	-9.04	-9.02	-8.98
5	-9.07	-9.05	-9.04	-9.02	-9.01
10	-9.06	-9.04	-9.03	-9.01	-9.01

Tableau 6.15 – Log-probabilité moyenne sur l'ensemble de validation de Semcor du modèle CSCS avec initialisation des paramètres par la décomposition en valeurs singulières et la translation. La log-probabilité est rapportée en fonction du nombre de concepts et de la taille de la fenêtre des mots de contexte (n).

	Nombre de concepts				
n	2	5	10	20	40
1	-9.05	-9.05	-8.95	-8.94	-8.95
5	-9.07	-9.05	-8.99	-8.97	-8.95
10	-9.06	-9.04	-8.94	-8.96	-8.94

Tableau 6.16 – Log-probabilité moyenne sur l'ensemble de validation de Semcor du modèle CSCS avec initialisation des paramètres par la décomposition en valeurs singulières et la softmax. La log-probabilité est rapportée en fonction du nombre de concepts et de la taille de la fenêtre des mots de contexte (n).

	Nombre de concepts				
n	2	5	10	20	40
1	-9.03	-8.99	-8.96	-8.95	-8.95
5	-9.05	-9.02	-8.99	-8.97	-8.95
10	-9.05	-9.02	-8.99	-8.96	-8-95

Références

AHA, D. W., D. KIBLER et M. K. ALBERT (1991), « Instance-Based Learning Algorithms », *Mach. Learn. 6*(1), p. 37–66.

BARTHOLOMEW, D. J. (1987), *Latent Variable Models and Factor Analysis*, New York : Oxford University Press.

BENGIO, Y. (1999), « Markovian Models for Sequential Data », *Neural Computing Surveys 2*, p. 129–162.

BERRY, M., T. DO, G. O. V. KRISHNA et S. VARADHAN (1993), « SVDPACKC (Version 1.0) User's Guide ».

CHEN, S. F. et J. GOODMAN (1996), « An Empirical Study of Smoothing Techniques for Language Modeling », *Proceedings of the Thirty-Fourth Annual Meeting of the Association for Computational Linguistics*, San Francisco, Morgan Kaufmann Publishers, p. 310–318.

COVER, T. M. et P. E. HART (1967), « Nearest neighbor pattern classification », *IEEE, Transactions on Information Theory, IT-13 1*, p. 21–27.

DEERWESTER, S., S. DUMAIS, G. FURNAS, T. LANDAUER et R. HARSHMAN (1990), « Indexing by latent semantic analysis ».

DEMPSTER, A. P., N. M. LAIRD et D. B. RUBIN (1977), « Maximum-likelihood from incomplete data via the EM algorithm », *Journal of Royal Statistical Society B 39*, p. 1–38.

DUDA, R. O. et P. E. HART (1973), *Pattern Classification and Scene Analysis*, New York : Wiley.

ESCUDERO, G., L. MÀRQUEZ et G. RIGAU (2000), « Boosting applied to word

sense disambiguation », *Proceedings of ECML-00, 11th European Conference on Machine Learning*, Barcelona, ES, Springer Verlag, Heidelberg, DE, p. 129–141.

ESCUDERO, G., L. MÀRQUEZ et G. RIGAU (2000), « A Comparison between Supervised Learning Algorithms for Word Sense Disambiguation », *Proceedings of CoNLL-2000 and LLL-2000*, Lisbon, Portugal, p. 31–36.

FELLBAUM, C. (1998), *WordNet : An Electronic Lexical Database*. MIT Press.

GALE, W. A., K. W. CHURCH et D. YAROWSKY (1992a), « Estimating upper and lower bounds on the performance of word-sense disambiguation programs », *ACL 30*, p. 249–256.

GALE, W. A., K. W. CHURCH et D. YAROWSKY (1992b), « A method for disambiguating word senses in a large corpus », *Computers and the Humanities 26*, p. 415–439.

GOLUB, G. H. et C. F. V. LOAN (1991), *Matrix Computation*. John Hopkins University Press, Second Edition.

GONZALO, J., F. VERDEJO, I. CHUGUR et J. CIGARRAN (1998), « Indexing with WordNet synsets can improve Text Retrieval », *Proceedings of the COLING/ACL '98 Workshop on Usage of WordNet for NLP*, Montreal, Canada, p. 38–44.

HOFMANN, T. (1999), « Probabilistic Latent Semantic Analysis », *Proceedings of the 15th Annual Conference on Uncertainty in Artificial Intelligence (UAI-99)*, San Francisco, CA, Morgan Kaufmann Publishers, p. 289–296.

IDE, N. et J. VÉRONIS (1998), « Word Sense Disambiguation : The State of the Art », *Computational Linguistics 24*, p. 1–40.

JOLLIFFE, I. (1986), *Principal Component Analysis*, New York : Springer-Verlag.

JORDAN, M. I. (1998), *Learning in graphical models*. MIT Press.

KATZ, S. (1987), « Estimation of probabilities from sparse data for the language model component of a speech recognizer », *IEEE Transactions on Acoustic, Speech and Signal Processing vol. 35*(num 3), p. 400–401.

KILGARRIFF, A. (1998), « SENSEVAL : An Exercise in Evaluating Word Sense

Disambiguation Programs », *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, Granada, Spain, p. 581–588.

LAPLACE, P. S. M. D. (1814), *Essai philosophique sur les probabilités*, Paris : Mme. Ve. Courcier.

LESK, M. E. (1986), « Automatic Sense Disambiguation Using Machine Readable Dictionaries : How to Tell a Pine Cone from an Ice Cream Cone », *Proceedings of the 1986 SIGDOC Conference*, p. 24–26.

MANNING, C. et H. SCHÜTZE (1999), *Foundations of Statistical Natural Language Processing*, Cambridge : MIT Press.

MILLER, G., M. CHODOROW, S. LANDES, C. LEACOCK et R. THOMAS (1994, March), « Using a Semantic Concordance for sense Identification », *Proceedings of the ARPA Human Language Technology Workshop*, Plainsboro, NJ, p. 232–235.

MOONEY, R. (1996, May), « Comparative Experiments on Disambiguating Word Senses : An Illustration of the Role of Bias in Machine Learning », *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, p. 82–91.

NG, H. T. et H. B. LEE (1996), « Integrating multiple knowledge sources to disambiguate word sense : an exemplar-based approach », *Proceedings of the 34th conference on Association for Computational Linguistics*, Association for Computational Linguistics, p. 40–47.

PEDERSEN, T. et R. BRUCE (1998, July), « Knowledge Lean Word Sense Disambiguation », *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, Madison, WI, p. 800–805.

RIVEST, R. L. (1987), « Learning Decision Lists », *Mach. Learn.* 2(3), p. 229–246.

SANDERSON, M. (1997), « Word Sense Disambiguation and Information Retrieval ».

SCHUTZE, H. (1998), « Automatic Word Sense Discrimination », *Computational Linguistics 24*(1), p. 97–123.

STOKOE, C., M. P. OAKES et J. TAIT (2003), « Word sense disambiguation in information retrieval revisited », *Proceedings of the 26th annual interna-*

tional ACM SIGIR conference on Research and development in informaion retrieval, ACM Press, p. 159–166.

WILKS, Y. et M. STEVENSON (1997), « Sense tagging : Semantic tagging with a lexicon ».

WILKS, Y. A., B. M. SLATOR et L. M. GUTHRIE (1996), *Electric words : dictionaries, computers, and meanings*. MIT Press.

YAROWSKY, D. (1992), « Word-sense disambiguation using Statistical models of Roget's categories trained on large corpora », *COLING 14*, p. 454–460.

YAROWSKY, D. (1994), « Decision Lists for Lexical Ambiguity Resolution : Application to Accent Restoration in Spanish and French », *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics (ACL-94)*, p. 88–95.

YAROWSKY, D. (1995), « Unsupervised Word Sense Disambiguation Rivaling Supervised Methods », *Meeting of the Association for Computational Linguistics*, p. 189–196.

YAROWSKY, D. (2000), « Hierarchical decision lists for word sense disambiguation ».