Université Paris 7 – DIDEROT

Laboratoire d'Ingénierie de la
Connaissance Multimédia et Multilingue

Désambiguïsation automatique

à partir d'espaces vectoriels multiples clutérisés

Rapport intermédiaire

Réalisé par :

Myriam RAKHO

Encadré par :

Guillaume PITEL

Claire MOUTON

Juin 2008

1 INTRODUCTION

La désambiguïsation sémantique des mots est une tâche intermédiaire fondamentale pour la plupart des applications de traitement automatique du langage telles que la traduction automatique, la recherche d'information, l'acquisition automatique de connaissances, la compréhension automatique, l'interaction homme-machine, le traitement de la parole, etc.

D'une manière générale, la désambiguïsation sémantique des mots consiste à associer une occurrence donnée d'un mot ambigu avec l'un des sens de ce mot. La résolution de ce problème s'effectue en deux étapes :

1. la *discrimination des différents sens du mot,* en regroupant les termes similaires, puis
2. l'*étiquetage sémantique* de chacune de ses occurrences.

La première sous-tâche a pour but de regrouper les différentes occurrences d'un mot en classes représentant chacune un *sens*, en décidant, pour deux occurrences données, si elles appartiennent au même sens ou non. Les méthodes supervisées de désambiguïsation sémantique se basent, à cette étape, sur une liste de sens pré-définie. Apparaît alors le problème de la définition du sens qui, malgré les nombreux débats qu'il a suscités dans la communauté, n'est toujours pas résolu. En effet, les ressources lexicales électroniques (dictionnaires, thesauri, etc.) telles que WordNet, du fait de leur disponibilité, ont été, un temps, très utilisées en désambiguïsation sémantique. Mais les approches basées sur ce type de ressources présentent trois principaux inconvénients. D'une part, le degré de granularité des sens requis dépend de l'application finale à laquelle sera intégré le module de désambiguïsation sémantique. Il n'existe pas de correspondance exacte entre un type d'application donné et le degré de granularité requis. Par exemple, le mot anglais *mouse* possède deux sens principaux (animal et périphérique), mais se traduit en français par *souris* dans les deux cas. Ce mot requérra une granularité moins importante pour la traduction automatique que pour la recherche d'information, où la distinction des sens sera nécessaire. En revanche, la distinction des sens pour cette dernière tâche ne sera pas nécessaire pour le mot *river* (français : *fleuve* ou *rivière*). D'autre part, il est souvent difficile de décider s'il faut distinguer deux emplois d'un même mot pour en faire deux sens différents. Selon [Pustejovsky 1995], un mot peut avoir une infinité de sens selon le contexte. Le sens des mots n'est donc pas fixé une fois pour toutes, ce qui signifie qu'on pourrait découper un mot en un nombre indéfini d'acceptions, souvent très variable selon les individus. Par conséquent, en désambiguïsation sémantique, le seuil de fusion ou de séparation des sens (seuil de similarité) varie d'une méthode à une autre. Enfin, parmi les théories du sens, deux courants principaux s'opposent. L'idée, ancienne, selon laquelle les mots correspondent à des objets et des concepts spécifiques (Aristote) et qui est la base des méthodes de construction des ressources

lexicales traditionnelles, a été remplacée au 20e siècle par une conception purement linguistique du sens. Dans un premier temps, *le sens d'un mot est défini exclusivement par la moyenne de ses usages linguistiques* (Saussure, Meillet, Hjemslev, Martinet, etc.). Puis, Wittgenstein rejette même l'idée de 'signification' pour ne garder que la notion d''usage' : *Don't look for the meaning, but for the use*. Et les théories les plus récentes ont des vues similaires sur le sens : [Bloomfield 1933] et [Harris 1954] le considèrent comme une fonction de distribution, [Barwise et Perry 1953], eux, le voient comme une abstraction du rôle que le mot joue systématiquement dans le discours (c'est la théorie de la *Situation Semantics*).

La seconde sous-tâche, l'étiquetage sémantique, assigne un sens à chacune des classes créées précédemment. Deux sources d'information sont requises pour cela : (i) le contexte du mot à désambiguïser (linguistique et/ou extra-linguistique), et (ii) selon le type d'approche, des connaissances issues de ressources linguistiques externes (lexicales, encyclopédiques, etc.) (approches supervisées, « basées sur des connaissances ») ou des informations sur les contextes d'instances du mot dans un corpus auparavant désambiguïsées (approches supervisées, « basées sur corpus »). Le principe général de cette étape est de comparer le contexte de l'instance à désambiguïser avec l'une de ces sources d'information, dans le but d'assigner un sens à chaque occurrence du mot.

Le coût de construction des ressources linguistiques manuelles, l'inadéquation de ces dernières dûe à la divergence entre les sens qu'elles représentent et les sens pragmatiques des mots, le haut degré de finesse dans la distinction des sens sont autant de limites à leur utilisation dans une tâche de désambiguïsation sémantique. Et le récent regain d'intérêt pour la linguistique de corpus (création et stockage de corpus de textes toujours plus larges) et, par conséquent, pour les méthodes statistiques, ne pouvaient que mener à l'apparition d'une nouvelle tendance dans le domaine de la désambiguïsation sémantique entre autres : les méthodes « basées sur corpus », avec deux orientations principales. D'une part, les approches supervisées, qui utilisent des corpus d'entraînement annotés. Mais la rareté de tels corpus et, une fois de plus, leur coût de construction, ont poussé les chercheurs du domaine à faire appel à des approches non supervisées, dans lesquelles les informations nécessaires à la désambiguïsation sont tirées de corpus non annotés par des méthodes de classification des sens. Des systèmes hybrides, à l'image de celui que nous nous proposons de construire, sont donc apparus, combinant plusieurs sources d'information (fréquence des mots, informations lexicales, syntaxiques, contextuelles, etc.). La détermination des sens des mots, première étape de tels systèmes, est primordiale pour la qualité de leurs résultats. Deux solutions ont été proposées : (i) l'enrichissement automatique des réseaux de type WordNet pour y introduire les informations permettant de répondre aux critiques formulées à leur encontre ([Agirre et Lopez de Lacalle 2003] [Mihalcea et Moldovan 2001]); (ii) l'extraction des sens des mots automatiquement à partir de corpus, sans utiliser de dictionnaires existants (désambiguïsation sémantique *automatique*) ([Schütze 1998]).

La désambiguïsation sémantique automatique se base ainsi sur les théories du sens les plus récentes, et en particulier sur l'Hypothèse Distributionnelle [Harris 1954], pour induire les sens des différentes occurrences d'un mot ambigu à partir de la similarité de leurs contextes dans un corpus

donné.

> *Hypothèse Distributionnelle :*
> Words that occur in the same contexts tend to have similar meanings.

Le sens recherché est donc un usage du mot, en contexte, plutôt que sa signification littérale.

Les trois grandes tendances de la désambiguïsation sémantique automatique sont :

- (Pantel & Lin 2002) : l'objectif premier de cette approche est de rassembler les mots en classes d'équivalence et donc plutôt de former des classes de synonymes. La découverte de sens est une conséquence indirecte : la méthode de classification utilisée, *Clustering by Committee*, autorisant l'appartenance d'un mot à plusieurs classes, chacune d'entre elles devient de facto un sens de ce mot.
- (Schütze 1998) (Pedersen et Bruce 1997) et (Purandare 2003) : cette seconde tendance caractérise chaque occurrence d'un mot par un ensemble de traits liés à son environnement plus ou moins proche et procède à une classification non supervisée de toutes les occurrences du mot sur la base de ces traits. Les différentes classes formées constituent autant de sens du mot.

- (Véronis 2003) (Dorow et Widdows 2003) (Rapp 2003) (Ferret 2004) : ces approches prennent comme point de départ les co-occurrents d'un mot enregistrés à partir d'un corpus et forment les différents sens de ce mot en regroupant ses co-occurrents suivant leur similarité ou au contraire leur dissimilarité.

2 PRINCIPALES PROBLÉMATIQUES ET APPROCHES EXISTANTES

2.1 Désambiguïsation

2.1.1 Problématique

L'ambiguïté inhérente aux langues naturelles est un problème récurrent dans le domaine du Traitement Automatique du Langage. On peut, en effet, rencontrer différents types d'ambiguïté en fonction du niveau d'analyse linguistique où l'on se situe : au niveau syntaxique, du fait des différentes manières possibles d'agencer ceux-ci dans une même langue (catégories syntaxiques, problèmes de rattachements), au niveau sémantique, avec les différents types d'ambiguïtés lexicales dues aux différents sens des mots (homonymies, polysémie, etc.), etc. S'ajoute à cela le fait qu'une langue peut faire l'objet de différents types d'usages, avec des conséquences importantes sur la manière de gérer les informations.

Dans son étude sur ces phénomènes d'ambiguïté, [Fuchs 2000] appelle « virtuelles » (par opposition aux ambiguïtés « réelles », qui subsistent pour l'humain) les ambiguïtés rencontrées par un système de traitement automatique des langues qui peuvent être levées par une analyse linguistique complète de leur contexte d'apparition, à l'aide de seules connaissances de la langue puisqu'un tel système n'a pas accès à autant d'informations que l'humain (connaissances extra-linguistiques). L'Analyse Sémanitque Latente [Landauer, T. et Dumais, S. 1997] (Latent Semantic Analysis ou LSA), produit, à partir d'un corpus, une représentation des mots de la langue sous forme d'espaces vectoriels sémantiques. On peut en extraire des informations de proximité (ou similarité) entre mots et les utiliser en désambiguïsation, en étudiant les regroupements de mots à proximité du mot à désambiguïser. Toutefois, ces espaces vectoriels présentent l'inconvénient d'être bruités du fait des ambiguïtés telles que la polysémie. Le propos de notre travail est d'améliorer la qualité du corpus à partir duquel est construit l'espace vectoriel sémantique en réduisant la polysémie des termes, et de mesurer l'impact d'une telle désambiguïsation sur la qualité des espaces de mot (voir section 2.2.1 sur la notion d'espace de mot).

Les espaces de mot servant à décrire la sémantique des unités lexicales à désambiguïser sont construits à partir de différents types d'informations sur leur contexte (lexicales, syntaxiques, sémantiques, etc.). Deux questions primordiales se posent alors, qu'il convient de résoudre avant tout. La première est la question du type de contexte, et par là, de la quantité des informations nécessaires à la désambiguïsation sémantique (voir section 2.2.1.1 *Deux types de contextes*), la seconde est celle du mode de représentation des données qui permet d'expliciter au mieux les informations les plus pertinentes pour la détermination du sens des mots (voir section 2.2.3.1 *Différentes méthodes de clustérisation*).

2.1.2 Deux méthodes de désambiguïsation

Parmi les nombreuses méthodes de désambiguïsation existantes, deux ont attiré notre attention :

- celle de [Schütze 1998], une méthode de désambiguïsation automatique et non-supervisée basée sur les résultats d'une clustérisation des données et qui prend en compte les co-occurrences du second ordre dans la représentation du contexte des mots, et
- celle de [Lesk 1986], une méthode supervisée qui calcule la similarité entre mots sur la base du chevauchement de leurs définitions respectives dans un dictionnaire électronique.

2.1.2.1. La méthode de Schütze

- **Principes**

[Schütze 1998] est la description d'un travail réalisé dans le cadre d'une application de recherche d'information. Dans ces travaux les réponses aux requêtes des utilisateurs sont calculées d'après le sens des mots de la requête, non d'après les mots eux-mêmes. Et puisque la mesure de similarité des documents est un processus interne au système, il n'est pas nécessaire de faire référence à des sens définis. On peut donc se contenter de la discrimination des sens des mots, sans se soucier de la seconde partir d'une tâche de désambiguïsation sémantique, à savoir l'assignation d'un sens aux mots (ce qui fait la particularité de ce travail).

Dans cet article, l'auteur présente l'algorithme CGD (Context-Group Discrimination), un algorithme de désambiguïsation sémantique automatique et non-supervisée basé sur les résultats d'une clustérisation des données. Pour cet algorithme, un cluster est une liste d'occurrences d'un mot similaires contextuellement, au second ordre. En effet, deux de ses occurrences sont assignées à un même cluster si les mots avec lesquels elles co-occurrent (leurs voisins) occurrent à leur tour avec des mots de contextes similaires dans un corpus d'entraînement. Cela permet d'obtenir des représentations moins éparses et plus robustes. L'auteur propose d'améliorer les performances de cet algorithme en intégrant des informations syntaxiques dans la représentation du contexte des mots et donc dans le calcul de la similarité entre mots.

- **L'algorithme CGD**

Dans CGD, les objets (mots, contextes et clusters) sont représentés dans un espace vectoriel du type Word Space (les dimensions de l'espace sont les mots).

Les ***vecteurs de mots*** contiennent les fréquences de co-occurrence entre un mot donné et tous ses voisins dans un corpus d'entraînement. Ils constituent l'espace sémantique vectoriel (*Word Space*, WS dorénavant) de départ de l'algorithme. Deux concepts proches étant le plus souvent exprimés par des ensembles de mots similaires, la similarité entre mots se mesure à la proportion de ressemblance entre les vecteurs des mots, en l'occurrence, c'est la mesure Cosinus (coefficient de corrélation normalisé) qui est utilisée.

Les contextes sont représentés par des ***vecteurs de contexte*** : le centroïde des vecteurs des mots qu'ils contiennent. Il s'agit en fait de la somme de ces derniers, ce qui résout le problème de la dispersion des données et permet d'obtenir des vecteurs de contexte denses. Dans ce calcul, les vecteurs de mots sont pondérés par leur potentiel discriminant. Les mots les plus fréquents sont des « discriminants faibles», par opposition aux « discriminants forts », qui ont une distribution plus solide (plusieurs d'occurrences dans un court intervalle) et donc une fréquence d'occurrence dans les documents qui est faible par rapport à leur fréquence absolue.

Enfin, les ***vecteurs de sens*** (groupes de contextes similaires) sont calculés : l'ensemble des vecteurs de contexte représentant les contextes d'apparition d'un mot ambigu donné dans le corpus d'entraînement est clustérisé à l'aide de l'algorithme Buckshot de [Cutting et al. 1992] (une méthode hybride entre EM et la clustérisation agglomérative). Un sens est alors représenté par le centroïd de son cluster.

L'inconvénient de l'algorithme EM de clustérisation (Expectation Maximization) est qu'il fournit des résultats satisfaisants au plan local uniquement, ce qui pose problème si les paramètres initiaux (déterminés aléatoirement) sont mauvais. L'auteur a donc choisi d'appliquer, dans un premier temps, l'algorithme GAAC (Group-Average Clustering Algorithm) sur un échantillon aléatoire de l'ensemble des vecteurs de contexte à clustériser. Les centroïdes des clusters ainsi calculés sont alors

pris comme paramètres initiaux de la première itération de EM (cinq itérations sont effectuées).

La qualité de la clustérisation dépend également de la représentation des vecteurs de contexte. L'espace multidimensionnel est donc réduit (à cent dimensions en l'occurrence) suivant la méthode SVD (Singular Value Decomposition, [Golub et van Loan 1989]), une méthode qui, comme LSA (Latent Semantic Analysis), permet de trouver les principaux axes de variation, c'est-à-dire les dimensions les plus pertinentes, dans WS. Les vecteurs de contexte sont alors représentés par leurs valeurs dans ces dimensions, et c'est à partir de ces dernières que les clusters sont calculés.

La désambiguïsation d'une occurrence d'un mot donné peut alors être réalisée par application de l'algorithme CGD à l'aide des vecteurs de mots et des vecteurs de contexte créés par l'étape précédente. Le cluster dont le vecteur de sens est le plus proche du vecteur de contexte représentant l'occurrence en question lui est assigné.

Application de CGD

L'algorithme CGD sélectionne le vecteur de sens (groupe de contextes) dont le centroïd est le plus proche du vecteur de contexte de l'occurrence à désambiguïser. Pour une occurrence *t* d'un terme ambigu *v* :

1. On compare *c*, le vecteur de contexte de *t*, avec le vecteur de contexte du mot *v* dans WS,
2. puis, on calcule tous les vecteurs de sens s_j de *v* (phase de clustérisation),
3. enfin, on assigne *t* au sens *j* dont le vecteur de sens s_j est le plus proche de *c*.

2.1.2.2. La méthode de Lesk

La seconde méthode, présentée dans [Lesk 1986], est une méthode de désambiguïsation automatique supervisée dont le but est de discriminer les sens des mots polysémiques à l'aide d'un dictionnaire électronique, en l'occurrence, le *Oxford Advanced Learner's Dictionary of Current English*. Le principe de base de cette méthode est de mesurer le chevauchement entre les différentes définitions, dans le dictionnaire, d'un mot ambigu et les définitions de ses voisins immédiats, dans une fenêtre de 10 mots. Par exemple, l'une des définitions proposées pour le mot anglais *ash* est :

ash
1 the solid residue left when combustible material is thoroughly burnd or is oxidized by chemical means fine particles of mineral matter from a volcanic vent.

Supposons que le mot qui le précède dans le texte est *coal*, dont la définition complète est :

coal
1 a piece of glowing carbon or charred wood **:** ember
2 charcoal

```
3 a black of brownish black solid combustible substance formed by the
partial decomposition of vegetable matter without free access of air
and under the influence of moisture and often increased pressure
and temperature that is widely used as a natural fuel pieces or a
quantity of the fuel broken up for burning
coal
1 to burn charcoal : char
2 to supply with coal
0 to take in coal
```

La définition de *ash* précédemment citée contient trois termes qui sont également utilisés dans celle du mot *coal*. D'autres termes de la définition de *coal* (*wood*, par exemple) apparaissent dans celle de *ash* mais dans des proportions moindres. La définition retenue pour une occurrence du mot *coal* dans l'expression *coal ash* sera donc celle citée plus haut.

Le caractère non-syntaxique de la méthode, et donc son éventuelle utilisation comme supplément d'une méthode de désambiguïsation par la syntaxe, est présenté par l'auteur comme son premier avantage. La combinaison des deux types de contextes (les voisins et les co-occurrents syntaxiques) est effectivement plus intéressante, comme nous l'avons vu auparavant. Le second atout de cette méthode est son indépendance par rapport à l'information tirée de contextes plus larges (par exemple, le fait qu'un mot apparaisse souvent dans le voisinage du mot ambigu relativement au nombre total de ses occurrences dans le corpus).

Les performances d'un tel système reposent avant tout sur le choix du dictionnaire utilisé, pour lequel la principale caractéristique à prendre en compte serait le volume d'informations fournies pour chaque définition, c'est-à-dire la longueur des entrées, la fréquence de chevauchement entre les définitions étant corrélée au nombre de termes utilisés pour les décrire. L'auteur pose d'ailleurs la question (sans y répondre) de savoir si cette fréquence doit, ou non, être pondérée par la longueur des entrées.

L'intérêt de cette méthode pour notre programme serait d'en utiliser une variante non-supervisée qui se baserait non pas sur un dictionnaire traditionnel mais sur les résultats du programme de construction des sens de [Ferret 2004]. Nous inclurions dans ce dernier nos informations de co-occurrences syntaxiques.

2.2 Espaces de mots

2.2.1 Principes

2.2.1.1. Les informations représentées dans les espaces de mots

- **Deux types de contextes**

On distingue en général deux principaux types d'information qui nécessitent chacun une taille de

contexte différente. Le **contexte thématique** (également appelé « sac de mots », *bag of words,* ou « ensemble de mots », *set of words*) représenté par les ***voisins*** du mot ambigu sans considération des relations qu'ils entretiennent avec lui ([Schütze 1998]), est extrait d'une fenêtre plus large que le **contexte relationnel**, défini en termes de relations entre le mot ambigu et ses ***cooccurrents*** (co-texte lexical, relations syntaxiques, cadre de sous-catégorisation, rôles sémantiques, collocations, distance au mot, etc.), et pour lequel l'unité maximale considérée est la phrase ([Yarowsky 1993], [Baili et al. Unsupervised learning of verb subcategorization frames], [Reifler 1995], [Jacquet polysémie verbale et construction syntaxique]). Selon [Yarowsky 1993], la sélection de cooccurrences ciblées par des relations syntaxiques (sujet-verbe, verbe-objet, adjectif-nom, nom-nom, etc.) peut être plus pertinente que la simple sélection des voisins du mot dans une fenêtre donnée. Selon [Goldberg 1995], en effet, l'identification sémantique des lexèmes doit être rapportée aux constructions dans lesquelles ils apparaissent (identification de leurs divers sens/emplois en contexte). L'hypothèse sous-jacente à cette théorie est qu'une construction syntaxique est porteuse d'un sens intrinsèque indépendamment des unités lexicales qui la composent. [Gross -verbes] prend l'exemple des verbes pour démontrer que, la plupart des prédicats étant polysémique, tout changement de sens d'un prédicat est corrélé à un changement de son schéma d'arguments, et donc que tous les éléments figurant dans une même position argumentale pour un sens déterminé appartiennent à la même classe d'objets. On peut, par exemple, faire ressortir les différents sens du verbe *jouer* à partir de ses différentes constructions argumentales : *Il joue de la trompette (pratiquer) ; Il joue avec son fils (s'amuser)* (exemple tiré de [Jacquet polysémie verbale et construction syntaxique]). Les informations obtenues à partir du contexte relationnel sont donc de nature pragmatique (les éléments étant définis essentiellement par l'ensemble des arguments qui leur sont appropriés, plutôt que par leurs traits sémantiques inhérents). Mais la syntaxe est insuffisante pour révéler, à elle seule, la structure sémantique des langues naturelles, car on associe rarement un sens à une construction syntaxique et une construction syntaxique à un sens (*jouer sur le canapé* vs. *jouer sur les mots* [Jacquet ????]). Il est donc nécessaire de considérer les constructions syntaxiques comme éléments du contexte qui contribuent en partie seulement au sens de l'unité lexicale à désambiguïser. Un troisième type d'information est parfois pris en compte : les informations sur le domaine, avec activation d'un mot uniquement s'il est pertinent avec le domaine du discours.

Le problème de la détermination de la taille de contexte optimale à utiliser a fait l'objet de plusieurs études ([Kaplan 1950], [Gougenhein and Michéa 1961][Yarowsky, 1993], [Choueka and Lusignan 1985], [Gale et al. 1993]). Il s'agit en fait de répondre à la question suivante : sachant que nous avons le mot cible et *n* mots avant et après lui, quelle valeur minimale de *n* permettra, au moins dans une fraction tolérable de cas, de mener au choix du sens correct du mot central ? La plupart des études réalisées sur le sujet démontre qu'il n'existe pas de taille optimale fixe qui soit adaptée à tous les mots : la valeur optimale de *n* varie en fonction du type d'ambiguïté et donc du type de contexte considéré (2 à 5 mots pour les ambiguïtés locales/grammaticales contre 20 à 50 mots pour les ambiguïtés thématiques ou sémantiques). [Crestan, El Bèze et De Loupy] proposent une méthode novatrice de sélection dynamique de la fenêtre optimale pour chaque phrase basée sur un système décisionnel probabiliste.

- **Informations de premier, deuxième et troisième ordre**

Les contextes thématique et relationnel, précédemment définis, permettent de capturer des informations de premier ordre sur la distribution des mots dans un texte. Mais l'irrégularité distributionnelle et sémantique des unités lexicales, l'essence même des langues naturelles, est à l'origine d'un problème commun à toutes les méthodes linguistiques non-supervisées : le caractère sporadique des informations extraites des corpus, avec pour conséquence une baisse des performances de l'application en termes de précision et donc de robustesse. Pour remédier à ce problème, des méthodes telles que [Grefenstette 1993], [Schütze 1998] et [Ferret 2004] ont choisi d'utiliser les informations de second ordre pour la construction des espaces de mots. La co-occurrence du second ordre est définie comme la cohésion entre les co-occurrents du mot basée sur leurs propres co-occurrents. Elle permet non seulement de capturer des similarités entre mots de même usage qui n'entretiennent pas forcément un lien de co-occurrence directe dans un corpus, mais qui occurrent dans des contextes similaires, mais aussi, pour les mots polysémiques, de détecter des dissimilarités entre les contextes d'usage de leurs diverses occurrences dans le corpus.

2.2.1.2. Représentation du sens dans les espaces vectoriels sémantiques

Dans un espace de mot, les divers sens d'un terme se distinguent par des valeurs différentes d'un certain nombre de paramètres (informations sur le contexte du mot, lexicales, syntaxiques, sémantiques, etc.). Chaque sens (ou emploi) du mot est donc représenté par une région de son espace sémantique, plus ou moins grande dans une dimension donnée, et les proximités de sens entre acceptions se traduisent dans l'espace par des relations de voisinage ou de recouvrement. La représentation en espaces vectoriels sémantiques est donc particulièrement adaptée pour rendre compte des phénomènes sémantiques tels que la polysémie car ils permettent de déterminer avec précision le sens de chaque acception d'un terme ambigu tout en conservant la notion de proximité, essentielle, selon l'approche continuiste du sens [Victorri et Fuchs 1996], dans la définition-même de la polysémie. La désambiguïsation consiste alors à étudier la position du vecteur représentant une acception donnée à désambiguïser dans l'espace sémantique du mot et à lui assigner le sens le plus proche.

2.2.2 La Semantic Map

Le modèle des cartes sémantiques (Semantic maps, SM, [Haspelmath 2003], [de Haan 2004]), également appelées « espaces sémantiques » est un outil de représentation linguistique des objets qui nous utilisons dans notre travail dans un but descriptif. La SM nous permet en effet de représenter les contextes thématique et relationnel des mots à désambiguïser, chacun nous permettant de découvrir des co-occurrences différentes et donc de saisir des informations différentes.

Dans notre modèle, trois tailles de fenêtre sont prises en compte pour la représentation du contexte thématique : cinq, dix et vingt mots avant et après le mot ambigu.

Pour la représentation du contexte relationnel, nous utilisons le système LIMA (Lic2m Multilingual

Analyzer), un ensemble d'outils d'analyse linguistique qui produit une analyse en dépendances des textes. Nous l'appliquons à notre corpus d'entraînement, un ensemble de quatre millions de pages extraites du Web. Nous obtenons ainsi l'ensemble des co-occurrents syntaxiques d'un mot donné, à partir duquel nous construisons, pour chaque relation syntaxique, une matrice de cooccurrences dont les valeurs sont l'Information Mutuelle (IM) entre le mot ambigu est ses cooccurrents.

A chaque construction syntaxique est donc associée une région de l'espace sémantique (une pour chaque relation syntaxique existante pour les co-occurrents, et une pour chaque taille de fenêtre pour les voisins) qui contient tous les sens compatibles avec cette construction. Chaque espace sémantique est ensuite clustérisé. Il s'agit alors de reconstituer les clusters à partir des sous-clusters obtenus pour chaque espace sémantique, ce qui nécessite la désambiguïsation de ces sous-clusters. Nous exploitons pour cela le contexte thématique (les voisins au second ordre) des éléments qui les composent.

2.2.3 Clustérisation et réduction de dimensions

2.2.3.1. La clustérisation : principes et méthodes

L'idée de base de la clustérisation est de regrouper ensembles des objets qui se ressemblent dans une ou plusieurs dimensions données, en l'occurrence des mots regroupés en classes de voisins/co-occurrents. En analyse des données, le but de la clustérisation est de créer un partitionnement d'un ensemble de données (mots, documents) en un ensemble de sous-classes pertinentes, appelées « clusters » (grappes), représentées par un centroïde (élément le plus représentatif ou moyenne de tout ou partie de leurs membres). Dans le cas de données à haute dimensionnalité, la clustérisation peut être utile pour la réduction des dimensions, par exemple. La désambiguïsation sémantique des mots consiste alors à comparer la représentation du mot avec chaque centroïde pour trouver le plus proche et assigner au mot le sens qu'il représente.

- **Différentes méthodes de clustérisation**

Le choix de la méthode de calcul de la similarité entre les objets du modèle (mots, documents) dépend du choix du modèle de représentation (espaces vectoriels, graphes, arbres de décision, etc.). Plusieurs méthodes ont été testées dans le domaine de la clustérisation, qu'on peut classer comme :

- **hiérarchiques** (*bottom-up*) : ces méthodes partent d'une partition totale des objets de la base de données (chaque objet est son propre cluster) et produisent une série de partitions (qualité supérieure de ces algorithmes) par fusion (agglomeratives) ou division (divisives) selon une mesure de similarité donnée. Leur résultat est un arbre de clusters appelé « dendogramme » et dont la racine est le cluster englobant (initial ou final selon la méthode). On fixe ensuite un seuil de similarité dans l'arbre au dessous duquel tous les composants connectés forment un cluster.
  exemples : AGNES, DIANA (Divisive ANAlysis)

- **partitionnelles** (*top-down*) : ces méthodes partent d'un cluster englobant pour produire une

partition unique des données par optimisation d'un certain critère pré-défini. Ces algorithmes effectuent une recherche combinatoire de toutes les clustérisations possibles afin de trouver la plus optimale (efficacité en temps pour des tailles de données importantes).
exemples : *K*-means, PAM, CLARA, CLARANS (Clustering Large Applications based on Randomized Search, [Ng et Han 1994])

Le point de départ de ces deux types de méthodes est un espace vectoriel sémantique de haute dimension construit à partir d'un corpus et dont la dimensionnalité est souvent réduite à l'aide de diverses techniques de décomposition en valeurs singulières (LSA, LSH, *Locality Sensitive Hashing*). De ceci découle leur principal inconvénient : la difficulté d'interprétation des dimensions de la représentation résultant de cette réduction, qui rend difficile la compréhension des clusters créés. De plus, les clusters pouvant exister dans les différents sous-espaces de l'espace vectoriel sémantique d'origine ne sont plus identifiables.

D'autres méthodes ont donc été récemment développées dans le but de résoudre ces problèmes. Parmi elles, dans le domaine de l'exploration de données (*data mining*) :

- les méthodes **par estimation de la densité**, empruntées à la statistique et dont l'idée de base de est que dans un cluster donné, le voisinage de chaque objet doit contenir au moins un certain nombre *MinPts* d'objets, i.e. la cardinalité du voisinage doit excéder le seuil *MinPts*. La distinction entre clusters et bruit est donc basée sur la densité des points : les zones de forte densité sont des clusters alors que les zones de faible densité sont du bruit.
  exemples : CLIQUE, DBSCAN (Density-Based Spatial Clustering of Applications with Noise, [Ester, Kriegel, Sander, Xu 1996]), RDBC (Recursive Density-Based Clustering Algorithm, [Su, Yang, Zhang, Xu et Hu 1999]), OPTICS (*Ordering Points to Identify the Clustering Structure,* [Ankerst, Markus, Breunig, Krieger et Sander, 1999]*)*

- **L'algorithme CBC (Clustering by Committee)**

Objets et représentations

Feature database. Pour représenter les mots et mesurer leur similarité, CBC utilise des vecteurs de traits (*feature vectors*, FV) qui contiennent une série de mesures les décrivant quantitativement (dans notre cas, les traits seraient les trois fenêtres de voisins et les contextes grammaticaux). Un trait est un triplet (*élément, trait, mot*). Par exemple, pour représenter nos relations syntaxiques, (*John, SUBJ_V, found*), qui signifie que *John* est sujet du verbe *found*. Les FV de tous les éléments sont enregistrées dans une première table de hachage, structure de donnée efficace pour l'insertion et la recherche d'éléments (temps constant). Une seconde table de hachage inversée est créée parallèlement, pour l'indexation des traits. Ces deux tables constituent la FD, qui sera ensuite utilisée pour accéder aux traits des éléments et à leurs mesures. Ce type de représentation s'est avéré efficace pour le traitement de quantité importantes d'éléments et de traits.

L'espace vectoriel des mots (les vecteurs de traits). La mesure utilisée pour calculer la valeur des traits dans les FV précédemment cités est l'Information Mutuelle (IM). Pour chaque mot, un vecteur de fréquences $C(e)$ de taille m (m le nombre de traits le décrivant) est construit, il contient la fréquence d'occurrence du mot dans chacun de ses traits. Puis un vecteur d'information mutuelle $MI(e) = (mi_{e1}, ..., mi_{em})$ est construit à partir de chaque $C(e)$, qui contient l'Information Mutuelle Point-à-point (IMP) entre le mot e et un trait f, mi_{ef}, définie ainsi :

$$mi_{ef} = (c_{ef} / c_{ef} + 1) * (\min (\Sigma^{n}_{i=1} c_{ei}, \Sigma^{m}_{j=1} c_{if}) / \min (\Sigma^{n}_{i=1} c_{ei}, \Sigma^{m}_{j=1} c_{if}) + 1)$$

C'est ce dernier vecteur qui sera utilisé pour les calculs de similarité entre mots.

La mesure similarité. L'évaluation de la similarité entre mots, $sim(e_i, e_j)$, est effectuée via le *Cosinus* (Salton et McGill 1983) entre leurs vecteurs d'IMP selon la formule suivante.

Déroulement de l'algorithme.

L'algorithme CBC de clustérisation est une combinaison des méthodes hiérarchique et partitionnelle. Les trois phases principales de l'algorithme sont :

1. *Recherche des k plus proches voisins :*
   - ENTRÉE :
     - la FD précédemment créée

   - Construction d'une matrice de similarité (table de hachage) qui enregistre, pour chaque mot, les scores des k (valeur prédéfinie) mots les plus similaires. Pour calculer les k plus proches voisins, on commence par trier les traits par ordre de PMI pour en extraire ceux dont l'information mutuelle est la plus élevée. Puis on calcule la Similarité par paires (SP) :

     - pour chaque objet e_i :
       - pour chacun de ses traits t_i :
         - on calcule la Similarité par Paire (SP) entre e_i et chacun des éléments e_j avec lesquels il partage le trait t_i

   Les traits à forte IM occurrent dans peu de mots, ce qui signifie qu'on effectue ce calcul sur une fraction seulement de toutes les combinaisons possible d'IP. Avec cette heuristique, les mots qui ne partagent que des traits à faible IM ne sont pas pris en compte par l'algorithme (sans que ceci affecte la qualité de la clustérisation, selon l'expérience décrite dans [Pantel 2003]).

   - SORTIE :
     La matrice de similarité : les mots et leurs k plus proches voisins

2. *Pour chaque élément, clustérisation de ses k plus proches voisins :*
   On effectue une clustérisation récursive des *k* plus proches voisins de chaque mot : à chaque étape, l'algorithme découvre un ensemble de clusters (les éléments de chaque cluster formant comité) et identifie les éléments résiduels (qui ne sont couverts par aucun comité déjà créé). Un élément est 'couvert' par un comité si la similarité entre l'élément et le centroïd du comité est supérieure à un seuil S_2 prédéfini. Par ailleurs, un comité est supprimé si sa similarité avec l'un des comités de créés auparavant est supérieure à un seuil S_1 prédéfini. Les valeurs de S_1 et S_2 sont déterminantes pour la qualité de la clustérisation.

   - ENTRÉE :
     - une liste *E* des éléments à clustériser,
     - une base de données de similarité (*k* plus proches voisins, phase I)
     - et deux seuils de similarité S_1 (seuil minimal de dissimilarité entre comités) et S_2 (seuil minimal de dissimilarité entre un élément et un comité).

   - CLUSTÉRISATION :
     - les objets de *E* sont clustérisés, puis, les comités sont calculés selon S_1, ainsi que les centroïds (la moyenne des membres du comité)
     - la liste *R* des résidus est à son tour clustérisée (clustérisation récursive)

   - SORTIE :
     L'union des comités découverts à chaque étape de la récursion, chacun représentant l'un des clusters de sortie de l'algorithme.

3. *Assignation des sens :*
   - ENTRÉE :
     La sortie de la phase précédente

   - Chaque élément e_i de *E* est assigné au cluster c_j le plus similaire.
   Dans sa version *soft-clustering* (possibilité d'assignation d'un mot à plusieurs clusters différents), CBC découvre les différents sens d'un mot en autorisant. La clé de cette version : lorsqu'un élément e_i est assigné à un cluster c_j, les traits intersectifs entre e_i et c_j sont supprimés de e_i, ce qui lui permet de découvrir les sens les moins fréquents d'un mot et lui évite de dupliquer les découvertes de sens.

   - SORTIE :
   Pour chaque mot, une liste de clusters : ses différents sens dans le corpus. Les éléments de chaque cluster sont triés par ordre descendant.

2.2.3.2. La réduction des dimensions : LSA et LSH

- **La LSA (Latent Semantic Analysis)**

La LSA ([Landauer et Dumais 1997]) est une méthode de représentation, sous la forme d'un espace sémantique de très grande dimension, du sens contextuel des mots à l'aide calculs statistiques sur un large corpus qui lui permettent d'inférer des relations profondes entre mots ou ensembles de mots. L'information de base utilisée par cette technique est la distribution des mots dans la somme de leurs contextes. L'idée sous-jacente est que la somme de tous les contextes d'apparition ou non d'un mot fournit un ensemble de contraintes mutuelles qui déterminent largement la similarité sémantique entre mots et ensembles de mots. L'expérience décrite dans [Deerwester et al.1990] est un exemple d'utilisation de cette technique pour l'analyse automatique du langage (indexation et extraction d'informations automatiques). Dans ce domaine, l'intérêt de cette technique est de permettre la construction automatique de connaissances sémantiques génériques (indépendantes du domaine) ([Bestgen 2004]).

Le point de départ de la LSA est une matrice de cooccurrences dont les dimensions sont les mots et leurs contextes d'apparition (fenêtres et contextes syntaxiques, dans notre cas), à laquelle on applique une décomposition en valeurs singulières (SVD, Singular Value Decomposition) qui produit une sorte de lissage des associations mot-à-mot. La matrice de cooccurrences est ainsi transformée en une matrice plus petite contenant la partie la plus pertinente de l'information contenues dans les cooccurrences initiales. Ceci permet de résoudre le problème de la disparité des fréquences de co-occurrence (probabilités nulles) entre mots entraînée par le fait que, même dans un grand corpus de textes, la plupart des mots sont relativement rares. Cela permet non seulement d'améliorer la complexité en temps (pour le calcul des distances ou des plus proches voisins) mais aussi en espace puisque la caractérisation d'un mot devient plus petite. Le positionnement des mots et de leurs sens, représentés par des vecteurs, dans l'espace sémantique ainsi obtenu permet toujours de mesurer leur proximité par le c*osinus*. Les clusters peuvent ensuite être construits à partir des vecteurs des mots proches dans l'espace.

Pour la construction de la matrice de cooccurrences initiale, plusieurs paramètres doivent être déterminés, tels que le type de contexte à prendre en compte, l'application de pré-traitements au corpus d'entraînement, comme la tokenisation, la lemmatisation des mots ambigus et/ou de leurs contextes, le filtrage des mots outils (déterminants, pronoms, etc.). A la suite des diverses expériences déjà menées, la lemmatisation s'est avérée inutile lorsque la taille de fenêtre étudiée est grande.

Les étapes de la LSA

1. *La matrice de cooccurrences :*
   Un tableau lexical M_{t*d} (termes * contextes) qui contient le nombre d'occurrences de chaque terme dans chaque document.

2. *Pondération des mots :*
   Les différentes occurrences des mots sont pondérées par une estimation de leur importance

dans leur contexte et par le degré d'information d'un mot sur les passages dans lesquels il apparaît, ce qui réduit l'impact des mots « peu informatifs » (à fréquence constante) :
- La valeur de chaque cellule de *M* (+1) est convertie en son *log*.
- l'entropie de chaque mot par rapport à tous ses contextes (toutes les cellules de sa ligne) est calculée (p *log* p), puis la valeur de chaque cellule de la ligne est divisée par l'entropie totale de la ligne.

3. *Réduction des dimensions par SVD :*
   La matrice *M* pondérée est décomposée en trois matrices :
   - T_{t*m} : matrice orthogonale gauche
   - D_{m*m} : matrice orthogonale droite
   - S_{m*d} : matrice diagonale des valeurs singulières

   (avec *d* le nombre de contextes/colonnes, *t* le nombre de mots/lignes et *m* le rang de *M*)
   Les valeurs singulières sont les racines des valeurs propres de M^*M (pour M^* le produit scalaire de *M*).
   Puis M est recomposée par factorisation de *T*, *D*, et *S* :
   $M = TDS$

4. *Emploi :*
   Calculs de proximité entre mots (représentés par leurs vecteurs) et clustérisation de l'espace vectoriel sémantique obtenu.

- **La LSH (Locality-Sensitive Hashing)**

La LSH est une technique de recherche des plus proches voisins. Selon cette technique, définie par [Indyk et Motwani 1998], la proximité entre deux éléments p et q est corrélée à leur probabilité de collision selon une fonction de hachage définie dans l'espace d'origine de ces éléments. Plus la distance entre p et q est grande, et plus la probabilité de collision est faible.

En pratique, pour les besoins des modèles d'espace de mots, la méthode est utilisée comme ceci :

- Tout d'abord, on définit un ensemble de K vecteurs aléatoires v_i qui vont découper notre espace en une série d'hyperplans.
- Pour un mot représenté par le vecteur v_M, on calcule sa signature composée de K bits selon la règle suivante :
  - bit(i) = { 1 si $\cos(v_M, v_i) > 0$,
    0 si $\cos(v_M, v_i) < 0$ }
- Selon la LSH la probabilité de collision de hachage entre deux points est directement proportionnelle à la distance entre ces points. Cela se traduit par le fait que le nombre de bits différant entre deux signatures est une approximation de l'angle entre les vecteurs à l'origine de ces signatures. On utilise donc la distance de Hamming sur les vecteurs hachés pour évaluer cet angle.

3 APPROCHE ÉVALUÉE

3.1 Présentation de l'approche théorique globale

Dans cette étude, nous proposons une approche hybride de la désambiguïsation automatique entre clustérisation et classification, non-supervisée puis supervisée. La première phrase de notre méthode consiste à clustériser automatiquement les mots de notre corpus à l'aide de leurs voisins et de leurs cooccurrents. Nous considérons ensuite les résultats de notre clustérisation comme une liste de classes, et donc comme données de base pour la classification de nouvelles instances de mots. Pour cette seconde partie, non encore réalisée, nous avons exploré plusieurs pistes (KNN, ANN, LSH, etc.).

La méthode KNN est une méthode de classification à partir de cas qui consiste à prendre des décisions (classer des objets) en recherchant un ou des cas similaires déjà résolus (notre liste de classes). Le but de la méthode est de construire un classifieur du type *Cl*: *O* -> *C*, qui affecte une classe *C* à un nouvel objet *O*. Les deux phases principales de la méthode sont :

- **Les deux étapes de l'algorithme KNN**

(1) ***l'apprentissage***, où il s'agit simplement de construire un modèle composé :

- d'une liste d'enregistrements (pour chaque mot) de la forme (*objet*, *classe*(*objet*)) construite à partir d'un échantillon d'apprentissage. Dans notre cas, ce dernier est le résultat de la clustérisation des cooccurrents du mot et les enregistrements du modèle sont de la forme (*cooccurrent/voisin*, *cluster*(*cooccurrent*)), un cooccurrent ou voisin et l'un des clusters/sens dans lesquels il co-occurre avec le mot.
- d'une fonction de distance (f_1),
- et d'une fonction de choix de la classe étant données les classes des *k* plus proches voisins du mot (f_2).

(2) ***la classification***, qui consiste à affecter une classe à une nouvelle instance.

- Dans un premier temps, l'algorithme identifie les k plus proches voisins (enregistrements) de o, la nouvelle instance à classer, selon f_1.
- Ensuite, il calcule $c(o)$, la classe correcte de o selon f_2.

- **Optimisations de la méthode KNN**

Dans [Kriegel, Pryakhin et Schubert 2005], les auteurs proposent une méthode de classification d'objets multi-représentés basée sur une version optimisée de l'algorithme KNN. L'inconvénient principal de cet algorithme de classification son efficacité en temps de sa phase de classification et la qualité de ses résultats sont proportionnels à la taille de la base de données d'entraînement. Les auteurs proposent donc de réduire cette dernière aux instances les plus représentatives : les objets de chaque classe sont clustérisés avec l'algorithme DBIR (Density-Based Instance Reduction), une méthode de clustérisation basée sur la densité, et seul le centroïd de chaque cluster découvert est inséré dans la nouvelle base de données d'entraînement réduite.

Déroulement de DBIR

ENTRÉE :

O, un espace de données d'entraînement composé de tuples du type $(o(r_1 * ... * r_m), c)$, avec o un objet de O, r une représentation de o, m le nombre de représentations, et c la classe de o.

1. Clustérisation des instances de chaque classe avec DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*). L'algorithme renvoie un ensemble de clusters $Clust = \{Clust_1, ..., Clust_l\}$, avec l le nombre de clusters créés, déterminé automatiquement par DBSCAN (voir *annexe 3* pour la description de cet algorithme);

2. Itération dans $Clust$ pour déterminer C_i, le centroïde de chaque $Clust_i$ avec DBIR;

3. Dans chaque représentation i :

dans chaque $Clust_i$:

tous les objets différents de C_i sont supprimés

tous les C_i sont insérés de i sont insérés dans une base de données d'entraînement Db_i

SORTIE :

Chaque représentation i est réduite à sa DB_i

Ainsi, dans chaque représentation, chaque classe ou concept est représenté par une unique instance

d'entraînement et les instances qui ne sont pas typiques d'une classe ne sont pas prises en compte. Par la suite, la classification d'un nouvel objet *o* est réalisée à l'aide de la méthode KNN.

3.2 Etapes préliminaires

3.2.1. Désambiguïsation manuelle

Pour l'évaluation de notre méthode de désambiguïsation, nous avons été menés à sélectionner neuf mots à caractère polysémique pour lesquels nous avons retenu deux à trois sens (*tableau 1*) :

- six proviennent de [Audibert 2002] (*compagnie*, *détention*, *observation*, *organe*, *solution* et *vol*),
- deux de [Sérichard 2004] (*barrage* et *lancement*) dont les sens sont créés automatiquement à l'aide de la méthode [Ferret 2004],
- et un (*formation*) avec une combinaison des sens proposés dans ces deux articles.

L'évaluation nécessitait un corpus d'instances des mots clustérisées manuellement. La première partie de mon travail était donc la constitution d'un corpus d'évaluation (*C1*) composé de phrases classées par sens pour chaque mot (voir l'*annexe 5* pour la DTD de *C1*). Ces phrases ont été extraites du corpus Europarl, ou du Web lorsqu'Europarl ne nous fournissait pas suffisamment d'exemples.

Mot	Sens	Définition	Nombre d'exemples
Vol	1	délit	20
	2	déplacement dans l'air	20
Compagnie	1	association de personnes	45
	2	présence de quelqu'un	44
Détention	1	incarcération, enfermement	43
	2	possession	37
Formation	1	instruction, études (formation initiale,continue, professionnelles, qualification, dispositif de formation	20
	2	formation de quelque chose	13
	3	groupe de personnes (parti, groupe, formation musicale)	20

Observation	1	surveillance, étude	20
	2	remarques, réflexions	20
	3	conformation à	20
Organe	1	groupe de personnes (politique, administratif)	20
	2	partie d'un corps	19
Solution	1	dénouement, réponse	20
	2	mélange liquide	18
Barrage	1	barrage hydraulique	19
	2	barrage routier de manifestation, blocage de la circulation	19
	3	barrage militaire, policier, de frontière	21
Lancement	1	lancement d'un produit boursier (OPA)	21
	2	lancement d'un produit média ou publicitaire (journal, émission, chaîne)	21
	3	lancement spatial, de missile ou de sous-marin	22

TABLEAU 1 – Les 9 mots de test et leurs sens

3.2.2. Clustérisation et désambiguïsation automatiques

Dans un deuxième temps, nous avons effectué une désambiguïsation automatique des phrases de *C1* basée sur les contextes syntaxiques. Pour cela, nous avons clustérisé manuellement les cooccurrents de chaque mot (voir l'*annexe 6* pour la DTD de *C2*, le résultat de la clustérisation manuelle) : à chaque cooccurrent syntaxique, nous avons assigné le(s) sens dans le(s)quel(s) il était probable qu'il occurre avec le mot traité. Puis, nous avons développé un programme (*BASE*) dont la fonction principale était de compter les cooccurrents syntaxiques, pour chaque phrase de *C1* et pour chaque sens du mot.

- *liste_coocc_synt_ph* : la table des cooccurrents syntaxiques :
  pour chaque mot m_i :
    pour chaque phrase p_j (*C1*) :
      tab_coocc_synt : liste des cooccurrents syntaxiques de m_i dans p_j

- *tab_ssm* : la table des scores par sens :
  pour chaque mot m_i :
    pour chaque phrase p_j (*C1*) :
      pour chacun de ses cooccurrents c_j (*liste_coocc_synt_ph*) :
        pour chaque sens s_k qui a été assigné manuellement à c_j (*C2*) :

$score\_s_k$++

- sélection du sens de chaque instance :
  pour chaque mot m_i dans *tab_ssm* :
    pour chaque phrase p_j (*C1*) :
      sélectionner le sens s_k qui a obtenu le score le plus élevé

3.3 Problème de l'évaluation

3.3.1 Evaluation d'un espace de mot

L'évaluation des espaces de mots est un problème pour lequel il n'existe pas de consensus clair. Les méthodes se basant sur la capacité d'un espace de mots à disposer, dans le voisinage des mots, de leurs synonymes, hyponymes, hyperonymes et autres ne sont pas utiles pour les applications thématiques, par exemple. Inversement, un espace de mots donnant de bons résultats thématiquement (par exemple chevalier/château fort) peut n'être d'aucune utilité pour une application de désambiguïsation. D'une manière générale il est quand même relativement admis qu'un espace de mots ne peut être évalué en soi, et que c'est son utilisation pour une tâche donnée qui peut être évaluée. Dans notre cas, l'évaluation nécessite d'utiliser l'espace de mots créé à une tâche de désambiguïsation « évaluable », c'est-à-dire portant sur des sens de mots issus de lexiques, et sur un corpus annoté manuellement.

De même, pour la clustérisation, il n'existe pas d'algorithme qui soit universellement supérieur aux autres : la meilleure stratégie de clustérisation est toujours fonction du problème à résoudre (type d'application) et du type des données traitées. Certains algorithmes ne sont d'ailleurs pensés que pour des types d'applications ou de données spécifiques. La seule manière de trouver le meilleur algorithme pour une tâche donnée est d'en expérimenter plusieurs, avec les mêmes données et mesures de distances, puis de comparer leurs résultats. Pour détecter les différences réelles entre deux modèles, il convient de comparer les paradigmes des deux méthodes, et non les détails de leur implémentation (nombre et type de leurs paramètres, mesure de distance choisie, etc.).

Les principales qualités requises d'une méthode de clustérisation sont, entre autres, selon [Pantel 2003] :

- l'évolutivité, c'est-à-dire la capacité à traiter des quantités importantes de données dans un espace de très haute dimension, pour cela l'algorithme se doit d'être efficace en temps et en espace,
- la capacité à classer des objets inconnus dans les clusters créés,
- la capacité à clustériser ou classifier avec peu de paramètres externes requérant des connaissances sur le domaine (nombre de clusters à découvrir par exemple)
- la capacité à traiter des données bruitées : l'algorithme se doit d'être suffisamment robuste pour minimiser son effet,
- la souplesse d'organisation, pour faciliter l'insertion ou la suppression de données nouvelles dans

la base d'entraînement.

Les trois méthodes d'évaluation les plus utilisées sont :

- l'**évaluation *in vivo*** [Ide et Véronis 1998], qui consiste à insérer la méthode de désambiguïsation dans une application de TAL, puis, à utiliser la méthode d'évaluation de l'application elle-même pour comparer ses performances avec et sans désambiguïsation.

- l'**évaluation *in vitro*** [Ide et Véronis 1998], pour laquelle on mesure tout simplement la précision, le rappel et la *F*-mesure des résultats de la clustérisation.

  - La *précision* d'un mot équivaut au pourcentage de clusters corrects auxquels il a été assigné; la précision d'un algorithme de clustérisation est la moyenne des précisions de tous les mots.
  Mais auparavant, il convient de définir la notion de 'sens correct'. Pour déterminer si un cluster correspond à un sens correct d'un mot, nous ne voyons pas d'autre solution que de le comparer avec les sens du mot proposés par une ressource externe comme WordNet, utilisé dans [Pantel 2003]. Il conviendra alors de définir une mesure pour calculer la similarité entre les clusters et les objets de la ressource utilisée.

  - Le *rappel* d'un mot correspond au rapport entre le nombre de clusters corrects auxquels le mot a été assigné et le nombre de sens du mot représentés dans le corpus; le rappel d'un algorithme de clustérisation est la moyenne des rappels de tous les mots.
  Le problème qui se pose ici tient au fait qu'on ne dispose jamais de la liste complète des sens d'un mot. La solution proposée par [Pantel 2003] consiste à appliquer au corpus un ensemble d'algorithmes de clustérisation, la liste des sens corrects d'un mot correspondrait alors à l'union des ensembles de clusters découverts par chaque algorithme.

- la **distance d'édition** ([Pantel 2003]), où il s'agit de comparer les clusters découverts automatiquement avec les résultats d'une clustérisation manuelle. La distance entre les deux se mesure au nombre d'opérations nécessaires pour transformer *C* (résultats automatiques) en *A* (résultats manuels) ([Pantel 2003] contient une description détaillée de cette méthode).

4 CONCLUSION

Les premières étapes de nos travaux ont consisté à mettre en place les outils permettant d'évaluer une méthode de désambiguïsation par rapport à des données issues d'une annotation manuelle. Outre la réalisation d'un programme fonctionnel et la production de plusieurs ressources utiles pour la désambiguïsation, cette étape a permis une familiarisation avec de nombreux outils et concepts tels que les espaces de mots, les techniques de réduction de dimensionnalité et la clustérisation.

La prochaine étape de ces travaux va consister à obtenir les clusters autour des mots ambigus, puis à les utiliser comme entrée de notre programme de désambiguïsation qui utilisait auparavant des clusters créés à la main. Quelques difficultés rencontrées dans l'équipe de recherche dans la construction de ces clusters, due à la nécessité d'utiliser de multiples espaces de mots de manière simultanée nous a conduit à choisir d'évaluer notre approche sur une clusterisation simple en fusionnant, dans un premier temps, tous les espaces de mots en un seul. Ceci, sans remettre en question la validité de la recherche que nous allons mener, nous permettra d'aboutir plus rapidement aux résultats attendus pour savoir si cette voie de recherche est effectivement pertinente.

5 BIBLIOGRAPHIE

[Agirre et Lopez de Lacalle 2003] [Mihalcea et Moldovan 2001]
E. AGIRRE, O. LOPEZ DE LACALLE. 2003. « Clustering WordNet word senses ». *Actes de RANLP 2003*.

[Ankerst, Markus, Breunig, Krieger et Sander, 1999]
M. ANKERST, M. MARKUS, BREUNIG, H. KRIEGAL et J. SANDER. 1999. *Actes de ACM SIGMOD' 99 Int.Conference on Management of Data*, Philadelphia.

[Audibert 2002]
Audibert, L. 2002. « Etude des critères de désambiguïsation sémantique automatique : présentation et premiers résultats sur les cooccurrences », *6ème Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL-2002)*, Nancy, p. 415-424.

[Bestgen 2004]
Y. BESTGEN. 2004. « Analyse sémantique latente et segmentation automatique des textes ». Dans G. Prunelle, F. Fairon et A. Dister (eds.). *Actes des 7e journées internationales d'analyse statistique des données textuelles (JADT04)*, Louvain-la-Neuve, 10-12 mars 2004, Presses Universitaires de Louvain, 171-181.

[Barwise et Perry 1983]
J. BARWISE, J. PERRY. 1983. S*ituations and Attitudes*, MIT Press, Cambridge.

[Basili et al. 1997]
R. BASILI, M. T. PAZIENZA, et M. VINDIGNI. 1997. « Corpus-driven unsupervised learning of verb subcategorization frames ». Actes du 5ième Congrès de l'Association Italienne d'Intelligence Artificielle, vol. 1321, p. 159-170.

[Bloomfield 1933]
Bloomfield, L. 1933. *Language*. New York: Henry Holt.

[Choueka and Lusignan 1985]

[Crestan, El Bèze et De Loupy 2003]
É. Crestan, M. El-Bèze, C. de Loupy. 2003. « Peut-on trouver la taille de contexte optimale ? ». *Actes de TALN 2003,* Batz-sur-Mer, pp. 85-94.

[Cutting et al. 1992]
D. R. CUTTING, J. O. PEDERSEN ET P.-K. HALVORSEN. 1992. « Scatter/gatter : A cluster-based approach to browsing large document collections, *Proceedings of SIGIR'92*, p. 318-329, Copenhague, Danemark.

[Deerwester et al. 1990]
S. DEERWESTER, S.T. DUMAIS, G.W. FURNAS, T.K. LANDAEUR et R. HARSHMAN. 1990. « Indexing by Latent Semantic Analysis », *Journal of the American Society for Information Science*, 41, 391-407.

[de Haan 2004]
F. de HAAN. 2004. *On representing semantic maps.*

[De Loupy 2002]
C. de Loupy. 2002. « Evaluation des taux de synonymie et de polysémie dans un texte ». *Actes de TALN 2002,* Nancy, France.

[Dorow et Widdows 2003]
B. DOROW et D. WIDDOWS. 2003. « Discovering corpus-specific word sens ». *Actes de EACL 2003*, p. 79-82.

[Ester, Kriegel, Sander et Xu 1996]
M. Ester, H. Kriegel, J. Sander, X. Xu. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise, *Proceedings of 2nd International Conference in Knowledge Discovery and Data mining*.

[Ferret 2004]
O. FERRET. 2004. « Découvrir des sens des mots à partir d'un réseau de cooccurrences lexicales », *Actes de TALN*, Fès, 19-22 avril 2004.

[Fuchs 2000]
C. FUCHS. 2000. *Les ambiguïtés du français*. Ophrys : Coll. L'Essentiel français.

[Gale et al. 1993]
Gale, W., K. Church, and D. Yarowsky,"A Method for Disambiguating Word Senses in a Large Corpus," in Computers and the Humanities, 1993.

[Goldberg 1995]

A. GOLDBERG. 1995. *Constructions : a construction grammar approach to argument structure*. Chicago et Londres, University of Chicago Press.

[Golub et van Loan 1989]
G. H. GOLUB et C. F. VAN LOAN. 1989. *Matrix computations*. The John Hopkins University Press, Baltimore and London.

[Gougenhein and Michéa 1961]

[Grefenstette 1993]
G. Grefenstette. 1993. « Evaluation techniques for automatic semantic extraction: comparing syntactic and window based approaches ». In Proceedings of the SIGLEX Workshop on Acquisition of Lexical Knowledge from Text, Columbus, Ohio.

[Gross 1989]
G. GROSS. 1989. *«* Désambiguïsation sémantique à l'aide d'un lexique-grammaire »*, Semantica*, Paris, Ladl et Univ. Paris 7.

[Harris 1954]
Z. HARRIS. 1954. « Distributional Structure ». in *Word*, p. 146-162.

[Haspelmath 2003]
M. HASPELMATH. 2003. « The geometry of grammatical meaning : semantic maps and cross-linguistic comparison. Dans M. Tomasello (ed.), *The new psychology of languge : cognitive and functional approaches to language structure*, vol. 2. Mahwah, NJ : Lawrence Erlbaum, 211-242.

[Ide et Veronis 1998]
IDE, Nancy et VERONIS, Jean (1998). « Word sense disambiguation : the state of the art ». *Computational Linguistics,* 24(1), 1-41.

[Indyk et Motwani 1998]
P. INDYK et R. MOTWANI. 1998. « Approximate Nearest Neighbor – Towards removing the curse of dimensionality ». *Actes du 30ème symposium Theory of Computing*, p. 604-613.

[Jacquet 2004]
JACQUET G. 2004. « Using the construction grammar model to disambiguate polysemic verbs in French », *Actes de ICCG3 (International Conference on Construction Grammar)*, Marseille.

[Kaplan 1950]
A. Kaplan. 1950. « An experimental study of ambiguity and context ». *Mechanical Translation*, (2:2), 39- 46.

[Kilgariff et Rosenzweig 2000]

Kilgariff A. et ROSENZWEIG J. 2000. « Framework and results for English SENSEVAL », *Computers and the Humanities*, 34, 15-48, Kluwer, 2000.

[Kriegel, Pryakhin et Schubert 2005]
H. KRIEGEL, A. PRYAKHIN, M. SCHUBERT. 2005. « Multi-represented *k*NN-Classification for Large Class Sets ». *Actes de DASFAA 05, p.* 511-522.

[Landauer, T. et Dumais, S. 1997]
Landauer, T. and S. Dumais (1997). A solution to plato's problem: The latent semantic analysis theory of acquisition. Psychological Review, 104(2):211–240.

[Lesk 1986]
LESK, M. 1986. « Automatic sense disambiguation : how to tell a pine cone from an ice cream cone », *Proceedings of the SIGDOC Conference*, 24-26.

[Ng et Han 1994]
Ng, R. T. et Han, J. 1994. Efficient and effective clustering methods for Spatial Data Mining, *Proceedings 20th International Conference on Very Large Data Bases*, 144-155. Santiage, Chile.

[Pantel 2003]
P. PANTEL. 2003. *Clustering by Committee*. Ph.D. Département d'Informatique, Université d'Alberta.

[Pantel & Lin 2002]
P. PANTEL et d. LIN. 2002. « Discovering word senses from text ». *Actes de ACM SIGKDD Conference on Knowledge Discovery and Data Mining 2002*, P. 613-619.

[Pedersen et Bruce 1997]
T. PEDERSEN et R. BRUCE. 1997. « Distinguishing word senses in untagged text ».*Actes de EMNLP'97*, p. 197-207.

[Purandare 2003]
A. PURANDARE. 2003. « Discriminating among word senses using Mcquitty's similarity analysis ». *Actes de HLT-NAACL 03 – Student Research Workshop*.

[Pustejovsky 1995]
Pustejovsky, James. 1995. *The generative lexicon*. Cambridge Massachusetts, Londres: MIT Press.

[Rapp 2003]
R. RAPP. 2003. « Word sense discovery based on sense descriptor dissimilarity ». *Actes de Machine Translation Summit IX*.

[Salton et McGill 1983]

Salton, G. and M. McGill. 1983. *Introduction to modern information retrieval*. McGraw-Hill, New York, NY.

[Schütze 1998]
H. Schütze. 1998. « Automatic Word Sense Discrimination », *Computational Linguistics*. Vol. 24 (1), p. 97-123.

[Sérichard 2004]
D. SERICHARD. 2004. *Désambiguïsation sémantique automatique appliquant une variante de l'algorithme de Lesk simplifié à des sens candidats définis à partir d'un réseau de cooccurrences lexicales*. Mémoire de DEA, Univ. Marne-la-Vallée.

[Su, Yang, Zhang, Xu et Hu 1999]
Z. SU, Q. YANG, H. ZHANG, X. XU et Y. HU. 1999. *Correlation-based document clustering using Web Logs*. Department of Computing Science, Tsinghua University, Beijing , China.

[Véronis 2003]
J. VERONIS. 2003. « Cartographie lexicale pour la recherche d'information ». *Actes de TALN 2003,* P. 97-123.

[Victorri et Fuchs 1996]
B. VICTORRI et C. FUCHS. 1996. *La polysémie, construction dynamique du sens*, Paris, Hermès.

[Yarowsky 1993]
D. Yaroswsky. 1993. « One sense per collocation », *Proceedings of the ARPA Workshop on Human Language Technology*, p. 266-271, 1993.

6 ANNEXES

Annexe 1 – *La DTD du corpus C1*

(voir fichier joint)

Annexe 2 – ***La DTD du fichier contenant les résultats de la clustérisation manuelle de C1***

(voir fichier joint)

Annexe 3 – ***Description de l'algorithme DBSCAN***

1. Les paramètres de DBSCAN

L'algorithme DBSCAN requiert deux paramètres d'entrée :

- *Eps* le seuil maximal de distance entre un objet et ses voisins
- *MinPts* le seuil minimal de densité de l'*Eps*-voisinage d'un point *p*

L'heuristique la plus courante consiste à déterminer une valeur globale pour ces deux paramètres sur la base des paramètres de densité du cluster le moins de dense de la base de données. Ces paramètres sont effectivement de bons candidats comme paramètres globaux de l'algorithme puisqu'ils spécifient la densité la plus faible qui n'est pas considérée comme du bruit.

2. Définition de la notion de 'cluster' selon les méthodes de clustérisation basées sur la densité

Définition 1 : (*Eps*-voisinage d'un point) L'*Eps*-voisinage d'un point p, noté $N_{Eps}(p)$, est défini tel que $N_{eps}(p) = \{ q / D \mid dist(p,q) <= Eps \}$.

Dans un cluster, on distingue deux types de points : les 'points centraux' (*core points*) et les 'points de frontière' (*border points*). De manière général, l'*Eps*-voisinage d'un point de frontière a une densité inférieure à celle de l'*Eps*-voisinage d'un point central. La valeur de MinPts pour l'*Eps*-voisinage d'un point p est donc relative au type de p. Pour chaque point p d'un cluster C, il existe un point q dans C tel que p / à $N_{Eps}(q)$ et $N_{eps}(q)$ contient au minimum *MinPts* points.

Définition 2 : (accessibilité directe selon la densité) Un point p est *directement accessible* à partir d'un q étant donnés *Eps* et *MinPts* si :

1) p / à $N_{eps}(q)$ et
2) $|N_{eps}(q)| >= MinPts$

L'accessibilité directe selon la densité n'est pas symétrique pour des paires formées d'un point central et d'un point de frontière.

Définition 3 : (accessibilité selon la densité) Un point p est *accessible selon la densité* à partir d'un point q étant donnés *Eps* et *MinPts* s'il existe une chaîne de points p_1, ..., p_n, avec $p_1 = p$, $p_n = q$, tels que p_{i+1} est directement accessible selon la densité à partir de p_i.
Il se peut toutefois que deux points de frontière appartenant à un cluster *C* ne soient pas

forcément accessibles selon la densité. Il doit alors exister un point central dans *C* à partir duquel tous deux sont accessibles selon la densité.

Définition 4 : (connectivité selon la densité) Un point p est *connecté selon la densité* à un point q étant donnés *Eps* et *MinPts* s'il existe un point *o* tel qu'à la fois p et q sont accessibles selon la densité à partir de o étant donnés *Eps* et *MinPts*. Cette relation est symétrique et réflexive.

On peut maintenant définir la notion de cluster basée sur la densité. Un cluster est défini comme l'ensemble maximal des points connectés selon la densité étant donnés *Eps* et *MinPts*, et le bruit est défini comme l'ensemble des points qui ne sont couverts par aucun cluster.

Définition 5 : (cluster) Etant donnée D une base de points. Un cluster C étant donnés *Eps* et *MinPts* est un sous-ensemble non vide de D satisfaisant les conditions suivantes :

- $\forall$ p, q : si p / C et q est accessible selon la densité à partir de q étant donnés *Eps* et *MinPts*, alors q / C (maximalité)
- $\forall$ p, q / C : p est connecté selon la densité à q étant donnés *Eps* et *MinPts* (connectivité)

Définition 6 : (bruit) Etant donnés C_1, ..., C_k les clusters de la base de données D étant donnés les paramètres Eps_i et $MinPts_i$, i = 1, ..., k. Le bruit est défini comme l'ensemble des points de D qui n'appartiennent à aucun cluster C_i, c'est-à-dire :

- bruit = { p / D | $\forall$ i : p // C_i }

Lemme 1 : Etant donné p un point de D, et $|N_{eps}(p)| >= MinPts$. Alors l'ensemble O = { o | o / D et o est accessible selon la densité à partir de p étant donnés *Eps* et *MinPts* } est un cluster étant donnés *Eps* et *MinPts*.

Un cluster ne doit pas être défini uniquement par l'un de ses points. Chaque point d'un cluster *C* est accessible selon la densité à partir de tout point central de *C*, donc *C* contient tous les points qui sont accessibles selon la densité à partir de n'importe quel point central de *C*.

Lemme 2 : Etant donné C un cluster étant donnés *Eps* et *MinPts* et p un point quelconque de C avec $|N_{eps}(p)| >= MinPts$. Alors C correspond à l'ensemble O = { o | o est accessible selon la densité à partir de p étant donnés *Eps* et *MinPts* }

3. Application de DBSCAN

Pour découvrir un cluster, DBSCAN sélectionne aléatoirement un point p / D et extrait tous les points accessibles selon la densité à partir de p étant donnés *Eps* et *MinPts*. Si p est un

point central, cette procédure découvre un cluster étant donnés *Eps* et *MinPts* (Lemme 2), si p est un point de frontière, aucun point est accessible selon la densité à partir de p. DBSCAN passe alors au point suivant dans D.

Les clusters de densité différente sont fusionnés selon la définition 5 s'ils sont proches l'un de l'autre. La distance entre deux ensembles de points S_1 et S_2 étant définie comme dist(S1, S2) = min { dist(p,q) | p / S_1, q / S_2 }. Par conséquent, deux ensembles de points dont la densité est au moins égale à la densité du cluster le moins dense sont séparés uniquement si la distance qui les sépare est supérieure à *Eps*.