N^o d'ordre: 2973

THÈSE

Présentée devant

devant l'université de Rennes 1

pour obtenir

le grade de : Docteur de l'université de Rennes 1
Mention Informatique

par

Vincent Claveau

Équipe d'accueil : TexMex/Irisa
École doctorale : Matisse
Composante universitaire : Ifsic

Titre de la thèse :

Acquisition automatique de lexiques sémantiques pour la recherche d'information

soutenue le 17 décembre 2003 devant la commission d'examen

M. :	Marie-Odile	Cordier	Président
MM. :	James	Cussens	Rapporteurs
	Béatrice	Daille	
MM. :	Mohand	Boughanem	Examinateurs
	Pascale	Sébillot	

Le silence vertébral
indispose la voile licite

L. Tesnière, 1953

Colorless green ideas
sleep furiously

N. Chomsky, 1957

Remerciements

Après un peu de plus trois années de thèse passées à l'IRISA, j'ai beaucoup de personnes à remercier pour leur participation directe ou indirecte à mes travaux.

Je tiens tout d'abord à remercier Marie-Odile Cordier qui m'a fait l'honneur de présider mon jury et qui m'a également donné des conseils avisés pour ma soutenance. Je remercie également Béatrice Daille et James Cussens d'avoir accepté la charge de rapporteur ; leurs relectures attentives de ce mémoire et leurs commentaires m'ont été très précieux. Mohand Boughanem, apportant son expertise en recherche d'information, m'a fait le plaisir de participer à mon jury ; je l'en remercie sincèrement.

Plusieurs personnes m'ont aidé lors de la rédaction de ce mémoire. Je tiens notamment à distinguer Fabienne Moreau et Catherine Belleannée pour leurs relectures de certaines parties de ce mémoire et leur commentaires éclairés. Caroline m'a quant à elle non seulement soutenu pendant la phase de rédaction de ce mémoire, mais également supporté pendant les trois années qui l'ont précédée ; sa patience n'a d'égal que mon affection envers elle.

Le travail présenté ici n'est pas uniquement le mien, loin de là. Pierrette Bouillon et Cécile Fabre ont en effet été partie prenante de l'ensemble de ces travaux, jouant le rôle ingrat de l'expert qui consiste à examiner des milliers de phrases, éplucher des résultats, préparer des données... Je leur adresse tous mes remerciements.

Les sympathiques discussions de cafet' sur l'apprentissage symbolique avec Daniel Fredouille et François Coste, pour détendues qu'elles fussent, m'ont été particulièrement profitables. Certaines des idées lancées à ces occasions — entre deux calembours improbables de François — sont d'ailleurs à l'origine d'une grande partie des travaux présentés au sein de ce mémoire.

Il me faut également remercier tous mes collègues du *patio* et de l'équipe TexMex. Impossible de tous les nommer, et j'ai épuisé ma collection de synonymes pour dire merci. Que chacun d'entre eux sache néanmoins que me lever le matin pour aller au labo (même tard !) a toujours été un plaisir, sachant l'ambiance tellement sympathique que j'allais y retrouver. Mes différents collègues de bureau, Irène Grosclaude, Roberto Bonato, Mathias Rossignol et Nicolas Bonnel, y sont certainement pour une grande part.

Enfin, *last but not least*, je tiens sincèrement à remercier Pascale Sébillot qui a dirigé, mais également participé activement, à mes travaux de thèse. Au-delà des formules de remerciements académiques, je voudrais lui exprimer toute ma gratitude pour son dynamisme, son indéflectible bonne humeur, sa compétence scientifique, sa patience, sa disponibilité (et j'en passe) durant ces quatre années passées en sa compagnie. C'est

grâce à elle que je me suis lancé dans cette aventure, contaminé rapidement par sa passion du domaine. Une thèse qui se déroule sans la moindre anicroche avec sa directrice de thèse, qui dit mieux ?

Table des matières

Liste des figures

Liste des tableaux

Introduction

Devant l'augmentation du nombre de documents électroniques, il est indispensable de mettre au point des outils permettant de gérer l'accès à l'information qu'ils contiennent. Cette phrase d'introduction est devenue un lieu commun tant il est vrai que depuis l'avènement d'Internet, la production et le volume de documents disponibles ont augmenté de manière exponentielle. Ainsi, le nombre de pages indexées sur le Web par le moteur de recherche réputé le plus complet (Google) dépasse à ce jour les 3,3 milliards; en 1999, son pendant (Northen Light) n'en comptait[1] seulement que 115 millions.

Cette explosion du nombre de documents disponibles s'accompagne d'une explosion du nombre des utilisateurs. Il s'agit bien sûr de particuliers cherchant des documents sur Internet, mais également de membres ou de clients d'entreprises ayant informatisé leurs documentations techniques et administratives. Pour répondre aux exigences de qualité des résultats de ces utilisateurs, certains systèmes d'information se tournent vers des représentations plus riches des documents manipulés et des informations qu'ils contiennent. Cette richesse doit ainsi permettre à ces systèmes d'améliorer à la fois leur pertinence et leur efficacité, voire également leur adaptabilité à de nouvelles tâches.

L'un des moyens d'obtenir ces représentations riches est d'utiliser des ressources sémantiques qui vont permettre, à travers le sens des mots contenus dans un texte, de représenter et gérer plus finement les informations qu'il renferme. Ces ressources peuvent être externes, c'est-à-dire issues d'une connaissance autre que les documents (expertises, dictionnaires ou autres ressources préexistantes), ou bien internes. Dans ce dernier cas, ce sont donc les documents eux-mêmes qui vont, à travers des techniques d'acquisition d'éléments sémantiques, fournir les informations nécessaires à la construction de ces bases de connaissances sur le sens des mots. C'est dans ce cadre — l'acquisition d'informations lexicales sémantiques à partir de textes — que se situent les travaux présentés dans ce mémoire.

Dans la plupart des études effectuées dans ce domaine, la donnée principale sur laquelle repose l'acquisition d'informations sémantiques lexicales est un ensemble de textes, constitués en corpus. La tâche d'acquisition peut avoir pour but, à partir de ce corpus, de collecter les unités sémantiques propres au domaine (par exemple des termes

[1] Ces chiffres sont tirés des estimations effectuées par Search Engine Showdown, disponibles à l'URL http ://www.searchengineshowdown.com.

techniques) mais également les relations existant entre ces unités (comme des liens de synonymie). Il est alors possible d'organiser, si besoin, ces unités et ces relations sous forme de lexiques.

La difficulté majeure réside dans le fait de devoir passer d'une simple séquence de symboles — les textes au format électronique — à une base de connaissances structurée manipulant le sens des mots. De plus, lorsque cette transformation d'une représentation de bas niveau en une représentation de haut niveau cognitif est appelée à être effectuée de nombreuses fois, l'accent doit également être mis sur son efficacité.

En pratique, cette efficacité se traduit par deux propriétés distinctes, parfois difficiles à concilier : la qualité des résultats et l'automaticité. La préférence est donnée soit à l'une, soit à l'autre, selon l'utilisation qui doit être faite des informations sémantiques acquises. Cette préférence se répercute sur les choix adoptés pour mener la tâche d'acquisition. L'expertise humaine, par exemple, est très coûteuse, mais produit bien sûr des résultats de très bonne qualité. Dans le cas où le processus d'acquisition doit être souvent répété, on opte toutefois plutôt pour des techniques plus automatiques ; elles permettent de traiter avec un minimum d'intervention humaine tout nouveau corpus mais, en contrepartie, peuvent produire des résultats de moins bonne qualité. Enfin, une dernière propriété attendue des techniques d'acquisition est leur interprétabilité. En effet, certaines approches donnent de bons résultats, mais leur aspect « boîte noire » a plusieurs désavantages. D'une part, elles ne permettent pas d'interpréter les résultats ; il est dès lors impossible de comprendre le processus ayant amené tel élément à être acquis ou non. D'autre part, elles n'offrent pas de définitions opérationnelles des informations sémantiques qu'elles acquièrent. Celles-ci sont pourtant intéressantes dans les cas où les éléments recherchés ne sont définis que de manière théorique ou par l'intermédiaire d'exemples.

Dans ce mémoire, nous présentons la méthode d'acquisition sur corpus d'informations sémantiques (unités ou relations sémantiques) que nous avons développé, qui tente de satisfaire ces différentes exigences. Cet outil logiciel, ASARES[2], doit donc répondre au triple objectif suivant :

1. assurer une bonne qualité des résultats d'extraction en acquérant de manière complète et précise l'ensemble des informations sémantiques voulues présentes dans un corpus ;
2. produire des résultats linguistiquement interprétables ;
3. être le plus générique et automatique possible et donc le plus facilement portable d'un corpus à un autre.

Nous avons choisi, dans le cadre des travaux présentés ici, d'appliquer cet outil à l'acquisition en corpus d'un type particulier de relations entre des noms et des verbes sémantiquement liés (comme le nom *couteau* et le verbe *couper*). Ces relations, appelées relations qualia, sont définies dans le modèle du Lexique génératif (Pustejovsky, 1995), mais de manière purement théorique. Leur acquisition revêt deux intérêts principaux. D'une part, la propriété d'interprétabilité d'ASARES doit permettre de mieux

[2]Cet acronyme signifie Acquisition Symbolique Automatique de REssources Sémantiques.

comprendre le fonctionnement de ces relations et de définir leurs modes de réalisations en contexte. D'autre part, ces relations sémantiques acquises automatiquement offrent la possibilité de vérifier l'intérêt de leur exploitation dans des applications de recherche d'information.

Pour vérifier ce second point, nous examinons la portée de l'apport des relations qualia à l'extension de requêtes dans un système de recherche documentaire. Dans ce domaine, dont le principe est de trouver un ou plusieurs documents à partir d'une requête, il paraît essentiel de pouvoir manipuler aisément le sens des mots contenus à la fois dans les documents et dans la requête. À ce titre, les unités répertoriées dans un lexique sémantique peuvent permettre d'accéder aux informations renfermées par les documents ; les relations entre ces unités, telles que les relations qualia, servent quant à elles à naviguer entre ces différentes informations en autorisant la prise en compte de formulations différentes d'un même même concept.

L'approche suivie pour le développement de notre méthode d'acquisition et son évaluation dans un cadre applicatif précis se situe au carrefour de plusieurs grands domaines de recherche. Nos travaux s'inscrivent bien sûr dans le cadre du traitement automatique des langues (TAL) et sont à mettre en parallèle avec les études existant en acquisition d'informations lexicales sémantiques sur corpus. Ils s'en distinguent cependant par la technique que nous avons adoptée. Le composant principal d'ASARES est en effet une méthode d'apprentissage artificiel symbolique : la programmation logique inductive. Celle-ci nous permet de découvrir (d'inférer) les patrons génériques définissant en contexte les informations lexicales sémantiques visées. Ces patrons sont ensuite utilisés pour découvrir et extraire ces informations lexicales au sein du corpus. Ils nous permettent de répondre à notre exigence d'interprétabilité, rarement satisfaite par les outils existants, en offrant une définition opérationnelle du concept régissant les éléments sémantiques recherchés. L'utilisation de cette technique place donc également nos travaux dans le cadre de l'apprentissage artificiel et plus généralement de l'intelligence artificielle. Notre approche répond également à une logique propre à la linguistique de corpus dans laquelle les connaissances linguistiques sont exclusivement dérivées d'exemples réels tirés de corpus. En effet, la programmation logique inductive est une méthode supervisée : les patrons sont donc inférés à partir d'exemples, c'est-à-dire à partir de quelques occurrences des informations désirées dans le corpus. Enfin, le dernier domaine de recherche que nous abordons est la recherche d'information (RI), et plus spécifiquement la recherche documentaire, que nous abordons à travers l'utilisation des connaissances lexicales sémantiques acquises sur corpus pour l'extension de requêtes.

L'originalité de nos travaux réside en plusieurs points concernant aussi bien l'approche adoptée dans ASARES que notre cadre applicatif d'acquisition de relations qualia et l'exploitation de ces relations en RI. Tout d'abord, l'utilisation de techniques d'apprentissage symbolique, telles que la programmation logique inductive employée au sein d'ASARES, en TAL et plus spécifiquement pour l'acquisition d'informations sur corpus, est rare. Les approches statistiques sont en effet très largement dominantes. Nous

montrons cependant que l'utilisation de ces techniques d'apprentissage symbolique et plus spécifiquement de la programmation logique inductive est parfaitement adaptée à de telles tâches d'acquisition. Nous mettons également en évidence les possibilités de combinaison des techniques numériques et symboliques permettant d'en conjuguer les avantages distincts. En particulier, nos travaux prouvent que l'automaticité des approches numériques et l'interprétabilité des symboliques peuvent être associées au sein d'un même outil d'acquisition.

La grande expressivité de la programmation logique inductive a pour contrepartie un coût calculatoire important. Nos travaux nous ont donc amené à adapter les caractéristiques de cette technique d'apprentissage à nos besoins spécifiques de manière à diminuer cette complexité. Notre contribution sur ce point, aussi bien théorique que pratique, permet ainsi à ASARES un temps d'acquisition plus faible, mais garantit aussi l'interprétabilité des résultats d'un point de vue linguistique.

Nous attestons par ailleurs que l'approche symbolique consistant à inférer des patrons d'extraction à partir d'exemples peut donner de très bons résultats, même lorsque les informations exploitées sont de bas niveau. Nous montrons par exemple que des patrons n'exploitant que des informations sur les catégories des mots (nom, verbe, préposition, *etc.*) donnent des résultats d'extraction à la fois précis et complets. Cette approche « *knowledge-poor* » permet de rendre notre technique d'acquisition aisément portable d'un corpus à un autre puisque peu de pré-traitements sur les données sont requis. De par sa portabilité, elle se veut donc générique et peut ainsi répondre à la diversité des besoins en acquisition d'informations sémantiques à partir de corpus.

Notre cadre applicatif, c'est-à-dire l'acquisition de relations sémantiques entre noms et verbes, est également original puisqu'il a fait l'objet de peu de travaux comparativement aux relations nom-nom. Plus encore, les relations de type qualia n'ont quasiment pas été étudiées dans une perspective d'acquisition. Cela est certainement dû à la relative jeunesse de ce modèle lexical, mais également au fait que, comme nous l'avons déjà dit, ces relations ne sont définies que d'une manière théorique ce qui rend difficile la mise en œuvre de leur extraction. À ce titre, l'interprétabilité de notre technique offre au linguiste de nombreux indices concernant les phénomènes linguistiques qui entrent en jeu dans la réalisation effective des relations qualia dans un texte.

L'application de ces relations nom-verbe à la recherche documentaire est également un thème de recherche peu abordé. Bien que plusieurs auteurs aient mentionné l'intérêt des liens nom-verbe en RI, les expériences utilisant des ressources sémantiques se sont principalement focalisées sur des extensions de requêtes (voire des représentations des documents) intra-catégorielles (reformuler un nom à l'aide d'un autre), mais peu sur des extensions inter-catégorielles comme celles que nous proposons. Nous examinons donc dans ce mémoire dans quelle mesure le lien nom-verbe qualia est pertinent pour cet usage.

Le triple objectif explicité ci-dessus — à savoir développer un outil d'extraction à la fois portable, produisant des résultats de bonne qualité, et dont les résultats et le processus soient linguistiquement interprétables — constitue le fil conducteur de ce mémoire. Dans le premier chapitre, nous présentons la problématique de l'acquisition

d'informations lexicales sémantiques sur corpus et nous examinons les différentes approches employées dans les techniques d'acquisition existantes. En s'appuyant sur les conclusions qui ressortent de ce panorama, le chapitre 2 motive les objectifs fixés et les choix adoptés dans notre outil ASARES ; les cadres techniques et applicatifs de nos travaux y sont également décrits à travers les présentations générales de la programmation logique inductive et du Lexique génératif définissant les relations qualia. Le chapitre 3 est quant à lui dédié à la présentation détaillée de l'utilisation que nous faisons de la programmation logique inductive au sein d'ASARES. Nous nous attachons notamment à décrire les contraintes imposées pour permettre l'inférence efficace de patrons linguistiquement interprétables. Nous présentons également les résultats obtenus pour l'acquisition de relations qualia. Nous examinons ensuite dans le chapitre 4 plusieurs adaptations possibles de notre outil, notamment en combinant techniques d'extraction numériques et symboliques, pour améliorer son automaticité et sa portabilité en maintenant la qualité des résultats d'acquisition. Le dernier chapitre de ce mémoire est consacré à l'exploitation des ressources sémantiques acquises automatiquement par ASARES dans une application de recherche d'information. Nous montrons que l'utilisation de telles ressources pour étendre des requêtes permet d'améliorer sensiblement la pertinence des documents retournés à un utilisateur interrogeant le système. Nous terminons en mettant en exergue les principales contributions de nos travaux et en proposant quelques pistes de recherche ouvertes à l'issue de ces derniers.

Chapitre 1

Acquisition d'informations lexicales sémantiques : problématique et état de l'art

Comme nous l'avons mentionné en introduction, donner les moyens à un utilisateur d'accéder au contenu d'une vaste base de documents électroniques, pour des besoins aussi divers que la veille technologique, la recherche documentaire mono ou interlingue, *etc.*, nécessite d'avoir une connaissance fine des informations contenues dans les documents, et plus particulièrement des informations sémantiques. Il est donc nécessaire de développer des ressources lexicales sémantiques suffisamment riches pour répondre aux spécificités des applications visées. Cependant, ces ressources étant *a priori* propres au domaine étudié, il convient plus particulièrement de mettre au point des méthodes permettant de faciliter ou même d'automatiser leur construction.

Selon les besoins de l'application et du domaine étudiés, deux types d'informations lexicales sémantiques sont recherchés : les unités lexicales sémantiques, et pour les structurer, les relations sémantiques entre ces unités. Les travaux cherchant à extraire les unités relèvent principalement de l'acquisition de terminologie, les termes étant les éléments privilégiés pour porter le sens d'un texte dans un domaine spécialisé. Ceux visant à acquérir les relations sémantiques s'intéressent essentiellement à des liens « classiques » tels que la synonymie ou l'hyperonymie.

Ce chapitre est dédié à un tour d'horizon des travaux existants dans le domaine de l'acquisition d'informations sémantiques lexicales sur corpus. Nous présentons dans un premier temps la problématique et les enjeux couverts par ce thème de recherche. Nous nous intéressons ensuite plus spécifiquement à l'acquisition de termes puis à celle de relations lexicales sémantiques ; nous nous attachons, à travers un panorama des travaux existants dans chacun de ces domaines, à faire ressortir les grandes familles de techniques utilisées. En particulier, l'opposition des méthodes exploitant l'aspect numérique des données ou leur aspect structurel est mise en avant, et les avantages et inconvénients de chacune sont détaillés.

1.1 Problématique et enjeux

Les ressources sémantiques lexicales sont nécessaires à de nombreuses applications du TAL, que ce soit pour la production, la diffusion, la gestion, la recherche, l'exploitation ou la traduction de documents (Bourigault & Jacquemin, 2000). Cependant, les informations sémantiques utilisées pour chacune de ces tâches sont spécifiques et rarement mutualisables. À cette spécificité relevant de l'utilisation devant être faite des ressources sémantiques s'ajoute une spécificité du domaine traité. En effet, chaque domaine de spécialité se traduit par l'emploi d'un vocabulaire propre et plus généralement d'un langage particulier, appelé sous-langage ou langage spécialisé (Pearson, 1998 ; Lerat, 1995).

Nous présentons dans un premier temps les enjeux que recouvrent l'utilisation de ressources sémantiques lexicales. Nous identifions en particulier quelques applications et domaines de recherche dans lesquels de telles bases de connaissances sémantiques sont déjà exploitées. Nous examinons ensuite deux approches opposées visant à satisfaire ces besoins d'informations sémantiques ; il s'agit de l'utilisation de bases généralistes préexistantes ou de celle de techniques d'acquisition. Les premières tentent de répondre aux besoins spécifiques par leur très large couverture, alors que les secondes visent à n'acquérir que les informations propres à la tâche et au domaine ciblés.

1.1.1 Enjeux applicatifs

Bien qu'il soit vain de vouloir lister l'ensemble des applications qui bénéficieraient de connaissances lexicales sémantiques, nous en proposons néanmoins quelques grandes familles pour lesquelles ce besoin d'apport de sémantique a été constaté et reconnu. Parmi celles-ci nous présentons plus particulièrement la recherche d'information qui sert de cadre applicatif à nos travaux.

La recherche d'information peut être vue comme la tâche consistant à fournir une ou plusieurs réponses à une ou plusieurs requêtes d'un utilisateur à partir d'une base documentaire. Ce cadre très général comprend de nombreux domaines de recherche très actifs. Par exemple, dans le cas de la recherche documentaire, la réponse fournie par le système sera un ensemble de documents, comme des pages Web ; dans le cas de systèmes QA[1] (acronyme anglais de *question answering*), la réponse à une question précise de l'utilisateur est un extrait de document, ou une phrase générée à partir d'informations extraites des documents.

Typiquement, un système de recherche de documents fonctionne de la manière suivante. À chaque document constituant la base sont attachés des termes d'indexation

[1]Certains ne considèrent pas les systèmes QA comme entrant dans la définition des systèmes de recherche d'information. Ainsi, C. van Rijsbergen (van Rijsbergen, 1979) reprend à son compte la définition de Lancaster (1968) : « *Information retrieval is the term conventially, though somewhat inaccurately, applied to the type of activity discussed in this volume. An information retrieval system does not inform (i.e. change the knowledge of) the user on the subject of his inquiry. It merely informs on the existence (or non-existence) and whereabouts of documents relating to his request.* »

(*indexing terms*) représentant au mieux son contenu. On assigne souvent à ces termes un poids indiquant leur adéquation avec le contenu du document. Les requêtes des utilisateurs subissent un traitement analogue; des termes de requêtes (*query terms*) éventuellement pondérés en sont donc extraits. Une procédure de mise en correspondance permet de retrouver les documents dont les termes d'indexation sont les plus proches des termes de requête.

Dans ce cadre, l'utilisation de ressources sémantiques permet, à travers les unités qu'elles contiennent, de choisir des termes d'indexation pertinents. Ces termes sont en effet souvent extraits à partir des documents (voir section 1.2.1); il est donc important d'avoir une connaissance pointue de la sémantique véhiculée par ces mots. Les relations sémantiques entre les unités lexicales donnent quant à elles accès à différentes formulations d'une même idée (utilisation de synonymes par exemple). Certains systèmes d'indexation automatique exploitent ainsi des thésaurus ou des réseaux lexicaux spécialisés, ou bien encore des outils comme FASTR (Jacquemin, 2001) qui permettent d'identifier les différentes variantes d'un même terme.

Outre la recherche d'information, de nombreuses autres applications peuvent bénéficier, soit directement, soit comme première étape de traitements plus complexes, de l'apport de ressources terminologiques ou plus généralement sémantiques lexicales (Bourigault & Jacquemin, 2000). Par exemple, les systèmes d'aide à la rédaction peuvent utiliser des terminologies de référence; de même les systèmes de gestion de données techniques exploitent avec intérêt des référentiels terminologiques. Dans les entreprises, les connaissances terminologiques sont utilisées lors de la construction d'ontologies pour les mémoires d'entreprises et au sein de systèmes d'aide à la décision. Elles servent également pour la constitution de glossaires de référence et de listes de termes pour les outils de communication inter-entreprise ou client-entreprise. Les bases de connaissances terminologiques, et les unités ou relations sémantiques qu'elles contiennent, peuvent être également exploitées dans des systèmes de résumé automatique. On peut également citer le filtrage d'information (Turenne, 2000), la classification de textes (Sebastiani, 2002), la vérification de texte (Faure, 2000), la traduction automatique (voir la description de TERMIGHT ci-dessous) ou encore la veille technologique (Polanco *et al.*, 1998).

1.1.2 Bases généralistes *versus* acquisition automatique

Pour répondre à ce besoin de ressources lexicales, deux approches ont vu le jour simultanément et continuent à s'opposer aujourd'hui. Il s'agit, comme nous l'avons déjà précisé, soit d'utiliser des bases généralistes soit d'acquérir des éléments de lexiques sémantiques spécialisés pour ensuite les exploiter.

Nous présentons ci-dessous les motivations qui nous ont amené à opter pour le deuxième choix, en détaillant certaines limites inhérentes au modèle même des bases généralistes. Nous rappelons ensuite l'origine des travaux dans le domaine de l'acquisition automatique d'informations sémantiques lexicales.

1.1.2.1 Limites des bases généralistes

Le domaine de la construction de lexiques sémantiques en TAL a connu un certain essor depuis les années 1990, avec l'apparition de grandes bases de connaissances ayant pour but de répondre à une demande croissante en ressources sémantiques lexicales élaborées. Pour tenter de couvrir le maximum de domaines, ces bases généralistes sont toutefois amenées à grossir sans cesse. La base WordNet (Miller *et al.*, 1989) a par exemple atteint une taille très importante, avec plus de 150 000 mots (dont 115 000 noms) regroupés en 115 000 *synsets*. Cette course à la complétude est cependant perdue d'avance : la couverture de ce type de base reste trop restreinte par rapport à l'ensemble des domaines dans lesquels s'expriment des besoins en ressources sémantiques. D'une part, se voulant indépendantes de tout domaine, elles sont en fait peu adaptées pour traiter les domaines spécialisés. Ces derniers manipulent en effet un vocabulaire particulier, prêtant des sens spécifiques à certains mots de la langue « générale » ; dans le même ordre d'idée, un mot non polysémique dans un certain domaine peut par exemple être noté comme polysémique dans de telles bases. D'autre part, la structure même de ces bases peut ne pas répondre de manière adéquate à des besoins particuliers, propres à une application visée. Comme nous le rappelons en section 2.1.1.2, on reproche par exemple à WordNet de ne pas indiquer systématiquement les relations entre des termes d'un même domaine, entre des termes morphologiquement liés... De telles attentes spécifiques ne peuvent donc trouver de réponse autre que le développement de ressources propres aux domaines et aux besoins. Ces considérations se retrouvent dans les résultats expérimentaux des systèmes exploitant de telles bases de connaissances généralistes[2] et de nombreux auteurs en soulignent les limites (voir par exemple (de Loupy & El-Bèze, 2002)).

1.1.2.2 Origines de l'acquisition automatique

Les problèmes inhérents aux bases généralistes, spécialement dans les domaines spécialisés, ont naturellement conduit à envisager d'autres alternatives pour répondre aux besoins en ressources sémantiques. Le développement de telles ressources par des experts n'étant pas envisageables pour de grandes masses de données et des demandes multiples, il a donc été examiné la possibilité de construire des outils d'aide à l'acquisition, voire d'acquisition automatique.

Des nécessités industrielles sont en grande partie à l'origine de l'acquisition de ressources sémantiques (Bourigault & Jacquemin, 2000). De nombreux outils d'extraction (entre autres ana, acabit, lexter, termight, *etc.*) ont ainsi été développés dans ce contexte pour répondre à des attentes précises telles que la traduction automatique, la gestion des connaissances, l'indexation automatique. Du fait de cette naissance en milieu industriel, ces besoins d'acquisition se sont exprimés pour des langages spécialisés et non pas pour la langue « générale ». Cela a conduit les travaux de sémantique lexicale à se porter sur l'étude du domaine connexe de la terminologie.

[2] WordNet a en particulier été utilisée dans un grand nombre d'applications du TAL (désambiguïsation de sens, annotation sémantique de texte, extraction d'information et bien sûr en recherche documentaire, voir (Fellbaum, 1998)).

Cependant, d'autres domaines de recherche ont également contribué aux développements de tels outils pour satisfaire leurs propres besoins. C'est notamment le cas de la recherche d'information dans laquelle la recherche d'une représentation adaptée des documents a conduit certains chercheurs à s'intéresser à la sémantique lexicale et plus particulièrement à la terminologie. Le domaine de l'ingénierie de connaissances a également contribué, à travers la construction d'ontologies, au développement de l'acquisition de terminologie, le terme étant généralement considéré comme porteur d'un concept et donc de la connaissance.

L'acquisition d'informations lexicales sémantiques sur corpus peut se découper artificiellement en deux types de travaux. Les premiers, que nous présentons dans la section suivante porte sur l'extraction des unités sémantiques, les seconds, que nous verrons en section 1.3, se concentrent sur les relations entre ces unités.

1.2 Unités des lexiques sémantiques

Beaucoup de travaux en acquisition d'unités sémantiques relèvent de l'extraction de terminologie. Outre les raisons historiques évoquées ci-dessus, cela s'explique par le fait que ces travaux se sont souvent focalisés sur des domaines spécialisés, et donc des langages de spécialité. Or, dans ce contexte, les unités sémantiques porteuses de sens sont principalement des termes.

Cependant, les techniques utilisées dans ce domaine ne précisent pas toujours le statut linguistique exact des unités acquises, leur utilisation à l'acquisition de termes étant principalement le fait de leur application aux textes spécialisés. On peut donc utiliser ces mêmes techniques pour acquérir d'autres unités linguistiques; nous les présentons donc comme des méthodes génériques d'acquisition d'unités de lexiques sémantiques.

Après une brève présentation de ce que recouvre, traditionnellement et de nos jours, la notion de terme, nous passons en revue quelques-uns des nombreux travaux effectués dans le domaine. Pour ce faire, nous les regroupons en trois grandes familles suivant les différents aspects du texte que ces techniques exploitent : fréquentiel, structurel ou les deux ensemble.

1.2.1 Qu'est-ce qu'un terme

Pour répondre à la question de ce qu'est un terme, la terminologie traditionnelle se heurte de nos jours à la linguistique de corpus. Les fondements théoriques classiques se révélant peu adaptés à cette pratique se trouvent revus. Dans la section suivante, nous montrons en effet que la définition classique du terme a été récemment critiquée et des définitions plus pragmatiques proposées. Nous nous intéressons ensuite aux différents modes de création des termes et terminons en rappelant quelques variantes de forme reconnues de ces unités sémantiques.

1.2.1.1 Définitions

La terminologie en tant que discipline est définie par l'ISO (Organisation international de normalisation) comme l'« étude scientifique des notions et des termes en usage dans les langues de spécialités » (ISO 1087, 1990). Ainsi au sein de ces langages de spécialités, le terme est généralement défini comme un objet linguistique à part entière (Lerat, 1995) utilisé principalement dans la littérature technique et scientifique, visant à faire référence à des concepts de façon consensuelle.

Plus précisément, selon E. Wüster, fondateur de la théorie générale de la terminologie, le terme désigne un concept scientifique, lui-même lié à d'autres concepts dans une organisation principalement taxinomique. Il est ainsi postulé que le terme est univoque et mono-référentiel (il y a donc une correspondance un à un entre termes et concepts) et universellement accepté comme tel parmi les utilisateurs de la langue de spécialité. La terminologie est donc la représentation parfaite du système conceptuel sous-jacent au domaine de connaissance.

L'approche wüsterienne, avec sa notion de label unique, est largement compatible avec le modèle aristotélicien représenté par le triangle sémiotique signe/concept/objet (Lerat, 1995 ; Rastier, 1995). D'autres linguistes, comme Rondeau, considère le terme comme un signe linguistique au sens saussurien (de Saussure, 1916), avec un signifiant (appelé encore dénomination) et un signifié (ou notion). Contrairement à la définition de Wüster, un terme désigne donc en même temps la dénomination et la notion. Quel que soit le point de vue adopté, la fonction de dénomination du terme semble impliquer naturellement l'utilisation de noms ou de syntagmes nominaux comme base du terme (voir (Rastier, 1995) pour une discussion sur ce point).

Cependant, en marge de cette définition traditionnelle, certains phénomènes inattendus, comme la variabilité de la terminologie même au sein d'un domaine, ont été constatés lors de travaux sur de grandes masses de données et contredisent la nécessaire fixité posée par la théorie. Il est notamment fait le constat qu'il n'y a pas *une* terminologie, qui représenterait *le* savoir sur le domaine, mais autant de terminologies que d'applications dans lesquelles ces terminologies sont utilisées (Bourigault & Slodzian, 1999). De plus, ces applications opérant le plus souvent sur des documents textuels (à des fins de traduction, d'indexation, *etc.*), et les connaissances d'un domaine s'exprimant généralement sous forme de documents écrits, la constitution de terminologies semble se définir désormais comme une tâche d'analyse de corpus textuels. Ces considérations contredisent donc la notion référentielle et unique du terme en l'inscrivant dans un cadre beaucoup plus pragmatique qu'est celui de l'utilité pour une application donnée. Par voie de conséquence, le nom, privilégié dans le cadre classique pour porter le terme, n'est plus forcément le seul élément de texte à prendre en compte dans cette nouvelle terminologie qualifiée de textuelle.

1.2.1.2 Différentes formes de termes

Distinction terme simple et terme complexe

On fait généralement la distinction entre termes simples et termes complexes. Les premiers sont composés d'un unique mot plein, comme un nom par exemple. Ils sont de ce fait plus susceptible d'être ambigus, mais en revanche ont un comportement syntaxique plus simple à modéliser, permettant de les acquérir et de les interpréter plus facilement. Les termes complexes sont quant à eux constitués d'au moins deux unités lexicales pleines. Ils ont des caractéristiques opposées à celles des termes simples : ils sont peu ambigus mais requièrent une analyse syntaxique plus fine pour être modélisés. Ils sont également plus difficiles à repérer, leurs différents constituants pouvant être séparés au sein de la phrase, et plus difficiles à interpréter. Les termes complexes, rendant mieux compte de la technicité d'un texte, sont plus couramment rencontrés dans un corpus spécialisé (Bourigault, 1992).

Variations de termes

La variation terminologique est un phénomène connu et très présent dans les textes et plutôt le fait des termes complexes. Les variantes d'un terme doivent bien sûr en partager la sémantique et donc renvoyer au même référent pour être valide. B. Daille (2002) propose une typologie des variations directement dérivée de ses travaux en acquisition de termes :

1. la variation graphique concernant principalement les changements de graphie (casse, absence ou présence d'un trait d'union...) ;
2. la variation flexionnelle (mise au pluriel d'un ou de plusieurs constituants d'un terme complexe) ;
3. la variation syntaxique faible, affectant les mots grammaticaux (comme *fixation d'azote* pour *fixation de l'azote*) ;
4. la variation syntaxique forte, modifiant la structure interne du terme (telle que *lait cru de brebis* pour *lait de brebis*) ;
5. la variation morphosyntaxique, modifiant la structure du terme et les mots qui le compose (comme *acidité du sang* et *acidité sanguine*) ;
6. la variation paradigmatique, échangeant un mot par un synonyme sans modification de la structure morphosyntaxique (*épuisement du combustible* pour *appauvrissement du combustible*) ;
7. la variation anaphorique, faisant référence à une mention préalable dans le texte (*procédé alimentaire* pour *procédé de conservation alimentaire*).

La prise en compte de la variation de termes, et plus généralement d'unités lexicales sémantiques, est un enjeu important de l'acquisition automatique. Il est en effet essentiel de pouvoir distinguer les variantes d'un même élément pour pouvoir, le cas échéant, les regrouper comme une seule unité.

1.2.2 Acquisition de termes

La plupart des définitions du terme données ci-dessus, même pragmatiques, sont non opératoires. Elles ne permettent donc pas de dériver une technique d'acquisition qui serait universelle. Cela explique sans doute la diversité des travaux effectués et des approches utilisées, ainsi que des communautés scientifiques s'intéressant à cette tâche d'acquisition (recherche d'information, ingénierie des connaissances...).

Nous tentons dans ce qui suit de définir un cadre unificateur à ces différentes approches. Nous examinons en particulier les présupposés communs à toutes ces techniques et définissons quelques notations utilisées par la suite. Nous présentons ensuite , sous un cadre formel commun, les grandes familles d'outils existants, en différenciant les approches numériques, structurelles et celles manipulant ces deux types d'information.

La tâche d'acquisition d'unités sémantiques lexicales (par exemple les termes d'un domaine) peut se représenter formellement par une fonction f telle que $f : \mathcal{D} \rightarrow T_{\mathcal{D}}$ où $\mathcal{D}$ est le domaine étudié et $T_{\mathcal{D}}$ l'ensemble des unités sémantiques lexicales de $\mathcal{D}$. Cette fonction f peut représenter le travail d'un expert du domaine, un processus automatique, ou bien encore un processus semi-automatique allié à un expert humain.

Dans ces deux derniers cas, la fonction f est en réalité utilisée sur un corpus $\mathcal{C}_{\mathcal{D}}$ représentatif du domaine. Même dans le cas où f représente une expertise humaine, celle-ci ne se faisant généralement pas *ex nihilo*, l'expert s'appuie sur un corpus. Il a en effet été constaté que l'hypothèse selon laquelle l'expert d'un domaine est le seul dépositaire d'un système conceptuel qu'il suffit de mettre au jour est non productive (Bourigault & Slodzian, 1999). Par ailleurs, la variabilité de terminologie au sein d'un même domaine évoquée ci-dessus semble contredire l'existence même d'une terminologie d'un domaine qui ne soit pas directement reliée à un corpus de ce domaine. Le problème de l'acquisition se récrit donc en $f : \mathcal{C}_{\mathcal{D}} \rightarrow T_{\mathcal{C}_{\mathcal{D}}}$ et, si la représentativité du domaine par le corpus est avérée, on espère avoir $T_{\mathcal{C}_{\mathcal{D}}} = T_{\mathcal{D}}$.

Enfin, la fonction f parfaite est en réalité impossible à atteindre, et ce même dans le cas où l'on fait appel à un expert humain. On approxime donc en pratique cette fonction d'acquisition par $\hat{f}$. Finalement, la problématique de l'acquisition est donc : $\hat{f} : \mathcal{C}_{\mathcal{D}} \rightarrow \hat{T}_{\mathcal{C}_{\mathcal{D}}}$ avec, on l'espère encore, $\hat{T}_{\mathcal{C}_{\mathcal{D}}} \simeq T_{\mathcal{C}_{\mathcal{D}}} \simeq T_{\mathcal{D}}$.

Cette pseudo-égalité montre les deux sources de problèmes auxquels on se heurte lorsque l'on souhaite construire une base d'informations sémantiques d'un domaine. Il faut dans premier temps constituer un corpus représentatif du domaine, c'est-à-dire couvrant exhaustivement le domaine ciblé et seulement ce domaine. Selon la méthode employée pour construire $\hat{f}$, le corpus $\mathcal{C}_{\mathcal{D}}$ peut nécessiter des caractéristiques supplémentaires comme par exemple la redondance (la même information est présente plusieurs fois dans le corpus sous des formes proches ; voir la section 1.2.2.1). Dans un deuxième temps, pour acquérir les informations voulues sur ce corpus, il faut trouver une méthode fiable, c'est-à-dire extrayant toutes les informations ciblées mais seulement celles-ci. Nous examinons ci-dessous plusieurs des diverses approches utilisées pour la construction de $\hat{f}$. La phase finale de la tâche d'acquisition est néanmoins commune à toutes ces approches : il s'agit de l'examen des propositions de $\hat{f}$, appelés candidats-

termes, par un expert qui les valide ou non en tant que termes.

On peut imaginer plusieurs façons de classer les différentes techniques d'acquisition d'informations, aucune d'elles n'offrant un découpage parfait. Nous choisissons pour notre part de considérer la nature des informations exploitées par les techniques pour les regrouper. Ainsi, nous examinons dans un premier temps les approches que l'on peut appelées numériques dans le sens où elles exploitent la nature fréquentielle des objets à acquérir et utilisent le plus souvent pour ce faire des techniques statistiques. Nous proposons ensuite un tour d'horizon des approches utilisant au contraire des informations structurelles ou symboliques pour acquérir les objets ciblés. Enfin nous présentons les techniques combinant explicitement ces deux approches.

Bien entendu, ce découpage du panorama des approches de l'acquisition, entièrement artificiel, ne saurait rendre compte de manière parfaite du caractère continu et complexe de ce spectre, certaines méthodes mélangeant de manière indissociable différentes méthodes et natures d'informations. Par ailleurs, cette section n'a pas vocation à présenter une liste exhaustive de tous les outils développés dans ce domaine, mais tente d'illustrer par des représentants pionniers ou célèbres les différentes techniques considérées.

1.2.2.1 Approche numérique

Les approches numériques d'acquisition de termes sur corpus ont été largement utilisées depuis de nombreuses années et elles continuent de connaître un grand succès. Elles sont aidées en cela par leur grande robustesse et par le fait que les documents informatisés sont de plus en plus facilement disponibles, rendant en cela la constitution de corpus volumineux — point de départ obligatoire de ces techniques purement quantitatives — plus aisée. Dans ces approches, $\hat{f}$ est en effet construite en exploitant la redondance de l'information terminologique. L'aspect fréquentiel utilisé est donc d'autant plus fiable et performant que le corpus est volumineux. Nous présentons ci-dessous deux familles de techniques très proches tirant parti de cet aspect fréquentiel : l'approche par cooccurrences et l'approche par segments répétés.

Approche par cooccurrences

De nombreux travaux cherchent à associer les mots apparaissant ensemble dans un texte de manière statistiquement significative. Les associations binaires ont notamment fait l'objet de travaux dans des buts lexicographiques (Church & Hanks, 1989 ; Church & Hanks, 1990 ; Church *et al.*, 1991). Les techniques utilisées reposent pour la plupart sur l'évaluation de la probabilité que les deux mots étudiés apparaissent ensemble dans une certaine fenêtre de texte plus souvent que le hasard ne l'aurait permis. Les entités trouvées par ce type de méthode sont d'ailleurs souvent dénommées *collocations*[3]

[3] Il existe de nombreuses définitions de collocation, parfois contradictoires ; nous proposons pour notre part de garder celle volontairement vague de J. Sinclair (1991) : « *collocation is the occurrence of two or more words within a short space of each other in a text* ».

pour mettre en relief cette notion d'apparition conjointe des composants dans le texte. L'hypothèse sur laquelle repose cette approche est que le contexte d'un mot apporte des informations sur le sens du mot, ce que J. R. Firth, cité par K. Church & P. Hanks (1989), exprime par : « *You shall know a word by the company it keeps* ».

Le principe opératoire de ces techniques est relativement simple :

1. il faut calculer pour chaque couple de mots un indice statistique — un score — mesurant la force du lien unissant ces deux mots ;
2. les couples finalement retenus sont ceux dont le score dépasse un seuil fixé.

Si la méthode est simple, les indices statistiques peuvent en revanche être extrêmement sophistiqués et variés. Parmi ceux utilisés dans ce cadre, un des plus célèbres est certainement celui de l'information mutuelle, adapté au besoin de l'acquisition de collocations par K. Church & P. Hanks (1989).

En reprenant les notations définies ci-dessus, ces techniques peuvent donc se formaliser sous la forme : $\hat{f}(\mathcal{C}_\mathcal{D}) = \hat{T}_{\mathcal{C}_\mathcal{D}} = \{\langle xy\rangle \in \mathcal{C}_\mathcal{D} \,|\, AS(P(x), P(y), P(x,y)...) > seuil\}$, où AS est un indice statistique mesurant l'association entre les constituants du candidat-terme binaire $\langle xy\rangle$ et les $P(x)$, $P(y)$, $P(x,y)$ sont les probabilités d'apparition de x, de y et de x et y ensemble[4] dans une certaine fenêtre sur $\mathcal{C}_\mathcal{D}$. Ces probabilités d'apparition étant *a priori* inconnues, elles sont généralement estimées à partir des fréquences relatives et conjointes des mots sur $\mathcal{C}_\mathcal{D}$ ou un extrait de $\mathcal{C}_\mathcal{D}$. Nous présentons dans la section 4.2.1.1 quelques indices usuellement utilisés pour l'extraction de cooccurrences.

Même si tous les candidats extraits sont effectivement de bons termes — $\hat{f}$ a alors une précision parfaite — on a néanmoins l'inclusion suivante : $\hat{T}_{\mathcal{C}_\mathcal{D}} \subseteq T_{\mathcal{C}_\mathcal{D}}$ (et plus certainement $\hat{T}_{\mathcal{C}_\mathcal{D}} \subset T_{\mathcal{C}_\mathcal{D}}$) puisqu'aucun des termes du domaine composés d'un mot ou de plus de deux mots ne peut être repéré par ce type de technique. Par ailleurs, des associations lexicales autres que les termes composés sont trouvés par cette approche, notamment les séquences répétitives telles que les locutions adverbiales ou prépositionnelles. Deux paramètres sont particulièrement influents dans ces techniques : le seuil à partir duquel un couple sera considéré pertinent, et la taille de la fenêtre choisie. Le premier paramètre est important puisqu'il détermine la qualité des résultats : on reproche souvent à ces techniques de ne pas réussir à détecter les phénomènes rares (associations pertinentes mais dont le nombre d'occurrences est trop faible pour dépasser le seuil du bruit). La taille de la fenêtre est également importante puisqu'il est en effet montré (Brown *et al.*, 1992) qu'une fenêtre petite (2 à 5 mots) favorise la détection de composés alors qu'une fenêtre plus grande (supérieure à 5 mots) permet d'observer des associations d'ordre paradigmatique ou sémantique entre les deux constituants. Ce dernier résultat explique que ce genre de méthodes soit également utilisé pour l'acquisition de relations sémantiques sur corpus.

[4] Suivant les cas, $P(x,y)$ représentera la probabilité d'apparition des deux mots ensemble quel que soit leur ordre, ou bien la probabilité d'apparition de x suivi de y au sein d'une fenêtre de texte (Church & Hanks, 1990).

Approche par segments répétés

Les techniques précédentes ont le défaut de n'extraire que des candidats-termes binaires. Pour contourner ce problème, l'approche dite des segments répétés (Lebart & Salem, 1994), développée en premier lieu dans un contexte lexicométrique, peut être utilisée. Son fonctionnement consiste à identifier dans le texte toute suite d'unités textuelles (segments répétés) reproduite sans variations à plusieurs endroits d'un corpus.

Ainsi, cette approche se formalise de la manière suivante : $\hat{f}(\mathcal{C}_\mathcal{D}) = \hat{T}_{\mathcal{C}_\mathcal{D}} = \{\langle x_1 x_2 ... x_n \rangle \in \mathcal{C}_\mathcal{D} \mid freq(\langle x_1, ..., x_n \rangle) > seuil\}$ où $\langle x_1 x_2 ... x_n \rangle$ est un segment répété de n composants, $freq(L)$ est la fonction indiquant la fréquence de la séquence L dans le corpus $\mathcal{C}_\mathcal{D}$, et *seuil* est une valeur numérique choisie par l'utilisateur. La forte similitude avec la formalisation des approches par cooccurrences souligne la parenté évidente de ces deux familles; on notera cependant que la notion de séquence, et donc d'ordre des mots, est explicitement formulée dans l'utilisation des segments répétés alors qu'elle n'est pas forcément prise en compte pour les cooccurrences.

Sans autre raffinement, comme notamment des restrictions sur les catégories de mots pouvant appartenir à un segment, cette méthode permet de repérer des objets linguistiques très hétérogènes comme des morceaux de syntagmes nominaux plus ou moins figés ou des fragments de texte récurrents mais peu intéressants (par exemple, le fragment *est un*) (Habert & Jacquemin, 1993). Les résultats sont donc trop bruités pour être utilisés directement dans un cadre de détections de termes, mais peuvent fournir un point de départ à d'autres techniques (voir par exemple l'approche utilisée par MANTEX page 33).

1.2.2.2 Approche symbolique

Nous l'avons vu, les définitions traditionnelles ou plus actuelles des termes sont non opératoires et ne peuvent donc directement être utilisées pour acquérir des candidats-termes. Néanmoins, un certain nombre de travaux s'appuient pour mener l'acquisition sur des indices structurels. Ces indices portent soit sur les constituants du terme, soit sur leur contexte, et peuvent être de nature différente : informations lexicales, morphologiques, syntaxiques ou sémantiques.

Les techniques d'acquisition structurelles exploitent principalement deux sources d'obtention de définitions des structures porteuses des termes. La première, la plus commune, est l'expertise linguistique; des définitions opérationnelles des termes, ou d'objets linguistiques proches, sont établies par des linguistes puis utilisées pour trouver les candidats-termes répondant à ces définitions. La seconde source de structures est moins utilisée; il s'agit de techniques d'apprentissage artificiel, qui proposent, en se basant souvent sur l'analyse d'exemples, des patrons d'extraction manipulant divers indices structurels.

Par expertise linguistique

TERMINO (David & Plante, 1990 ; David & Plante, 1991) est un outil pionnier de l'acquisition sur corpus de terminologie. Il est basé sur les travaux de É. Benveniste sur les synapsies, structures composées binaires (ou récursivement binaires) constituées d'un déterminé (tête) et d'un déterminant (expansion) qui se définissent selon un ensemble de traits (Benveniste, 1974) :

- la nature syntaxique (non morphologique) de la liaison entre les membres composant la synapsie ;
- l'emploi de joncteurs à cet effet, notamment *de* et *à* ;
- l'ordre déterminé et déterminant des membres ;
- leur forme pleine, et le choix libre de tout substantif ou adjectif ;
- l'absence d'article devant le déterminant ;
- la possibilité d'expansion pour l'un ou l'autre membre ;
- le caractère unique et constant du signifié.

Ainsi, à la différence de *garde-malade*, qui est un composé, *gardien d'asile* est une synapsie, ainsi que *gardien d'asile de nuit* (dont la décomposition synaptique en arbre binaire est ambiguë).

TERMINO acquiert dans un premier temps tous les syntagmes nominaux d'un texte à l'aide d'une analyse syntaxique des phrases. Ces syntagmes sont ensuite examinés pour en extraire les synapsies avec une grammaire dédiée opérant sur les catégories morphosyntaxiques des mots et sur les informations syntaxiques fournies (notamment les dépendances entre déterminés et déterminants). Enfin, un deuxième jeu d'heuristiques est utilisé pour supprimer, le cas échéant, certains compléments non pertinents au sein de ces synapsies.

Dans le prototype TERMS, J. Justeson et S. Katz (Justeson & Katz, 1995) utilisent une technique symbolique proche pour l'anglais. Ils proposent d'extraire les composés à partir d'une expression régulière sur les étiquettes catégorielles des mots. Cette même approche de patrons catégoriels a également été utilisée par I. Dagan et K. Church pour construire le module d'extraction de candidats-termes de leur outil TERMIGHT (Dagan & Church, 1997). L'expression caractérisant les termes peut également parfois porter sur d'autres types d'informations que les catégories. C'est le cas par exemple dans les travaux de U. Heid *et al.* (1996 ; 2000), qui proposent à l'utilisateur de spécifier lui-même l'expression recherchée grâce à leur système CQP, ou de A. Voutilainen (1993). L'outil d'acquisition mis au point par ce dernier, NPTOOL, utilise une approche à bases de règles pour extraire des syntagmes nominaux. Ces règles s'appuient sur l'analyse morphologique et la description syntaxique des phrases obtenues grâce à une technique à base de grammaires à contraintes (écrites à la main). Les règles sont des expressions régulières portant sur ces informations, les syntagmes obtenus sont les séquences maximales répondant à ces règles.

LEXTER (Bourigault, 1992 ; Bourigault, 1994) extrait lui aussi des candidats-termes sur des corpus étiquetés morphosyntaxiquement (à chaque mot est assigné sa catégorie :

nom, verbe, adjectif, *etc.*). Il utilise pour ce faire une approche duale des précédentes puisque les termes sont définis *en négatif*, c'est-à-dire en spécifiant les catégories de mots ne pouvant pas entrer dans la composition d'un terme. On a donc là encore une approche à base de règles dont la plupart sont fixées sur des considérations linguistiques mais aussi, pour quelques-unes, générées au besoin à partir du corpus. C'est une approche similaire qui est employée dans SYNTEX (Bourigault & Fabre, 2000) qui étend la couverture des dépendances prises en compte par LEXTER aux syntagmes verbaux et adjectivaux. Ces travaux sur LEXTER sont aussi à rapprocher de ceux de J. Royauté *et al.* (1992) développés dans une perspective d'extraction de descripteurs textuels pour l'indexation de documents.

Ces différentes approches ont pour point commun d'exploiter une sorte de langage, pour caractériser ce qu'est ou n'est pas un terme ; ce langage (noté $\mathscr{L}_G$) est la plupart du temps défini par une ensemble de règles G. Elles peuvent donc se formaliser par des formules du type : $\hat{f}(\mathcal{C}_\mathcal{D}) = \hat{T}_{\mathcal{C}_\mathcal{D}} = \{\langle x_1 ... x_n \rangle \in \mathcal{C}_\mathcal{D} \mid info(\langle x_1 ... x_n \rangle) \in \mathscr{L}_G\}$, où $info(S)$ est la fonction donnant les informations exploitées dans G (par exemple, les catégories des mots dans TERMS) d'une suite de mots S.

Par apprentissage

Dans ses travaux de thèse, É. Naulleau (1997 ; 1999) propose une approche originale pour extraire automatiquement d'un texte des syntagmes nominaux pertinents pour l'indexation de documents. Le principe de sa méthode est de généraliser par une technique d'apprentissage artificiel des *filtres* positifs (exemples de syntagmes intéressants) et négatifs (exemples de syntagmes non pertinents) fournis par l'utilisateur. Ces filtres et leurs généralisations exploitent des informations lexicales, morphologiques, catégorielles et sémantiques ajoutées au texte, et, appliqués au texte, doivent ainsi permettre de proposer des syntagmes conformes aux filtres positifs et ne répondant pas au filtres négatifs.

La technique de généralisation des filtres est assez rudimentaire et très contrainte pour limiter les problèmes de combinatoire. Elle se situe en effet dans un cadre propositionnel (les exemples sont décrits par des ensembles d'attributs-valeurs[5]), et l'expressivité des généralisations est restreinte. L'absence de formalisation de l'espace de recherche des généralisations ainsi défini conduit à certaines redondances et à un coût calculatoire heureusement réduit par de fortes contraintes sur la forme de ces généralisations.

Cette relative faiblesse de la technique d'apprentissage employée se traduit par des résultats seulement moyens alors que le nombre d'exemples (filtres) positifs et négatifs utilisés est gigantesque : certaines expériences comptent en effet près de 20 000 filtres positifs et autant de négatifs. Ces derniers chiffres semblent donc interdire toute portabilité aisée de cette approche.

[5] Un exemple de ce type de description est donné en section 2.2.1.4.

Ce type d'approche symbolique par apprentissage artificiel se formalise de la même façon que les autres approches symboliques : $\hat{f}(\mathcal{C}_\mathcal{D}) = \hat{T}_{\mathcal{C}_\mathcal{D}} = \{\langle x_1...x_n\rangle \in \mathcal{C}_\mathcal{D} \mid info(\langle x_1...x_n\rangle) \in \mathscr{L}_G\}$ à ceci près que les règles G définissant le langage $\mathscr{L}_G$ sont apprises à partir d'exemples (des mots de $\mathscr{L}_G$) et non plus définies manuellement par un expert.

1.2.2.3 Approche mixte

Certains outils combinent les deux approches présentées précédemment pour en conjuguer les avantages respectifs. Cette combinaison peut être réalisée par une simple juxtaposition des deux techniques — soit structurelle puis numérique, soit l'inverse — ou bien par un couplage plus intime.

Ces techniques sont de ce fait plus difficiles à modéliser simplement à l'aide de nos notations. On peut néanmoins considérer que dans le premier cas, on se trouve dans un phénomène de conjonction de deux fonctions d'acquisition f_1 et f_2, soit : $\hat{f} = \hat{f}_1 \wedge \hat{f}_2$ avec $\hat{f}_1 : \mathcal{C}_\mathcal{D} \rightarrow \hat{T}^{\hat{f}_1}_{\mathcal{C}_\mathcal{D}}$, $\hat{f}_2 : \mathcal{C}_\mathcal{D} \rightarrow \hat{T}^{\hat{f}_2}_{\mathcal{C}_\mathcal{D}}$ et $\hat{T}_{\mathcal{C}_\mathcal{D}} = \hat{T}^{\hat{f}_1}_{\mathcal{C}_\mathcal{D}} \cap \hat{T}^{\hat{f}_2}_{\mathcal{C}_\mathcal{D}}$. Dans le second cas, il s'agit plutôt d'une composition des fonctions : $\hat{f} = \hat{f}_1 \circ \hat{f}_2$ avec $\hat{f}_2 : \mathcal{C}_\mathcal{D} \rightarrow E$ et $\hat{f}_1 : E \rightarrow \hat{T}_{\mathcal{C}_\mathcal{D}}$ et $\hat{f}_1 \circ \hat{f}_2(x) = f_1(f_2(x))$, où E est un espace de représentation intermédiaire.

Approche structurelle suivie d'une approche statistique

Acabit de B. Daille (Daille, 1994) est un outil d'acquisition terminologique sur corpus dont le processus se décompose en deux étapes :

- un repérage linguistique des termes à l'aide de règles simples (par exemple, la règle $N(Prep(Det)^*)^*N$, ou N *à* $Vinf$) appliquées par des transducteurs au corpus étiqueté ; des mécanismes de variation permettent aussi d'extraire des variantes (*cf.* section 1.2.1.2) des termes (Daille, 1999 ; Daille, 2001) ;
- un filtrage statistique des candidats-termes retenus à l'étape précédente.

Plusieurs indices statistiques usuels ont été testés pour cette deuxième phase, et leurs performances ont été comparées et rapportées dans (Daille, 1994).

Formellement, en utilisant les notations définies précédemment, l'approche se modélise alors de la façon suivante : $\hat{f}(\mathcal{C}_\mathcal{D}) = \hat{T}_{\mathcal{C}_\mathcal{D}} = \{\langle x_1...x_n\rangle \in \mathcal{C}_\mathcal{D} \mid (cat(\langle x_1...x_n\rangle) \in \mathscr{L}_G) \wedge freq(\langle x_1, ..., x_n\rangle) > seuil\}$, où $cat(S)$ indique la séquence d'étiquettes catégorielles correspondant à la séquence de mots S.

Ana (pour Acquisition Naturelle Automatique) est un système d'extraction de candidats-termes (Enguehard, 1992 ; Enguehard & Pantera, 1995) reposant ouvertement sur une procédure excluant toute analyse linguistique. Il n'utilise en effet ni lexique, ni informations syntaxiques, et se veut donc indépendant de la langue. La reconnaissance en corpus des termes est effectuée à l'aide d'une observation de répétitions de patrons, identifiés au départ à partir d'un petit ensemble de termes complexes, et d'un calcul d'égalités souples (basé sur la distance d'édition ou distance de Levenshtein) entre mots permettant ainsi de se passer de lemmatisation.

MANTEX (Oueslati, 1999) s'inscrit également dans une utilisation d'une approche numérique suivie d'une technique structurelle. Plus précisément, il emploie la méthode d'extraction des segments répétés présentée précédemment mais seules les séquences de mots ne contenant pas de mots grammaticaux, de ponctuations ou de verbes sont retenues. Il s'agit donc là d'un filtrage préalable sur des informations structurelles des segments candidats qui sont ensuite répertoriés, avec leur fréquence d'apparition, dans une liste proposée à un expert du domaine si cette fréquence dépasse 2. Cette légère amélioration de l'approche de L. Lebart et A. Salem a malheureusement le même défaut de reposer en grande partie sur la fréquence d'apparition, ce qui élimine toute possibilité de découverte de phénomènes rares dans le corpus.

Approche statistique suivie d'une approche structurelle

L'outil d'extraction XTRACT (Smadja, 1993a ; Smadja, 1993b) emploie une technique inverse à celles que nous venons de voir. Il effectue en effet dans un premier temps un repérage statistique de mots cooccurrents puis un étiquetage de ces collocations grâce aux informations syntaxiques fournies par l'étiqueteur CASS (Abney, 1990). Plus précisément, le processus comporte les trois étapes suivantes :

1. extraction des collocations par une technique semblable à celle exposée en 1.2.2.1 ;
2. expansion des collocations en répétant itérativement l'étape précédente avec les collocations déjà trouvées ;
3. étiquetage des collocations repérées à l'aide des informations syntaxiques données par l'analyseur CASS et de patrons spécifiés par l'utilisateur (du type verbe-objet). Un couple dont les occurrences sont majoritairement dans une certaine relation (ce seuil est fixé à 80% dans XTRACT) est retenu comme représentant de cette relation.

Bien que XTRACT soit principalement un extracteur de collocations et non pas seulement de termes, son fonctionnement est prototypique des approches mêlant, dans cet ordre, les techniques statistiques aux techniques symboliques.

Imbrication complexe des approches structurelles et numériques

Développé en premier lieu dans un but d'indexation de documents, le système CLARIT (Evans & Zhai, 1996) cherche à acquérir des séquences de mots décrivant au mieux le contenu d'un document. À ce titre, il tente d'extraire des unités telles que des termes complexes et présente de nombreux points communs avec des techniques plus spécifiquement dédiées à l'acquisition de terminologie.

Le principe de CLARIT est le suivant :

1. extraction de tous les syntagmes nominaux et étiquetage catégoriel des constituants ;
2. analyse en dépendances des syntagmes en s'appuyant sur les catégories des mots et sur les formes attestées trouvées par ailleurs dans le corpus ;

3. génération des termes possibles à partir de l'analyse des syntagmes.

Durant l'étape 1, un processus itératif permet de repérer ce que les auteurs nomment des atomes lexicaux (comme *hot dog*, *part of speech*) en comparant les fréquences d'apparition de leurs constituants. De plus, seules certaines successions de catégories sont autorisées dans ces atomes lexicaux (nom-nom, adjectif-nom, atome lexical-nom...). Dès qu'une paire est détectée comme atomique, elle est ensuite considérée comme un seul mot et le processus est relancé; cela permet de trouver des atomes lexicaux de taille de plus en plus importante. L'analyse en dépendances de l'étape 2 a pour but de regrouper deux à deux les mots adjacents d'un syntagme pour trouver la configuration la plus restrictive et informative au vu du corpus. Enfin, l'étape 3 génère, en se basant sur l'analyse du syntagme, les termes d'indexation répondant à certains schémas jugés intéressants par les auteurs dans leur cadre de recherche d'information. Les résultats obtenus sont de bonne qualité et évalués directement par rapport à leur besoin applicatif; ils montrent ainsi que les performances de leur système sont améliorés par l'emploi de ces termes d'indexation complexes.

Ce mélange entre nature structurelle et numérique des données textuelles se retrouve dans les travaux de K. Church (Church, 1988). Ce dernier décrit une méthode permettant de trouver les frontières des syntagmes nominaux à partir de textes en utilisant une technique similaire à celle employée pour construire des étiqueteurs morphosyntaxiques stochastiques. Ces travaux sont à rapprocher de ceux de L. Ramshaw et M. Marcus (1995), qui se proposent également de faire le repérage de syntagmes nominaux en le ramenant à un exercice d'étiquetage. Cependant, la technique utilisée dans leur cas s'appuie sur l'utilisation des règles de transformation comme cela est fait dans l'étiqueteur de E. Brill (Brill, 1992 ; Brill, 1994). Les travaux de K. Frantzi *et al.* (1996 ; 2000) s'inscrivent aussi dans ce type d'approche hybride puisque leur outil d'acquisition de termes exploite à la fois des informations structurelles et numériques sur le corpus.

1.3 Relations sémantiques

Un lexique sémantique est composé d'unités lexicales, dont nous venons de présenter les principales approches d'extraction, mais aussi de relations entre ces unités. Celles-ci structurent le lexique (Czap & Nedobity, 1990 ; Skuce & Meyer, 1991) et exhibent les liens sémantiques entre les différentes unités lexicales. Ces relations sémantiques permettent donc également d'accéder au sens d'une unité lexicale en la comparant, à travers ces relations, à d'autres unités (Cruse, 1986).

Nous proposons ci-dessous de définir les relations sémantiques et leurs propriétés, linguistiquement dans un premier temps, puis plus formellement. Nous examinons ensuite les différentes approches existantes utilisées pour l'aide à l'acquisition ou l'acquisition automatique de ces relations, en nous attachant notamment à mettre en valeur leurs avantages et défauts.

1.3.1 Définition des relations sémantiques

Les relations sémantiques, et plus particulièrement certaines d'entre elles, ont été largement étudiées, tant d'un point de vue formel que dans une optique de linguistique de corpus. Nous en proposons ci-dessous une définition usuelle, mettant en particulier en exergue la différence entre relations syntagmatiques et paradigmatiques (Cruse, 1986). Nous présentons ensuite, à travers la notion mathématique de relations entre ensembles, le formalisme permettant de modéliser les relations sémantiques et leurs propriétés.

1.3.1.1 Types de relations sémantiques

Il existe plusieurs types de relations sémantiques, offrant ainsi différents liens pour structurer les bases de connaissances lexicales. On distingue en particulier les relations sémantiques portées dans le texte par un prédicat — celui-ci pouvant être explicite ou implicite — et celles dont le prédicat n'est pas exprimé (Cruse, 1986). Le premier type de lien, reposant sur les propriétés syntaxiques des constituants du couple en relation ou de leur contexte, est qualifié de syntagmatique. Dans le second cas, on parlera plutôt de relations paradigmatiques.

Relations syntagmatiques

Au sein des textes, certaines formes syntaxiques suggèrent l'existence d'un lien sémantique entre deux mots. Ainsi, les verbes accompagnés des éléments de leurs structures argumentales peuvent être considérés comme des prédicats dotés d'arguments dénotant donc d'une relation particulière. D'autres structures syntaxiques sont également l'indice de relations syntagmatiques. Par exemple, une phrase contenant *l'effet de X sur Y* indique clairement une relation entre les entités X et Y. On retrouve ce même lien dans des variantes syntaxiques impliquant des verbes support (*X a un effet sur Y*), ou d'autres schémas syntaxiques. Il y a ainsi plusieurs patrons équivalents pour dénoter une même relation prédicative entre mots.

Dans cette optique, H. Robison (Robison, 1970) a étudié près de 8 000 relations syntagmatiques explicites (qu'il appelle patrons primaires) et leurs variantes (patrons secondaires). Les patrons primaires indiquent les prédicats reliant les mots considérés mais peuvent être ambigus; l'étude du patron secondaire sert alors à lever l'ambiguïté sur le sens de ce prédicat.

Le prédicat reliant plusieurs mots n'apparaît pas toujours aussi clairement dans un texte. Une relation sémantique peut par exemple s'établir entre les constituants d'un composé multinominal. L'explicitation du prédicat requiert alors une analyse fine des constituants du composé.

C'est dans ce cadre que se placent les travaux de C. Fabre (Fabre, 1996 ; Fabre & Sébillot, 1999) qui a développé une technique permettant l'interprétation automatique des composés de la forme nom-nom pour l'anglais et nom-préposition-nom pour le français. Plus précisément, l'analyse du composé doit fournir le prédicat et les rôles

sémantiques de chacun des constituants. Dans le cas le plus simple, le prédicat apparaît sous forme d'un déverbal comme dans *tondeuse à gazon*, *filtre à air* ou *détecteur de choc*. Pour ce dernier exemple, l'analyse produit alors une représentation de la relation de ce type : détecter(instrument : *détecteur*, objet : *choc*). Si le prédicat n'est pas directement accessible, il faut alors disposer d'une connaissance sémantique supplémentaire pour pouvoir interpréter le composé. Considérons par exemple le composé *bread knife* ; l'interprétation naturelle que l'on voudrait lui associer est cut(instrument : *knife*, objet : *bread*). Cela n'est possible que si l'on sait que la fonction typique de *knife* est *to cut*. C. Fabre propose d'utiliser les connaissances sémantiques codées dans le Lexique génératif (Pustejovsky, 1995) pour avoir accès à ces informations prédicatives implicites (voir section 2.3.1).

Relations paradigmatiques

Certaines relations sémantiques entre mots n'apparaissent pas sous forme de lien syntaxique standard au sein des textes, mais relèvent d'une association paradigmatique (Grefenstette, 1994a) (comme la synonymie par exemple). Dans ces relations, la notion précédente de prédicat n'est pas pertinente ; elles sont donc parfois également appelées relations non prédicatives (Morin, 1999).

A. Cruse (1986) s'appuie sur une interprétation ensembliste pour définir des relations primaires qu'il nomme *congruences* permettant de caractériser certaines relations paradigmatiques. Ainsi, la synonymie correspond à la relation d'identité entre ensembles de mots ; l'hyponymie correspond quant à elle à une relation d'inclusion.

Il existe bien sûr beaucoup de relations paradigmatiques, opérant essentiellement sur des unités de même catégorie (de nom à nom par exemple). Le modèle de A. Cruse permet de les représenter en combinant les opérations ensemblistes ; il les appelle *variantes de congruence*. Parmi ces dernières, notons que les quasi relations permettent de modéliser des relations entre unités lexicales de catégories différentes. Ainsi, le participe passé *coloré* est quasi hyperonyme de *rouge* et *jaune*.

Les relations paradigmatiques induisent le plus souvent une structure particulière sur l'espace des unités sémantiques (Cruse, 1986). Ainsi, une relation hiérarchique telle que l'hyperonymie, l'hyponymie, la méronymie (relation partie-de) sur des termes impose une représentation sous forme de taxinomie des unités lexicales. On parle parfois pour ce type de relation de liens verticaux. Ces relations hiérarchiques sont d'ailleurs parfois considérées comme les plus intéressantes (Condamines & Amsili, 1993). Elles permettent en effet, grâce à l'organisation arborescente qu'elles induisent, de refléter l'organisation hiérarchique des concepts portés par les unités sémantiques que l'on retrouve dans les ontologies (Rastier, 1995). D'autres relations sont au contraire symétriques (ou quasi-symétriques), comme par exemple la synonymie et l'antonymie. Les liens qu'elles permettent d'établir dans l'espace des unités sémantiques sont alors qualifiés de transversaux ou d'horizontaux.

L'acquisition de ce type de relations est difficile : il faut non seulement les détecter, mais aussi les identifier, c'est-à-dire préciser le lien sémantique existant entre les divers

constituants. Comme nous le montrons ci-après en section 1.3.2, un tel travail peut soit exploiter les similitudes de contextes de certains mots, soit recourir à des patrons lexicaux, morphosyntaxiques ou sémantiques.

1.3.1.2 Représentation formelle d'une relation

On peut représenter la notion de relation, notamment de relation sémantique, par un formalisme issu des mathématiques. Nous proposons ci-dessous une définition abstraite de cet objet mathématique.

Définition 1 *Soient A et B deux ensembles. On appelle relation sur $A \times B$ tout prédicat défini sur $A \times B$. Si $(x, y) \in A \times B$ et si $\mathcal{R}$ est un prédicat à valeur dans $A \times B$, on notera $\mathcal{R}(x, y)$ ou plus simplement $x\mathcal{R}y$ pour signifier que x est en relation $\mathcal{R}$ avec y.*

Une relation définit donc un sous-ensemble de $A \times B$. Dans le cas où l'on s'intéresse aux relations entre termes d'un domaine, les ensembles A et B sont égaux à l'ensemble des termes T. La relation $\mathcal{R}$ se définit alors sur T^2 ; une telle relation sur T^2 est dite binaire.

Une relation sur $A \times B$ définit un graphe inclus dans $A \times B$, formellement présenté dans la définition 2.

Définition 2 *On appelle graphe d'une relation $\mathcal{R}$ le sous-ensemble $G_\mathcal{R}$ de $A \times B$, déterminé par :*

$$G_\mathcal{R} = \{ (x, y) \in A \times B \mid x\mathcal{R}y \}$$

c'est-à-dire l'ensemble des couples dont les composantes sont en relation.

Il est important de noter que se donner un sous-ensemble de $A \times B$ revient exactement à définir une relation. Ainsi, dans le cadre des relations sémantiques, se fixer le graphe d'une relation correspond à définir cette relation en extension (c'est-à-dire uniquement par l'exemple) sans avoir besoin d'expliciter le lien ciblé (ce qui correspondrait à la définir en intention).

On définit ci-dessous quelques propriétés classiques des relations appliquées à notre cadre de relations sémantiques.

Propriété 1 *Soient $\mathcal{R}$ une relation entre unités sémantiques et T l'ensemble des unités sémantiques ; on définit les propriétés suivantes :*

- *$\mathcal{R}$ est symétrique si $\forall (x, y) \in T^2, x\mathcal{R}y \Rightarrow y\mathcal{R}x$*
- *$\mathcal{R}$ est antisymétrique si $\forall (x, y) \in T^2, x\mathcal{R}y \wedge y\mathcal{R}x \Rightarrow x = y$*
- *$\mathcal{R}$ est réflexive si $\forall x \in T, x\mathcal{R}x$*
- *$\mathcal{R}$ est antiréflexive si $\forall x \in T, \neg(x\mathcal{R}x)$*
- *$\mathcal{R}$ est transitive si $\forall (x, y, z) \in T^3, x\mathcal{R}y \wedge y\mathcal{R}z \Rightarrow x\mathcal{R}z$*

□

Parmi la variété des relations possibles, deux types de relations particulières sont très utilisés :

- $\mathcal{R}$ est une *relation d'équivalence* si elle est réflexive, symétrique et transitive ;
- $\mathcal{R}$ est une *relation d'ordre* si elle est réflexive, antisymétrique et transitive.

Lorsque $\mathcal{R}$ est une relation d'équivalence, on définit par ailleurs les *classes d'équivalence* de x modulo $\mathcal{R}$ par : $\bar{x} = \{y \in T | x\mathcal{R}y\} = \{y \in T | y\mathcal{R}x\}$. Par exemple, si la relation $\mathcal{R}$ représente la synonymie, $\bar{x}$ est l'ensemble des synonymes d'un terme x.

On définit également, pour une relation binaire d'équivalence, les *ensembles quotients* $T/\mathcal{R}$, avec $\mathscr{P}(T)$ l'ensemble des parties de T par : $T/\mathcal{R} = \{\mathcal{C} \in \mathscr{P}(T) \mid \exists x \in T$ tel que $\mathcal{C} = \bar{x}\}$. Si $\mathcal{R}$ désigne comme dans l'exemple précédent la synonymie, les ensembles quotients de $\mathcal{R}$ représentent une partition de T en classes (*clusters*) de termes synonymes.

Les relations hiérarchiques comme l'hyperonymie ou la méronymie correspondent dans ce cadre plutôt à des relations d'ordre (partiel) sur T. En effet, si $\mathcal{R}$ est une relation d'hyperonymie, en supposant que l'on ait *végétal* $\mathcal{R}$ *arbre* (un arbre est une sorte de végétal) et *arbre* $\mathcal{R}$ *chêne* (un chêne est une sorte d'arbre), alors on sait que *végétal* $\mathcal{R}$ *chêne* (transitivité), que l'on n'a pas *arbre* $\mathcal{R}$ *végétal* (antisymétrie) et que *arbre* $\mathcal{R}$ *arbre* (réflexivité).

1.3.2 Acquisition de relations sémantiques

Nous présentons dans cette section les différentes familles de travaux portant sur l'acquisition de relations sémantiques sur corpus, en nous efforçant comme précédemment d'utiliser un cadre formel commun à leur description. Nous reprenons également l'approche utilisée en section 1.2.2, en distinguant les méthodes exploitant l'aspect fréquentiel du corpus et celles s'appuyant sur des indices structurels pour détecter les relations sémantiques.

1.3.2.1 Approche numérique

Beaucoup de travaux exploitant des techniques statistiques ont été menés en acquisition de relations sémantiques (pour une vue de domaine, voir (Grefenstette, 1994b ; Pichon & Sébillot, 1997 ; Habert *et al.*, 1997)). L'idée sur laquelle reposent ces méthodes est de détecter les associations statistiquement significatives, c'est-à-dire plus fréquentes que du fait du hasard. Ces associations permettent soit de mettre au jour directement des mots en relation sémantique — c'est l'approche statistique décrite ci-dessous —, soit d'étudier les mots partageant les mêmes associations — cette approche est alors qualifiée de distributionnelle et est décrite en section 1.3.2.1.

Statistique

Comme nous l'avons souligné en section 1.2.2.1, certains indices statistiques d'association sont parfois utilisés pour la détection non plus de termes complexes mais de

relations sémantiques. Le principe méthodologique est le même que celui précédemment exposé, à ceci près que les fenêtres utilisées pour calculer les cooccurrences sont de tailles souvent plus importantes. Les relations mises au jour par ce type de méthodes sont généralement syntagmatiques ; on les note $\mathcal{R}_\mathcal{S}$ ci-après.

Par ce type d'approche, on estime donc la relation $\mathcal{R}_\mathcal{S}$ par $\hat{\mathcal{R}_\mathcal{S}}$ que l'on définit par $x\hat{\mathcal{R}_\mathcal{S}}y$ si $f(P(x), P(y), P(x,y)...) > seuil$; ces probabilités sont elles-mêmes estimées à l'aide des fréquences d'apparition dans le corpus : $\hat{P}(x) = freq(x)$. On fait ensuite le postulat que $\forall(x,y) \in T^2, x\hat{\mathcal{R}_\mathcal{S}}y \Rightarrow x\mathcal{R}_\mathcal{S}y$. Il faut noter que suivant les indices statistiques utilisés, la relation $\mathcal{R}_\mathcal{S}$ sera symétrique ou non. Ce type d'approche produit généralement des résultats bruités et hétérogènes puisqu'il ne permet pas de typer les relations obtenues.

Analyse distributionnelle

À partir de l'hypothèse harrissienne (Harris *et al.*, 1989) selon laquelle une analyse distributionnelle des *propriétés contextuelles* (nous précisons ci-après ce que ces propriétés peuvent être) des mots fait apparaître des classes de concepts (regroupant les mots partageant les mêmes propriétés) et des relations entre elles, beaucoup de travaux ont vu le jour (voir (Grefenstette, 1994b)). Ces travaux s'attachent donc à faire ressortir des textes des relations dites paradigmatiques (que l'on note $\mathcal{R}_\mathcal{P}$ par opposition au relations syntagmatiques précédentes).

La plupart de ces méthodes s'appuient sur une procédure en trois temps (que nous adaptons à partir de celle proposée par G. Grefenstette (1994a)) :

1. recherche des propriétés contextuelles de chaque mot du corpus ;
2. mise en relation deux à deux de mots partageant les mêmes propriétés contextuelles ;
3. construction de classes à partir des relations découvertes à l'étape 2.

Cette définition volontairement large ne précise ni les propriétés contextuelles étudiées, ni la façon dont elles sont recherchées, ni la mise en relation des mots partageant les mêmes propriétés, ni enfin ce qui est précisément entendu par la construction de classes.

Concernant les deux premiers points, les propriétés considérées varient selon les travaux. Ce peut être le contexte syntaxique des mots (Faure & Nédellec, 1999) ou les relations de dépendance tête-expansion au sein de syntagmes nominaux (Bouaud *et al.*, 1997 ; Habert & Fabre, 1999) ou de tout syntagme (Bourigault, 2002), ou d'une relation quelconque spécifiée par l'utilisateur (Grefenstette, 1992). Cela peut également être les mots cooccurrant dans une certaine fenêtre (Grefenstette, 1994b ; Pichon & Sébillot, 1999 ; Rossignol & Sébillot, 2002) ou les segments répétés (Rousselot *et al.*, 1996) ou encore les mots d'un même domaine sémantique (de Chalendar & Grau, 2000 ; de Chalendar, 2001).

L'appariement deux à deux des mots suivant leurs propriétés partagées (et aussi celles non partagées) est une phase complexe influençant ensuite la phase de construction de classes homogènes. En effet, si l'on note $Prop(x)$ l'ensemble des propriétés

considérées du mot x (par exemple l'ensemble des dépendances syntaxiques dans lesquelles il apparaît), alors on définit la relation $\mathcal{R}_\mathcal{P}$ sur base d'analyse distributionnelle par la formule[6] : $x\mathcal{R}_\mathcal{P}y \iff (Prop(x) = Prop(y))$. Ainsi définie, la relation trouvée par analyse distributionnelle serait une relation d'équivalence (c'est-à-dire symétrique, réflexive et transitive). La phase 3 consiste donc à construire les classes d'équivalence et à produire ainsi l'ensemble quotient censé représenter la partition de l'espace des termes en classes conceptuelles.

Malheureusement, la définition précédente, trop contraignante sur le partage complet des propriétés contextuelles, n'est pas celle utilisée en pratique car elle aurait pour effet de n'assembler que très peu de termes. Généralement, on emploie une définition plus lâche indiquant qu'une *majorité* des propriétés contextuelles doit être partagée pour que deux mots soient déclarés en relation ; cela peut se traduire plus formellement par : $x\mathcal{R}_\mathcal{P}y \iff |Prop(x) \cap Prop(y)| > seuil$ avec $|\cdot|$ indiquant le cardinal d'un ensemble. Cette définition plus souple a bien sûr pour effet de mettre en relation plus de mots mais la transitivité de la relation est perdue (*i.e.* on peut avoir $x\mathcal{R}_\mathcal{P}y$ et $y\mathcal{R}_\mathcal{P}z$ et pas $x\mathcal{R}_\mathcal{P}z$). Le regroupement en classes ne peut donc plus se faire par classes d'équivalence ; c'est pourquoi ont été proposées des structures plus faibles que ces dernières pour modéliser les classes conceptuelles attendues (cliques, composantes connexes, *etc.*).

Chacune de ces formes de regroupements semble faire ressortir des informations de nature différente (Bouaud *et al.*, 1997 ; Bourigault, 2002) mais aucune ne permet de n'obtenir que des classes homogènes ni d'isoler un type de relation sémantique fixé. L'interprétation des classes obtenues est dans la plupart des cas laissée au soin de l'utilisateur. Notons enfin que les différents regroupements obtenus peuvent également se faire sous forme d'arbres ou de pyramides à l'aide de techniques de classification hiérarchique (Assadi, 1998 ; Rossignol & Sébillot, 2002 ; Agarwal, 1995 ; Faure & Nédellec, 1999) : la racine contient une unique classe conceptuelle (certainement très hétérogène) et à l'inverse les feuilles sont des classes très spécialisées.

1.3.2.2 Approche symbolique

Par opposition aux techniques numériques précédentes, nous présentons ci-dessous des approches symboliques d'acquisition de relations sémantiques. Ces approches peuvent elles-mêmes se classer en deux grandes familles : les approches linguistiques, où les indices structurels exploités sont donnés *a priori* (par une analyse linguistique par exemple) et les approches basées sur une notion d'apprentissage (artificiel ou non). Il faut noter, comme pour l'acquisition d'éléments terminologiques, que ce type d'approche travaille le plus souvent au niveau des occurrences des couples de mots en relation. Chaque occurrence est donc individuellement classée comme porteuse d'une relation ; on note $occ_i(x, y)$ la i^{e} occurrence d'un couple (x, y) dans un corpus.

[6] On remarque que dans le cas où $Prop(x)$ représente l'ensemble des cooccurrents de x (l'ensemble $\{z \mid x\mathcal{R}_\mathcal{S}z\}$ en reprenant les notations précédentes), $x\mathcal{R}_\mathcal{P}y \iff (\forall z(x\mathcal{R}_\mathcal{S}z \iff y\mathcal{R}_\mathcal{S}z))$. La relation recherchée par l'analyse distributionnelle se construit donc à partir d'une première relation syntagmatique. C'est cette approche en deux temps qui justifie l'appellation d'*affinités du deuxième ordre* donnée par G. Grefenstette (1994a) à ce type de relation.

Approche linguistique

Le système d'acquisition SEEK (Jouis, 1995) est prototypique des systèmes fondés sur une expertise linguistique. Il fonctionne à partir d'un corpus lemmatisé et nécessite une grande intervention humaine. À l'aide de règles dites d'exploration contextuelle (Jouis, 1993), le système détecte des couples de mots en relation binaire. Ces relations sont variées (plus d'une vingtaine au total) — elles représentent par exemple la notion d'inclusion, d'identification, d'appartenance, de localisation ou de tout à partie — mais statiques. Les couples détectés sont ensuite proposés à l'utilisateur. Ce dernier les rejette ou les valide comme représentant de la relation proposée ; dans ce dernier cas, il doit manuellement indiquer quels sont les arguments de cette relation. Le système propose enfin un graphe représentant l'ensemble des relations ainsi acquises.

Les règles d'exploration contextuelles reposent sur l'identification de marqueurs linguistiques (principalement lexicaux) et sont construites manuellement. Ce sont ainsi plus de 220 règles de la forme SI <condition de co-présence de marqueurs linguistiques> ALORS <actions> OU <conclusions>, manipulant plus de 3 300 marqueurs linguistiques qui sont utilisées par SEEK.

Plus récemment, D. Garcia propose à travers son système COATIS (Garcia, 1998 ; Garcia *et al.*, 2000) une approche similaire mais en se focalisant sur la relation de causalité. Avec ATERM, R. Oueslati (1999) propose également une technique proche pour la détection de couples en relations sémantiques. Seul le degré d'interactivité de son outil le différencie de SEEK ou COATIS. En effet, ATERM est une interface permettant au linguiste de spécifier à tout moment les patrons qu'il souhaite voir utilisés. Ces patrons sont exprimés dans un langage dédié, LEXICA, manipulant les lemmes, catégories ou formes fléchies des mots du corpus. Enfin, les travaux de C. Jacquemin (Jacquemin, 1996 ; Jacquemin, 1997 ; Jacquemin, 2001) sur FASTR peuvent également s'inscrire dans cette approche. Ce dernier acquiert des variantes morphosyntaxiques de termes (la variation peut donc être vue comme une relation d'équivalence) à l'aide de plusieurs niveaux de règles (les règles opérant sur d'autres règles sont appelées méta-règles).

Dans l'ensemble de ces travaux, on estime la relation $\mathcal{R}$ par $\hat{\mathcal{R}}$ définie comme un ensemble de règles données par un expert. Plus généralement, $\hat{\mathcal{R}}$ est souvent la réunion de n règles R_1,... R_n, définissant chacune un aspect de la relation et manipulant des indices pouvant être de différente nature (marqueurs lexicaux, catégories morphosyntaxiques...). Ainsi, un couple de mots (x, y) est considéré en relation si une (ou un nombre minimal) de ses occurrences répond à une des règles définies. En notant $Desc(o)$ la description d'une occurrence o (par exemple la séquence d'étiquettes catégorielles de cette occurrence), cela peut se transcrire par :

$$x\hat{\mathcal{R}}y \iff \exists i \; Desc\,(\,occ_i(x,y)\,) \in \mathscr{L}_R \;\text{ avec }\; \mathscr{L}_R = \bigcup_{1 \leq l \leq n} \mathscr{L}_{R_l}.$$

Il faut remarquer par ailleurs que suivant la relation étudiée, $\mathcal{R}$ n'est pas forcément symétrique.

Le postulat de base de ces outils est de supposer que les relations statiques décrites sont suffisamment génériques pour ne pas dépendre d'un domaine en particulier. Malheureusement, les expériences rapportées dans (Jouis *et al.*, 1997) montrent que les résultats d'extraction sur un plus gros volume de textes sont entachés de bruit (relations détectées et non pertinentes). Il semble donc que certains de ces travaux soient en réalité difficilement portables d'un domaine à un autre sans opérer de lourdes modifications manuelles dans la base de règles. De la même façon, l'ajout d'un nouveau type de relation nécessite de découvrir et d'insérer dans ce type de système complexe de nouvelles règles contextuelles décrivant cette relation. La portabilité et l'utilisation à grande échelle sur des textes variés de ces techniques semblent donc très problématiques et coûteuses. Enfin, dans ces travaux, aucune discussion n'est faite de l'expressivité des règles à base de marqueurs lexicaux et de leur pouvoir de représentation des relations.

Approche par apprentissage symbolique

L'acquisition de relations sémantiques par apprentissage de patrons d'extraction lexico-syntaxiques (ou LSPE en anglais pour *lexico-syntactic pattern extraction*) est l'une des techniques les plus connues. L'idée principale de cette approche est d'identifier dans un corpus les marqueurs ou indices d'une relation sémantique sur un petit ensemble d'exemples pour ensuite les réutiliser pour extraire de nouvelles unités en relation. Cette approche a été initiée par M. Hearst (Hearst, 1992 ; Hearst, 1998) et formalisée en cinq étapes :

1. choisir une relation cible $\mathcal{R}$;
2. réunir une liste de paires en relation $\mathcal{R}$ (par exemple les extraire d'un thésaurus, d'une base de connaissances) ;
3. retrouver les phrases du corpus contenant ces paires et enregistrer leurs contextes lexical et syntaxique ;
4. trouver les points communs entre ces contextes et supposer que cela forme un schéma lexico-syntaxique de $\mathcal{R}$;
5. appliquer les schémas pour obtenir de nouvelles paires et retourner en 3.

Elle se différencie de la méthode exposée précédemment par le fait que les marqueurs de la relation (ici les informations lexicales et catégorielles) sont issus d'une analyse d'exemples et non plus d'une connaissance linguistique *a priori*.

La relation $\mathcal{R}$ n'est donc pas définie explicitement au départ mais par la donnée d'exemples. Cela permet de s'intéresser à des relations connues partiellement en extension (par des instances) mais pas en intention (non formalisée par une règle). L'algorithme, et en particulier la phase 4 qui reste la plus vague, propose d'estimer une définition $\hat{\mathcal{R}}$ explicite de $\mathcal{R}$ en généralisant les exemples. On fait ensuite le postulat que $\forall (x, y) \in T^2, x\hat{\mathcal{R}}y \Rightarrow x\mathcal{R}y$.

Cette technique a été utilisée avec succès pour la relation d'hyponymie. Pour cela, M. Hearst s'est servi de WordNet pour générer une liste de paires candidates en relation d'hyponymie (étape 2). En revanche, l'étude d'autres types de relations (comme

la méronymie) semble avoir donné de moins bons résultats du fait de l'obtention de patrons trop généraux.

La phase 4 est entièrement manuelle dans (Hearst, 1992) ; la généralisation des structures lexico-syntaxiques des phrases en patrons d'extraction est donc faite par l'utilisateur. M. Hearst suggère une automatisation de cette étape (Hearst, 1998) par différentes techniques dont l'apprentissage artificiel mais qui ne sont pas mises en œuvre.

C'est cette phase de généralisation que E. Morin se propose d'automatiser (Morin, 1999 ; Morin, 1997) avec son système PROMÉTHÉE. Pour ce faire, il s'appuie sur un calcul de similarité (Morin, 1998) deux à deux entre les contextes lexico-syntaxiques de deux occurrences de paires. Plusieurs mesures de similarité sont proposées, mais toutes s'appuient sur une description lexicale de la phrase. Des classes de contextes lexico-syntaxiques sont ainsi constituées. Un schéma représentatif de chaque classe est ensuite choisi (il s'agit d'un des contextes lexico-syntaxiques de la classe que l'on généralise en supprimant tous les attributs non communs aux autres contextes de la même classe).

Cette généralisation est cependant assez mal contrôlée et induit trop de bruit dans les résultats, notamment du fait de l'absence de définitions formelles de notions de généralité qui aideraient à stopper le processus lorsque les patrons ne sont plus assez précis. Pour résoudre ce problème, E. Morin & E. Martienne (1999) emploient une technique d'apprentissage pour tenter d'inférer des restrictions sur les patrons trop généraux.

Dans son logiciel CAMELEON (Séguéla & Aussenac-Gilles, 1999 ; Séguéla, 2001), P. Séguéla propose une approche à l'intersection de celles de M. Hearst et C. Jouis puisqu'il suggère de réutiliser, en les adaptant si besoin, certains marqueurs de relations dits génériques, et d'acquérir d'autres marqueurs spécifiques au corpus étudié. Même si l'idée de la réutilisabilité des patrons et de leur adaptation est intéressante, ce travail souffre du même manque d'automaticité que la méthode de M. Hearst. L'utilisateur intervient en effet de façon centrale dans le processus : c'est lui qui examine la pertinence des marqueurs génériques pour le corpus, et au besoin c'est lui qui les adapte ; enfin, c'est également lui qui doit construire l'ensemble des marqueurs spécifiques. Ces travaux s'inscrivent de ce fait plutôt dans une approche d'aide à l'extraction de relations que d'une extraction entièrement automatique.

Il faut noter que le découpage en cinq étapes proposé par M. Hearst sur lequel repose tous ces travaux est en réalité le processus habituel d'un apprentissage supervisé (*i.e.* à partir d'exemples) : trouver un concept à apprendre, trouver un ensemble d'exemples répondant à ce concept, décrire les exemples par des attributs, inférer une définition du concept à l'aide des exemples et de leur attributs, trouver d'autres objets répondant à la définition du concept. La relation $\mathcal{R}$ à apprendre est alors estimée par un classifieur. Ce dernier est généré par inférence à partir d'exemples. Dans le cas de M. Hearst ou de CAMELEON cette inférence est manuelle alors qu'elle est automatisée (quoique « rudimentaire ») chez E. Morin.

Ce type d'approche par apprentissage se formalise donc de la même manière que précédemment : chaque occurrence de couple (et son contexte) est examinée pour constater

si elle répond ou non à l'une des règles. On a donc encore pour un couple (x, y) et avec les mêmes notations que précédemment : $x\hat{\mathcal{R}}y \iff \exists i\, Desc\,(\,occ_i(x,y)\,) \in \mathscr{L}_R$ avec $\mathscr{L}_R = \bigcup_{1\leq l\leq n} \mathscr{L}_{R_l}$. La différence avec l'approche précédente est que ces règles sont inconnues *a priori* (ou du moins certaines) et dérivées d'exemples de couples dont les constituants sont en relation. Cette technique d'acquisition permet donc d'apprendre de nouvelles règles pour tout nouveau corpus qui sont pertinentes car adaptées au corpus.

1.4 Bilan

Nous l'avons vu, de nombreuses techniques d'acquisition d'informations lexicales sémantiques existent et reposent sur des approches très diverses. Certaines de ces approches sont d'ailleurs également utilisées pour extraire sur corpus d'autres sortes d'informations que les unités sémantiques et les relations les structurant. Ainsi, le repérage d'entités nommées, de dates, de lieux, *etc.*, fait aussi l'objet de travaux d'acquisition relevant du domaine de l'extraction d'informations. Parmi les techniques développées dans ce cadre, une grande place est notamment faite aux méthodes d'acquisition structurelles s'appuyant sur des patrons. Nous ne présentons pas les nombreuses réalisations de ce thème de recherche que le lecteur intéressé pourra étudier en se reportant par exemple aux actes des conférences MUC (MUC-7, 1998).

Les travaux d'acquisition que nous avons présentés, aussi bien de relations sémantiques que d'unités lexicales, peuvent se grouper (approximativement) selon deux familles : les méthodes exploitant l'aspect numérique des données textuelles et celles exploitant leur aspect structurel.

Les avantages des approches numériques sont principalement leur automaticité et leur portabilité : elles sont relativement faciles à mettre en place car elles ne nécessitent souvent aucune donnée autre que le corpus. Elles sont de ce fait adaptées aux traitement de nouveaux textes, et les résultats produits sont propres au domaine étudié. À l'inverse, les techniques d'acquisition symboliques nécessitent des connaissances supplémentaires pour fonctionner. Elles doivent par exemple disposer en plus du corpus d'un ensemble de patrons d'extraction, ou bien, si ces patrons sont appris automatiquement, d'exemples d'apprentissage. La constitution de ces données supplémentaires demandent donc un investissement humain qu'il est le plus souvent nécessaire de reproduire à chaque nouveau domaine que l'on souhaite traiter.

En retour, les méthodes numériques souffrent d'un manque d'interprétabilité. Il est en effet souvent difficile de comprendre pourquoi un certain élément sémantique a été retenu et pas un autre, le seul indice fourni à ce sujet étant généralement un score statistique. La détection se passe au niveau du corpus et non pas de l'occurrence ; il n'est donc pas possible de « revenir » au texte pour l'expliquer. C'est également pour cette raison que ces méthodes produisent parfois des résultats très hétérogènes, la seule nature fréquentielle des objets linguistiques n'étant pas toujours à même de les différencier (voir section 1.2.2.1). Enfin, toujours à cause du manque d'interprétabilité du

processus, ces méthodes n'offrent aucun retour sur la définition de l'information recherchée. Elles ne permettent pas d'en avoir une compréhension plus précise ou d'en donner une définition opérationnelle. Les approches structurelles sont en revanche plus facilement interprétables à plusieurs titres. La granularité de détection des éléments sémantiques intéressants est très fine puisqu'elle s'opère au niveau de l'occurrence. Cela permet de comprendre plus naturellement pourquoi une information sémantique a été retenue ou non. Par ailleurs, les règles ou patrons employés au sein de ces techniques permettent également de définir de manière très pragmatique l'information recherchée. Cette définition peut s'avérer intéressante lorsque les éléments que l'on tente d'acquérir, et plus précisément leurs réalisations en corpus, sont mal connus. Cette dernière propriété d'interprétabilité est encore plus intéressante lorsque les patrons sont appris automatiquement à partir d'exemples dans le corpus, comme dans l'approche par apprentissage présentée précédemment.

C'est dans le cadre des approches structurelles, et plus précisément celui de l'apprentissage artificiel symbolique de patrons, que nous avons choisi de nous placer pour développer le système ASARES. Ce positionnement, qui doit nous permettre de répondre aux besoins multiples et variés en ressources lexicales sémantiques, est réalisé en pratique par l'utilisation d'une technique éprouvée d'apprentissage supervisé : la programmation logique inductive. Celle-ci permet à ASARES la production (on parlera alors d'inférence) de patrons décrivant et définissant l'unité ou la relation (le concept) ciblée à partir d'exemples. Ce mode de fonctionnement permet ainsi de répondre en partie à notre triple objectif exposé en introduction puisqu'il nous assure, comme toute méthode structurelle, l'interprétabilité des résultats, mais aussi des performances d'extraction satisfaisantes et une bonne généricité de notre outil. Ce dernier point s'explique en effet par l'aspect supervisé de la technique d'apprentissage qui permet à ASARES de produire des patrons d'extraction conformes aux exemples fournis par un expert (*cf.* chapitre 3) et donc à l'information cherchée par ce dernier. Cette étape de constitution d'exemples, qui peut sembler pénalisante, est cependant plus facile à effectuer par un expert qu'une formalisation directe de l'information sémantique recherchée sous forme de patrons ; nous proposons d'ailleurs une méthode pour l'automatiser (*cf.* chapitre 4) nous permet de parachever l'ensemble de notre triple objectif.

Le chapitre suivant présente plus avant les cadres méthodologique, technique et applicatif dans lesquels se place le développement de notre outil. Les différents choix présidant au fonctionnement d'ASARES sont notamment explicités et motivés, les principes de la technique d'apprentissage artificiel utilisée et l'application choisie pour évaluer notre outil — l'acquisition sur corpus de relations qualia — sont également exposés.

Chapitre 2

Le système ASARES : positionnement et cadre applicatif

L'objectif des travaux présentés dans ce mémoire est de répondre aux besoins en ressources sémantiques spécifiques de nombreuses applications du TAL. en construisant un système générique d'acquisition sur corpus d'informations lexicales sémantiques aisément portable d'un corpus à un autre et produise des résultats de bonne qualité et interprétables. Pour ce faire, nous avons développé ASARES, un outil d'acquisition s'appuyant sur une technique d'apprentissage symbolique supervisé — la programmation logique inductive (PLI) —, cette approche semblant la plus à même de répondre à notre triple objectif.

Pour évaluer la validité de notre méthode, nous l'avons appliquée à un type de relations sémantiques entre noms et verbes définies au sein du Lexique génératif appelées relations qualia. Cette tâche d'acquisition est particulièrement intéressante puisqu'il n'existe pas de définition opérationnelle (c'est à dire permettant une acquisition) de ces relations qualia; on ne sait donc pas quelles sont en corpus les structures morphosyntaxiques et sémantiques susceptibles de les porter. L'interprétabilité des patrons produits doit donc permettre de répondre en partie à cette question.

L'objectif de ce chapitre est de présenter en détail les choix et les techniques présidant au développement d'ASARES ainsi que le cadre applicatif auquel nous le confrontons. Nous détaillons dans la première partie les motivations et le positionnement de nos travaux, et nous expliquons l'architecture globale de notre système. Nous présentons ensuite le cadre de l'apprentissage artificiel supervisé, et plus spécifiquement la programmation logique inductive qui est au cœur de notre système d'acquisition. Nous terminons enfin par une description succincte du modèle du Lexique génératif, et plus particulièrement des relations sémantiques qualia et par la présentation des intérêts applicatifs qu'elles offrent.

2.1 Positionnement et motivations de nos travaux

Nos travaux se placent dans un cadre d'apprentissage artificiel. La souplesse de ce cadre nous permet d'offrir la possibilité pour un but applicatif quelconque d'acquérir des éléments de lexiques sémantiques qui lui sont nécessaires. C'est donc une approche générique que nous avons tenté de mettre en œuvre dans ASARES , avec cependant les contraintes exposées précédemment concernant l'automaticité du processus, la qualité et l'interprétabilité des résultats.

Nous détaillons dans la première partie les motivations sous-tendant le triple objectif que nous nous sommes fixé pour le développement d'ASARES. Nous indiquons ensuite le positionnement de notre approche par rapport aux techniques proches existantes et précisons le rôle de la théorie linguistique définissant les éléments sémantiques à acquérir. Nous terminons en présentant l'architecture globale de notre système et détaillons son fonctionnement.

2.1.1 Triple objectif

Nous revenons dans cette partie sur le triple objectif qualité-interprétabilité-automaticité, exposé dès l'introduction, que nous fixons pour nos travaux. Ces trois critères influencent les choix opérés à chaque étape de la construction de notre système d'acquisition et sont *a priori* difficilement conciliables. Nous nous attachons ici à les développer et les motiver successivement, et nous présentons pour chacun d'eux le positionnement d'ASARES.

2.1.1.1 Automaticité et portabilité

Des obstacles à la portabilité

La pérennité des outils développés en TAL, comme dans tout autre domaine, est fortement liée à leur portabilité. Une technique développée pour une application précise dans un domaine particulier ne sera pas utilisée dans un autre contexte si sa phase d'adaptation et son utilisation dépassent un seuil de tolérance. Ce seuil de tolérance est propre à l'utilisateur (le chercheur, le particulier ou encore l'entreprise) et fonction de la qualité attendue des résultats.

Par ailleurs, la viabilité d'un tel produit est également conditionnée par sa facilité de maintenance pour des raisons similaires. En effet, même dans le cadre d'une application donnée et fixée, les connaissances du domaine, ou le domaine lui-même, peuvent évoluer sensiblement et nécessiter une mise à jour ou une adaptation de l'outil.

Il paraît donc important de mettre en œuvre des techniques dont la portabilité soit étudiée. La capacité d'adaptation de ces techniques et leur degré d'automatisation doivent en conséquence être évalués au même titre que la qualité des résultats qu'elles produisent[1].

[1]Nous rejoignons par cette remarque les conclusions d'Y. Wilks (1999) concernant les outils développés dans le domaine connexe de l'extraction d'informations.

On peut distinguer trois familles principales d'obstacles à la portabilité des outils. Tout d'abord, les mises au point et les réglages de paramètres nécessaires à leur bon fonctionnement peuvent être fastidieux, demandant beaucoup de temps à l'utilisateur, mais peuvent également dépasser ses compétences et ses connaissances. Il convient donc, lors de la construction de tels outils, de favoriser les approches « presse-bouton ». Cela peut se faire soit en utilisant des techniques ne nécessitant pas de réglages complexes et fins, soit en prévoyant un module dédié à l'automatisation du choix des différents paramètres du système. Il faut noter qu'opter pour une approche « presse-bouton » ne signifie pas pour autant que l'outil fonctionne comme une *boîte noire*. Il peut être en effet intéressant de proposer à l'utilisateur averti une lisibilité complète de l'ensemble du processus. Libre à lui ensuite d'exploiter cette connaissance de l'outil pour l'adapter à une nouvelle tâche ou de nouveaux besoins.

Un deuxième obstacle majeur à l'automaticité, très présent dans le domaine du TAL, est l'utilisation de ressources externes. Ces ressources peuvent être de différentes natures : ontologie, thésaurus ou plus simplement corpus d'un volume minimum. Cela peut également s'exprimer en termes de format ou d'informations sur les données traitées par le logiciel. Par exemple, de nombreuses techniques d'acquisition d'informations sémantiques emploient des connaissances syntaxiques. L'utilisation de ces méthodes requiert donc d'étiqueter les textes du nouveau domaine à traiter, soit manuellement, ce qui est une tâche coûteuse en temps et en compétences, soit par l'emploi d'outils spécifiques, pas toujours disponibles surtout dans certaines langues, parfois peu performants, voire financièrement non abordables.

Enfin, un dernier type d'obstacle à la portabilité est l'intervention humaine. Beaucoup d'outils ont été développés dans une perspective d'aide à l'acquisition d'informations sémantiques. Ils s'appuient de ce fait sur l'expertise humaine. Celle-ci peut intervenir à différents niveaux : en amont de l'utilisation de l'outil (par exemple sélectionner des phrases intéressantes), durant son utilisation (guider le processus en édictant des règles d'extraction), ou en aval (filtrer ou valider les résultats). Si ce type d'approches permet de se prémunir contre des résultats de trop mauvaise qualité — c'est d'ailleurs la raison principale de ce choix d'architecture —, il est aussi un frein évident à leur utilisation à grande échelle. Ainsi, la portabilité de ces outils est-elle liée à la disposition d'un expert du domaine et de la technologie employée. Enfin, même si le niveau d'expertise requise est à la portée de l'utilisateur, son intervention empêche l'application entièrement automatique (donc peu coûteuse et souvent plus rapide) à n'importe quel domaine.

Les choix retenus dans ASARES

Ces diverses remarques justifient pleinement l'approche que nous avons adoptée pour la construction de notre système d'acquisition. Notre souhait est en effet de pouvoir traiter le plus aisément possible n'importe quel corpus, avec un minimum d'intervention

humaine et de ressources externes. Nous avons donc tenté à chaque étape du système de diminuer voire supprimer les différentes sources de coûts telles que les réglages manuels de paramètres et l'alimentation en données.

Ainsi, certains réglages de paramètres d'ASARES sont réalisés automatiquement. C'est notamment le cas pour les paramètres utilisés lors de la phase d'inférence des patrons d'extraction par le logiciel de programmation logique inductive (voir la section 3.3.1 concernant la quantité de bruit autorisée dans les patrons produits).

D'autres sources de coûts sont inhérents à notre technique : ce sont les données qui lui sont nécessaires. La principale ressource utilisée par notre système est bien sûr le corpus. Ce dernier doit évidemment répondre aux critères de représentativité du domaine et d'homogénéité. Notre approche étant principalement symbolique, elle ne nécessite toutefois pas d'utiliser un corpus d'une taille importante comme dans les techniques statistiques. Cette condition est un avantage dans les domaines où les ressources textuelles sont peu nombreuses ou difficiles d'accès.

Les informations dérivées de ce corpus sont issues d'un étiquetage catégoriel (morphosyntaxique) et, pour l'expérience décrite au chapitre 3, d'un étiquetage sémantique. L'étiquetage catégoriel est une procédure entièrement automatique, ne nécessitant ni ressources propres au domaine, ni intervention humaine (voir section 2.1.3.2 pour une description des outils utilisés lors de nos expériences). Étape préliminaire à la majorité des applications du TAL, l'étiquetage catégoriel a été largement étudié et offre d'excellentes performances dans une grande variété de langues. Elle ne constitue donc pas un frein à l'utilisation de l'ensemble de notre système. L'étiquetage sémantique que nous utilisons est une procédure originale dont le fonctionnement est calqué sur celui de certains étiqueteurs catégoriels (voir la présentation de cette technique en section 4.1.1). Contrairement à l'étiquetage catégoriel, il nécessite toutefois la définition de ressources propres au corpus étudié et constitue donc potentiellement un frein à une complète automatisation du système. Ce coût supplémentaire et son influence sur la qualité des résultats ont donc été étudiés et sont présentés en section 4.1. Le système ASARES, construit de manière à bénéficier de toute information supplémentaire de ce type, n'en est néanmoins pas dépendant ; il fonctionne de manière très performante sans ces étiquettes sémantiques (et est d'ailleurs utilisé de cette façon dans les expériences rapportées au chapitre 5). ASARES suit donc une approche que l'on pourrait qualifiée de *knowledge-poor* — le système pourrait même être utilisé sans aucune autre information que les formes fléchies des mots — qui doit permettre une portabilité plus aisée; nous rejoignons en cela la philosophie à la base de certains outils d'acquisition tels que ANA (Enguehard, 1992 ; Enguehard & Pantera, 1995). Cependant, notre approche n'exclut pas que d'autres types d'informations (syntaxiques par exemple) puissent facilement être inclus dans le système pourvu qu'ils respectent certaines conditions énoncées en section 3.1.1.3.

Le dernier type de ressources utilisées par notre système est l'ensemble d'exemples des éléments à acquérir. Cela constitue indubitablement une importante intervention humaine. Il faut néanmoins noter que cette façon d'introduire de la connaissance lin-

guistique dans le système d'acquisition relève d'un niveau de compétence assez bas au regard des systèmes d'acquisition basés sur des ensembles de règles données par un linguiste. Il est en effet plus facile de fournir des exemples de réalisations d'un phénomène linguistique que de formaliser dans un premier temps ce phénomène et, dans un second temps, de rendre cette formalisation opérationnelle dans un système. Cette approche est le propre de systèmes d'apprentissage supervisé, qui à partir d'exemples vont induire une définition du concept qui les relie. C'est également le principe fondateur de notre système. Ce principe est d'autant plus indiqué lorsqu'aucune définition pratique d'un phénomène linguistique n'est connue, comme c'est le cas pour les relations qualia, l'exemple d'acquisition que nous présentons dans ce mémoire. La phase de supervision, inhérente à l'emploi de la programmation logique inductive, peut néanmoins être réduite voire supprimée en utilisant conjointement à cette technique d'apprentissage symbolique une technique d'acquisition reposant cette fois-ci sur une approche numérique. Nous présentons en section 4.2 deux modifications de notre système mettant en pratique cette idée. Les systèmes hybrides ainsi obtenus ne nécessitent donc plus aucune intervention de l'utilisateur et sont donc entièrement automatiques.

2.1.1.2 Qualité des résultats

Obtenir des résultats de bonne qualité est le but affiché de toute tâche d'acquisition. En pratique, cette qualité se traduit principalement par deux propriétés :

- la complétude du résultat, c'est-à-dire que la totalité (ou presque) des éléments devant être acquis le sont effectivement ; cela se manifeste donc par un bon taux de rappel ;
- la précision du résultat, c'est-à-dire que seuls (ou presque) les éléments devant être acquis le sont effectivement ; cela se manifeste par un bon taux de précision.

Ces deux propriétés sont cependant difficiles à obtenir ensemble, puisque la recherche de l'exhaustivité entraîne souvent une perte de la précision et inversement. Il n'est pas rare que des outils privilégient ouvertement l'une de ces deux propriétés pour leurs résultats, influençant par là leurs choix de conception. Par exemple, certaines techniques fonctionnant avec l'aide d'un expert comme dernière étape de validation peuvent se permettre de mettre l'accent sur la complétude puisque l'expert humain sert de filtre ; les résultats sont donc au final complets et précis. Malheureusement, outre le fait que ce type d'approche n'est pas compatible avec notre objectif d'automaticité, il n'est guère envisageable à grande échelle. En effet, l'examen de données bruitées par un expert ne peut se concevoir que si leur volume n'excède pas une taille limite.

Par ailleurs, le besoin conjoint de précision et de complétude dans les tâches d'acquisition peut être primordial. Les performances des applications exploitant ensuite les ressources lexicales acquises dépendent en effet de leur qualité. Cela apparaît à travers les résultats médiocres de systèmes utilisant des ressources bruitées ou inadéquates (de Loupy & El-Bèze, 2002). Il est par exemple souvent reproché à une base généraliste comme WordNet d'être inadaptée à une application sur un domaine précis car les *synsets* sont jugés trop hétérogènes, ou certains liens is_a impossibles dans le domaine

étudié ou encore des distinctions de sens non pertinentes (Palmer, 1998) — il s'agit donc là d'un problème de précision. Mais l'on critique aussi le fait que ces bases ne fassent pas de liens inter-catégoriels (rien ne relie *to cook* et *cooking* par exemple) (Gonzalo *et al.*, 1998), ou ne relient pas certains termes d'un même domaine (comme *tennis*, *racket*, *ball* et *tennis player*) (Fellbaum *et al.*, 1996) — il s'agit dans ce cas d'un problème de complétude.

Qualité des résultats et ASARES

Il semble donc qu'il soit impossible de privilégier la précision aux dépends de la complétude ou le contraire sous peine de produire des résultats finalement inexploitables dans le cadre d'applications réelles. Par ailleurs, si l'on souhaite développer des outils devant être utilisés de nombreuses fois et sur des gros volumes de données, comme c'est le cas pour nous, il semble également indispensable qu'ils produisent des résultats suffisamment bons pour ne pas nécessiter une phase de validation manuelle ou une phase d'enrichissement par un expert qui serait trop coûteuse en pratique. Finalement, notre souci d'automaticité semble donc aller à l'encontre de notre exigence concernant la qualité attendue des résultats. Cependant, comme nous le montrons au travers des résultats exposés en section 3.3.2.2, notre système, grâce notamment à l'emploi de techniques symboliques performantes, réussit à parvenir à un très bon compromis entre le taux de rappel et de précision, sans qu'aucun post-traitement humain ne soit effectué.

2.1.1.3 Interprétabilité des résultats

Le dernier objectif que nous nous fixons pour la réalisation de notre système d'acquisition est l'interprétabilité. Apanage des approches symboliques, cette dernière doit notamment se traduire dans le système par les trois propriétés suivantes :

- la granularité de la détection, qui permet de décider occurrence par occurrence si un élément recherché doit être acquis ;
- la compréhension des résultats qui permet d'expliquer pourquoi tel élément est extrait et pas tel autre ;
- l'explicativité du système qui permet de donner une définition intentionnelle et opérationnelle du concept ciblé par l'acquisition.

Granularité de la détection

La première propriété, la granularité de la détection, est liée à la nature des informations généralement exploitées dans les approches structurelles. En effet, contrairement à l'acquisition numérique où l'élément est extrait à partir d'informations collectées sur le corpus pris dans son intégralité, les approches structurelles examinent souvent les éléments occurrence par occurrence. Ainsi, une occurrence sera considérée comme relevant du concept recherché ou non sur la base d'indices recueillis dans son contexte, indépendamment de ce qui est décidé pour les autres occurrences du même élément.

Les approches structurelles disposent dans un premier temps d'un ensemble d'occurrences jugées pertinentes ou non. Dans un deuxième temps, un élément est défini comme pertinent si une condition sur ses occurrences est remplie. Par exemple, cette condition sera que le nombre d'occurrences pertinentes (c'est-à-dire détectées par le système) doit dépasser un certain seuil, ou encore que le taux d'occurrences pertinentes par rapport à l'ensemble des occurrences dépasse un score minimal. En fin de processus, ce type de système a alors deux choix pour la présentation des résultats : soit ils sont détaillés, c'est-à-dire que chaque occurrence valide est proposée à l'utilisateur, soit ils sont résumés et seuls les éléments, hors de toute occurrence, sont présentés.

En jouant sur la condition permettant de juger un élément comme pertinent à partir des occurrences détectées par le système d'acquisition, on a ainsi la possibilité de pouvoir s'attacher à la détection de phénomènes rares. Dans les approches numériques, ces phénomènes, n'offrant que peu de réalisations en corpus ou occultés par des phénomènes beaucoup plus massifs, sont le plus souvent « noyés » sous le niveau du bruit et donc non détectés. Ils peuvent pourtant se révéler d'intérêt dans une application nécessitant des ressources sémantiques de très bonne qualité (*cf.* section précédente).

Compréhension des résultats

La propriété de compréhension des résultats est intimement liée à la précédente. On reproche en effet souvent le fonctionnement « boîte noire » des méthodes purement statistiques, donnant des résultats hors de tout contexte permettant de les interpréter. Les approches structurelles, de par les indices symboliques qu'elles manipulent, permettent au contraire de comprendre pourquoi le système a accepté ou rejeté telle ou telle occurrence. Cette propriété importante en TAL permet donc d'interpréter plus facilement les résultats et de comprendre les limitations du système, les causes de ses erreurs et surtout les phénomènes linguistiques sous-jacents.

Explicativité : vers une définition linguistiquement interprétable, opérationnelle et fondée sur des réalisations réelles

Enfin, la dernière propriété attendue, l'explicativité, est certainement la moins commune des systèmes d'acquisition. En effet, alors que les deux propriétés précédentes sont le fruit de l'utilisation d'une approche structurelle, l'obtention d'une définition du concept recherché ne peut être que le fait de l'emploi d'une technique d'apprentissage symbolique artificiel. Comme nous l'avons dit précédemment, une telle technique permet, à partir d'exemples (et éventuellement de contre-exemples) de ce que l'on cherche à acquérir, d'inférer une définition du concept représenté par ces exemples. Cette définition, utilisée ensuite pour découvrir de nouvelles instances répondant au concept, manipule des symboles, c'est-à-dire des indices structurels, interprétables par l'utilisateur.

Cette définition obtenue par induction a trois caractéristiques essentielles :

– elle est linguistiquement interprétable ;

- elle est opérationnelle ;
- elle est fondée sur des réalisations réelles du phénomène linguistique que l'on cherche à caractériser.

Ces trois avantages font des techniques d'apprentissage symbolique artificiel un outil particulièrement intéressant dans le domaine du TAL où les définitions des concepts linguistiques peuvent manquer, être inadaptées à une implémentation efficace, ou encore ne pas être fondées sur les données langagières réelles et donc peu productives.

Intérêt de l'interprétabilité dans ASARES

Notre système s'inscrit pleinement dans ce cadre symbolique. Nous inférons, grâce à la programmation logique inductive, des patrons sémantiques et morphosyntaxiques qui servent de support à une analyse linguistique des réalisations en corpus des éléments sémantiques acquis. Dans les travaux présentés dans ce mémoire, ces éléments sont les relations qualia pouvant exister entre un nom (N) et un verbe (V) ; on appelle par la suite couples N-V qualia ou couples qualia ou encore couples en relation qualia les paires nom-verbe telles que le nom et le verbe soient en relation qualia. La définition existante de ces relations dans le Lexique génératif est purement théorique est non opérationnelle. L'examen des patrons générés est donc très instructif d'un point de vue linguistique, puisqu'il permet de préciser le modèle théorique.

2.1.2 Positionnement de nos travaux

Comme nous l'avons vu précédemment, beaucoup de travaux d'acquisition d'informations lexicales sémantiques existent. Nos travaux s'appuient quant à eux sur une technique d'apprentissage supervisé, la programmation logique inductive, pour générer des patrons d'extraction à partir d'exemples. Ces patrons permettent ainsi d'acquérir de nouveaux éléments sémantiques répondant aux mêmes critères que ceux régissant les exemples. Nous nous positionnons donc ci-après par rapport aux autres travaux d'acquisition s'appuyant sur des patrons appris pour extraire des informations sémantiques, et par rapport au rôle du cadre linguistique dans ces travaux.

2.1.2.1 Extraction par patrons

Notre approche est très similaire à celles de M. Hearst ou de E. Morin (*cf.* section 1.3.2.2). Elle s'en distingue néanmoins sur deux points. Tout d'abord, comme nous l'avons déjà signalé, nous nous plaçons complètement dans le cadre d'un problème d'apprentissage. L'inférence des patrons d'extraction est donc menée par une technique d'apprentissage établie (la programmation logique inductive) et non pas manuellement (Hearst, 1998) ou par une technique *ad hoc* (Morin, 1999). Ensuite, les informations prises en compte dans les patrons produits par ASARES sont morphosyntaxiques et sémantiques. Bien que d'autres types d'informations pourraient être facilement inclus dans notre méthode, ce choix s'explique par la volonté de produire des patrons donnant

des indications générales. Cela explique pourquoi ces patrons n'utilisent pas d'informations de nature lexicale.

2.1.2.2 Rôle du cadre linguistique

Le cadre linguistique que dans lequel s'opère une tâche d'acquisition joue un rôle important dans le développement même d'un système d'acquisition, même s'il est trop rarement mis en avant. Il permet à la fois de guider le processus d'acquisition, d'une manière plus ou moins directe, et de valider les résultats produits et ainsi de mesurer les performances du système. Ces deux rôles principaux du cadre linguistique, que nous détaillons ci-dessous, sont ouvertement exploités dans ASARES.

On peut distinguer deux façons dont une théorie linguistique intervient dans le développement d'un système d'acquisition. Soit la théorie donne une définition opérationnelle de l'objet que l'on tente d'acquérir ; cette définition peut alors être implémentée dans un système d'acquisition comme c'est le cas pour l'acquisition de synapsies par TERMINO. Soit la théorie linguistique ne fournit aucun indice directement exploitable par une technique d'acquisition — c'est le cas le plus commun — mais un expert de cette théorie peut identifier des exemples de ce que l'on tente d'acquérir. Il est alors possible d'exploiter ces exemples pour construire ou du moins guider le processus d'acquisition. C'est dans ce deuxième cas de figure que se placent nos travaux.

Il est important d'évaluer le plus finement possible les résultats d'un système d'acquisition, d'autant plus s'il se veut automatique, c'est-à-dire sans intervention humaine pour filtrer les résultats. À ce titre, la théorie linguistique définissant l'objet linguistique que l'on cherche à acquérir joue un rôle essentiel. En effet, elle permet d'examiner les résultats selon des critères rigoureux et établis préalablement à la tâche d'acquisition. En contrepartie, l'analyse en corpus et l'étude en grandeur nature effectuée lors du processus d'acquisition peuvent amener à préciser des points de la théorie, notamment si, comme nous l'avons vu, la technique d'acquisition est interprétable. L'application d'ASARES à l'acquisition d'éléments du Lexique génératif (les couples N-V qualia) se place dans ce cadre puisque les éléments acquis sont examinés et comparés à ceux obtenus manuellement par un expert. Les patrons générés, permettant de préciser les modes de réalisations en contexte de ces éléments, sont également étudiés dans une perspective linguistique.

2.1.3 Architecture du système

Nous présentons dans cette section les différents modules composant le système ASARES. Après un descriptif schématique de son organisation générale, nous décrivons la phase d'étiquetage préliminaire à l'emploi d'ASARES, puis détaillons le fonctionnement de chacun des modules.

2.1.3.1 Survol d'ASARES

Le découpage en module de notre système d'acquisition correspond aux différentes phases du processus d'acquisition de schémas d'extraction de M. Hearst (*cf.*

section 1.3.2.2). Le premier module est dédié à l'acquisition (automatique ou manuelle) d'occurrences dans le corpus (étiqueté) d'éléments recherchés qui serviront d'exemples. Ces exemples sont ensuite exploités par le deuxième module qui va inférer, à l'aide d'un algorithme de programmation logique inductive, un ensemble de règles décrivant le concept recherché sous forme de clauses de Horn. Le dernier module utilise les règles produites comme des patrons d'extraction, qui, appliqués au corpus, vont permettre de détecter des occurrences de nouveaux éléments. Ces éléments sont finalement réunis dans un « lexique » de couples qualia ou bien à leur tour réutilisés comme exemples pour affiner la recherche des couples dans le corpus.

La figure 2.1 résume cette architecture générale. Les icônes représentant un ordinateur indiquent que le module est entièrement automatique, ce qui est le cas de tous les modules à l'exception de celui de génération des exemples qui peut être soit manuel (voir chapitre suivant) soit automatique (*cf.* section 4.2).

2.1.3.2 Étiquetage

La première phase nécessaire à l'utilisation d'ASARES est l'étiquetage du corpus textuel sur lequel on doit effectuer la tâche d'acquisition. Cette phase, commune à la plupart des applications du TAL, est précédée d'un nettoyage consistant à supprimer du corpus tous les éléments non textuels (figures, tableaux, table des matières) et d'une segmentation du texte en différentes unités (paragraphes, phrases, mots). Ensuite, trois étapes, non nécessairement distinctes, réalisent l'étiquetage morphosyntaxique : l'assignation des étiquettes, la lemmatisation, et la désambiguïsation des mots ayant reçu plusieurs étiquettes.

Assignation des étiquettes morphosyntaxiques

L'étiquetage morphosyntaxique assigne à chaque mot, sous forme d'étiquette (*tag*), sa ou ses catégories possibles. Ainsi, pour le mot *souris*, les deux étiquettes possibles sont nom et verbe. Il peut également adjoindre à ces étiquettes des informations morphologiques issues de l'étape précédente telles que le genre (masculin ou féminin), le nombre (singulier ou pluriel), ou pour les verbes leur mode, temps et personne.

Lemmatisation

La phase de lemmatisation, souvent couplée à la phase d'étiquetage morphosyntaxique, consiste à assigner à chaque mot une forme de référence (l'infinitif pour un verbe, le masculin singulier pour un adjectif, *etc.*). Si plusieurs lemmes sont possibles, ils sont tous retenus. Ainsi, pour le mot *souris*, au moins deux lemmes peuvent être retenus : *souris* et *sourire*. C'est généralement pendant cette étape que sont identifiées les informations morphologiques sur les mots.

Désambiguïsation

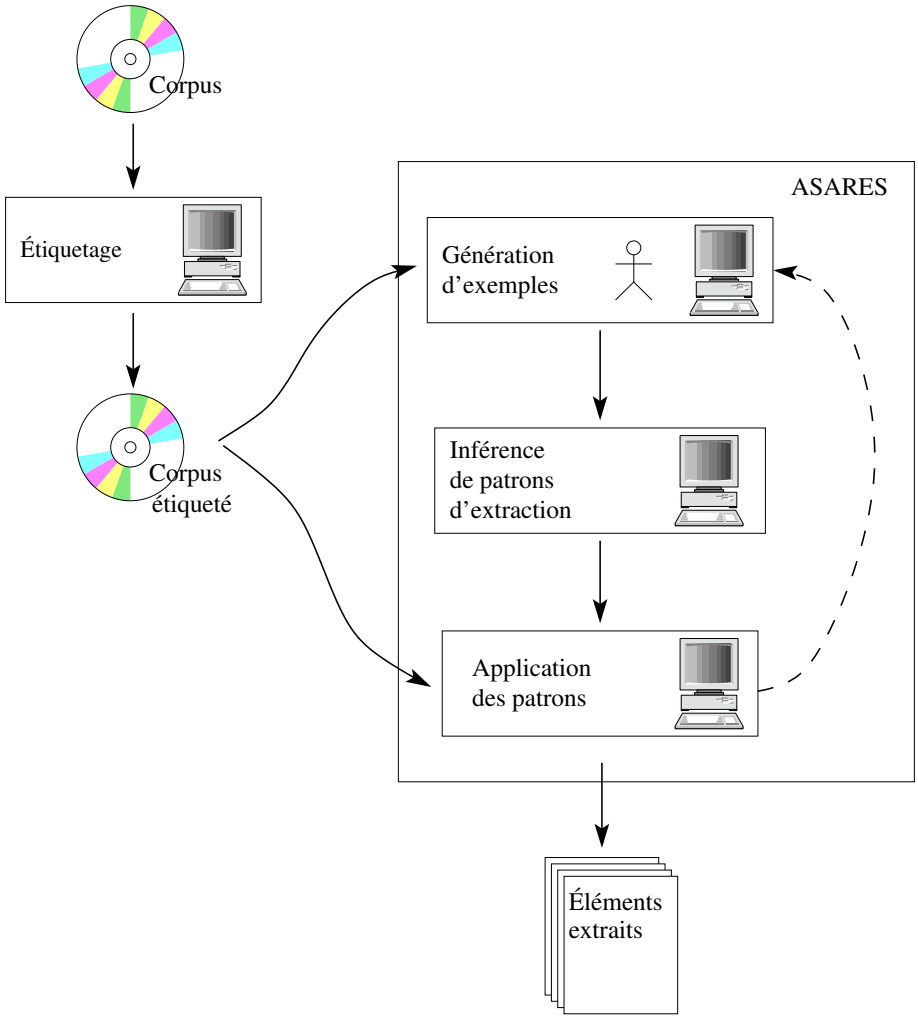


FIG. 2.1 – Architecture globale du système ASARES

La dernière étape est celle de la désambiguïsation. Les étapes de lemmatisation et d'étiquetage pouvant retenir plusieurs alternatives par mots, la désambiguïsation doit permettre de choisir parmi elles la bonne solution. Plusieurs approches sont possibles mais reposent toutes sur un examen des séquences de mots précédant ou suivant le mot ambigu. Les deux techniques les plus courantes sont l'approche par règles ou par chaînes de Markov cachées.

2.1.3.3 Génération d'exemples

La phase de génération d'exemples a pour but de trouver des phrases dans le corpus de référence contenant des occurrences des informations (unités ou relations) sémantiques recherchées. Dans le cas étudié dans ce mémoire, il s'agit par exemple de repérer

des phrases contenant des couples nom-verbe en relation qualia.

Cette phase peut elle-même se dérouler en deux étapes. La première consiste alors à identifier un ensemble d'éléments recherchés dans le corpus. Cette connaissance peut être obtenue par un expert du domaine (c'est cette technique qui est employée dans l'expérience rapportée au chapitre 3) ou automatiquement (voir les expériences en section 4.2). Dans un second temps, on recherche dans le texte les occurrences des éléments obtenus. Lors de cette deuxième phase, on vérifie éventuellement que les occurrences sont effectivement conformes à ce que l'on recherche.

Ce sont donc finalement les occurrences et leur contexte qui constituent la base d'exemples. Des contre-exemples, ou exemples négatifs, peuvent également être trouvés de la même manière ou en collectant les occurrences d'exemples potentiellement positifs qui sont rejetés à la phase de vérification mentionnée ci-dessus. Ces exemples et contre-exemples sont formatés de manière à être accessibles à notre technique d'inférence utilisée pour la production de patrons (voir ci-dessous) et donc transformés en un ensemble de prédicats (*cf.* section 3.1.1.2).

2.1.3.4 Inférence de patrons

Cette phase de production de patrons est bien sûr la plus importante du système. L'approche que nous avons adoptée est de considérer cette étape comme un problème d'apprentissage artificiel symbolique. Nous tentons donc d'inférer à partir des exemples (les occurrences d'éléments recherchés en contexte) et de contre-exemples (éléments non pertinents également en contexte) un ensemble de règles qui serviront ensuite de patrons d'extraction.

La procédure d'apprentissage que nous avons choisie dans ce cadre est la programmation logique inductive. Grâce à son formalisme basé sur la logique du premier ordre, elle nous permet de décrire aisément notre problème d'acquisition et de générer des patrons linguistiquement interprétables. Nous présentons en section 2.2 les principes de la programmation logique inductive et au chapitre suivant son utilisation dans le système ASARES pour l'acquisition de couples N-V qualia.

2.1.3.5 Application de patrons

Les patrons obtenus à l'étape précédente sont des clauses logiques. Leur application au corpus nécessite donc de décrire ce dernier par des ensembles de prédicats et de faits logiques à partir desquels les clauses inférées vont permettre de détecter de nouveaux éléments conformes à ceux donnés en exemples. Cette étape est donc en réalité identique à la phase de description des exemples (voir en section 3.1.1.2).

2.2 La programmation logique inductive

Comme nous l'avons dit précédemment, le cœur de notre système d'acquisition repose sur l'utilisation de la programmation logique inductive (PLI). Elle nous permet d'inférer des règles définissant en intention le concept que l'on cherche à acquérir et qui

est décrit en extension (au moins partiellement) à l'aide d'exemples. Les règles — des clauses de Horn — sont ensuite utilisées comme patrons d'extraction pour découvrir au sein du corpus du domaine tous les éléments répondant au concept décrit.

Après une présentation générale de l'apprentissage symbolique supervisé, nous donnons dans cette section quelques fondements et notions de la programmation logique inductive. Pour une introduction plus complète à cette technique d'apprentissage, le lecteur intéressé peut se reporter aux articles introductifs de (Džeroski *et al.*, 2000) et aux chapitres correspondants des ouvrages de référence sur l'apprentissage artificiel (Cornuéjols & Miclet, 2002 ; Mitchell, 1997). Nous présentons ensuite quelques particularités inhérentes à l'utilisation en pratique de la PLI, notamment dans le domaine du TAL. Les éléments et notions de logique utilisés dans cette partie sont décrits dans l'annexe A.

2.2.1 Apprentissage symbolique supervisé

La programmation logique inductive que nous nous proposons d'utiliser est une des nombreuses techniques existantes permettant d'apprendre automatiquement à partir d'exemples, c'est-à-dire de manière supervisée. Nous présentons ci-dessous le cadre général de ce domaine de recherche à travers la définition de l'apprentissage de T. Mitchell puis nous exposons la notion sur laquelle repose la plupart des techniques d'apprentissage : l'induction. Nous définissons ensuite quelques principes et notations utilisés dans la suite de ce chapitre. Enfin, nous développons un exemple mettant en évidence l'intérêt de la PLI pour l'apprentissage dit relationnel (voir page 64) et justifions le choix de cette technique particulière d'apprentissage dans le cadre de notre travail d'acquisition d'informations lexicales sémantiques à partir de corpus.

2.2.1.1 Définition

T. Mitchell (1997) donne une définition précise de l'apprentissage : un programme informatique *apprend* la tâche $\mathcal{T}$ à partir de l'expérience $\mathcal{E}$ et de la mesure de performance $\mathcal{P}$, si sa capacité à exécuter la tâche $\mathcal{T}$, mesurée par $\mathcal{P}$, augmente avec $\mathcal{E}$. Par exemple, la reconnaissance d'écriture manuscrite peut se formaliser comme ceci :

- la tâche $\mathcal{T}$ est la reconnaissance et la classification de mots manuscrits à partir d'images ;
- la mesure de performance $\mathcal{P}$ est le pourcentage de mots correctement classés ;
- l'expérience $\mathcal{E}$ est une base de données contenant des images de mots manuscrits déjà classés.

Toute tâche d'apprentissage nécessite donc de préciser plusieurs points. Tout d'abord, il faut soigneusement définir la tâche ($\mathcal{T}$) à apprendre ; il s'agira par exemple, dans les expériences présentées aux chapitres suivants, d'apprendre à caractériser en fonction de leur contexte les couples N-V en relation qualia. Il faut ensuite constituer un ensemble d'exemples d'apprentissage ($\mathcal{E}$) sur lequel le programme d'apprentissage se base pour *apprendre* ; dans notre cas, ces exemples sont formés à partir de phrases du

corpus annotées par étiquetage contenant des couples de noms et de verbes en relation qualia (exemples positifs) ou non (exemples négatifs). Enfin, il faut définir une mesure de performance ($\mathcal{P}$) qui permet à l'algorithme de se guider dans l'espace de recherche des solutions possibles (voir le chapitre suivant sur ce point).

2.2.1.2 Notion d'induction

L'induction est souvent présentée comme le processus inverse de la déduction. Son principe est effectivement de « remonter » des faits aux lois qui les régissent. Ce type d'inférence, ainsi que la déduction et l'abduction, peuvent s'illustrer à l'aide du célèbre syllogisme donné en figure 2.1 (Mayer, 1999) ; chacune des trois propositions est accompagnée de sa représentation Prolog standard.

	Langage naturel	Prolog
(a)	Tous les hommes sont mortels	mortel(X) :- homme(X).
(b)	Or Socrate est un homme	homme(socrate).
(c)	Donc Socrate est mortel	mortel(socrate).

TAB. 2.1 – Exemple d'induction à partir d'un syllogisme

Nous disposons de trois éléments : (a) est une règle représentant une implication, (b) est un fait, de même que la conclusion (c). On a l'implication suivante :

$$(a) \wedge (b) \models (c).$$

Examinons les différents types d'inférences possibles à partir de ce syllogisme :

1. Si l'on connaît (a) et (b), on peut trouver (c) par déduction, en appliquant par exemple le *modus ponens*. C'est ce type d'inférence, assez naturel chez un humain, qui est utilisé dans les démonstrateurs automatiques.
2. Si l'on connaît (a) et (c), on trouve (b) par abduction ; cela correspond à retrouver à partir d'une observation et connaissant les lois d'un système, l'hypothèse manquante expliquant l'observation. C'est donc ce type d'inférence qui est principalement utilisé dans le domaine du diagnostic automatique.
3. Enfin, si l'on connaît les faits (b) et (c), on dit que l'on infère la règle (a) expliquant ces faits par induction.

C'est donc cette dernière opération qui est utilisée dans les techniques d'apprentissage artificiel supervisé : les exemples sont des propositions de type (b) et les classes qui leur sont assignées représentent les faits (c). Dans le cadre de la programmation logique inductive, (b) et (c) sont exprimés par des faits Prolog et les règles inférées sont, comme dans notre exemple, des clauses de Horn.

2.2.1.3 Principes et notations

L'apprentissage artificiel a pour but d'expliquer au travers de généralisations, que l'on appelle classifieurs, un ensemble d'observations. Ce processus d'induction doit donc manipuler deux classes d'objets distincts : l'espace des exemples (ou instances ou observations) E et l'espace des généralisations (ou hypothèses, ou encore règles) $\mathscr{E}_H$. Les observations peuvent parfois se découper en plusieurs classes ; en particulier, dans nombre de problèmes, on considère souvent le cas d'un ensemble E composé d'exemples positifs E^+ et d'exemples négatifs E^- du concept que l'on cherche à acquérir. La description d'un problème d'induction nécessite donc de définir deux langages pour manipuler ces deux classes, le langage des exemples ($\mathscr{L}_E$ par la suite) et le langage des hypothèses ($\mathscr{L}_H$).

Pour permettre de trouver l'hypothèse la plus plausible étant données les observations, il faut pouvoir faire le lien entre les deux espaces E et $\mathscr{E}_H$, c'est-à-dire pouvoir faire correspondre une hypothèse h décrite dans le langage $\mathscr{L}_H$ à des éléments de E décrits avec $\mathscr{L}_E$. Cette « traduction » de $\mathscr{L}_H$ à $\mathscr{L}_E$ se fait par une relation $covers : \mathscr{E}_H \rightarrow \mathscr{P}(E)$ ($\mathscr{P}(E)$ est l'ensemble des parties de E) indiquant l'adéquation d'une hypothèse avec les exemples. Cette fonction est souvent appelée relation de couverture. La figure 2.2 illustre cette notion de lien entre l'espace des exemples et celui des hypothèses.

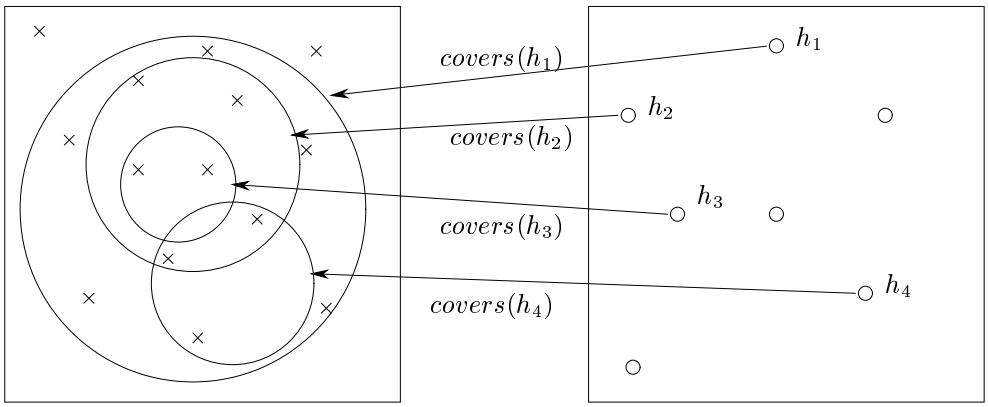


FIG. 2.2 – Représentation de la problématique en apprentissage artificiel

La problématique de l'apprentissage artificiel peut donc se résumer, à la manière de problèmes classiques en intelligence artificielle, à la recherche d'un ou plusieurs éléments (les hypothèses) dans un espace ($\mathscr{E}_H$) (Mitchell, 1982). Pour se guider plus facilement dans l'exploration de cet espace $\mathscr{E}_H$, on peut souvent le doter d'une relation de généralité permettant de comparer ses éléments entre eux. Cette relation, que l'on note $\succeq$: $\mathscr{E}_H \rightarrow \mathscr{E}_H$ ($h_1 \succeq h_2$ se lit « h_1 est plus générale que h_2 »), doit bien sûr être compatible avec la relation de couverture *covers* pour être utile. On voudra en particulier qu'une

hypothèse h_1 plus générale qu'une hypothèse h_2 couvre plus d'exemples que h_2, soit :

$$h_1 \succeq h_2 \Leftrightarrow covers(h_1) \supseteq covers(h_2)$$

Ainsi, le schéma précédent peut en fait être enrichi comme illustré dans la figure 2.3.

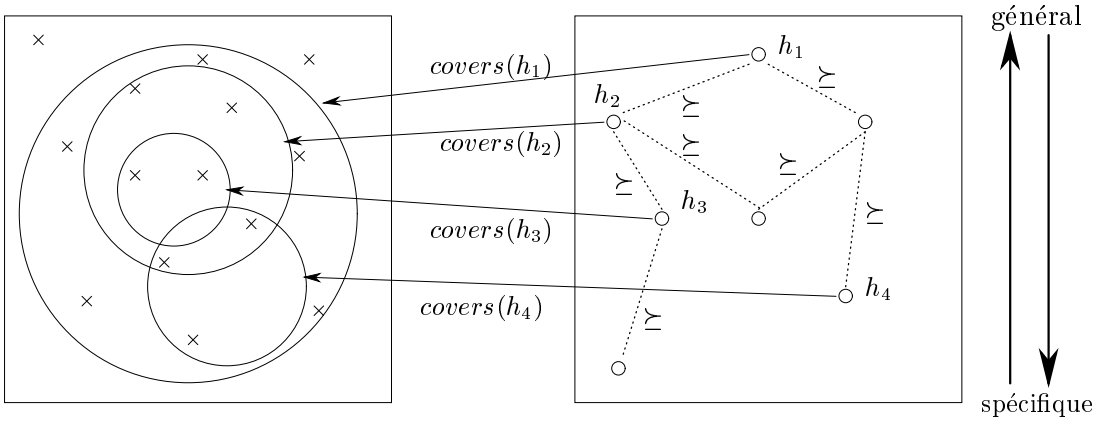


FIG. 2.3 – Représentation de la problématique en apprentissage artificiel sachant une notion de généralité dans $\mathscr{E}_H$

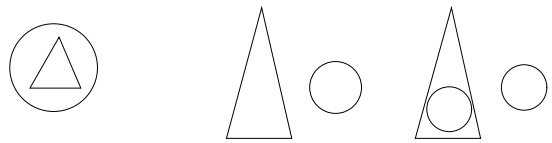
Grâce à cette notion de généralité, la recherche de solutions dans l'espace $\mathscr{E}_H$ peut être menée intelligemment, dans une approche que l'on qualifiera de descriptive, en exploitant la structure induite par la relation d'ordre retenue. Par exemple, on pourra explorer $\mathscr{E}_H$ en utilisant une stratégie « du plus général au plus spécifique », et exploiter le fait que l'ensemble des exemples couverts par un successeur d'un élément de $\mathscr{E}_H$ est nécessairement un sous-ensemble des exemples couverts par cet élément.

Dans une approche plus constructive, cette notion de généralité peut même être utilisée pour construire l'ensemble $\mathscr{E}_H$ à partir d'un ou plusieurs de ses éléments. L'espace de recherche est ainsi construit au fur et à mesure de son exploration en se servant de $\succeq$ pour générer ses éléments. Cela est notamment utilisé lorsque l'espace $\mathscr{E}_H$ est trop gros pour être donné extensivement *a priori*, ce qui est le cas général en apprentissage artificiel.

2.2.1.4 Intérêt de l'apprentissage en logique du premier ordre

Comme nous venons de le voir, outre la donnée d'exemples, un problème d'apprentissage se définit aussi par la donnée de langages de description, en particulier celui des exemples ($\mathscr{L}_E$). C'est un paramètre important puisqu'il doit permettre la représentation la plus adaptée possible des exemples, et est intimement lié au choix de la méthode d'apprentissage utilisée. En effet certaines techniques sont plus à même de manipuler certains langages de description que d'autres.

Considérons les trois exemples adaptés du problème « Bongard[2] » numéro 47, représentés en figure 2.4. Avant de s'intéresser à résoudre le problème, il faut le modéliser.



Exemple 1 Exemple 2 Exemple 3

FIG. 2.4 – Trois exemples tirés du problème Bongard n°47

Pour cela, une façon de le faire est de décrire chaque exemple par des ensembles de couples attribut-valeur. C'est cette modélisation qui est utilisée par la plupart des algorithmes d'apprentissage artificiel. Un autre langage de description couramment utilisé, la logique propositionnelle, peut-être vu comme une sous-classe du langage attribut-valeur : les attributs sont des propositions logiques et leurs valeurs sont soit vrai, soit faux.

En langage attribut-valeur, les différentes informations peuvent être rassemblées dans un tableau tel que le tableau 2.2 : les exemples sont donnés en ligne, les colonnes représentant les attributs décrivant les deux premiers exemples. On remarque d'ores et

	forme objet 1	**taille objet 1**	**orientation objet 1**	**forme objet 2**	**taille objet 2**	**orientation objet 2**
exemple 1	triangle	petit	haut	cercle	grand	*sans objet*
exemple 2	triangle	grand	haut	cercle	petit	*sans objet*

TAB. 2.2 – Description en langage attribut-objet

déjà que cette modélisation n'est pas idéale puisque certains attributs (orientation 2) ne sont pas renseignés. Cela est dû au fait que la représentation attribut-valeur impose que l'ensemble d'attributs soit le même pour tous les exemples. Ce n'est donc pas adapté à la description d'objets ayant un nombre variable de caractéristiques comme c'est le cas ici pour le triangle et le cercle.

Si l'on veut rendre compte des relations existant entre les différents objets, comme par exemple leur position relative, il est nécessaire d'ajouter autant d'attributs que de types de relations existantes (*cf.* tableau 2.3).

Par ailleurs, l'exemple 1 est décrit comme possédant une forme 1 qui est un triangle et une forme 2 qui est un cercle. Une autre description parfaitement valable de cet exemple pourrait être de considérer que la forme 1 est le cercle et la forme 2 le triangle, et

[2]Les problèmes Bongard sont des problèmes de reconnaissance de patrons visuels inventés par le scientifique russe M. Bongard en 1967. Chacun de ces problèmes est constitué de deux classes représentées par 6 images chacune. Le but est de trouver la différence entre les deux classes, c'est-à-dire quelle caractéristique essentielle différencie les schémas appartenant à une classe des autres.

	...	**1 dans 2**	**1 à gauche de 2**	**...**
exemple 1	...	oui	non	...
exemple 2	...	non	oui	...

TAB. 2.3 – Description de relations par attribut-valeur

donc d'inverser les positions des différents attributs qui leur sont liés (voir tableau 2.4). Cela peut conduire à un problème lors de la phase d'apprentissage si une des règles

	forme objet 1	**taille objet 1**	**orientation objet 1**	**forme objet 2**	**taille objet 2**	**orientation objet 2**
exemple 1	triangle	petit	haut	cercle	grand	*sans objet*
exemple 1	cercle	grand	*sans objet*	triangle	petit	haut

TAB. 2.4 – Ambiguïté des descriptions attribut-valeur

générées est du type : si forme 1 = triangle ... ou encore si une des règles manipule un attribut relatif à un de ces objets géométriques. Encore une fois, la description attribut-valeur se révèle donc peu adaptée à la description d'exemples manipulant plusieurs objets.

Considérons maintenant l'exemple 3 donné en figure 2.4. Il contient un objet géométrique de plus que les deux précédents ; il est donc nécessaire d'ajouter de nouveaux attributs pour rendre compte de cet objet. Les deux exemples précédents n'ayant pas de troisième objet, les valeurs des attributs correspondants ne sont donc pas renseignées. Outre le manque de commodité de ces ajouts d'attributs grossissant la description de l'ensemble des exemples de la base, le fait que leur valeur ne soit pas renseignée peut causer des problèmes lors de l'exploitation de ces données par des algorithmes d'apprentissage.

Lorsque la description d'un problème d'apprentissage se heurte aux limitations de la modélisation attribut-valeur soulignées ci-dessus, on se tourne souvent vers une modélisation plus riche s'appuyant par exemple sur la logique du premier ordre et ses variantes ou vers d'autres logiques aussi expressives. Cette modélisation permet notamment de décrire facilement les exemples ayant des nombres variables d'objets, ces objets pouvant eux-mêmes avoir un nombre variable d'attributs. Elle permet également de rendre compte plus naturellement des relations entre les objets ; c'est d'ailleurs pourquoi les techniques d'apprentissage exploitant ces modélisations sont dites relationnelles.

Parmi ces techniques, nous présentons dans la section suivante la programmation logique inductive. Elle exploite la logique des prédicats (voir l'annexe A) comme langage de description des exemples ($\mathscr{L}_E$), mais également comme langage de description des hypothèses ($\mathscr{L}_H$), et offre ainsi un cadre logique unifié à la tâche d'induction. Dans ce cadre, les exemples sont décrits comme dans le tableau 2.5, et le concept induit est représenté par un ensemble de règles sous forme de clauses de Horn.

	modélisation logique
exemple 1	triangle(o1,petit,haut). cercle(o2,grand). in(o1,o2).
exemple 2	triangle(o3,grand,haut). cercle(o4,petit). a_gauche(o3,o4).
exemple 3	triangle(o5,grand,haut). cercle(o6,petit). cercle(o7,petit). in(o7,o5). a_gauche(o5,o6). a_gauche(o7,o6).

TAB. 2.5 – Description en logique des prédicats

ASARES et l'apprentissage relationnel

Comme nous venons de le voir, l'apprentissage relationnel est nécessaire lorsque les exemples consistent en un nombre variable d'objets et lorsque les relations entre ces objets sont importantes.

Dans notre cas, le système doit être capable de générer des patrons à partir de toutes les informations disponibles sur l'élément à acquérir (par exemple les mots en relation sémantique) et son contexte. Cela signifie en particulier qu'il doit pouvoir prendre en compte tous les mots apparaissant dans une phrase avec l'occurrence de l'élément sémantique recherché. Ce nombre de mots est susceptible de varier d'une phrase à l'autre, ainsi que le nombre d'informations sur chacun de ces mots. Enfin, les relations qu'entretiennent les mots au sein de la phrase, ne serait-ce que leur position relative, est importante pour produire des patrons d'extraction pertinents.

La modélisation par ensembles attributs-valeurs (ou en logique propositionnelle) s'avère donc inadéquate pour rendre compte de ces spécificités de nos données d'apprentissage. Cela nous amène par conséquent à considérer pour notre tâche d'acquisition en corpus des techniques d'apprentissage relationnel telles que la PLI.

2.2.2 Induction en logique des prédicats

La programmation logique inductive est une technique d'apprentissage artificiel permettant de mettre en œuvre l'induction en logique des prédicats. Cette induction repose sur certains principes qui sont regroupés sous le nom de sémantique ; nous les présentons dans la sous-section suivante. En pratique, le processus d'induction se ramène à une recherche dans un espace ordonné. La sous-section 2.2.2.2 est dédiée à la présentation des ordres les plus couramment utilisés dans ce contexte. Nous introduisons ensuite les différents biais permettant de délimiter cet espace de recherche. Nous terminons en présentant en sous-section 2.2.2.4 quelques-unes des stratégies d'exploration de l'espace de

recherche effectivement utilisées au sein de systèmes de PLI. Les choix retenus dans le cadre de notre application à l'acquisition de relations qualia sont comme précédemment précisés pour chacun de ces points.

2.2.2.1 Principes et sémantiques

Le principe de la programmation logique inductive est très similaire à la plupart des techniques d'apprentissage supervisé. Il s'agit, étant donné un ensemble d'observations (des exemples positifs et négatifs), un ensemble de connaissances (*background knowledge*) B, un langage d'hypothèses $\mathscr{L}_H$ et une relation de couverture, de trouver une hypothèse (ou un ensemble d'hypothèses) $H \in \mathscr{L}_H$ telle que, sachant B, H couvre tous les exemples positifs et aucun négatif. Dans le cadre de la PLI, H est soit une clause, soit un programme logique, c'est-à-dire un ensemble de clauses (on note alors h_i ses éléments).

On impose par ailleurs que les deux conditions suivantes soit vérifiées par les données d'apprentissage :

- la consistance (ou satisfiabilité) *a priori* est assurée si les exemples négatifs, représentés par des clauses de Horn sans tête, ne sont pas en contradiction avec les connaissances données dans le *background knowledge*, ce qui se note : $B \wedge E^- \not\models \square$;
- la nécessité *a priori* est la traduction du besoin d'une connaissance autre que B, d'une hypothèse, pour expliquer les exemples positifs, ce que l'on traduit par $B \not\models E^+$.

L'hypothèse trouvée doit ensuite vérifier deux autres conditions :

- la consistance (ou satisfiabilité) *a posteriori* impose qu'aucune contradiction ne soit trouvée entre B, H et E^-, ce que l'on note : $B \wedge H \wedge E^- \not\models \square$;
- la complétude consiste à s'assurer que l'hypothèse, combinée au *background knowledge*, permet bien d'expliquer tous les exemples positifs, soit $B \wedge H \models E^+$.

Les quatre conditions précédentes, définissant la problématique de la PLI, sont énoncées pour des programmes normaux ; on parle alors de *sémantique normale*. Les algorithmes de PLI se placent pour la plupart dans le cadre de la logique des clauses définies. Les quatre conditions précédentes se traduisent dans ce formalisme, en sémantique définie, sous la forme suivante (Muggleton & De Raedt, 1994) :

- $\forall e^- \in E^-, e^-$ est faux dans $\mathcal{M}(B)$;
- $\exists e^+ \in E^+, e^+$ est faux dans $\mathcal{M}(B)$;
- $\forall e^- \in E^-, e^-$ est faux dans $\mathcal{M}(B \wedge H)$;
- $\forall e^+ \in E^+, e^+$ est vrai dans $\mathcal{M}(B \wedge H)$.

où $\mathcal{M}(\Phi)$ est le plus petit modèle de Herbrand associé au programme défini Φ dans lequel toute formule logique est soit vraie soit fausse et où les exemples négatifs sont maintenant représentés par des faits faux.

D'autres sémantiques présentant la problématique de la PLI sous d'autres aspects ont été proposées. Parmi ces alternatives, on peut citer la sémantique non monotone

proposée par N. Helft (1989) et P. Flach (1992) où les exemples font partie intégrante du *background knowledge* (les négatifs sont définis implicitement par l'hypothèse du monde clos). Dans ces conditions, les trois contraintes suivantes doivent être respectées :

- la validité : $\forall h \in H$, h est vraie dans $\mathcal{M}(B)$;
- la complétude : si la clause générale g est vraie dans $\mathcal{M}(B)$, alors $H \models g$;
- la minimalité : aucun sous-ensemble strict de H n'est à la fois valide et complet.

Le lecteur intéressé peut se reporter aux articles cités ainsi qu'à (Muggleton & De Raedt, 1994) pour une description détaillée de cette sémantique et une comparaison à la sémantique normale. Dans la suite de ce mémoire nous nous plaçons, sauf s'il est explicitement précisé le contraire, dans le cadre de la sémantique définie.

2.2.2.2 Notion de généralité dans l'espace des hypothèses

La problématique de la PLI peut se ramener à une recherche de clauses dans un espace satisfaisant un certain nombre de propriétés. Malheureusement, en contrepartie de l'expressivité du langage employé, cet espace d'hypothèses est généralement trop grand pour être exploré extensivement. Cependant, il peut être organisé hiérarchiquement, à l'aide d'une notion de généralité entre clauses. L'exploration peut ainsi être menée plus efficacement, guidée par cette généralité.

Comme pour les autres techniques d'apprentissage, la relation attendue pour rendre compte de la généralité (notée $\succeq$) doit être cohérente avec la relation de couverture des exemples (voir section 2.2.1.3).

Dans le cas des logiques dérivées du premier ordre, trouver un tel ordre de généralité est bien plus difficile que dans le cas propositionnel où l'inclusion des attributs pouvait servir d'ordre. L'implication logique serait certainement la solution idéale mais des résultats d'indécidabilité en empêche l'utilisation dans le cas général (Schmidt-Schauß, 1988 ; Nienhuys-Cheng & de Wolf, 1996), et même si, comme généralement en PLI, l'on se restreint aux clauses de Horn (Marcinkowski & Pacholski, 1992).

Un autre ordre, largement utilisé dans les systèmes de PLI et défini ci-dessous, est la θ-subsomption (Plotkin, 1970).

Définition 3 (θ-subsomption) *Une clause C_1 θ-subsume une clause C_2 ($C_1 \succeq_\theta C_2$) si et seulement si (ssi) il existe une substitution θ telle que $C_1\theta \subseteq C_2$ (en considérant les clauses comme des ensembles de littéraux).* □

Par exemple, la clause $p(a,b) \leftarrow r(b,a)$ est subsumée par la clause $C = p(Y_1,Y_2) \leftarrow r(Y_2,Y_1)$. En effet, on a bien $\{p(Y_1,Y_2), \neg r(Y_2,Y_1)\}\theta_1 \subseteq \{p(a,b), \neg r(b,a)\}$ avec $\theta_1 = \{Y_1/a, Y_2/b\}$. La clause $p(X_1,X_2) \leftarrow r(X_2,X_1), q(X_1)$ est également subsumée par la clause C, avec par exemple la substitution $\theta_2 = \{Y_1/X_1, Y_2/X_2\}$.

La θ-subsomption est réflexive, transitive et mais pas antisymétrique ; c'est donc un quasi-ordre. On peut néanmoins définir une relation d'équivalence sous θ-subsomption : si $C_1 \succeq_\theta C_2$ et $C_2 \succeq_\theta C_1$, on notera cette équivalence $C_1 \sim_\theta C_2$. On a ainsi un ordre partiel (antisymétrique) sur les classes d'équivalence. Ces dernières possèdent un unique représentant, appelé clause réduite, qui le plus petit ensemble de littéraux équivalent

aux autres clauses sous θ-subsomption (Nienhuys-Cheng & de Wolf, 1997). Les classes d'équivalence ainsi formées sont organisées en treillis.

Cet ordre induit est plus faible que l'implication ($C_1 \succeq_\theta C_2 \Rightarrow C_1 \models C_2$ mais l'inverse n'est toujours vrai (Plotkin, 1971)) mais décidable (Robinson, 1965), et donc plus facilement manipulable bien qu'il soit encore NP-difficile (Kapur & Narendran, 1986). Il est cependant montré au travers des lemmes de Gottlob (Gottlob, 1987) et de Lee (Lee, 1967 ; Muggleton, 1992) que ces deux ordres sont très proches, et même équivalents si l'on considère des clauses non récursives (c'est-à-dire n'intervenant pas pour leur propre résolution).

Différentes notions de généralité ont été proposées. Certaines visent à pallier les différences entre implication et θ-subsomption (par exemple la T-implication (Idestam-Almquist, 1995)). D'autres ont été définies dans le but d'améliorer la faisabilité du parcours de $\mathscr{E}_H$ (comme la subsomption empirique (Champesme *et al.*, 1995a ; Champesme *et al.*, 1995b)), ou bien de rendre la complexité de la comparaison de deux hypothèses polynomiale (*e.g.* la subsomption stochastique (Rouveirol & Sebag, 1997)), ou encore de contraindre les substitutions possibles (*e.g.* la subsomption sous identité objet (Esposito *et al.*, 1996 ; Badea & Stanciu, 1999), *cf.* section 3.1.2.2).

La relative absence de prise en compte des informations contenues dans le *background knowledge* par la θ-subsomption a également mené à des notions de généralité telles que la (θ-)subsomption relative de G. Plotkin (1971) ou encore la subsomption généralisée de W. Buntine (1988) (voir section 3.1.2.2). Une définition opérationnelle de la généralité entre clauses définies et permettant naturellement la prise en compte de B est la SLD-subsomption. Elle repose sur le principe de résolution de Robinson (1965) : $h_1 \succeq h_2$ ssi $B \wedge h_1 \vdash_{SLD} h_2$.

Organisation de l'espace de recherche dans ASARES

Il est important d'utiliser une relation de généralité qui soit adaptée au problème traité. Dans le cadre de notre application, les hypothèses manipulées utilisent des informations linguistiques (les étiquettes des mots) qui sont structurées : par exemple, un nom commun pluriel est une sorte de nom commun. Il est donc nécessaire que cette structure d'informations se retranscrive hiérarchiquement dans l'espace de recherche. De plus, les clauses obtenues doivent ensuite servir de patrons d'extraction et permettre d'identifier les schémas linguistiques portant la relation ciblée. De ce fait, leur forme est contrainte, ce qui a une incidence également sur l'organisation de l'espace de recherche. Ces deux conditions nous amènent donc à définir un ordre de généralité particulier à notre tâche mélangeant subsomption sous identité objet et subsomption généralisée (voir section 3.1.2.2).

2.2.2.3 Biais de langage

En PLI, l'espace de recherche $\mathscr{E}_H$ manipulé est dans la plupart des cas pratiques gigantesque et peut même être infini (contrairement aux espaces des problèmes attribut-

valeur). Il est donc important pour la faisabilité de la tâche d'apprentissage d'imposer des contraintes sur $\mathscr{E}_H$ pour en réduire la taille et ainsi rendre possible son exploration. Ces contraintes portent le plus souvent sur les éléments qui le composent, c'est-à-dire sur les hypothèses susceptibles d'être proposées par l'algorithme de PLI et sur la façon de les chercher. Ces restrictions sont appelées biais puisqu'elles vont en effet biaiser le résultat en privilégiant certaines hypothèses par rapport à d'autres sur des considérations autres que leur simple adéquation avec les exemples. Il existe de nombreux types de biais (Nédellec *et al.*, 1996) ; la définition donnée par T. Mitchell (1980) est d'ailleurs très large : *« the term bias refers to any basis for choosing one generalization over another, other than strict consistency with the observed training instances ».*

Outre l'aspect combinatoire de ces biais, leur autre effet majeur est de permettre d'améliorer la qualité des hypothèses en ajoutant au système de la connaissance sur le concept recherché. Il est d'ailleurs bien connu que loin d'être une entrave à l'apprentissage, ces biais en assurent au contraire la faisabilité puisqu'un système fonctionnant absolument sans aucun biais est incapable de généraliser les exemples et donc de produire l'induction nécessaire à la généralisation (Mitchell, 1980).

Parmi les différents types de biais, les biais déclaratifs s'attachent à définir la forme attendue des hypothèses, c'est-à-dire à préciser le langage $\mathscr{L}_H$. Ainsi, seules les règles réellement pertinentes sont proposées. En PLI, on distingue usuellement au sein des biais déclaratifs deux grandes familles : les biais syntaxiques et les biais sémantiques (Muggleton & De Raedt, 1994). Une hypothèse de $\mathscr{E}_H$ répondant aux descriptions des biais est alors dite bien formée.

Biais syntaxiques

Les biais syntaxiques permettent de restreindre la taille des clauses produites, c'est-à-dire de limiter le nombre de littéraux qui y apparaissent. Le système a ainsi tendance à produire des hypothèses courtes et générales. On peut de la même manière imposer des restrictions sur l'utilisation des variables dans les clauses : il peut être intéressant de limiter le nombre de variables apparaissant dans les clauses, ou leur ordre d'apparition ou la façon dont elles sont reliées les unes aux autres à travers les littéraux.

Tous les systèmes de PLI, depuis les tous premiers, utilisent ce type de biais de manière plus ou moins sophistiquée. Certains utilisent même des langages dédiés pour leur description, comme $\mathcal{D}$LAB (Declarative LAnguage Bias) (Dehaspe & De Raedt, 1995) ou les schémas d'hypothèses de MOBAL (Kietz & Wrobel, 1992).

Biais sémantiques

L'autre type de biais déclaratif largement utilisé sont les biais sémantiques. Également utilisés dès les premiers systèmes de PLI (comme MIS (Shapiro, 1981)), ils sont désormais utilisés par la quasi-totalité des algorithmes de PLI (notamment PROGOL (Muggleton, 1995) et ses successeurs). Ils permettent plus spécifiquement de préciser le fonctionnement attendu des littéraux d'une clause. C'est le rôle des déclarations de

types et de modes. Les premiers indiquent de quels types sont les variables apparaissant dans les prédicats. Ainsi, si dans une hypothèse, un littéral utilise une variable d'un certain type t, les autres littéraux ne peuvent utiliser cette même variable que s'ils sont déclarés comme pouvant fonctionner avec des variables de type t. Les déclarations de modes permettent quant à elles de préciser le fonctionnement attendu des prédicats utilisés en termes de variables d'entrée et de sortie. Cela autorise à se restreindre aux clauses dont les littéraux utilisent en variables d'entrées les variables de sortie des littéraux précédents par exemple.

Biais de recherche

Outre les deux types de biais précédents, d'autres formes d'informations peuvent être utilisées pour influencer la tâche d'apprentissage. C'est notamment le cas des heuristiques de recherche, appelées aussi biais de préférence, qui vont diriger le parcours de $\mathscr{E}_H$ et ainsi favoriser certaines hypothèses par rapport à d'autres.

Ce type de biais est bien sûr intimement lié au choix de parcours de $\mathscr{E}_H$ (voir la section suivante). Une heuristique largement employée, notamment dans FOIL (Quinlan & Cameron-Jones, 1995) est l'utilisation d'une mesure de gain de compression s'appuyant sur la théorie de l'information.

Langage d'hypothèses dans ASARES

L'algorithme de PLI que nous utilisons au sein de notre système d'acquisition d'éléments sémantiques est ALEPH (Srinivasan, 2001), un successeur de PROGOL. Nous bénéficions donc de toutes les possibilités de biais déclaratifs tels que les déclarations de types et de modes. Ces biais et d'autres permettent de définir très précisément le langage $\mathscr{L}_H$ que nous utilisons pour que les hypothèses produites soient non seulement valides vis-à-vis des exemples mais aussi pertinentes d'un point de vue linguistique. Tous les choix de biais restreignant le langage d'hypothèses, définissant les règles que l'on dira bien formées au regard de notre tâche, sont décrits et motivés en section 3.1.2.1.

Comme les biais sont omniprésents en PLI, on suppose par la suite que, même si cela n'est pas explicitement précisé, les hypothèses considérées sont bien formées au regard de ces biais.

2.2.2.4 Stratégies de recherche

Les biais présentés ci-dessus permettent de définir précisément la forme des hypothèses recherchées. La notion de généralité retenue entre ces dernières autorise quant à elle l'espace de recherche $\mathscr{E}_H$ à être organisé hiérarchiquement. Différentes stratégies pour parcourir cet espace des hypothèses bien formées sont possibles. La mise en œuvre de ces stratégies est bien sûr intimement liée à la notion de généralité entre hypothèses retenue et également au langage $\mathscr{L}_H$ précisé par le biais, c'est-à-dire à la forme des hypothèses attendues.

Nous présentons quelques-unes des approches utilisées en pratique au sein des différents systèmes de PLI existants. Le lecteur intéressé par une description fine de ces systèmes ou à leur comparaison suivant d'autres critères peut se rapporter aux articles cités et à (Muggleton & De Raedt, 1994).

Recherche descendante

Un premier type de stratégie pour l'exploration de l'espace de recherche des hypothèses peut être une approche descendante (ou *top-down* en anglais). Une telle approche peut se décomposer en deux niveaux :

1. trouver une règle h par spécialisation (voir ci-après) à partir des ensembles d'exemples E^+ et E^- ;
2. trouver un ensemble de règles H en réitérant l'étape 1 et en mettant à jour les ensembles d'exemples autant de fois que nécessaire.

La mise à jour de E^+ et E^- consiste par exemple à ôter les exemples déjà couverts par une règle h trouvée précédemment.

L'étape 1 correspond à une exploration de l'espace des hypothèses du plus général au plus spécifique. Le principe utilisé est donc celui de l'inférence déductive puisqu'il s'agit d'obtenir une formule S à partir d'une formule G avec $G \models S$. Le point de départ de cette recherche est la définition la plus générale du concept, par exemple $\square$ (la clause vide), ou bien de manière plus pratique, la clause $(ccpt(X_1, ..., X_n) \leftarrow)$ où *ccpt* est le prédicat représentant le concept.

L'exploration se fait de manière constructive : à chaque hypothèse, un opérateur propose toutes les clauses qui lui sont plus spécifiques (selon la notion de généralité retenue). L'opérateur produisant ces spécialisations de clauses est appelé, selon le terme proposé par E. Shapiro (1981), opérateur de raffinement (voir section 3.2.1). Ainsi, dans un espace organisé par la θ-subsomption, l'opérateur de raffinement effectuera basiquement deux opérations sur une hypothèse pour en générer des hypothèses plus spécialisées :

1. appliquer une substitution, et ainsi transformer des variables de la clause de départ en constantes ou unifier plusieurs variables ;
2. ajouter des littéraux à la clause de départ.

Les hypothèses ainsi générées sont donc ensuite testées, c'est-à-dire que l'on examine leur couverture des ensembles E^+ et E^-. Pour répondre aux spécifications établies en sémantique définie, on voudra en particulier qu'aucun $e^- \in E^-$ ne soit couvert par l'hypothèse h testée, mais on voudra également que h couvre le maximum d'exemples positifs pour faire converger au mieux l'étape 2. Enfin, le choix des spécialisations effectivement suivies (raffinées à leur tour) peut se faire à l'aide d'heuristiques, souvent calculées à l'aide des taux de couverture de E^+ et E^-. Ce type d'approche de la PLI appartient donc à la famille des algorithmes dits *generate-and-test* (générer et tester), dont le système $\mathcal{HAIKU}$ (Nédellec & Rouveirol, 1994) représente un cadre unifié puisqu'il peut s'identifier à chacun de ces algorithmes selon les paramétrages choisis.

Ce type de fonctionnement a connu beaucoup de succès, et de nombreux algorithmes de PLI ont été développés en s'appuyant sur une philosophie proche. Outre le précurseur MIS (pour Model Inference System) de E. Shapiro (1981), on peut citer FLIPPER (Cohen, 1995), LINUS (Lavrač & Džeroski, 1994), ML-SMART (Bergadano *et al.*, 1988), et le très célèbre FOIL (Quinlan & Cameron-Jones, 1995) et ses nombreuses variantes et adaptations (Pazzani *et al.*, 1991 ; Cohen, 1994 ; Džeroski & Bratko, 1992)

Se plaçant dans un cadre proche, MOBAL (Kietz & Wrobel, 1992) utilise une extension de la θ-subsomption compatible avec ses modèles de règles (voir section précédente) pour générer ses hypothèses des plus générales aux plus spécifiques. Notons également que d'autres logiciels, se positionnant dans une sémantique autre que la sémantique normale, utilisent également une approche descendante pour parcourir leur espace de recherche sous θ-subsomption. C'est le cas de CLAUDIEN (De Raedt & Bruynooghe, 1993 ; Dehaspe *et al.*, 1994 ; De Raedt & Dehaspe, 1996), de TILDE (Blockeel & De Raedt, 1998 ; Blockeel, 1998), et d'ICL (De Raedt & van Lear, 1995).

Recherche ascendante

De nombreux systèmes de PLI reposent également sur une exploration ascendante (*bottom-up*) de l'espace de recherche. Le point de départ de la recherche est donc un sous-ensemble des exemples positifs (souvent des paires d'exemples) que l'on cherche à généraliser en une clause. Symétriquement au cas précédent, les règles utilisées pour ce faire sont donc des règles d'inférence inductive qui se définissent formellement comme permettant d'obtenir un ensemble de formules G à partir de S telles que $G \models S$.

Ces méthodes utilisent la notion de moindre généralisé, c'est-à-dire le plus petit pas inductif dans $\mathscr{E}_H$, pour explorer étape par étape cet espace. Plus formellement, le moindre généralisé se définit comme ci-dessous (Torre, 2000).

Définition 4 (Moindre généralisé) *Un moindre généralisé G de H_1, H_2, ..., H_n sous un ordre de généralité noté $\succeq$ doit vérifier les deux conditions suivantes :*

- *pour tous les H_i, on a $G \succeq H_i$;*
- *s'il existe G' tel que $(\forall H_i : G' \succeq H_i) \wedge (G \succeq G')$ alors $G \sim G'$.* □

Comme dans la recherche descendante, c'est un opérateur qui a pour fonction de produire à partir d'une clause ses généralisations. Cet opérateur est également appelé opérateur de raffinement bien que les clauses produites ne soient en aucun cas plus spécifiques que l'hypothèse d'origine.

À la base de la plupart des algorithmes ascendants se trouvent les moindres généralisés correspondant à la relation de θ-subsomption. Ces moindres généralisés sont appelés de leur acronyme anglais *lgg* (*least greater generalization*) et ont été proposés par G. Plotkin (1970 ; 1971) en même temps que la θ-subsomption (il propose également un algorithme polynomial pour les calculer).

L'utilisation des *lgg* en PLI se fait au travers de leurs variantes permettant de prendre en compte le *background knowledge*. Ces variantes sont donc qualifiées de *relatives* (au *background knowledge*), et notées *rlgg* (Plotkin, 1971). La *rlgg* de deux clauses

e_1 et e_2 se définit par : $rlgg(e_1, e_2) = lgg(e_1 \leftarrow \mathcal{M}(B), e_2 \leftarrow \mathcal{M}(B))$, où $\mathcal{M}(B)$ est un modèle du *background knowledge* B. C'est l'approche utilisée dans un des logiciels pionniers de la PLI, GOLEM (Muggleton & Feng, 1990), suivi par de nombreux autres. Une discussion sur la complexité de ces moindres généralisés suivant les spécificités des systèmes de PLI est proposée par J.-U. Kietz (1993).

L'utilisation des *rlgg* pose cependant un certain nombre de problèmes. Tout d'abord, les modèles du *background knowledge* B (et plus particulièrement le plus petit modèle de Herbrand) peuvent être infinis. Dans ce cas, la *rlgg* de deux clauses pourra donc être également infinie. Même dans le cas où un sous-ensemble fini d'un modèle de B est utilisé, la *rlgg* de deux clauses peut être de taille très importante et croître exponentiellement avec le nombre d'exemples. Enfin, les concepts contenant plusieurs clauses ne peuvent pas être appris par des algorithmes dont la phase d'induction repose exclusivement sur l'utilisation des *rlgg* puisque cette dernière ne produit qu'une seule règle à partir d'un ensemble d'exemples.

Une autre approche utilisant des règles d'inférence inductive pour explorer $\mathscr{E}_H$ du plus spécifique au plus général repose sur l'idée qu'il est possible d'inverser le principe de résolution de Robinson (1965). Cette approche, appelée *inverting resolution*, repose donc sur la SLD-subsomption vue précédemment. Elle a donné lieu à de nombreux logiciels, exploitant chacun différentes règles d'induction : DUCE (Muggleton, 1987), CIGOL (Muggleton & Buntine, 1988), MARVIN (Sammut & Banerji, 1986). Certains se sont notamment attachés à restreindre le non déterminisme de l'opération de résolution inverse — véritable obstacle combinatoire de cette approche —, ou à renverser en un pas plusieurs étapes de résolution (LFP2 (Wirth, 1989), IRES (Rouveirol & Puget, 1990)). D'autres (Rouveirol, 1992 ; Rouveirol, 1994) permettent de gérer plus facilement le cas des clauses contenant des fonctions (Nienhuys-Cheng & de Wolf, 1997 ; Hirata, 1999).

Stratégie de recherche mixte

Enfin, des stratégies hybrides peuvent être utilisées pour trouver plus efficacement les hypothèses intéressantes au sein de l'espace de recherche. Les systèmes développant ce type de stratégie combine donc des approches *bottom-up* et *top-down* similaires à celles exposées précédemment.

Par exemple, le logiciel CHILLIN (Zelle *et al.*, 1994) conduit successivement une recherche *bottom-up* puis *top-down*. La première fonctionne comme GOLEM par *rlgg* : deux exemples sont choisis, et une clause $C = rlgg(e1, e2)$ est construite. Si cette clause est jugée trop générale au regard des exemples négatifs, elle est raffinée par ajout de littéraux à la manière de FOIL. Enfin, si après les ajouts de tous les littéraux valides, la clause est encore jugée trop générale, un prédicat est inventé et lui est ajouté.

L'approche retenue dans PROGOL (Muggleton, 1995) et ses successeurs combine elle aussi, successivement et dans cet ordre, une approche ascendante avec une approche descendante. Dans la première, l'inférence inductive s'appuie non pas sur la résolution inverse mais sur ce que S. Muggleton appelle l'implication (entre modèles) inverse (*inverting entailment*). Cette approche étant celle utilisée par ALEPH, le programme

d'inférence adopté au sein d'ASARES, nous en détaillons en conséquence quelque peu le fonctionnement.

Le principe de l'implication inverse repose sur la constatation que pour un exemple e, on veut trouver une clause h telle que $B \wedge h \models e$, ce qui est équivalent à $B \wedge \overline{e} \models \overline{h}$ en notant $\overline{x}$ la négation de x. Puisque h et e sont de simples clauses, leurs négations sont des clauses unitaires skolemisées, c'est-à-dire un programme logique clos. Soit $\overline{\bot}$ la conjonction (potentiellement infinie) de littéraux sans variables qui sont vrais pour tous les modèles de $B \wedge \overline{e}$. Puisque $\overline{h}$ doit être vraie dans tous les modèles de $B \wedge \overline{e}$, elle doit être un sous-ensemble de $\overline{\bot}$; ainsi $B \wedge \overline{e} \models \overline{\bot} \models \overline{h}$. Soit encore pour tout h, on a $h \models \bot$.

Dans le cas général, $\bot$ peut être de taille infinie; PROGOL (et ses variantes) utilise donc un ensemble de biais syntaxiques (limitation de la taille des hypothèses) et surtout des biais sémantiques (déclarations de modes) pour définir $\mathscr{L}_H$. On définit la clause la plus spécifique répondant à $\mathscr{L}_H$, notée $\bot_{\mathscr{L}_H}$, comme étant la clause la plus spécifique de $\mathscr{L}_H$ telle que $\bot_{\mathscr{L}_H} \succeq \bot$ (voir (Muggleton & De Raedt, 1994 ; Muggleton, 1998) pour le détail de sa construction). Une recherche descendante peut alors être utilisée pour trouver un sous-ensemble des solutions pour h en considérant les clauses qui θ-subsument $\bot_{\mathscr{L}_H}$. Pour ce faire PROGOL applique un algorithme A^* avec pour critère la compression maximale de la théorie résultante.

Enfin, d'autres stratégies d'exploration plus complexes peuvent être utilisées pour parcourir $\mathscr{E}_H$. Citons par exemple les travaux récents de A. Tamaddoni-Nezhad & S. Muggleton (2000 ; 2002) qui proposent d'exploiter le formalisme des algorithmes génétiques pour représenter les opérations d'exploration possibles dans le treillis de clauses sous θ-subsomption. Les opérateurs de raffinement génétiques qui en sont déduits permettent de parcourir le treillis efficacement par généralisation ou spécialisation.

2.2.3 La PLI en pratique

Les techniques de PLI, bien que relativement récentes, ont été appliquées à un grand nombre d'applications réelles (Lavrač & Džeroski, 1994), dans des domaines variés : la découverte de connaissances dans des bases de données, la découverte scientifique (Muggleton, 1999b), la modélisation biomoléculaire... De ce fait, comme nous le voyons en section suivante, certaines des définitions théoriques présentées ci-dessus se trouvent assouplies pour répondre de manière plus adéquate aux problèmes traités et aux particularités des données. Nous présentons ensuite quelques-unes des familles d'applications du TAL exploitant le cadre formel et les capacités d'inférence de la PLI.

2.2.3.1 Gestion de bruit et données imparfaites

Dans des applications réelles d'apprentissage, les conditions expérimentales sont rarement aussi parfaites que ce qui est attendu en théorie. Il est donc important que les techniques d'apprentissage puissent s'adapter à ces conditions dégradées et préserver au mieux leurs caractéristiques théoriques. L'une des causes les plus évidentes de ces condi-

tions d'apprentissage dégradées est l'imperfection des données d'apprentissage (Lavrač & Džeroski, 1992). Ces imperfections peuvent être dues à un manque de données, ou à des erreurs dans les données : le *bruit*.

En apprentissage supervisé, le bruit dans les données peut être principalement de trois natures différentes. Il peut soit s'agir d'erreurs dans la classe attribuée à un exemple, c'est-à-dire une erreur de supervision due à l'expert. Cela peut également être une erreur dans la description d'un exemple, comme assigner une mauvaise valeur à un attribut dans le cas d'un langage de description attribut-valeur. Enfin, cela peut être une description erronée des connaissances externes manipulées par le système d'apprentissage (par exemple une erreur dans le *background knowledge*), ou simplement inadaptée (par exemple, un langage d'hypothèses trop contraint pour permettre l'expression et donc l'apprentissage du concept).

Comme beaucoup de techniques d'apprentissage symbolique, la PLI s'intéresse à des phénomènes structurels trouvés dans les exemples sans prendre en compte *a priori* d'informations numériques sur la fréquence de ces phénomènes. Elle est donc plus sensible au bruit que les méthodes d'apprentissage statistique par exemple. La définition donnée en section 2.2.2.1 impose notamment, que ce soit en sémantique normale ou définie, que tous les exemples positifs soient couverts par l'hypothèse et que tous les négatifs soient rejetés. Une telle assertion n'est bien sûr pas appliquée en pratique puisqu'elle interdit toute erreur dans les données d'apprentissage. Un unique exemple mal classé par l'expert peut ainsi empêcher toute induction du concept cherché. On adopte, dans le cadre d'applications réelles, une définition du principe de fonctionnement plus faible mais plus adaptée à la gestion de données bruitées. Une telle définition peut s'exprimer de la manière suivante; étant donnés :

- un ensemble d'observations (exemples positifs E^+ et exemples négatifs E^-) ;
- un ensemble de connaissances préalables B (le *background knowledge*) ;
- un langage d'hypothèses $\mathscr{L}_H$;
- une relation de couverture ;
- un critère de qualité (une fonction de score) Sc ;

trouver l'ensemble d'hypothèses $H \in \mathscr{L}_H$ tel que (sachant B) H est optimal par rapport au critère Sc calculé à partir de la couverture de H sur E^+ et E^-.

Le critère de qualité peut être utilisé pendant la recherche d'hypothèses (comme le gain de compression dans FOIL ou LINUS (Gamberger & Lavrač, 1996 ; Quinlan, 1990 ; Muggleton *et al.*, 1992)) ou bien intervenir après la recherche pour élaguer les hypothèses non pertinentes (comme dans FOCL (Pazzani & Brunk, 1991)). Le lecteur intéressé peut se reporter à (Lavrač & Džeroski, 1994) pour un panorama la gestion des données imparfaites en PLI.

Données imparfaites dans ASARES

Les outils du TAL travaillant sur corpus se doivent d'être résistants aux données bruitées. Notre système est de ce point particulièrement exposé. En effet, outre les erreurs « classiques » telles que les fautes (d'orthographe, grammaticales) potentiellement présentes dans le corpus et les erreurs d'étiquetage des mots, notre méthode dépend d'exemples. Ces derniers peuvent être également sources de bruit puisque l'expert peut assigner une mauvaise classe à une occurrence. C'est ainsi à la fois la description des exemples et leur classement au sein des ensembles E^+ et E^- qui peuvent être erronés. Nous verrons dans le chapitre suivant que cela est principalement géré par une notion de score des clauses permettant à quelques exemples négatifs d'être couverts par une hypothèse.

2.2.3.2 Traitement automatique des langues et PLI

Le traitement automatique des langues bénéficie de plusieurs réalisations impliquant la PLI ou plus largement l'exploitation d'inductions dans des formalismes logiques. Ce domaine, à la croisée de la logique, du TAL et de l'apprentissage artificiel, est appelé en anglais *Learning Language in Logic* (LLL).

La PLI a notamment été appliquée avec succès à des problèmes du TAL dans lesquels une représentation logique est naturelle ; la plupart d'entre eux sont décrits dans (Cussens & Džeroski, 2000 ; Cussens, 1998 ; Mooney, 1997). Nous en distinguons ci-dessous quelques-uns parmi ceux-ci selon leur champ d'application :

- l'étiquetage syntaxique (Cussens & Pulman, 2000) et l'apprentissage de grammaires catégorielles ;
- l'apprentissage de cadres de sous-catégorisation de verbes (Faure & Nédellec, 1999) ;
- l'étiquetage morphosyntaxique ou phonologique (Cussens, 1996 ; Eineborg & Lindberg, 2000) ;
- la segmentation de mots (Kazakov & Manandhar, 2001) ;
- la représentation sémantique (logique) (Zelle, 1995 ; Mooney, 1999) ;
- l'extraction d'informations (Thompson & Califf, 2000) ;
- la catégorisation de texte (Junker *et al.*, 2000).

L'utilisation de la PLI pour de tels problèmes du TAL a de multiples intérêts (Muggleton, 1999a ; Džeroski *et al.*, 2000) :

- les règles inférées sont interprétables par un linguiste ;
- l'intégration de connaissances externes est réalisée très naturellement par l'utilisation du *background knowledge* ;
- la représentation des problèmes bénéficie du formalisme de logique du premier ordre.

D'une manière générale, l'emploi de la PLI est particulièrement intéressant pour les problèmes manipulant des informations structurées complexes (voir section 2.2.1.4). Il est cependant nécessaire de disposer d'exemples en nombre suffisant, ce qui représente souvent un frein à l'utilisation de telles techniques supervisées (Brill, 2000). Les

exemples négatifs notamment, même s'ils ne sont pas indispensables, ne sont pas toujours disponibles pour certains problèmes. Il est heureusement possible d'utiliser des techniques permettant de contourner ces difficultés : réduction du nombre d'exemples nécessaires, adjonction de techniques externes (voir l'approche que nous adoptons en section 4.2), ou collaboration interactive avec un expert.

Pour notre part, nous utilisons la PLI pour la génération de patrons morphosyntaxiques et sémantiques décrivant les réalisations en corpus d'informations sémantiques lexicales. L'aspect structurel exploité dans notre approche est donc principalement la structure de la phrase (la succession des mots de la phrase). Les travaux les plus proches sont donc ceux effectués dans le domaine de l'extraction d'informations (voir par exemple le système RAPIER (Califf & Mooney, 1997)). Dans ceux-ci, la PLI est en effet utilisée pour produire des patrons permettant d'extraire les informations nécessaires au remplissage de cadres prédéfinis (*templates*) à partir de phrases étiquetées. Outre une finalité différente, ces travaux se distinguent de notre approche par les exigences que nous imposons sur la forme des patrons attendus et leurs implications sur la phase d'inférence.

2.3 Cadre applicatif

Comme nous l'avons dit précédemment, la tâche à laquelle nous avons choisi de confronter notre système ASARES est l'acquisition sur corpus de relations qualia définies au sein du modèle du Lexique génératif (Pustejovsky, 1995). Ces relations sémantiques entre noms et verbes relativement peu étudiées permettent de valider l'approche suivie lors du développement de notre outil, notamment en ce qui concerne le besoin d'interprétabilité.

Nous présentons dans un premier temps ce modèle et ses motivations, et nous détaillons plus particulièrement la notion de couples qualia. Nous mettons ensuite en évidence quelques-uns des intérêts applicatifs que ce type de lexique peut revêtir, notamment en recherche d'information, domaine dans lequel nous utilisons effectivement les couples qualia acquis (voir chapitre 5).

2.3.1 Le Lexique génératif

J. Pustejovsky a proposé un modèle de représentation de données lexicales nommé Lexique génératif (Generative Lexicon, (Pustejovsky, 1995)). Nous présentons dans un premier temps ce qui a motivé son travail et en particulier les insuffisances des lexiques existants tels ceux de type SEL (*Sense Enumeration Lexicon* : lexique à énumération de sens), puis nous détaillons les principales caractéristiques de son modèle.

2.3.1.1 Limites des lexiques traditionnels

Le langage naturel regorge de phénomènes linguistiques intuitivement compréhensibles mais extrêmement difficiles à formaliser avec précision. Nous allons en présenter

quelques-uns et voir comment les lexiques de type SEL, jusqu'alors communément utilisés, en tenaient difficilement compte.

Le SEL est un modèle de lexique contenant une entrée pour chaque sens des mots. Ainsi le mot *banque* est référencé une fois en tant qu'institution financière et une autre fois comme local. Cependant un tel lexique se heurte à l'usage créatif de sens. Il semble en effet difficile de prévoir *a priori* toutes les utilisations possibles d'un mot en contexte, comme cela est illustré par l'adjectif *rapide* dans les phrases de l'exemple 2.1. Un lexique traditionnel imposerait une entrée pour chacun de ces sens pour capturer les différentes facettes sémantiques de cet adjectif.

Exemple 2.1 (Différences aspectuelles)

- *une voiture rapide* fait référence au fait que le conducteur conduit vite ou que la voiture en elle-même est puissante ;
- *le tennis est un jeu rapide* veut dire que le tennis implique de bouger vite ;
- *une route rapide* désigne une route où les voitures roulent vite.

Un autre défaut important des lexiques de type SEL est qu'ils ne prennent pas en compte la perméabilité des sens d'un mot. En effet, ils n'indiquent pas comment les sens des différentes entrées communiquent. Par exemple, dans la phrase *Il s'évada par la fenêtre*, le mot *fenêtre* fait référence à la fois à l'ouverture et à l'objet. Ceci est insaisissable avec les lexiques usuels.

De plus, les différentes constructions argumentales possibles pour un sens d'un mot nécessitent plusieurs entrées dans les lexiques de type SEL alors que le sens est le même, comme le verbe *tuer* dans l'exemple suivant.

Exemple 2.2 (Alternance argumentale)

- *Jean tua Marie*
- *le pistolet tua Marie*
- *la guerre tua Marie*

Enfin, et c'est peut-être le plus important, ce genre de lexique ne tient absolument pas compte du caractère compositionnel du langage, c'est-à-dire qu'on ne considère pas que les sens des mots interagissent les uns sur les autres au sein d'une phrase. Cela empêche donc toute analyse correcte de métaphores par exemple. Ces quelques remarques ont donc motivé J. Pustejovsky à construire un modèle de lexique plus adapté à ces phénomènes.

2.3.1.2 Le modèle de Pustejovsky

Dans le formalisme du Lexique génératif, les mots sont présentés sous plusieurs aspects simultanément. La représentation est donc plus complexe que pour les lexiques traditionnels, mais elle est aussi plus complète. Nous présentons dans un premier temps les structures capturant les différentes propriétés des mots, puis les mécanismes permettant d'expliquer le sens d'un mot en contexte à l'aide de ces structures.

La représentation d'un mot

Les entrées du Lexique génératif sont constituées d'ensembles structurés de prédicats typés définissant un mot. Ces représentations lexicales peuvent donc être considérées comme des réserves de types sur lesquelles différentes stratégies peuvent être utilisées pour interpréter un mot en contexte. Cette théorie générative du lexique inclut trois niveaux de représentation d'une entrée lexicale : la structure argumentale ($ArgStr$), la structure événementielle ($EventStr$), et la structure des qualia ($QualStr$) comme nous l'illustrons en figure 2.5 pour un mot générique **M**.

$$\left[\begin{array}{l} \mathbf{M} \\ ArgStr = \left[\begin{array}{l} \text{ARG}_1 = \ldots \\ \text{D-ARG}_1 = \ldots \end{array}\right] \\ EventStr = \left[\begin{array}{l} \text{E}_1 = \ldots \\ \text{E}_2 = \ldots \\ \text{RESTR} = \text{relation temporelles entre événements} \\ \text{HEAD} = \text{relation de proéminence} \end{array}\right] \\ QualStr = \left[\begin{array}{l} info \cdot M - lcp \\ \text{CONSTITUTIF} = \ldots \\ \text{FORMEL} = \ldots \\ \text{TELIQUE} = \ldots \\ \text{AGENTIF} = \ldots \end{array}\right] \end{array}\right]$$

FIG. 2.5 – Entrée lexicale générique du Lexique génératif

Toutes les catégories syntaxiques reçoivent le même type de description. Comme leur nom l'indique, les structures argumentale et événementielle contiennent les arguments et les événements nécessaires à la définition d'un mot. Ces éléments peuvent être décrits soit explicitement de manière syntaxique, soit implicitement — dans ce dernier cas, ils sont appelés arguments par défaut (D-ARG) ou événements par défaut (D-E). La structure des qualia met en relation ces différents arguments et événements et définit la façon dont ils prennent part à la sémantique du mot décrit.

Plus précisément, dans la structure des qualia (ou structure qualia ou encore plus simplement la qualia), les mots sont décrits en termes de rôles sémantiques :

1. le rôle *télique* indique le but ou la fonction de l'item (par exemple, *couper* pour *couteau*) ;
2. le rôle *agentif* son mode de création (*construire* pour *maison*) ;
3. le rôle *constitutif* ses parties constitutives (*poignée* pour *tasse*) ;
4. le rôle *formel* décrit la façon dont l'item se définit au sein du ou des concepts auxquels il se rapporte (*(objet physique) contenir (de l'information)* pour *livre*).

La structure des qualia d'un nom est donc principalement composée d'associations verbales, codant des informations relationnelles, notamment en ce qui concerne les rôles télique, agentif et formel. Le rôle constitutif est quant à lui plus généralement porté par des relations avec d'autres noms. Ces quatre rôles correspondent ainsi à des attributs interprétés qui forment le vocabulaire de base de la description d'un mot. Ce sont ces couples formés d'un nom et des verbes apparaissant dans sa structure des qualia que nous nommons couples N-V qualia, ou plus simplement couples qualia.

La figure 2.6 présente la représentation lexicale de *livre*, donnée comme exemple par Pustejovsky (Pustejovsky, 1995), dans lequel l'item apparaît à la fois comme un objet physique et comme un objet contenant de l'information (ce que l'on note par *info.physobj-lcp* où *lcp* signifie *lexical conceptual paradigm*). Cette représentation peut

$$\left[\begin{array}{l} \textbf{livre} \\ ArgStr = \left[\begin{array}{l} \text{ARG}_1 = \text{x : physobj} \\ \text{ARG}_2 = \text{y : info} \end{array}\right] \\ EventStr = \left[\begin{array}{l} \text{D-E}_1 = e_1 \\ \text{D-E}_2 = e_2 \end{array}\right] \\ QualStr = \left[\begin{array}{l} info \cdot physobj - lcp \\ \text{CONSTITUTIF} = \text{partie\_de(x} \cdot \text{y, z : couverture, pages, ...)} \\ \text{FORMEL} = \text{contenir(x, y)} \\ \text{TELIQUE} = \text{lire}(e_1, \text{ w, x} \cdot \text{y}) \\ \text{AGENTIF} = \text{écrire}(e_2, \text{ v, x} \cdot \text{y}) \end{array}\right] \end{array}\right]$$

FIG. 2.6 – Exemple d'entrée lexicale du Lexique génératif

être interprétée comme :
$\lambda x.y\,[livre(x : \texttt{physobj}.y : \texttt{info}) \wedge contenir(x,y) \wedge \lambda w\,\lambda e_1[lire(e_1, w, x.y)] \wedge \exists e_2\,\exists v\,[ecrire(e_2, v, x.y)]]$.

Un réseau de relations est ainsi défini pour chaque nom, par exemple *livre-écrire*, *livre-lire*, *livre-contenir* pour *livre*. Ces relations ne sont pas définies empiriquement mais sont linguistiquement motivées : elles sont nécessaires pour expliquer le comportement sémantique du nom en contexte.

Les mécanismes génératifs

Ces structures des qualia ne seraient rien sans les opérateurs permettant d'en tirer parti. Ce sont notamment ces opérateurs qui doivent assurer l'aspect génératif de ce modèle de lexique en exploitant les informations présentes dans les représentations lexicales, et plus particulièrement dans la structure des qualia, pour permettre d'interpréter les mots en contexte. On distingue :

la **coercition de types** qui permet de *décaler* le type d'un item pour analyser correctement une expression. Cela sert par exemple pour le sous-typage : en supposant que le verbe *conduire* prenne comme argument un objet de type véhicule et que le nom *twingo* est de type voiture, l'énoncé *conduire une twingo* sera correctement interprété car le type de *twingo* est voiture et le type de voiture est véhicule. C'est un exemple très simple, mais la coercition permet aussi d'expliquer que la phrase *Jean commence un livre* signifie *Jean commence à lire un livre*, voire *Jean commence à écrire un livre* en allant chercher les éléments permettant l'interprétation dans les rôles (télique et agentif ici) de la structure des qualia. Bien sûr, pour éviter des reconnaissances abusives d'énoncés faux, il convient de contraindre très fortement cette coercition (Bouillon & Pustejovsky, 1995) ;

la **co-composition** qui permet de résoudre la polysémie en composant les structures qualia du verbe et de son complément. Par exemple, le verbe *couper* a des sens différents dans *couper une pomme* (transformation de l'état de la pomme) et *couper des quartiers de pomme* (création de quartiers par l'action de couper). Ce phénomène est capturé par le fait que la qualia agentive de *quartier* contient le verbe *couper* pour indiquer que c'est son mode de création (ce qui n'est pas le cas de *pomme*) ;

le **liage sélectif** qui est particulièrement adapté pour résoudre la polysémie due aux adjectifs. Il s'agit en effet de chercher, étant donné un nom et un adjectif, le rôle qualia du nom auquel s'applique l'adjectif. Ainsi, en rappelant que le rôle télique de *couteau* est *couper*, l'expression *un bon couteau* est interprété comme *un couteau qui coupe bien*.

Critiques du Lexique génératif

Le Lexique génératif a fait l'objet de plusieurs critiques, portant principalement sur l'hypothèse de généricité. Plusieurs auteurs ont en effet remarqué que, contrairement à ce que semble indiquer les ouvrages de J. Pustejovsky, les mécanismes génératifs ne permettent pas d'interpréter le sens de certains mots en contexte.

D'un point de vue linguistique, A. Copestake (2001) note que le modèle de J. Pustejovsky ne permet pas de prendre en compte certains cas de métonymie logique. Elle en donne le contre-exemple reproduit ci-dessous. Dans le cas 1, les mécanismes génératifs fonctionnent comme attendu, et grâce au verbe *read* contenu dans la structure des qualia, la phrase sera interprétée comme *Kim began reading the book*. En revanche, dans le second cas, la seule interprétation possible serait *Kim began building the tunnel* à partir de l'agentif; mais rien ne permet d'interpréter cette phrase comme *Kim began driving through the tunnel*.

Exemple 2.3 (Métonymie logique)

1. *Kim began the book*
2. *Kim began the tunnel*

D'autres phénomènes non pris en compte à travers le Lexique génératif, tels que les collocations, sont également présentés et discutés dans ces mêmes travaux.

Plus pragmatique, A. Kilgariff (2001) tente de vérifier et de mesurer le nombre de fois où les mécanismes génératifs permettent réellement de trouver le sens d'un mot utilisé d'une manière non standard (se reporter à l'article pour les détails du protocole utilisé). Les résultats sont très faibles : seulement 2% de ces usages sont effectivement interprétables selon le Lexique génératif.

Ces critiques, bien que fondées, ne remettent pas en cause l'intérêt de la structure des qualia. Elles semblent montrer au contraire que plus d'informations sémantiques ou de mécanismes génératifs seraient nécessaires pour prendre en compte de nouveaux phénomènes linguistiques.

2.3.1.3 Acquisition d'éléments du Lexique génératif

Du fait de la relative nouveauté du modèle du Lexique génératif, peu de travaux ont été entrepris pour construire à grande échelle des bases lexicales (pour des applications réelles) répondant aux critères énoncés par J. Pustejovsky (1995). Néanmoins, la structure des qualia a fait l'objet de quelques travaux d'acquisition dans un but très similaire au nôtre.

J. Pustejovsky *et al.* (1993) proposent ainsi d'acquérir les éléments des structures de qualia à partir d'un texte étiqueté syntaxiquement. Pour ce faire, les auteurs utilisent tout d'abord une technique d'extraction statistique de cooccurrences qui est ensuite couplée à un jeu d'heuristiques sous forme de patrons syntaxiques. C'est donc une approche mixte mêlant techniques numériques et structurelles.

Malheureusement, ces travaux ne sont pas clairement évalués, et aucune indication sur la qualité des résultats obtenus n'est donnée. Par ailleurs, la question de la portabilité de la méthode d'un texte à un autre n'est pas abordée, notamment en ce qui concerne l'étiquetage syntaxique. Enfin, les auteurs utilisent des heuristiques syntaxiques porteuses des liens qualia définies *a priori* alors même que de telles structures syntaxiques ne sont pas connues et sont susceptibles de varier selon les corpus.

Se servant des verbes qualia des noms pour définir un cadre à l'interprétation de de la métonymie logique, A. Lapata & A. Lascarides (2003) proposent une technique d'acquisition de couples N-V qualia. Cette technique repose sur un apprentissage probabiliste dérivé du Bayésien naïf (Mitchell, 1997), mais les probabilités d'apparition conjointe des noms et des verbes sont estimées à partir des occurrences où ces composants sont en relation verbe-sujet ou verbe-objet (ces liens syntaxiques sont obtenus par l'analyseur CASS de S. Abney). Il s'agit donc comme précédemment d'une approche mixte utilisant des informations structurelles et numériques. Le corpus utilisé est un corpus généraliste en anglais, mais l'absence d'informations spécifiques à la langue autres que celles issues de l'analyseur syntaxique laisse à penser que cette technique pourrait aisément être adaptée à d'autres langues.

Ces travaux n'offrent toutefois pas de résultats chiffrés concernant la capacité du sys-

tème à trouver des couples N-V en relation qualia. Les auteurs proposent en revanche une évaluation de la plausibilité d'interprétation de métonymies à l'aide des couples extraits. Les performances obtenues, évaluées à l'aide d'un ensemble de soixante personnes, montrent la très bonne adéquation des couples extraits avec l'interprétation « naturelle » des métonymies. Ces résultats valident donc à la fois le modèle théorique de l'utilisation des verbes de la qualia pour l'interprétation d'énoncés et aussi la technique statistique utilisée pour trouver les verbes qualia.

Les travaux que nous venons de décrire soulèvent tous deux la même critique, à savoir que les auteurs se focalisent sur une relation syntaxique pour trouver les couples N-V en relation qualia. Or, s'il semble évident que des couples dont les éléments sont en relation qualia peuvent se trouver liés syntaxiquement, tous les couples liés syntaxiquement ne sont pas des couples qualia et inversement, aucun indice théorique ni expérimental n'assure que les couples qualia soient toujours reliés syntaxiquement. Ces hypothèses sont d'ailleurs en partie invalidées par les résultats de l'analyse comparée des couples N-V en relation qualia et des couples liés syntaxiquement dans le corpus utilisé lors de nos propres expérimentations (*cf.* section 3.3.3.2).

2.3.2 Intérêts applicatifs du Lexique génératif

L'intérêt majeur du Lexique génératif réside certainement dans sa capacité à modéliser aisément de nombreux phénomènes linguistiques contenant de l'information implicite (Bouillon & Kanzaki, 2001 ; Bouillon & Kanzaki, 2003). Cette propriété est notamment exploitée dans les deux familles d'application exposées ci-dessous.

2.3.2.1 Interprétation des composés

Comme nous l'avons vu en page 1.3.1.1, certaines séquences complexes résistent à une interprétation directe par simple examen de leurs constituants. En effet, alors qu'il est relativement aisé de mettre sous forme prédicative un composé contenant un déverbal, d'autres nécessitent de faire appel à une connaissance sémantique qui semble externe.

C. Fabre (Fabre, 1996) a montré que dans le cas de séquences binominales françaises ou anglaises, cette connaissance sémantique était en fait contenue dans la structure des qualia des constituants. Ainsi, le composé *gas pipeline* peut être interprété comme conduct(instrument : *pipeline*, objet : *gas*) grâce au rôle télique de *pipeline*. De la même manière pour les autres rôles de la structure des qualia, *milk disease* s'interprète en cause(objet : *disease*, agent : *milk*) avec le rôle agentif de *disease*, *pea soup* en partie-de(objet : *soup*, agent : *pea*) à l'aide du rôle constitutif de *soup* et *livre de contes* en contient(agent : *livre*, objet : *conte*) grâce au rôle formel de *livre*.

L'interprétation des composés est cruciale dans de très nombreuses applications du TAL comme par exemple la traduction automatique, où de telles séquences supportent rarement une traduction séparée de leurs constituants, ou encore la recherche d'information sur laquelle nous revenons ci-après.

2.3.2.2 La recherche d'information

La recherche d'information est un domaine où l'apport de sémantique constitue clairement un atout. À ce titre, un Lexique génératif peut être exploité pour donner accès à des informations qu'il serait impossible d'obtenir sans la description riche des mots.

Plus précisément, la structure des qualia définit par des formules prédicatives essentiellement verbales les différentes facettes de la sémantique des noms. Certains auteurs ont déjà souligné l'intérêt de tels liens nomino-verbaux pour la RI (Grefenstette, 1997 ; Bouillon *et al.*, 2000b). Les relations qualia sont à ce titre particulièrement indiquées pour permettre l'accès à des variantes des termes d'indexation inhabituelles et peu usitées. Par exemple, le lien télique entre le N *jaugeur* et le V *mesurer* permet d'accéder à des extensions inter-catégorielles des termes du type *jaugeur de carburant* ⇒ *mesurer du carburant*. Par ailleurs, sur le plan théorique, le Lexique génératif est une théorie des mots en contexte. Les informations qu'il contient rendent donc compte des emplois réels des mots dans les documents manipulés par un système de recherche d'information. Nous examinons plus complètement l'intérêt du lien N-V qualia en section 5.2.2, où nous évaluons également la technique d'extension de requêtes à partir de couples qualia.

2.4 Synthèse de notre approche

Pour obtenir le plus automatiquement possible des résultats à la fois de bonne qualité et interprétables, notre système d'acquisition d'éléments lexicaux sémantiques s'appuie principalement sur la PLI. Celle-ci permet grâce à sa souplesse d'utilisation de modéliser relativement aisément notre tâche d'acquisition, mais aussi, à l'aide du *background knowledge*, les connaissances externes sur cette tâche comme les informations morphosyntaxiques et sémantiques issues de l'étiquetage.

Pour tester l'efficacité de notre approche, nous appliquons cette technique à l'acquisition d'éléments du Lexique génératif, les couples qualia. Le système ASARES apprend donc, dans un premier temps, le concept de couples qualia à partir d'exemples de ces couples grâce à la PLI, c'est-à-dire trouve une définition contextuelle en termes d'informations catégorielles et sémantiques. Puis, dans un second temps, ASARES applique cette définition pour retrouver en contexte d'autres couples.

L'utilisation que nous souhaitons faire de ces couples qualia influence la tâche d'acquisition. En effet, il serait possible d'acquérir les éléments de la structure des qualia séparément, c'est-à-dire faire la distinction entre les rôles télique, formel et agentif des verbes. Cela impliquerait de mener un apprentissage multi-classes ; un couple ne serait donc pas seulement qualia ou non qualia, mais télique, agentif, formel ou non qualia. Cependant, cette distinction entre les différents rôles qualia n'est pas pertinente pour l'emploi que nous souhaitons faire des couples acquis. Ces derniers doivent en effet servir pour étendre des requêtes au sein d'un système de recherche d'information ; or dans ce contexte, il n'est pas nécessaire de connaître le rôle exact des verbes utilisés pour ce faire. C'est pourquoi nous nous contentons dans ce mémoire de présenter l'acquisition

de couples qualia et donc de mener un apprentissage du concept de rôle qualia dans sa globalité.

Cet apprentissage, qui constitue le cœur d'ASARES, est présenté en détails dans le chapitre suivant. Nous y montrons comment les différentes propriétés souhaitées pour notre outil que nous avons présentées dans ce chapitre sont formalisées et mises en œuvre dans l'algorithme d'inférence par PLI. Nous y exposons également les résultats obtenus par ASARES dans notre cadre applicatif d'acquisition de relations qualia.

Chapitre 3

Apprentissage de patrons d'extraction par programmation logique inductive

Notre système ASARES repose sur la programmation logique inductive (PLI) pour inférer des patrons d'extraction. Il est cependant nécessaire d'adapter à notre tâche cette technique d'apprentissage. Il faut en particulier qu'elle génère des patrons qui soit linguistiquement pertinents, c'est-à-dire reflétant un phénomène linguistique identifiable par un expert. Pour cela, de nombreuses contraintes sont fixées sur le format des patrons attendus. Par ailleurs, comme beaucoup de techniques d'apprentissage, la PLI a un coût calculatoire élevé qu'il convient de maîtriser. Il faut donc adapter les caractéristiques internes de cette méthode pour qu'elle tire parti au mieux des informations disponibles sur les exemples et que l'inférence de patrons soit réalisée en un minimum de temps.

Ce chapitre décrit la formalisation et la mise en œuvre du module d'inférence qui est au centre d'ASARES et constitue une part essentielle des travaux développés dans ce mémoire. Les principes génériques régissant ASARES sont ici appliqués à notre cadre applicatif d'acquisition de couples qualia, mais les formalismes exposés sont bien sûr valides pour l'acquisition d'autres types d'informations lexicales sémantiques. Nous présentons dans la première section le cadre opérationnel dans lequel doit se dérouler le processus d'induction. Nous y détaillons les étapes de constitution des exemples nécessaires à l'apprentissage et les contraintes qui sont imposées sur la forme attendue des patrons à travers le langage d'hypothèses. Nous étudions ensuite en section 3.2 les problèmes de complexité et de coûts calculatoires. Nous présentons notamment la technique utilisée pour parcourir efficacement l'espace de recherche et les moyens d'éviter l'exploration de portions de cet espace ne pouvant pas conduire à de bonnes solutions. Nous terminons enfin par l'évaluation de notre approche sur la tâche d'acquisition de couples qualia. Nous examinons notamment les performances du système d'un point de vue quantitatif, mais aussi qualitatif en analysant dans une perspective plus linguistique

les couples acquis et les patrons produits.

3.1 Inférence de patrons pertinents

Cette section est dédiée à la description du processus d'apprentissage par PLI et plus particulièrement aux adaptations de l'algorithme d'inférence utilisé, ALEPH (Srinivasan, 2001), nécessaires à la production de clauses pertinentes. Cette pertinence doit se traduire en pratique par trois propriétés des patrons produits :

- la performance pour l'extraction qui assurera la construction d'un lexique contenant l'ensemble des éléments recherchés et sans erreurs ;
- l'interprétabilité et le bien-fondé des schémas morphosyntaxiques et sémantiques mis en valeur par ces patrons ;
- la pertinence du point de vue de l'apprentissage, c'est-à-dire la qualité du processus d'induction.

Nous examinons par ailleurs la gestion du dilemme expressivité/efficacité de l'induction en décrivant les contraintes définissant le langage $\mathscr{L}_H$, l'opérateur de raffinement et les techniques d'élagage de l'espace de recherche $\mathscr{E}_H$.

3.1.1 Constitution des données d'apprentissage

Comme nous l'avons vu, la programmation logique inductive est une technique d'apprentissage supervisée. Cela signifie qu'elle nécessite la constitution d'ensembles d'exemples et de contre-exemples du concept que l'on cherche à inférer. Dans notre cas, nous cherchons à apprendre comment repérer en contexte des occurrences de couples nom-verbe en relation qualia. Il est donc nécessaire de fournir à notre algorithme d'apprentissage certaines occurrences de ces couples, avec une description de leur contexte, et, pour les contre-exemples, des occurrences de couples n'étant pas en relation qualia.

Nous décrivons dans cette section les différentes données nécessaires à l'inférence de patrons d'extraction. La première sous-section présente à ce titre le corpus dont nous nous servons lors de nos expériences et les différentes informations qui doivent être exploitées pour construire ces patrons. La sous-section 3.1.1.2 expose quant à elle le mode de constitution des exemples et contre-exemples qui sont extraits de ce corpus, et la sous-section 3.1.1.3 détaille le contenu du *background knowledge* utilisé en conjonction de ces exemples lors du processus d'inférence.

3.1.1.1 Description du corpus

Le corpus utilisé lors de nos expérimentations est un manuel de maintenance d'hélicoptères, en français, qui nous a été fourni par MATRA-CCR Aérospatiale. Il contient environ 104 000 mots, soit une taille de près de 700 Koctets. Ce corpus technique présente des caractéristiques se prêtant bien à notre tâche d'acquisition : il est très homogène (aussi bien pour ses structures syntaxiques que pour son vocabulaire) ; il

contient également de nombreux termes concrets apparaissant fréquemment au sein des phrases avec des verbes indiquant leur rôle télique ou agentif.

Ce corpus a tout d'abord subi un étiquetage catégoriel automatique. À l'aide des outils développés dans le projet MULTEXT (Armstrong, 1996 ; Ide & Véronis, 1994 ; Petitpierre & Russell, 1994), il a donc été segmenté en phrases et en mots, puis lemmatisé et analysé, et enfin désambiguïsé avec TATOO[1], un outil basé sur des chaînes de Markov cachées. Chaque mot a ainsi reçu une étiquette (ou *tag*) indiquant sa catégorie morphosyntaxique, son genre, son nombre. La précision de cet étiquetage, évaluée à l'aide d'un extrait de 4 000 mots étiquetés à la main est très bonne : moins de 2% d'erreurs ont été détectées.

Le corpus a également subi un étiquetage sémantique. Chaque mot a ainsi reçu une étiquette sémantique le décrivant, avec un taux de réussite très élevé mais néanmoins non totalement parfait. Nous décrivons le processus complet de cet étiquetage en section 4.1.1, en examinant son influence sur la qualité des résultats et les coûts qu'il induit sur l'ensemble du processus d'acquisition.

Il est important de noter que ces informations, aussi bien sémantiques que catégorielles, sont organisées de manière hiérarchique. Chaque mot peut donc être décrit par une série d'étiquettes de la plus spécifique à la plus générale. Ainsi un verbe étiqueté comme un participe passé peut être considéré comme un simple participe ou encore plus simplement comme un verbe. Il en va de même pour l'étiquetage sémantique ; un extrait de la hiérarchie sémantique de la classe des noms est donnée en figure 3.1. D'après cette structure arborescente, un nom ayant reçu l'étiquette d'artefact peut aussi être interprété comme un objet ou encore plus généralement une entité. La hiérarchie complète des informations sémantiques est détaillée en section 4.1.1.2.

Ainsi, la phrase « *L'installation se compose : de deux atterrisseurs* ***protégés*** *par des* ***carénages****, fixés et articulés...* » est transformée en :

11123/1	LSPLIT	L'	BOS	le#det_sg/ddef
11123/3	TOK	installation		installation#noun_sg/pro
11123/16	TOK	se		se#pron/ppers
11123/19	TOK	compose		composer#verb_sg/posv
11123/27	PUNCT	:		:#wpunct)/ponct
11124/1	TOK	de		de#prep/rde
11124/4	TOK	deux		deux#num/quant
11124/9	TOK	atterrisseurs		atterrisseur#noun_pl/art
11124/35	TOK	protégés		protéger#verb_adj/acp
11124/44	TOK	par		par#prep/rman
11124/48	TOK	des		un#det_pl/dindef
11124/52	TOK	carénages		carénage#noun_pl/art
11124/62	PUNCT	,		,#wpunct/virg
11125/1	TOK	fixés		fixer#verb_adj/acp
11125/7	TOK	et		et#conj_coord/rconj
11125/10	TOK	articulés		articuler#verb_adj/acp

[1] Disponible à l'URL http://www.issco.unige.ch/staff/robert/tatoo/tatoo.html.

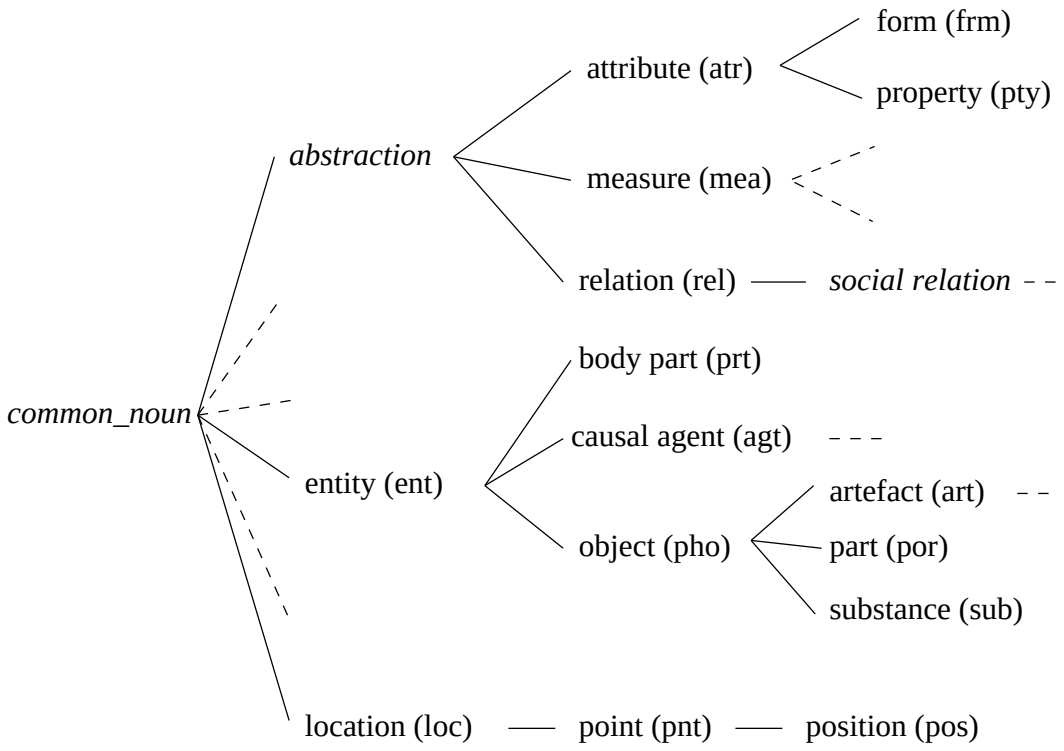


FIG. 3.1 – Extrait de la hiérarchie des classes sémantiques des noms

où la première colonne donne un identifiant unique à chaque mot, et où la dernière colonne précise le lemme, la catégorie et la classe sémantique de chaque mot (carénages est ainsi renseignés comme un nom commun au pluriel (noun_pl) désignant un artefact (art)).

3.1.1.2 Construction des exemples

Notre première tâche consiste à construire les ensembles d'exemples positifs E^+ et négatifs E^- nécessaires à ALEPH pour inférer des clauses généralisées représentant le concept de couple qualia en contexte, c'est-à-dire expliquant ce qui distingue en termes de contexte morphosyntaxique et sémantique les paires N-V qualia des non qualia.

Voici la procédure suivie pour construire ces deux ensembles de supervision. Tout d'abord, on considère chaque nom commun du corpus MATRA-CCR. Plus précisément, on se focalise sur un sous-ensemble du corpus de 81 314 mots formé de toutes les phrases contenant au moins un nom et un verbe. Ce sous-corpus contient 1 489 noms différents (29 633 occurrences) et 567 verbes différents (9 522 occurrences). Pour chaque nom, les 10 verbes les plus fortement associés au sens du Φ^2 sont sélectionnés[2]. Cette

[2]Le Φ^2 est une mesure d'association statistique calculée à partir des fréquences relatives des mots, *cf.* la section 4.2.1.1.

première étape produit des paires dont les composants sont effectivement liés par une relation de type qualia (telles que *roue–gonfler* par exemple) mais également des paires non pertinentes, c'est-à-dire non liées pas une relation qualia (telles que *roue–prescrire*).

Chaque paire ainsi obtenue est donc ensuite manuellement annotée comme pertinente ou non selon les principes définissant la structure des qualia dans la théorie du Lexique génératif. Un programme Perl est ensuite utilisé pour retrouver toutes les occurrences de ces paires dans le corpus.

La totalité des occurrences des couples annotés négatifs (non qualia) par l'expert servent à constituer les exemples négatifs. En revanche, pour les couples annotés globalement positifs (qualia), chacune des occurrences est vérifiée pour s'assurer qu'elle correspond effectivement, dans la phrase considérée, à la relation attendue. Après ce contrôle, un second programme Perl produit un exemple positif en ajoutant à l'ensemble E^+ une clause de la forme : is_qualia(identificateur_du_nom,identificateur_du_verbe). où identificateur_du_nom est une constante définissant de manière univoque l'occurrence du nom considéré (*idem* pour le verbe). Des informations supplémentaires sont également ajoutées au *background knowledge* décrivant chaque mot de la phrase contenant l'occurrence ainsi que la position relative du couple N-V. Par exemple, pour une phrase de 5 mots dont les identificateurs seraient w_1 ... w_5, et le couple N-V w_4-w_2, les clauses suivantes seraient ajoutées :

```
sentence_beginning(w_1).
tags(w_1,tag catégoriel,tag sémantique).
tags(w_2,tag catégoriel,tag sémantique).
pred(w_2,w_1).
tags(w_3,tag catégoriel,tag sémantique).
pred(w_3,w_2).
tags(w_4,tag catégoriel,tag sémantique).
pred(w_4,w_3).
tags(w_5,tag catégoriel,tag sémantique).
pred(w_5,w_4).
sentence_end(w_5).
distances(w_4,w_2,distance en mots,distance en verbes).
```

où pred(x,y) indique que le mot y apparaît juste avant le mot x dans la phrase, le prédicat tags/3 donne les étiquettes catégorielle et sémantique d'un mot, sentence_beginning/1 et sentence_end/1 marquent les mots apparaissant en début ou en fin de la phrase, et distances/4 spécifie le nombre de mots et de verbes entre N et V (une distance négative indique que N apparaît avant V, une valeur positive que V apparaît avant N ; ces valeurs sont décalées d'une unité pour refléter l'ordre entre N et V lorsque les distances sont nulles).

Pendant cette étape, certaines catégories de mots ne sont pas prises en compte. C'est notamment le cas des déterminants et de certains adjectifs, qui ne sont pas considérés pertinents pour caractériser la nature qualia d'une paire N-V au sein d'une phrase ; ces mots sont soulignés dans l'exemple ci-dessous. Par exemple, la phrase contenant le couple N-V qualia en gras « *L'installation se compose : de deux atterrisseurs* ***protégés***

par *des* ***carénages****, fixés et articulés...* » est transformée[3] en

```
is_qualia(m11124_52,m11124_35).
```

et

```
sentence_beginning(m11123_3).
tags(m11123_3, tc_noun_sg, ts_pro).          installation
tags(m11123_16, tc_pron, ts_ppers).          se
pred(m11123_16, m11123_3).
tags(m11123_19, tc_verb_sg, ts_posv).        compose
pred(m11123_19, m11123_16).
tags(m11123_27, tc_wpunct_pf, ts_ponct).     :
pred(m11123_27, m11123_19).
tags(m11124_1, tc_prep, ts_rde).             de
pred(m11124_1, m11123_27).
tags(m11124_4, tc_num, ts_quant).            deux
pred(m11124_4, m11124_1).
tags(m11124_9, tc_noun_pl, ts_art).          atterriseurs
pred(m11124_9, m11124_4).
tags(m11124_35, tc_verb_adj, ts_acp).        protégés
pred(m11124_35, m11124_9).
tags(m11124_44, tc_prep, ts_rman).           par
pred(m11124_44, m11124_35).
tags(m11124_52, tc_noun_pl, ts_art).         carénages
pred(m11124_52, m11124_44).
tags(m11124_62, tc_wpunct, ts_virg).         ,
pred(m11124_62, m11124_52).
tags(m11125_1, tc_verb_adj, ts_acp).         fixés
pred(m11125_1, m11124_62).
tags(m11125_7, tc_conj_coord, ts_rconj).     et
pred(m11125_7, m11125_1).
tags(m11125_10, tc_verb_adj, ts_acp).        articulés
pred(m11125_10, m11125_7).
...                                          ...
distances(m11124_52, m11125_35, 2,1).
```

Les exemples négatifs subissent le même traitement à l'aide du même programme Perl. Ils sont donc automatiquement construits à partir des paires N-V fortement associées au sens du Φ^2 manuellement annotées comme non pertinentes, auxquelles s'ajoutent les paires rejetées lors de la construction de E^+. Par exemple, la paire non qualia en gras dans la phrase suivante : « *Au montage : gonfler la* ***roue*** *à la pression* ***prescrite****, ...* » est ajoutée à l'ensemble E^- en tant que is_qualia(m7978_15,m7978_31). et les informations suivantes sont ajoutées au *background knowledge* :

[3] Pour faciliter la compréhension, nous avons noté à droite les mots de la phrase décrits par les prédicats tags/3.

```
sentence_beginning(m7977_1).
tags(m7977_1,tc_prep,ts_ra).                    Au
tags(m7977_3,tc_noun_sg,ts_acy).                montage
pred(m7977_3,m7977_1).
tags(m7977_11,tc_wpunct_pf,ts_ponct).           :
pred(m7977_11,m7977_3).
tags(m7978_7,tc_verb_inf,ts_acp).               gonfler
pred(m7978_7,m7977_11).
tags(m7978_15,tc_noun_sg,ts_ins).               roue
pred(m7978_15,m7978_7).
tags(m7978_20,tc_prep,ts_ra).                   à
pred(m7978_20,m7978_9).
tags(m7978_22,tc_noun_sg,ts_phm).               pression
pred(m7978_22,m7978_20).
tags(m7978_31,tc_verb_adj,ts_acc).              prescrite
pred(m7978_31,m7978_22).
tags(m7978_41,tc_wpunc,ts_virg).                ,
pred(m7978_41,m7978_31).
...                                             ...
distances(m7978_15,m7978_31,-3,-1).
```

Ce sont ainsi 3 099 exemples positifs et 3 176 négatifs qui sont produits automatiquement selon ce processus à partir du corpus MATRA-CCR.

3.1.1.3 Connaissances préalables

Des clauses décrivant les relations hiérarchiques entre les étiquettes sémantiques ou catégorielles sont aussi ajoutées au *background knowledge* d'ALEPH. Ces relations indiquent par exemple que l'étiquette tc_verb_pl correspond à un verbe conjugué au pluriel (conjugated_plural), qui peut être considéré comme un verbe conjugué (conjugated) ou simplement un verbe (verb). Ces prédicats décrivent en quelque sorte la nature linguistique des mots en s'appuyant sur les informations fournies par leurs étiquettes catégorielles et sémantiques. Voici un exemple de ces prédicats et des relations qu'ils entretiennent :

```
verb( W ) :- conjugated( W ).
verb( W ) :- infinitive( W ).
...
conjugated( W ) :- conjugated_plural( W ).
conjugated( W ) :- conjugated_singular( W ).
conjugated_plural( W ) :- tags( W, tc_verb_pl, _ ).
...
```

Le *background knowledge* décrivant l'ensemble de ces relations et toutes les définitions de prédicats est donné dans l'annexe B.1.

Nous définissons ci-après certains termes utilisés dans la suite dans ce chapitre :

- Nous appelons par la suite « littéraux les plus généraux » les littéraux décrivant les mots qui n'apparaissent pas dans le corps d'une clause du *background knowledge* (par exemple, common_noun/1, verb/1). Tous les mots peuvent être décrits par au moins un littéral plus général.
- Nous appelons « littéraux catégoriels » (resp. « littéraux sémantiques ») les littéraux décrivant la nature morphosyntaxique (sémantique) d'un mot.
- Pour deux littéraux l_1 et l_2 tels qu'il existe une clause l_1 :- l_2. dans le *background knowledge*, l_1 est appelé la « généralisation immédiate » de l_2 et l_2 est la « spécialisation immédiate » de l_1. La généralisation immédiate de tout littéral est unique.
- Le littéral l_1 est dit plus général que le littéral l_2 s'il existe une suite de spécialisations immédiates de l_1 amenant à l_2 ; l_2 est alors dit plus spécifique que l_1.

Ces prédicats de « description des mots », tous unaires, sont définis de telle manière que tous ceux décrivant un mot donné peuvent être organisés sous forme d'une forêt, c'est-à-dire sous forme de plusieurs arbres reflétant les relations hiérarchiques des différentes sources d'informations. Par exemple, dans la figure 3.2, un nom commun désignant un artefact et apparaissant en début de phrase peut-être décrit par plusieurs prédicats portant soit sur sa position dans la phrase (arbre 1) soit sur ses étiquettes catégorielles et sémantiques (arbre 2) et éventuellement par d'autres prédicats. L'arbre 2 de cet exemple retranscrit notamment la structure hiérarchique de nos informations apportées par les étiquetages.

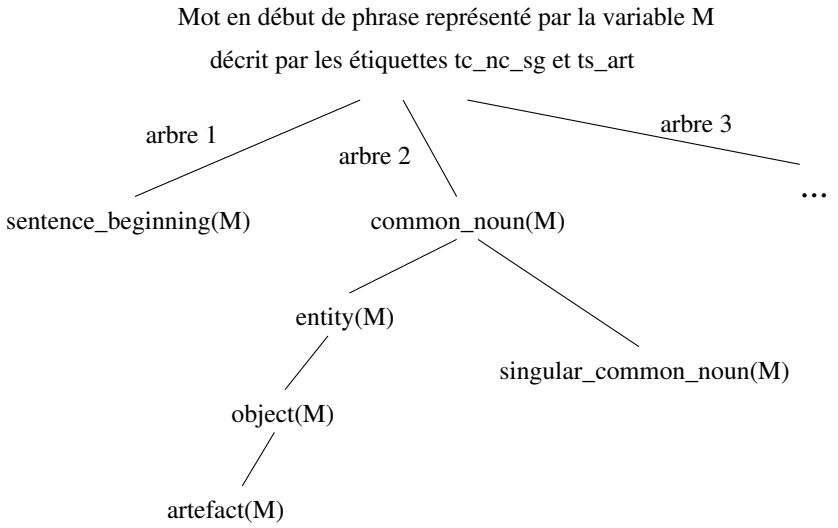


FIG. 3.2 – Arbres de prédicats s'appliquant à une variable

Cette structure particulière de nos informations est exploitée dans le processus même

d'inférence pour améliorer l'efficacité de la recherche dans l'espace $\mathscr{E}_H$ et l'interprétabilité des règles produites.

D'autres prédicats utiles sont également stockés dans le *background knowledge*. Par exemple, le prédicat suc/2 est défini par suc(X,Y) :- pred(Y,X).; suc/2 est uniquement utilisé pour une lecture des clauses plus aisée mais est considéré, spécialement lors de la recherche de la meilleure hypothèse (voir section suivante), comme l'équivalent de pred/2. Ainsi, les deux clauses is_qualia(A,B) :- suc(A,B). et is_qualia(A,B) :- pred(B,A). sont considérées comme une unique hypothèse.

3.1.2 Espace de recherche de patrons pertinents

Beaucoup de tâches d'apprentissage artificiel peuvent être considérées comme de simples problèmes de recherche dans un espace. Dans le cadre de la PLI, le programme logique que l'on cherche à apprendre doit vérifier les conditions données en section 2.2.3.1. En particulier, chaque clause inférée doit répondre à un critère de qualité Sc, calculé en fonction des exemples positifs et négatifs qu'elle couvre.

Dans notre cas, H, l'ensemble résultant de l'apprentissage, peut être composé de plusieurs clauses représentant chacune un patron morphosyntaxique et sémantique portant une relation nom-verbe qualia. Chaque élément de cet ensemble de clauses H est cherché *a priori* à travers l'espace des clauses de Horn selon l'algorithme 1.

Algorithme 1 Algorithme d'ALEPH

Itération jusqu'à $E^+ = \emptyset$

1. choisir aléatoirement un exemple positif e^+ dans E^+ ;
2. construire la clause la plus spécifique $\perp$ couvrant e^+ par implication inverse ;
3. parcourir l'espace de recherche $\mathscr{E}_H$ basé sur $\perp$ à la recherche de la clause h maximisant une fonction de score Sc ;
4. ajouter h à l'ensemble H et ôter de E^+ les exemples couverts par h.

Fin itération

L'espace de recherche $\mathscr{E}_H$ est généralement très grand voire même infini. Une recherche exhaustive des hypothèses possibles, et leur test de couverture, à travers tout l'espace est donc exclue. Des biais sont donc généralement utilisés à la fois pour réduire cet espace de recherche mais également pour assurer un bon processus d'induction (voir section 2.2.2.3). Comme nous l'avons vu précédemment, parmi les différents types de biais existants, l'un des plus naturels et des plus utilisés est le langage d'hypothèses qui permet de définir des contraintes syntaxiques sur les clause recherchées (Nédellec *et al.*, 1996). Nous présentons ci-dessous ceux que nous imposons à notre tâche d'apprentissage. L'espace de recherche défini par ce langage d'hypothèses peut être organisé grâce un ordre de généralité, et ainsi permettre une recherche des hypothèses plus efficace. Nous détaillons en section 3.1.2.2 cet ordre de généralité et la structure qu'il induit sur

notre espace de recherche.

3.1.2.1 Langage d'hypothèses

Le biais de langage imposé lors de la phase d'apprentissage limite considérablement le nombre de solutions potentielles, prévient les phénomènes de sur-généralisation (*overfitting*) et assure que seules des hypothèses bien formées (au regard de la tâche d'acquisition visée) seront générées.

Dans notre cadre d'apprentissage de patrons d'extraction de couples qualia, une hypothèse bien formée est définie comme étant une clause donnant des informations (sémantiques ou catégorielles) sur les mots (N,V ou n'importe quel autre mot apparaissant dans son contexte) ou des informations sur les positions respectives de N ou de V dans la phrase. Par exemple, is_qualia(A,B) :- artefact(A), pred(B,C), suc(A,C), auxiliary(C). – qui signifie qu'un couple N-V (ici représenté par les variables A et B) est qualia si N est un artefact, V est précédé d'un verbe auxiliaire et N est suivi par ce verbe – est une hypothèse bien formée. Il nous faut donc indiquer dans les paramétrages d'ALEPH que les prédicats artefact/1, pred/2, suc/2, auxiliary/1... peuvent être utilisés pour construire une hypothèse.

Critère de concision

Une contrainte importante que nous imposons sur le langage d'hypothèses est qu'il ne peut pas y avoir deux informations (prédicats unaires) sur un mot telles que l'une des deux soit plus générale que l'autre. Cela signifie qu'une hypothèse telle que is_qualia(A,B) :- pred(B,C), participle(C), past_participle(C). n'est pas considérée comme étant bien formée puisque deux informations catégorielles sont données sur le mot C. De telles informations redondantes sur un mot sont superflues du fait de l'organisation hiérarchique des prédicats : l'un des littéraux est en effet par définition plus spécifique que les autres et donc suffisant pour décrire le mot de manière précise; les autres littéraux sont donc sans intérêt. Il est en effet inutile dans notre exemple d'indiquer que C est un participe (participle(C)) si l'on sait par ailleurs qu'il s'agit d'un participe passé (past_participle(C)). À l'inverse, les hypothèses is_qualia(A,B) :- pred(B,C), participle(C), action_verb(C). ou is_qualia(A,B) :- pred(B,C), past_participle(C), physical_action_verb(C). ou encore is_qualia(A,B) :- pred(B,C), suc(A,C). sont considérées comme bien formées au regard de notre tâche. Cette condition sur les littéraux s'appliquant aux mots est appelée par la suite critère de concision.

Critère d'utilisation des variables

De plus, nous voulons éviter d'obtenir des clauses contenant des variables non contraintes. Par exemple, l'hypothèse is_qualia(A,B) :- infinitive(B), pred(A,C). peut être plus simplement exprimée par is_qualia(A,B) :- infinitive(B). puisque pred(A,C) n'apporte aucune information linguistique jugée intéressante. Cependant, is_qualia(A,B) :- suc(A,C),

suc(C,D), object(D). est considérée comme bien formée puisqu'il y a une contrainte sémantique sur le mot D, et C est contraint par les deux suc/2. Cette condition que nous imposons est très similaire à la contrainte bien connue de liaison (*linkedness*) : selon (Helft, 1987), une clause est dite liée (*linked*) si toutes ses variables sont liées ; une variable V est liée dans une clause C si et seulement si V apparaît dans la tête de C, ou s'il existe un littéral l de C contenant V et une autre variable W ($V \neq W$) telle que W soit liée dans C. Cela correspond également à la contrainte de connexion (Quinlan, 1990), aux clauses *i1*-déterminées dans le cadre de l'*ij*-détermination (Muggleton & Feng, 1990) ou au concept de clause-chaîne (Rieger, 1996) mais, dans notre cas, toutes les variables doivent non seulement être connectées aux variables de tête (avec les prédicats pred/2 et suc/2), mais doivent aussi être *utilisées* autre part dans le corps de la clause. C'est cette contrainte qui est appelée par la suite critère d'utilisation des variables.

Une hypothèse réunissant ces deux conditions, concision et utilisation des variables, est dite *bien formée* au regard de notre tâche. Sauf mention contraire, les clauses que nous manipulons dans la suite de ce chapitre sont supposées être bien formées.

3.1.2.2 Espace des hypothèses et ordre de généralité

Malgré ce biais de langage, notre espace d'hypothèses demeure immense. Heureusement, les hypothèses qui le composent peuvent être organisées par une relation de généralité (à l'aide d'une relation de quasi-ordre entre les hypothèses) qui permet de parcourir intelligemment cet espace de solutions. Plusieurs quasi-ordres ont été étudiés dans le cadre de la PLI ; nous en avons présenté certains au chapitre précédent. Parmi ceux-ci, la θ-subsomption (voir section 2.2.2.2) est certainement le plus utilisé dans les systèmes de PLI.

Subsomption sous identité objet

La θ-subsomption demeure cependant trop forte, c'est-à-dire trop générale, pour notre application. En effet, considérons $h_1 \equiv$ is_qualia(X_1,Z_1) :- suc(X_1,Y_1), pred(Z_1,W_1), verb(Y_1), verb(W_1). et $h_2 \equiv$ is_qualia(X_2,Z_2) :- suc(X_2,Y_2), pred(Z_2,Y_2), verb(Y_2).. Nous avons alors $h_1 \succeq_\theta h_2$ avec $\theta = [\ X_1/X_2, Y_1/Y_2, Z_1/Z_2, W_1/Y_2\ ]$ et, puisque dans notre application les variables représentent des mots, ceci signifie que la θ-subsomption permet de considérer un même mot comme deux différents au sein d'une clause, comme c'est le cas avec le mot Y_1/W_1 dans h_1. Cela est donc équivalent à donner un patron d'extraction pour le couple X1-Z1 de la forme X1 + verbe + (tout mot)* + verbe + Z1 où le verbe suivant X1 peut désigner également le verbe précédant Z1. De tels phénomènes rendent les patrons difficilement interprétables et empêchent une bonne compréhension des manifestations linguistiques qu'ils devraient traduire. Cette propriété n'est donc pas jugée pertinente pour notre tâche d'apprentissage ; nous nous intéressons de ce fait à une forme plus coercitive de la θ-subsomption : la *θ-subsomption sous identité objet* (θ_{OI}-subsomption par la suite) (Esposito *et al.*, 1996) définie ci-dessous.

Définition 5 (d'après (Badea & Stanciu, 1999)) *Une clause C_1 θ_{OI}-subsume une clause C_2 ($C_1 \succeq_{OI} C_2$) ssi il existe une substitution θ telle que $C_1\theta \subseteq C_2$ et que θ soit injective (c'est-à-dire que θ n'unifie pas des variables de C_1).*

La θ_{OI}-subsomption est clairement plus faible que la θ-subsomption ($C_1 \succeq_{OI} C2 \Rightarrow C_1 \succeq_\theta C_2$ mais l'inverse n'est pas toujours vrai) mais préserve la propriété attendue que $h_1 \not\succeq_{OI} h_2$ avec h_1 et h_2 définies ci-dessus. Cette contrainte supplémentaire est gérée par ALEPH par la génération d'hypothèses contenant un ensemble d'inégalités imposant que deux variables d'une clause ayant des noms différents ne puissent être unifiées. Par exemple, h_1 est représentée au sein du système de généralisation par is_qualia(X,Z) :- suc(X,Y), pred(Z,W), verb(Y), verb(W), X≠Z, X≠Y, Z≠Y, X≠W, Y≠W, Z≠W. Pour des raisons de lisibilité, dans la suite de ce chapitre et dans les chapitres suivants nous n'écrivons pas ces ensembles d'inégalités et nous supposons donc de manière implicite que deux variables nommées différemment représentent des mots distincts. Par ailleurs nous supposons dans le reste de ce chapitre, comme c'est le cas en Prolog, que deux clauses distinctes possèdent des variables distinctes.

Cette restriction de la θ-subsomption a quelques propriétés intéressantes, présentées ci-dessous, que nous exploitons lors de l'exploration de notre espace d'hypothèses. La propriété suivante montre par exemple que l'application des substitutions telles que celles ayant cours au sein dans un espace ordonné par θ_{OI}-subsomption n'unifie pas de littéraux d'une même clause.

Propriété 2 *Soit C une clause et θ une substitution injective remplaçant toutes les variables de C par de nouvelles variables (c'est-à-dire n'apparaissant pas déjà dans C), alors $C\theta$ contient autant de littéraux que C ; on a donc $|C\theta| = |C|$.* □

Preuve

Supposons qu'il existe une substitution injective remplaçant les variables de C par de nouvelles variables et deux littéraux l_1 et l_2 de C tels que $l_1 \neq l_2$ mais $l_1\theta = l_2\theta$. Puisque l_1 et l_2 ne diffèrent que par leurs variables, il existe donc au moins une variable X dans l_1 et Y dans l_2 telles que $X \neq Y$. Or $l_1\theta = l_2\theta$, et comme θ s'applique à toutes les variables de C, cela signifie que θ a unifié X et Y sous une même variable. Cela contredit le fait que θ soit injective. □

Nous démontrons deux corollaires de cette propriété ci-dessous. Ils traduisent le fait qu'en pratique cet ordre de généralité repose uniquement sur l'ajout ou la suppression de littéraux, c'est-à-dire qu'une hypothèse peut être construite en utilisant des littéraux de l'un de ses subsumants stricts (à un renommage de variables près) une seule fois chacun.

Corollaire 1 *Soient C et D deux clauses telles que $C \succ_{OI} D$, alors il existe une substitution injective telle que $C\theta \subset D$, et donc $|C| < |D|$.* □

Preuve

Soient C et D telles que $C \succ_{OI} D$, alors il existe une substitution injective θ_1 telle que $C\theta_1 \subseteq D$, et il n'existe aucune substitution injective θ_2 telle que $D\theta_2 \subseteq C$. Supposons que l'on ait $C\theta_1 = D$. Puisque les clauses C et D ne partagent aucune variable commune, θ_1 s'applique à toutes les variables de C et les transforme de manière unique (injectivité) en variables de D. Il est donc possible de construire θ_1^{-1} qui soit également injective et telle que $C = D\theta_1^{-1}$. Cela contredit le fait que $C \succ_{OI} D$. □

Corollaire 2 *Soient C et D deux clauses telles que $C \succeq_{OI} D$, alors C peut être construite en utilisant des littéraux de D (à un renommage de variables près) une seule fois chacun.* □

Preuve

Soient C et D telles que $C \succeq_{OI} D$, alors il existe une substitution injective θ telle que $C\theta \subseteq D$. Supposons qu'il existe dans C deux littéraux (différents) l_1 et l_2 issus du même littéral de D, c'est-à-dire tels que $l_1\theta = l_2\theta$. Les clauses C et D sont supposées contenir des variables distinctes; or d'après la propriété 2, une substitution injective n'unifie pas de littéraux, soit encore $|C\theta| = |C|$, ce qui rend impossible le fait que $l_1\theta = l_2\theta$. □

Subsomption généralisée

La notion de généralité que nous utilisons (que nous appelons θ_{NV}-subsomption) est dérivée de la θ_{OI}-subsomption et adaptée aux besoins de notre application. La θ_{OI}-subsomption, telle que définie ci-dessus, ne capture pas exactement la notion de généralité que nous voulons utiliser pour notre espace d'hypothèses. Nous voudrions en effet prendre en compte l'organisation hiérarchique des informations disponibles sur les mots (telles que les informations sémantiques et catégorielles). Nous souhaitons donc que notre notion de généralité exploite au mieux la théorie du domaine décrite dans le *background knowledge*, suivant les idées développées dans le cadre de la *subsomption généralisée* (Buntine, 1988). Par exemple, nous souhaitons que l'hypothèse is_qualia(A,B) :- entity(A). soit considérée comme plus générale que is_qualia(A,B) :- object(A). qui elle même doit être considérée comme plus générale que is_qualia(A,B) :- artefact(A). (voir section 3.1.1.1).

Ainsi, nous définissons ci-dessous la relation de généralité adaptée à nos besoins.

Définition 6 *Une clause C_1 θ_{NV}-subsume une clause C_2 ($C_1 \succeq_{NV} C_2$) relativement au background knowledge B ssi il existe une substitution ssi il existe une substitution injective θ et une fonction f_D telle que $f_D(C)\theta \subseteq D$, où f_D est telle que $\forall l \in C,\ \ B, f_D(l) \models l$, avec $f_D(\{l_1, l_2, ..., l_m\})$ signifiant $\{f_D(l_1), f_D(l_2), ..., f_D(l_m)\}$.*

Intuitivement, cela signifie qu'une clause D peut être plus spécifique que C si :

1 – D contient des littéraux en plus de ceux de C ;

2 – D contient des littéraux plus spécifiques (en fonction des hiérarchies d'informations décrivant les mots) sur les mêmes variables que C.

Tout comme la θ-subsomption et la θ_{OI}-subsomption, la θ_{NV}-subsomption induit une relation de quasi-ordre sur l'espace des hypothèses bien formées telles que définies ci-dessus et de notre *background knowledge*, comme l'attestent les deux résultats suivants :

- $C \succeq_{NV} C$ (réflexivité)
- $C_1 \succeq_{NV} C_2$ et $C_2 \succeq_{NV} C_3 \Rightarrow C_1 \succeq_{NV} C_3$ (transitivité)

Ce quasi-ordre peut être classiquement étendu à un ordre partiel grâce à l'antisymétrie sur les classes d'équivalence des clauses sous $\succeq_{NV}$:

- $C_1 \succeq_{NV} C_2$ et $C_2 \succeq_{NV} C_1 \Rightarrow C_1$ et C_2 sont équivalentes (ce que l'on note $C_1 \sim_{NV} C_2$) ; dans notre cas (de même que pour la θ_{OI}-subsomption), $C_1 \sim_{NV} C_2$ signifie $C_1 = C_2$ au renommage des variables près.

Preuve

1 – Réflexivité : trivial.

2 – Antisymétrie : $C_1 \succeq_{NV} C_2$ et $C_2 \succeq_{NV} C_1$, donc il existe f_1, f_2, θ_1 et θ_2 telles que $f_1(C_1)\theta_1 \subseteq C_2$ et $f_2(C_2)\theta_2 \subseteq C_1$, avec $\forall l \in C_1, B, f_1(l) \models l$ et $\forall l \in C_2, B, f_2(l) \models l$. Ainsi, $\forall l \in C_1, B, f_2(f_1(l)) \models f_1(l)$ et donc $\forall l \in C_1, B, f_2(f_1(l)) \models l$ avec $f_2(f_1(l)) \in C_1$. D'après notre *background knowledge* et puisque C_1 est considérée bien formée, nous avons $\forall l \in C_1, f_2(f_1(l)) = l$ et $f_1(l) = l$; de la même manière, $\forall l \in C_2, f_2(l) = l$. Cela signifie que $C_1\theta_1 \subseteq C_2$ et $C_2\theta_2 \subseteq C_1$ et puisque θ_1 et θ_2 sont injectives, C_1 et C_2 sont simplement des variantes alphabétiques.

3 – Transitivité : $C_1 \succeq_{NV} C_2$ et $C_2 \succeq_{NV} C_3$, il existe donc f_1, f_2, θ_1 et θ_2 telles que $f_1(C_1)\theta_1 \subseteq C_2$ et $f_2(C_2)\theta_2 \subseteq C_3$. Nous avons $f_2(f_1(C_1))\theta_1\theta_2 \subseteq C_3$, et $f_2 \circ f_1$ (composition de f_2 et f_1) et $\theta_1\theta_2$ sont injectives, donc $C_1 \succeq_{NV} C_3$.

□

Grâce à la représentation de nos exemples et au *background knowledge* utilisé, tous les littéraux pouvant apparaître dans les hypothèses sont déterministes ; de telles hypothèses sont dites *clauses déterministes*. Avec ces clauses liées et déterministes, le quasi-ordre de la θ_{NV}-subsomption implique que l'espace des hypothèses est structuré en treillis (la démonstration en est donnée en appendice B.2 pour la θ_{OI}-subsomption et la θ_{NV}-subsomption). Au sommet de ce treillis se trouve la clause la plus générale ($\top$) et au bas la clause la plus spécifique (appelée MSC pour *most specific clause* ou *bottom* et notée $\bot$ par la suite). Dans notre cas, $\top$ est la clause is_qualia(A,B). indiquant que toutes les paires N-V sont qualia, et $\bot$ est une clause sans constante contenant tous les littéraux pouvant être utilisés pour décrire l'exemple en cours de généralisation (voir (Muggleton, 1995) pour les détails de la construction de $\bot$) à l'exception des littéraux superflus (littéraux donnant des informations plus générales que d'autres littéraux déjà présents dans $\bot$).

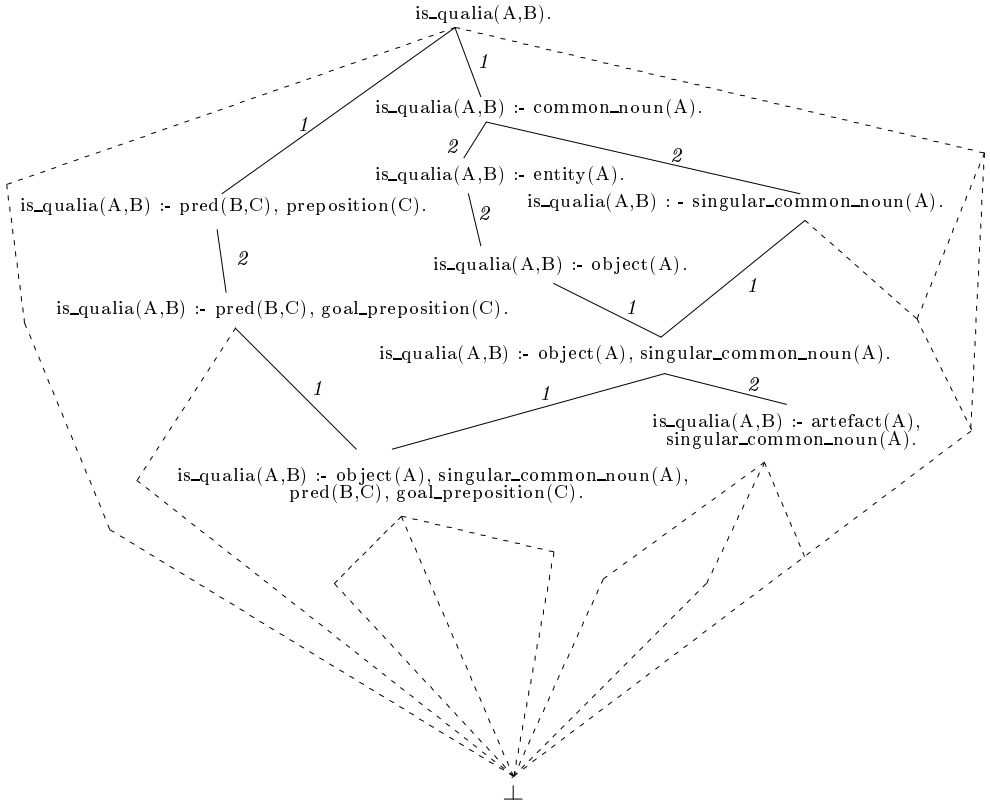


FIG. 3.3 – Treillis d'hypothèses sous la θ_{NV}-subsomption

La figure 3.3 montre un exemple simple d'un treillis ordonnant des clauses bien formées sous θ_{NV}-subsomption. Les numéros présents sur les arcs indiquent à laquelle des deux interprétations intuitives données en page 99 se rapporte la notion de généralité entre les deux clauses reliées par cet arc.

3.2 Exploration de l'espace des hypothèses

La recherche d'hypothèses au sein de l'espace présenté ci-dessus est une tâche très coûteuse. En effet, pour trouver la meilleure hypothèse de cet espace, il est nécessaire de tester chaque hypothèse rencontrée par rapport aux exemples positifs et négatifs. C'est ce test qui est le point le plus coûteux du processus d'inférence. Il est donc nécessaire de parcourir intelligemment cet espace, en tirant notamment parti de sa structure, et de ne pas poursuivre l'exploration de partions de cet espace que l'on sait ne pas contenir de solutions intéressantes.

Le premier point, le parcours de l'espace de recherche, est réalisé par un opérateur de raffinement que nous présentons dans la section suivante. Le second point relève quant à lui de l'élagage ; nous décrivons l'approche que nous utilisons en section 3.2.2.

3.2.1 Opérateur de raffinement

La façon dont le treillis est parcouru est extrêmement importante pour trouver la meilleure hypothèse (relativement à une fonction de score choisie) en un minimum de temps. Notre *background knowledge* contenant des informations structurées en forêt (un ensemble d'arbres) et les prédicats relationnels introduisant de nouvelles variables dans une clause (pred/2 et suc/2) étant déterministes, nous montrons ci-dessous que l'on peut construire un opérateur de raffinement *parfait* (Badea & Stanciu, 1999) permettant un parcours efficace de cet espace d'hypothèses.

3.2.1.1 Propriétés des opérateurs de raffinement

Un opérateur de raffinement définit la façon dont l'espace des solutions est exploré. Cet espace étant généralement gigantesque, l'opérateur doit réaliser son parcours le plus efficacement possible, mais sans *oublier* de solutions potentielles. Cette dernière propriété de complétude est certainement l'une des plus importantes des opérateurs de raffinement.

Un opérateur de raffinement peut être formellement défini comme une fonction de $\mathscr{L}_H$, le langage d'hypothèses, vers $\mathscr{P}(\mathscr{L}_H)$, l'ensemble des parties de $\mathscr{L}_H$. On peut schématiquement regrouper les opérateurs en deux familles : les opérateurs ascendants, qui produisent des hypothèses plus générales qu'une hypothèse donnée, et les opérateurs descendants qui produisent des spécialisations d'une hypothèse. L'opérateur que nous avons défini, appelé ρ_{nv}, que nous présentons en section 3.2.1.3, est un opérateur descendant, c'est-à-dire que pour toute clause C, $\rho_{nv}(C) \subseteq \{D \mid C \succeq_{NV} D\}$.

Nous définissons quelques notations qui sont utilisées par la suite. Si $\succeq$ est un quasi-ordre (par exemple $\succeq_\theta$, $\succeq_{OI}$ ou $\succeq_{NV}$), et si $C \succeq D$, C est dite *plus générale ou équivalente à* D. D est donc un *raffinement descendant* de C et C est un *raffinement ascendant* de D. On note $C \succ D$ lorsque $C \succeq D$ mais que $D \not\succeq C$; cela signifie que C est *strictement plus générale* que D (D est *strictement plus spécifique* que C) et D est un *raffinement descendant strict* de C (C est un *raffinement ascendant strict* de D). D est appelée *subsumé immédiat* (*downward cover*) de C si et seulement si $C \succ D$ et il n'existe pas de E satisfaisant $C \succ E \succ D$.

Certaines propriétés importantes des opérateurs de raffinement assurent d'obtenir les meilleures hypothèses acceptant (couvrant) les exemples positifs et rejetant les négatifs. On définit plusieurs d'entre elles comme suit (Badea & Stanciu, 1999) :

- ρ est un opérateur de raffinement *(localement) fini* ssi $\rho(C)$ est fini et calculable pour toute clause C.
- ρ est opérateur de raffinement *strict* pour tout C ssi $\rho(C)$ ne contient pas de D telle que $D \sim C$.
- ρ est un opérateur de raffinement *faiblement complet* ssi, en notant $\rho^* = \rho \circ \rho \circ ... \circ \rho$, $\rho^*(\top)$ = l'ensemble des clauses de l'espace.
- ρ est un opérateur de raffinement *complet* ssi pour toutes C et D, $C \succ D \Rightarrow \exists E \in \rho^*(C)$ telle que $E \sim D$.

- ρ est un opérateur de raffinement *non redondant* ssi pour toutes C_1, C_2 et D, $D \in \rho^*(C_1) \cap \rho^*(C_2) \Rightarrow C_1 \in \rho^*(C_2)$ ou $C_2 \in \rho^*(C_1)$.
- ρ est un opérateur de raffinement *idéal* ssi il est fini, strict et complet.
- ρ est un opérateur de raffinement *optimal* ssi il est fini, non redondant et faiblement complet.
- ρ est un opérateur de raffinement *minimal* ssi pour toute C, $\rho(C)$ contient seulement des subsumés immédiats de C et tous ses éléments sont incomparables (D_1, $D_2 \in \rho(C) \Rightarrow D_1 \not\succeq D_2$ et $D_2 \not\succeq D_1$).
- ρ est un opérateur de raffinement *parfait* ssi il est minimal et optimal.

Puisque généralement la recherche est réalisée dans un ensemble ordonné ou un treillis, une hypothèse peut *a priori* être atteinte par plusieurs chemins. Cela implique qu'une telle hypothèse sera évaluée plusieurs fois. Pour des raisons d'efficacité, les opérateurs de raffinement optimaux sont donc souvent préférés aux idéaux (un opérateur ne pouvant être à la fois complet et non redondant, il ne peut donc pas être parfait et idéal). De tels opérateurs n'existent pas forcément pour les espaces ordonnés par θ-subsomption (van der Laag & Nienhuys-Cheng, 1994a) ou l'implication logique (van der Laag & Nienhuys-Cheng, 1994b), mais existent dans le cadre de la θ_{OI}-subsomption (Esposito *et al.*, 1996 ; Semeraro *et al.*, 1997).

3.2.1.2 Exploration du treillis sous θ_{OI}-subsomption

L'exploration de notre treillis d'hypothèses est un point crucial pour l'efficacité de notre processus d'inférence de patrons. Nous avons développé pour ce faire un opérateur de raffinement parfait. Cet opérateur s'appuie principalement sur la structure particulière de l'espace de recherche ordonné par θ_{OI}-subsomption ; comme nous le voyons en sous-section 3.2.1.3, la subsomption généralisée est prise en compte en pratique comme une contrainte sur cet espace particulier. Nous considérons donc dans un premier temps le treillis des hypothèses ordonné par la θ_{OI}-subsomption sans la contrainte d'être bien formées pour notre tâche, c'est-à-dire sans besoin de subsomption généralisée. Par souci de lisibilité, nous notons dans cette section $\succeq$, $\succ$ et $\sim$ les relations d'ordre induites par la θ_{OI}-subsomption. Nous définissons ci-dessous un opérateur permettant le parcours de ce treillis, puis nous le modifions pas à pas pour prendre en compte divers critères, jusqu'à obtenir un opérateur parfait adapté à notre espace de clauses bien formées ordonné par θ_{NV}-subsomption.

Opérateur de raffinement optimal

Dans le cadre de notre application, chaque variable représente un mot d'une phrase. Or, le nombre de mots contenus dans une phrase et le nombre de prédicats pouvant s'appliquer à ces mots sont finis. Il s'en suit que le nombre de littéraux contenus dans toute clause du treillis ordonné par la θ_{OI}-subsomption est fini et est un sous-ensemble des littéraux de $\perp$ à un renommage des variables près. Cela est traduit dans le corollaire 2 présenté précédemment.

On peut construire un opérateur de raffinement exploitant cette propriété pour parcourir l'espace de recherche. Pour comprendre le principe de cet opérateur, observons d'abord la structure du treillis des hypothèses ordonné par la subsomption sous identité objet. Ce dernier a une structure particulière permettant une exploration aisée. Plus précisément, nous établissons dans la propriété suivante que ce treillis des hypothèses est isomorphe à une algèbre de Boole et donc équivalent à un treillis des parties d'un ensemble.

Propriété 3 *Le treillis des hypothèses liées et déterministes (sans la contrainte d'être bien formées pour notre tâche) ordonnées par la θ_{OI}-subsomption est isomorphe à un treillis de Boole.* □

Preuve

D'après le corollaire 2, toute hypothèse du treillis peut être construite (à un renommage des variables près) à partir d'un sous-ensemble des littéraux d'une clause qu'elle subsume sous identité objet. Or par définition, toutes les clauses subsument $\perp$, qui contient donc le nombre maximum de littéraux pouvant apparaître dans une hypothèse ; on note ce nombre n.

Il est possible de construire une fonction f qui soit un isomorphisme de l'ensemble de littéraux de $\perp$ dans $\{1, ..., n\}$, c'est-à-dire qui numérote chacun des littéraux. D'après le théorème de Stone, notre treillis d'hypothèses est donc isomorphe au treillis de l'algèbre de Boole $(\{1, ..., n\}, \subseteq)$. □

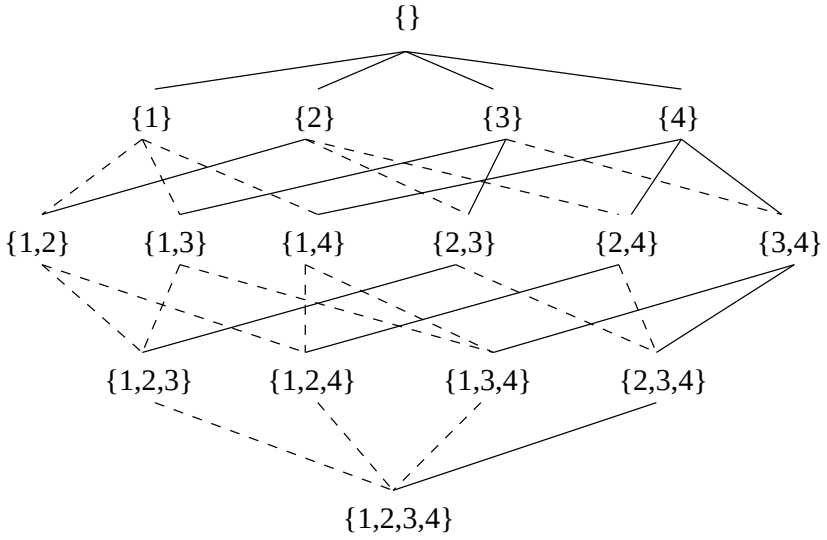


FIG. 3.4 – Treillis de l'algèbre de Boole ({1–4},⊆)

On peut aisément construire un opérateur de raffinement parcourant ce type de treillis de haut en bas. Il suffit pour cela de considérer un opérateur ρ ajoutant un

singleton à un ensemble tel que ce singleton soit un numéro plus petit que le minimum des numéros de l'ensemble à spécialiser. Un tel parcours est illustré en figure 3.4 sur un treillis des parties de {1–4} (*i.e.* {1, ..., 4}) muni de l'inclusion ensembliste; les arcs (pleins ou en pointillés) représentent la relation de généralité entre les éléments du treillis et les arcs pleins sont ceux effectivement exploités pour explorer le treillis.

On peut donc explorer de la même manière le treillis des hypothèses liées et déterministes muni de la θ_{OI}-subsomption puisqu'il est isomorphe au treillis ({1–n},$\subseteq$) avec n le nombre de littéraux de $\perp$. Dans notre cas, un opérateur de raffinement s'appuyant sur cette même technique consiste alors à choisir un littéral dans $\perp$ dont le numéro par un isomorphisme f est plus petit que le minimum des littéraux de la clause à raffiner. Par souci de lisibilité, on note $\min(C)$ pour $\min_{l_i \in C\theta_\perp}(f(l_i))$ avec $\theta_\perp$ la substitution injective telle que $C\theta_\perp \subseteq \perp$. Plus formellement, en notant θ_r un renommage de toutes les variables par de nouvelles variables, le raffinement D d'une clause C peut être défini par : $D \in \rho(C)$ ssi $D = (C\theta_\perp \cup \{l\})\theta_r$ avec le littéral $l \in (\perp \setminus C\theta_\perp)$ et $f(l) < \min(C)$ ou plus simplement $l \in \{l_i \in \perp | f(l_i) < \min(C)\}$.

L'opérateur ainsi construit est clairement localement fini puisque n est fini et (faiblement) complet, puisque toutes les combinaisons de littéraux sont essayées. Il est en revanche redondant. Considérons en effet le cas où $\perp = \{\mathsf{p(A)}, \mathsf{p(B)}, \mathsf{q(C)}, ...\}$ avec $f(\mathsf{p(A)}) = 1$, $f(\mathsf{p(B)}) = 2$ et $f(\mathsf{q(C)}) = 3$. Le raffinement de $\{\mathsf{q(V)}\}$ peut produire $D_1 = \{\mathsf{q(W)}, \mathsf{p(X)}\}$ (ajout du littéral 1) et $D_2 = \{\mathsf{q(Y)}, \mathsf{p(Z)}\}$ (ajout du littéral 2). Or D_1 et D_2 sont équivalentes : $D_1 \sim D_2$.

Examinons maintenant au travers des propriétés 4 et 5 s'il répond au critère de minimalité.

Propriété 4 *L'opérateur ρ défini précédemment ne produit que des subsumés immédiats.* □

Preuve

Supposons que $D \in \rho(C)$ et qu'il existe E telle que $C \succ E \succ D$. Alors, d'après le corollaire 1 et la propriété 2, on a le jeu d'inégalités : $|C| < |E| < |D|$. En utilisant les mêmes notations que précédemment, on a également $D = (C\theta_\perp \cup \{l\})\theta_r$ et donc $|D| = |C| + 1$. Finalement, on aboutit à la contradiction suivante : $|C| < |E| < |C| + 1$. □

On montre par ailleurs dans la propriété 5 que deux clauses produites par le raffinement d'une même hypothèse ne peuvent se subsumer strictement l'une l'autre.

Propriété 5 *Pour toutes clauses C, D_1 et D_2 telles que $D_1, D_2 \in \rho(C)$, alors $D_1 \not\succ D_2$ et $D_2 \not\succ D_1$.* □

Preuve

Supposons que $D_1, D_2 \in \rho(C)$ et que $D_1 \succ D_2$. Par définition de ρ, on a $D_1 = (C\theta_\perp \cup \{l_1\})\theta_{r1}$ et $D_2 = (C\theta_\perp \cup \{l_2\})\theta_{r2}$. Comme $D_1 \succ D_2$, alors $|D_1| < |D_2|$, ce qui

contredit la définition de D_1 et D_2. Même démonstration pour $D_2 \succ D_1$. □

Malgré ces deux propriétés, l'opérateur ρ n'est pas minimal. En effet, il est possible qu'il génère, lors du raffinement d'une hypothèse, deux clauses équivalentes. Ce cas particulier se produit lorsque ce sont des clauses non liées qui sont manipulées, comme le montre l'exemple utilisé précédemment pour montrer la redondance de l'opérateur.

L'absence de non redondance et de minimalité est due à la production de clauses non liées (*cf.* page 97). Or, dans notre application, ces clauses sont jugées non pertinentes. Il est donc possible de proposer un opérateur de raffinement ρ_o ne produisant pas de telles clauses, c'est-à-dire travaillant dans l'espace des clauses liées, mais toujours en restant dans le cadre de la θ_{OI}-subsomption. Pour ce faire, nous utilisons une information supplémentaire qui est associée à chaque clause, d'une manière similaire à ce qui est appelé contexte par L. Badea (2000). L'idée est de ne pas autoriser la sélection des littéraux conduisant à une clause non liée mais, pour préserver la complétude, de les conserver pour les employer lors d'un raffinement ultérieur, s'ils répondent à ce moment au critère de liaison. Ce critère de liaison peut s'exprimer à l'aide des fonctions $vars(C)$, $vars^+(l)$ et $vars^-(l)$ indiquant respectivement l'ensemble des variables contenues dans une clause C, l'ensemble des variables d'entrée du littéral l et l'ensemble des variables de sortie de ce même littéral. L'ensemble des littéraux $Eligible(C)$ contient les littéraux qui n'ont pas été applicables à un moment de la construction de C. Sa définition est donnée de manière récursive ci-dessous ; on suppose que $Eligible(\top) = \emptyset$.

À l'aide de ces nouvelles notations et de celles utilisées précédemment, on peut définir plus formellement le raffinement D d'une clause C par : $D \in \rho_o(C)$ ssi $D = (C\theta_\perp \cup \{l_i\})\theta_r$ avec $l_i \in (\{l \in \perp | f(l) < \min(C)\} \cup Eligible(C))$ et $vars^+(l_i) \in vars(C\theta_\perp)$. On définit également dans le même temps :
$Eligible(D) = \{l \in \perp | f(l_i) < f(l) < \min(C)$ et $vars^+(l) \notin vars(C\theta_\perp)\} \cup Eligible(C) \setminus \{l \in Eligible(C) | f(l) \geq f(l_i)$ et $vars^+(l) \in vars(C\theta_\perp)\}$.

Cela équivaut en fait à un réordonnancement dynamique des littéraux de $\perp$. De ce fait, toutes les hypothèses liées sont effectivement atteintes par raffinement. L'opérateur est donc comme précédemment localement fini et faiblement complet.

Cet opérateur est de plus non redondant si on se limite à des littéraux déterministes comme l'atteste la propriété 6 que nous démontrons ci-dessous.

Propriété 6 *L'opérateur ρ_o défini précédemment est non redondant pour l'espace des clauses liées et déterministes.* □

Preuve

Supposons qu'il existe deux clauses C et D identiques mais obtenues par deux séquences de raffinement différentes. D'après la propriété 8 donnée en annexe B.2, il n'existe qu'une seule substitution θ telle que $C\theta \subseteq \perp$ et $D\theta \subseteq \perp$. C et D ont donc été obtenues par ajout d'un même ensemble de littéraux de $\perp$ mais dans des ordres différents. Comme ρ_o n'ajoute qu'un littéral à chaque raffinement, cela signifie qu'il existe dans C et D deux littéraux l_i et l_j tels que l_i ait été ajouté à la clause par un

raffinement et l_j par un des raffinements suivants dans C et tels que l_j ait été ajouté avant l_i dans D. Considérons la clause C ; dans cette dernière, l_j a pu être ajouté par raffinement de l'ensemble de littéraux $A \cup \{l_i\} \cup B$ soit parce que $f(l_j) < min(A)$, soit parce que $l_j \in Eligible(A \cup \{l_i\} \cup B)$.

Supposons que l'on soit dans le cas où $f(l_j) < min(A \cup \{l_i\} \cup B)$; cela signifie que $f(l_j) < f(l_i)$. Cela est contredit par l'ajout de l_i dans D.

Supposons que l'on soit dans le cas où $l_j \in Eligible(A \cup \{l_i\} \cup B)$. Alors, comme l_j est applicable à A (c'est-à-dire $A \cup \{l_j\}$ est liée), on a $\forall l \in \{l_i\} \cup B$, $f(l) > f(l_j)$ et en particulier $f(l_i) > f(l_j)$. Cette dernière inégalité est impossible puisque l_i ne pourrait être ajouté dans D. □

L'opérateur est donc non redondant, localement fini et faiblement complet, c'est-à-dire optimal. Par ailleurs, il ne produit que des subsumés immédiats (même raisonnement qu'en page 105) et deux clauses issues du raffinement d'une même hypothèse sont incomparables (voir propriété 7) ; il est donc minimal. La combinaison de toutes ces propriétés fait de ρ_o un opérateur parfait.

Propriété 7 *Pour toutes clauses C, D_1 et D_2 telles que $D_1, D_2 \in \rho_o(C)$, alors $D_1 \not\succeq D_2$ et $D_2 \not\succeq D_1$.* □

Preuve

Supposons que $D_1, D_2 \in \rho_o(C)$ et que $D_1 \succeq D_2$. Soit on a $D_1 \succ D_2$; on aboutit alors à la même contradiction que pour la propriété 5 (même raisonnement pour $D_2 \succ D_1$). Soit on est dans le cas où $D_1 \sim D_2$, c'est-à-dire qu'il existe θ injective telle que $D_1\theta = D_2$. Comme $D_1, D_2 \in \rho_o(C)$, alors $D_1 = (C\theta_\perp \cup \{l_1\})\theta_{r1}$ et $D_2 = (C\theta_\perp \cup \{l_2\})\theta_{r2}$. On peut alors écrire $D_1\theta\theta_{r2}^{-1} = D_2\theta_{r2}^{-1} \subset \perp$, soit encore d'après la propriété 8 donnée en annexe B.1 $D_1\theta_{r1}^{-1} = D_2\theta_{r2}^{-1}$. Ainsi, on a $C\theta_\perp \cup \{l_1\} = C\theta_\perp \cup \{l_2\}$, soit encore $l_1 = l_2$. Chaque littéral ne pouvant être choisi qu'une fois lors d'un raffinement, on aboutit à une contradiction. □

3.2.1.3 Exploration du treillis sous θ_{NV}-subsomption

Le treillis précédent contient des hypothèses qui ne sont pas bien formées au regard de notre tâche et ne prend pas directement en compte la notion de généralité entre les littéraux décrivant les mots. C'est cependant ce treillis qui forme la base de l'espace exploré par notre opérateur de raffinement. Nos deux critères (concision et utilisation des variables) nécessitent l'adaptation de la stratégie de recherche décrite ci-dessus, et en particulier la prise en compte de la subsomption généralisée. Nous proposons ci-dessous de considérer cette dernière comme une contrainte ajoutée à l'espace des clauses ordonné par θ_{OI}-subsomption.

Critère d'utilisation des variables

Pour prendre en compte notre critère d'utilisation des variables, nous devons empêcher la production de clauses contenant des variables inutilisées. Il faut donc de nouveau adapter l'opérateur de raffinement pour y inclure cette nouvelle contrainte. Pour ce faire, nous procédons dans un premier temps comme le propose L. Badea & M. Stanciu (1999), c'est-à-dire nous allons utiliser l'opérateur ρ_o itérativement pour générer les raffinements d'une clause jusqu'à ce que l'un de ces raffinements soit valide (*i.e.* réponde à notre critère). Ainsi, un pas de raffinement avec ce nouvel opérateur correspond en réalité à plusieurs pas avec ρ_o. On a donc : $\rho_{nv}(C) = \{D_n | D_1 \in \rho_o(C), D_2 \in \rho_o(D_1), ..., D_n \in \rho_o(D_{n-1})$ tel que D_n est valide mais $D_1, ..., D_{n-1}$ ne le sont pas $\}$.

L'opérateur obtenu est optimal, mais clairement non minimal du point de vue de la θ_{OI}-subsomption, puisqu'il peut ajouter plusieurs littéraux en une seule fois (Badea & Stanciu, 1999). Il n'est pas non plus minimal du point de vue de la θ_{NV}-subsomption puisque le raffinement d'une clause peut produire deux hypothèses D_1 et D_2 telles que $D_1 \succ_{NV} D_2$, comme c'est le cas pour les clauses $D_1 = C \cup \{\mathsf{pred(A,C)}, \mathsf{verb(C)}\}$ et $D_2 = C \cup \{\mathsf{pred(A,C)}, \mathsf{pred(C,D)}, \mathsf{verb(C)}, \mathsf{common\_noun(D)}\}$ si on suppose que les littéraux ont été ajoutés par ρ_o dans l'ordre dans lequel ils sont écrits ici.

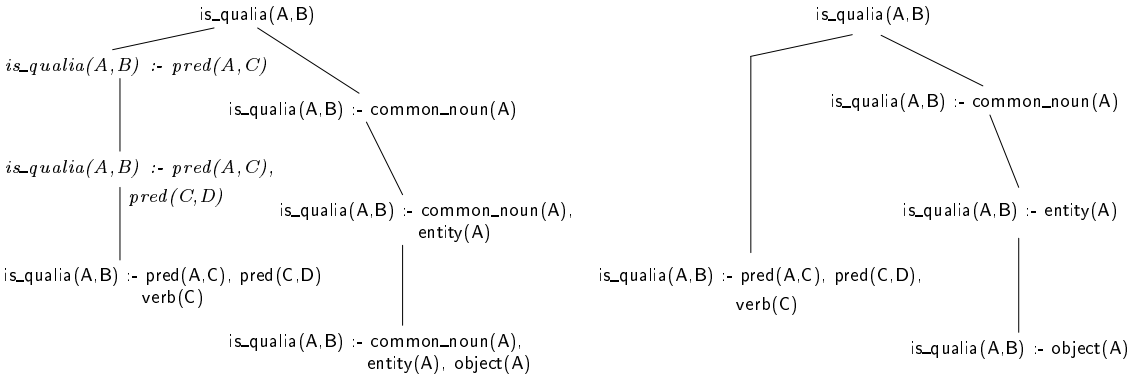
Pour éviter ce phénomène nous procédons dans un second temps comme précédemment pour le critère de liaison. On interdit donc lors du raffinement d'une clause l'ajout d'un littéral unaire sur une variable si cela produit une clause ne répondant pas au critère d'utilisation des variables. Pour préserver la complétude, de tels littéraux doivent cependant pouvoir être choisis ultérieurement ; c'est le rôle de l'ensemble *Eligible*. On redéfinit donc le raffinement par ρ_o d'une clause par : $D \in \rho_o(C)$ ssi $D = (C\theta_\perp \cup \{l_i\})\theta_r$ avec $l_i \in (\{l \in \perp | f(l) < \min(C)\} \cup Eligible(C))$ et $vars(l_i) \in vars^+(C\theta_\perp)$ si $|vars(l_i)| = 1$, et $vars^+(l_i) \in vars(C\theta_\perp)$ sinon. On définit également dans le même temps :
$Eligible(D) = \{l \in \perp | f(l_i) < f(l) < \min(C)$ et $vars(l) \notin vars^+(C\theta_\perp)$ si $|vars(l)| = 1$ et $vars^+(l) \notin vars(C\theta_\perp)$ sinon $\} \cup Eligible(C) \setminus \{l \in Eligible(C) | f(l) \geq f(l_i)$ et $vars(l) \in vars^+(C\theta_\perp)$ si $|vars(l)| = 1$ ou $vars^+(l) \in vars(C\theta_\perp)$ sinon $\}$.

Comme précédemment, ce réordonnancement des littéraux n'a pas d'influence sur les propriétés de minimalité, de complétude faible, de non redondance de ρ_o, et il est bien sûr localement fini. Il permet ainsi à l'opérateur de raffinement ρ_{nv} d'être minimal pour l'espace des clauses respectant le critère d'utilisation des variables.

Critère de concision et subsomption généralisée

Considérons maintenant les clauses répondant au critère d'utilisation des variables mais telles que pour toutes les variables décrites (c'est-à-dire apparaissant dans au moins un littéral unaire), les littéraux de chaque arbre présents dans la clause soient accompagnés de tous leurs littéraux parents jusqu'à la racine. Par exemple, dans une clause C de ce type, s'il apparaît le littéral $\mathsf{artefact(M)}$, il doit également y avoir dans C les littéraux $\mathsf{object(M)}$, $\mathsf{entity(M)}$ et $\mathsf{common\_noun(M)}$, et de même pour les autres littéraux et les autres variables. Ces clauses ne répondent donc pas au critère de concision, mais si l'on suppose que l'on ne considère dans ces clauses que les littéraux les plus spécifiques, et si l'on supprime du treillis les clauses ne répondant pas à cette condition,

on peut se ramener, par cette procédure « d'allègement », à notre treillis des clauses bien formées. Par exemple, une clause dont le corps est pred(A,C), common_noun(C), entity(C), object(C) sera considérée, d'un point de vue externe comme la clause pred(A,C), object(C) ; en revanche, pour la suite de la recherche, l'opérateur de raffinement prendra bien en compte tous ses littéraux et pourra donc produire une clause de corps pred(A,C), common_noun(C), entity(C), object(C), artefact(C) qui sera elle-même interprétée comme pred(A,C), artefact(C). On a ainsi une équivalence entre l'espace des clauses ordonné sous θ_{OI}-subsomption et l'espace des clauses bien formées sous θ_{NV}-subsomption comme l'illustre la figure 3.5 dans laquelle les clauses en italiques sont les étapes intermédiaires de ρ_o.



Espace de recherche sous θ_{OI}-subsomption

Espace de recherche sous θ_{NV}-subsomption

FIG. 3.5 – Espaces d'hypothèses sous θ_{OI} et θ_{NV}-subsomption

Comme pour les critères de liaison et d'utilisation des variables, cette contrainte est prise en compte dans l'opérateur intermédiaire ρ_o en empêchant l'ajout de littéraux unaires sur une variable à une clause qui ne contient pas tous ses littéraux plus généraux. Pour assurer la complétude, comme précédemment, les littéraux qui ne répondent pas à ce critère sont ajoutés à *Eligible* pour être sélectionnables dans un raffinement ultérieur.

On peut démontrer de la même manière que précédemment que ce réordonnancement dynamique des littéraux conserve les propriétés de non redondance et de complétude. L'opérateur ρ_{nv} résultant de l'application successive de ρ_o est ainsi minimal, non redondant, faiblement complet et localement fini, c'est-à-dire parfait, pour la θ_{OI}-subsomption, mais il l'est également pour la θ_{NV}-subsomption si les clauses sont interprétées dans leur forme « allégée » décrite ci-dessus.

3.2.2 Propriétés privées et élagage

Notre opérateur de raffinement, grâce à sa (faible) complétude, nous permet d'explorer potentiellement l'espace d'hypothèses complet. Cependant, une telle exploration est trop coûteuse calculatoirement. Nous évitons donc volontairement de parcourir cer-

taines parties de cet espace (c'est-à-dire des raffinements d'hypothèses) si nous savons par avance qu'elles ne contiennent aucune bonne solution potentielle.

Cela se traduit donc par un élagage de l'espace de recherche des clauses ne vérifiant pas une certaine propriété P. Un tel élagage ne peut cependant se faire que s'il n'y a aucun risque « d'oublier » une bonne hypothèse. Or si une hypothèse viole une propriété P quelconque, rien n'assure *a priori* que ses raffinements violent également P. La section suivante présente un cadre formel garantissant un élagage sûr ; la section 3.2.2.2 applique ce cadre à notre cas.

3.2.2.1 Propriétés privées

Certaines propriétés, dites *propriétés privées* (Torre & Rouveirol, 1997c ; Torre & Rouveirol, 1997b ; Torre & Rouveirol, 1997a), assurent un élagage sûr en fonction d'un opérateur de raffinement donné. Elles permettent l'arrêt du raffinement d'une hypothèse donnée ne satisfaisant pas une certaine propriété privée sans craindre de manquer une solution potentielle. On est en effet dans ce cas assuré qu'aucun des raffinements de l'hypothèse ne vérifie cette propriété.

Définition 7 (d'après (Torre & Rouveirol, 1997c)) *Une propriété P est dite privée au regard de l'opérateur de raffinement ρ dans l'espace de recherche $\mathscr{E}_H$ ssi :*

$$\forall h, h' \in \mathscr{E}_H : \forall(h' \in \rho^*(h) \wedge \overline{P(h)} \Rightarrow \overline{P(h')})$$

où $\overline{X}$ indique la négation de X et $\forall F$, avec F une formule, dénote la clôture universelle de F, qui est la formule close obtenue en ajoutant un quantificateur universel à toutes les variables ayant une occurrence libre dans F.

Examinons une propriété privée très simple et connue (utilisée comme exemple par (Torre & Rouveirol, 1997c)) qui permet d'élaguer l'espace de recherche de manière sûre : la longueur d'une clause. Formellement, la propriété qui limite la longueur des clauses à k littéraux peut être exprimée par $|h| \leq k$ ($|C|$ indique le nombre d'éléments dans la clause C en tant qu'ensemble de littéraux, c'est-à-dire sa longueur). Cette propriété est privée pour l'opérateur de raffinement ρ et l'espace de recherche $\mathscr{E}_H$ ssi $\forall h, h' \in \mathscr{E}_H$: $\forall k \in \mathbb{N} : (h' \in \rho^*(h) \wedge |h| > k \Rightarrow |h'| > k)$. Cette condition est remplie par les opérateurs usuels qui raffinent par ajout de littéraux.

3.2.2.2 Élagage du treillis

Nous utilisons ce formalisme de propriétés privées pour effectuer de nombreux élagages de notre espace de recherche et sommes donc assuré de ne pas supprimer de bonnes hypothèses potentielles.

Notre opérateur consiste basiquement à ajouter des littéraux ($h' \in \rho_{nv}(h) \wedge |h'| > |h|$) ou à en remplacer par de plus spécifiques (alors $h' \in \rho_{nv}(h) \wedge |h'| = |h|$). La propriété de longueur de clause est donc privée au regard de ρ_{nv} et permet un élagage sûr dès qu'une hypothèse dépasse un nombre maximum de littéraux.

Plusieurs autres propriétés privées sont exploitées pour élaguer l'espace des hypothèses. Nous utilisons par exemple un nombre minimum d'exemples positifs à couvrir; ainsi, une clause n'expliquant pas assez d'exemples de couples qualia n'est pas jugée suffisamment représentative et est donc supprimée. Cette propriété est clairement privée d'après notre opérateur de raffinement ρ_{nv} puisque le nombre d'exemples positifs (et également négatifs) décroît au fur et à mesure des spécialisations.

Dans les systèmes de PLI, des propriétés sur la fonction de score sont souvent utilisées pour élaguer l'espace de recherche. Cette fonction permet de décider quelle hypothèse est la meilleure pour la tâche d'apprentissage visée et s'appuie souvent sur la couverture des exemples par l'hypothèse. Nous notons par la suite E_h^+ et E_h^- les ensembles d'exemples et contre-exemples couverts par une hypothèse h. La fonction de score que nous utilisons est $Sc(h) = (P - N, |h|)$ où $P = |E_h^+|$ est le nombre d'exemples positifs couverts par l'hypothèse h et $N = |E_h^-|$ le nombre de négatifs. Ainsi, h_1 est dite meilleure hypothèse que h_2 (avec $Sc(h_1) = (P_1 - N_1, |h_1|)$ et $Sc(h_2) = (P_2 - N_2, |h_2|)$) ssi $P_1 - N_1 > P_2 - N_2$ ou $P_1 - N_1 = P_2 - N_2 \wedge |h_1| < |h_2|$. Malheureusement, puisque $P - N$ n'est pas monotone, il est en général impossible de dire quoi que ce soit sur une hypothèse h qui ne satisferait pas un certain critère de score tel que $Sc(h) < k$, où k peut être le meilleur score trouvé à ce moment de la recherche. Cette propriété permettrait un élagage très intéressant mais puisqu'elle n'est pas privée, elle pourrait conduire à supprimer abusivement de bonnes hypothèses. La propriété privée que nous utilisons pour élaguer en fonction du score est donc plus faible : $Sc^{opt}(h) \geq Sc^{best}$ où Sc^{best} est la plus grande différence $P - N$ trouvée durant la recherche et $Sc^{opt}(h) = P_{current} - N_{\perp}$. $P_{current}$ est le nombre d'exemples positifs couverts par l'hypothèse courante, $N_{\perp}$ est le nombre d'exemples négatifs couverts par $\perp$ (évalué au moment de sa construction). $\forall h, h' \in \mathscr{E}_H : \forall Sc^{best} \in \mathbb{N} : \quad (h' \in \rho^*(h) \wedge Sc^{opt}(h) < Sc^{best} \Rightarrow Sc^{opt}(h') < Sc^{best})$ puisque P décroît par raffinement et $N_{\perp}$ est constant.

Tous ces élagages sûrs nous assurent de trouver la meilleure solution, au regard du critère de score Sc, en un minimum de temps pour chaque treillis exploré. Nous testons dans la section suivante la validité et l'efficacité de toutes ces adaptations apportées à la phase d'inférence d'ASARES à travers l'utilisation du classifieur (l'ensemble des patrons d'extraction inférés) produit à l'acquisition de relations qualia sur notre corpus MATRA-CCR.

3.3 Évaluation

Cette section est dédiée à l'examen des performances de notre système pour notre tâche d'acquisition de couples qualia à partir d'un corpus. Ces performances doivent se traduire par plusieurs points, que nous examinons dans les sous-sections suivantes. Tout d'abord, le processus d'inférence doit s'être bien déroulé et avoir produit des patrons ni trop généraux, sous peine de ne pas produire des résultats assez précis, ni trop spécifiques, pour détecter un grand nombre de couples. Ensuite, les résultats de l'extraction doivent être à la fois précis et complets. Pour les évaluer, nous avons

donc construit un jeu de test empirique ; nous exposons les performances obtenues par ASARES sur celui-ci en section 3.3.2. Enfin, dans une perspective plus linguistique, les résultats produits doivent être pertinents, à la fois en termes de couples qualia trouvés et en termes de patrons générés ; cette analyse est présentée en section 3.3.3.

3.3.1 Évaluation de la qualité de l'apprentissage

Comme toute technique d'apprentissage artificiel, il est important de contrôler le niveau de généralisation des règles (hypothèses) obtenues. Une trop grande généralisation engendrerait une faible précision de l'extraction, et une faible généralisation (apprentissage par cœur) un faible rappel.

3.3.1.1 Validation croisée

Ce contrôle et le réglage des paramètres qui en découle — principalement le bruit autorisé (proportion d'exemples négatifs pouvant être couverts par une hypothèse) — se font de manière automatique grâce à une validation croisée en 10 blocs (Kohavi, 1995) : l'ensemble des exemples et contre-exemples d'apprentissage (E^+ et E^-) est divisé aléatoirement en 10 sous-ensembles ; chaque sous-ensemble sert alternativement de jeu de test pour évaluer l'apprentissage effectué à partir des 9 autres sous-ensembles. Les résultats de chaque évaluation sont résumés dans une matrice de confusion similaire à celle de la figure 3.1[4].

	qualia réel	non qualia réel	Total
prédit qualia	TP	FP	PrP
prédit non qualia	FN	TN	PrN
Total	AP	AN	S

TAB. 3.1 – Matrice de confusion

On peut ainsi calculer le coefficient Φ qui traduit en une seule grandeur toutes les informations de cette matrice de confusion :

$$\Phi = \frac{(TP * TN) - (FP * FN)}{\sqrt{PrP * PrN * AP * AN}}$$

3.3.1.2 Ajustement des paramètres

Pour chaque valeur possible du bruit, une validation croisée est donc effectuée et un Φ moyen calculé à partir des 10 matrices obtenues sur les 10 sous-ensembles de test.

[4]La signification des variables est donnée par la combinaison des lettres : A signifie *actual* (réel), Pr *predicated* (prédit), T *true* (vrai), F *false* (faux), P *positive* (positif), N *negative* (négatif) ; S (*Sum*) est le total.

Les valeurs de bruit retenues sont celles qui maximisent ce Φ moyen. Le tableau 3.2[5] présente les résultats obtenus avec cette valeur optimale.

	Temps (secondes)	Précision (%)	Rappel (%)	Coefficient Φ
Moyenne	2 198	81.09	88.81	0.70
Écart-type	830	2.34	1.74	0.01

TAB. 3.2 – Résultats de la validation croisée

L'emploi de la validation croisée nous apporte donc deux précieuses indications. Elle permet d'une part d'évaluer la qualité intrinsèque de l'apprentissage effectué, et d'autre part de régler automatiquement certains des paramètres nécessaires à l'algorithme de PLI. Ces paramètres, difficiles à appréhender par un utilisateur non averti, se trouvent ainsi pris en compte de manière transparente d'un point de vue extérieur. Cela répond à notre souci affiché de concevoir un outil le plus automatique possible, et donc le plus facilement portable d'un corpus à un autre par un utilisateur quelconque. Dans les expériences présentées, seul le taux de bruit autorisé est ainsi réglé, mais d'autres paramètres pourrait l'être de la même façon (par exemple du nombre minimal d'exemples positifs devant être couverts par une clause pour qu'elle soit acceptée).

Un dernier apprentissage est enfin relancé, avec ces mêmes réglages, et cette fois-ci la totalité des exemples. Les clauses produites sont celles finalement conservées pour l'extraction de couples qualia; dans notre expérience, elles sont au nombre de neuf. Nous les présentons en section 3.3.3 et discutons notamment des schémas morphosyntaxiques qu'elles traduisent. Auparavant, nous évaluons dans la section suivante leurs performances d'extraction sur un jeu de test.

3.3.2 Évaluation des performances d'extraction

L'un des trois objectifs affichés pour notre technique est la qualité des résultats. Elle doit en effet permettre d'acquérir des éléments sémantiques en corpus avec de bons taux de rappel et de précision. Pour évaluer ces performances, nous avons construit un jeu de test permettant de vérifier de manière empirique la qualité de la tâche d'acquisition réalisé par ASARES. Nous présentons ci-dessous ce jeu, puis les résultats obtenus.

3.3.2.1 Construction du jeu de test

Le jeu de test sur lequel nous évaluons les performances de notre système d'acquisition de couples qualia est un extrait de 32 000 mots du corpus MATRA-CCR étiqueté catégoriellement et sémantiquement. Malgré sa taille relativement petite, examiner manuellement toutes les occurrences de couples N-V de ce sous-corpus pour les annoter comme qualia ou non qualia est impossible. Nous nous sommes donc concentré sur

[5] La machine utilisée lors de ces mesures est un PC Pentium IV 2.4 GHz, 512 Mo de RAM sous Red Hat Linux 8.0.

7 noms particulièrement représentatifs du vocabulaire du corpus : *vis, écrou, porte, voyant, prise, capot, bouchon*. Pour ne pas fausser les mesures, aucun de ces noms n'a évidemment été utilisé lors des phases d'apprentissage.

Un programme Perl présente toutes les occurrences de couples N-V, où N est l'un des 7 noms recherchés, apparaissant au sein d'une phrase du sous-corpus, à quatre experts qui annotent alors ces paires comme qualia ou non qualia. Les divergences sont discutées jusqu'à ce qu'un accord se dégage. Finalement, parmi les 286 couples différents trouvés, 66 d'entre-eux sont notés comme étant qualia. Ce jeu de test est ensuite utilisé pour comparer les résultats d'extraction de notre système à ceux des experts.

3.3.2.2 Évaluation des résultats empiriques

La comparaison entre le jeu de test et les couples obtenus par notre système d'extraction sur le sous-corpus se fait à l'aide de matrices de confusion telles que celle donnée précédemment.

Courbes rappel-précision

On peut ainsi grâce à ces matrices calculer les taux de rappel et de précision ; cependant, il faut préalablement choisir un seuil (noté s par la suite), c'est-à-dire un nombre minimum d'occurrences détectées par nos patrons appris, à partir duquel un couple extrait sera effectivement considéré comme qualia. Ainsi, les taux de rappel R et de précision P du système d'acquisition, calculés grâce à notre jeu de test, s'expriment en fonction de s (mêmes notations que précédemment) par :

$$R(s) = \frac{TP(s)}{TP(s) + FN(s)},\ P(s) = \frac{TP(s)}{TP(s) + FP(s)}.$$

Un seuil bas aura évidemment tendance à favoriser le rappel au détriment de la précision et un seuil élevé produira l'effet l'inverse. Pour représenter les performances en fonction des différentes valeurs de s possibles (on note $\mathcal{S}$ cet ensemble), on utilise usuellement les courbes rappel-précision dans lesquelles chaque point représente la précision du système étant donné son rappel pour un seuil s donné.

La figure 3.6 représente la courbe rappel-précision obtenue par notre système sur le jeu de test décrit précédemment. La référence utilisée comme point de comparaison (*baseline*) est la densité de couples qualia parmi tous les couples N-V étudiés du sous-corpus, c'est-à-dire $\frac{AP}{S} = \frac{66}{286} = 0.231$. Cette densité représente la précision moyenne qu'obtiendrait un système choisissant au hasard les couples qualia.

Mesures globales

Les courbes rappel-précision, bien qu'informatives sur les performances globales du système ne donne pas d'informations sur le seuil à choisir ; de plus, la comparaison de deux systèmes à l'aide de ces courbes peut se révéler difficile si les courbes s'entrecroisent. Pour faciliter ces comparaisons, on cherche donc parfois à assigner à ces

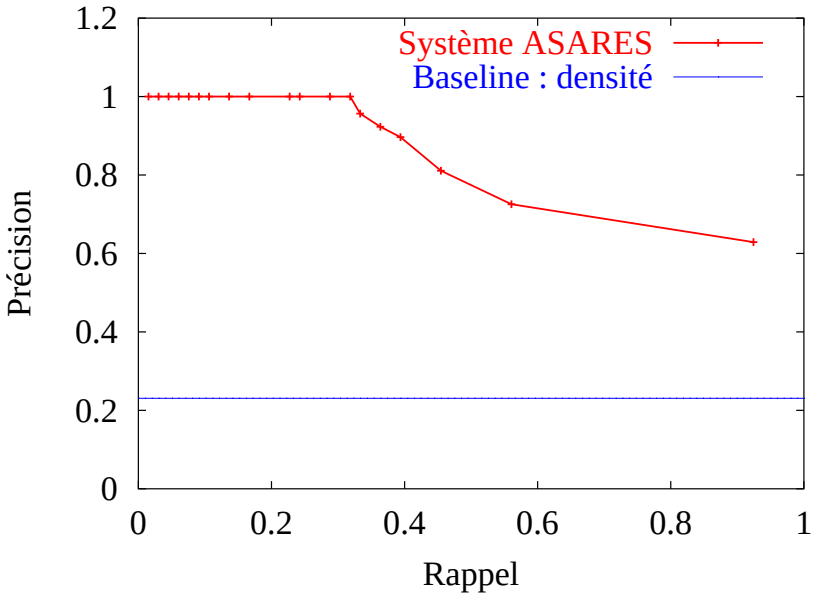


FIG. 3.6 – Courbes rappel-précision du système ASARES

systèmes une mesure unique. Nous utilisons dans ce but la mesure Φ définie précédemment, qui synthétise en une seule grandeur les caractéristiques du système pour un seuil donné. Nous calculons également la F-mesure, fréquemment utilisée[6] dans le domaine de la recherche d'information.

La F-mesure est historiquement issue de la E-mesure (le E est posé pour *efficiency*) proposée par C. Rijsbergen (1979). Cette dernière doit rendre compte de l'efficacité du système en combinant les résultats de rappel ($R(s)$) et de précision ($P(s)$) : $E_\beta(s) = 1 - \frac{1+\beta^2 P(s)R(s)}{\beta^2 P(s) + R(s)}$. La valeur β est fixée par l'utilisateur et traduit l'importance relative qu'il souhaite donner au rappel et à la précision. En notant $\alpha = \frac{1}{\beta^2+1}$, la E-mesure se récrit :

$$E_\alpha(s) = 1 - \frac{1}{\alpha\frac{1}{P(s)} + (1-\alpha)\frac{1}{R(s)}} = 1 - \frac{P(s)R(s)}{(1-\alpha)P(s) + \alpha R(s)}$$

La F-mesure est définie comme $1-E$, et est préférée à la E-mesure car, contrairement à cette dernière, elle croît avec les performances du système. Plus précisément, elle représente la moyenne harmonique pondérée du taux de rappel et du taux de précision :

$$F_\alpha(s) = \frac{P(s)R(s)}{(1-\alpha)P(s) + \alpha R(s)},\ 0 \leq \alpha \leq 1.$$

[6] Bien que très largement utilisée dans ce contexte, la F-mesure, donnée ici à titre indicatif, n'est pas considérée suffisamment informative des performances du système puisque, contrairement au coefficient Φ, elle n'est calculée qu'à partir de trois composantes de la matrice de confusion.

Dans le cas où un poids égal est donné au rappel et à la précision — le cas le plus courant — alors $\alpha = 0.5$ et la F-mesure s'écrit :

$$F(s) = \frac{2P(s)R(s)}{P(s) + R(s)}.$$

Le tableau 3.3 présente les taux de rappel et de précision, la F-mesure et le coefficient Φ de notre système ASARES calculés à partir du jeu de test. Le seuil s_{opt} retenu est choisi tel que $\Phi(s_{opt})$ soit maximal ($s_{opt} = \underset{s \in \mathcal{S}}{\text{argmax}}(\Phi(s))$). Dans notre cas on a $s_{opt} = 1$, c'est-à-dire on obtient les meilleures performances en considérant un couple qualia dès qu'il est détecté au moins une fois.

	Rappel (%)	Précision (%)	F-mesure	Coefficient Φ
ASARES	92.4	62.2	0.744	0.671

TAB. 3.3 – Performances du système d'acquisition ASARES

3.3.3 Évaluation linguistique

Cette section est dévolue à une discussion des différents résultats présentés précédemment dans une perspective linguistique. Plus précisément, nous détaillons dans un premier temps les résultats de l'application des règles apprises sur le jeu de test en faisant en particulier ressortir les causes des mauvaises détections et des absences de détections. Nous examinons ensuite au travers des résultats d'une petite expérience la relation existant entre la notion de couples qualia et celle de lien syntaxique nom-verbe. Nous terminons cette discussion par un examen des 9 règles apprises par PLI et par une comparaison de ces structures sémantiques et morphosyntaxiques à celles identifiées manuellement par une observation linguistique traditionnelle du même corpus.

3.3.3.1 Examen des couples extraits

Les résultats d'extraction obtenue par l'application des patrons appris par PLI sont prometteurs. D'une part, cette technique permet de détecter, sur notre jeu de test, quasiment la totalité des couples qualia. Les cinq couples non détectés apparaissent dans des constructions très rares dans notre corpus, comme *prise-relier* dans *la citerne est **reliée** à l'appareil par des **prises*** où un syntagme prépositionnel est inséré entre le verbe et le syntagme débutant par *par*. D'autre part, au seuil optimal, seulement huit paires sur les 36 non qualia incorrectement détectées qualia ne sont pas liées syntaxiquement. Cela signifie que l'apprentissage par PLI a réussi à faire la distinction de manière très satisfaisante entre les couples reliés syntaxiquement et ceux non reliés.

On peut isoler certains types d'erreurs commises par cette technique d'extraction par PLI. Quelques erreurs sont par exemple causées par des constructions ambiguës dans lesquelles le N et le V considérés peuvent ou non être liés syntaxiquement, comme

enlever-prises dans *enlever les shunts sur les prises*. Ces couples ne peuvent pas être désambiguïsés par des indices contextuels superficiels tels que nos étiquettes sémantiques et catégorielles. De telles constructions sont néanmoins très rares dans notre corpus (8 couples dans notre jeu de test). Cela met au jour les limites de notre approche qui n'exploite que des informations d'assez bas niveau (étiquetage catégoriel et étiquetage sémantique) et est volontairement *knowledge-poor*.

3.3.3.2 Comparaison à une approche syntaxique

Nous avons comparé les résultats obtenus par notre système d'extraction à une approche entièrement manuelle : une annotation syntaxique du corpus. Chaque occurrence de paire N-V de notre jeu de test apparaissant au sein d'une phrase du corpus est étiquetée comme ayant des constituants liés syntaxiquement (c'est-à-dire, le nom est sujet, objet ou modificateur du verbe) ou non. L'idée sous-jacente de cette technique est qu'un lien syntaxique fréquent entre un nom et un verbe dans un texte peut indiquer un lien sémantique entre ce nom et ce verbe, par exemple un lien qualia.

À partir de ces annotations, il nous est possible de construire un système d'acquisition simple. Celui-ci consiste à considérer une paire N-V qualia si plus d'un certain nombre de ses occurrences ont été notées comme étant liées syntaxiquement.

On peut comme précédemment choisir différents seuils fixant le nombre minimal d'occurrences d'un couple devant être liées syntaxiquement pour être considéré qualia. On représente le rappel et la précision de ce système d'extraction pour chacun des seuils possibles dans la courbe rappel-précision 3.7 ; la courbe obtenue par notre système symbolique est donnée pour comparaison. On remarque que les deux systèmes obtiennent

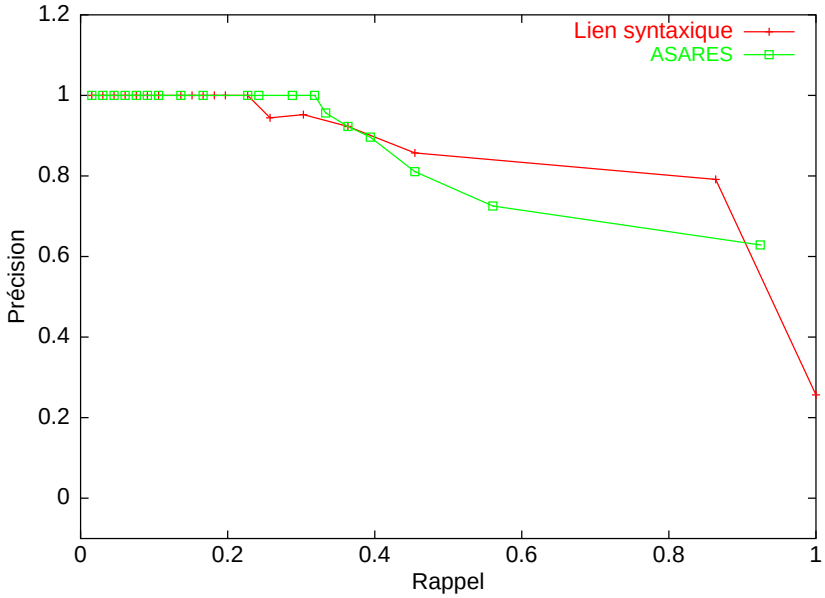


Fig. 3.7 – Courbe rappel-précision du système d'extraction syntaxique

des performances assez proches, avec néanmoins un avantage à l'extraction basée sur le lien syntaxique. Celle-ci jouit notamment d'une meilleure précision que le système ASARES pour de grands rappels.

On recherche également comme précédemment le seuil maximisant le coefficient Φ ; la valeur trouvée est 1. La table 3.4 donne les performances pour ce seuil de ce système manuel.

	rappel (%)	précision (%)	F-mesure	Coefficient Φ
Lien syntaxique	86.4	79.2	0.826	0.772

TAB. 3.4 – Résultats de l'extraction par lien syntaxique

Comme nous l'avons déjà remarqué sur le graphe rappel-précision, ces résultats indiquent un taux de rappel légèrement inférieur à notre système symbolique mais une bien meilleure précision. On constate également que le rappel est inférieur à 100%, ce qui tendrait à montrer que, dans notre corpus, un lien qualia est plus qu'un simple lien syntaxique. Les 13.6% de couples qualia non liés syntaxiquement sont essentiellement des paires N-V apparaissant dans des tournures elliptiques, ou dont les constituants sont séparés par des ponctuations fortes. Par exemple, le couple qualia *voyant-allumer* n'est pas considéré comme étant lié syntaxiquement dans *éteindre le voyant ; allumer*, de même que le couple *poser-vis* dans *poser l'ensemble : rondelle, vis et serrer au couple*.

Cependant, les meilleurs résultats de l'extraction par liens syntaxiques laissent à penser que notre système améliorerait singulièrement ses performances, et plus précisément son taux de précision, en considérant, en plus des informations catégorielles et sémantiques, des informations syntaxiques.

Les techniques d'annotation syntaxique automatique demeurent néanmoins de trop faible précision pour du texte tout-venant pour être utilisées sans supervision humaine, et une annotation manuelle n'est pas envisageable pour une quantité importante de texte. Un compromis doit donc être fait entre des résultats de très haute qualité demandant beaucoup d'intervention humaine et une méthode d'extraction plus automatique mais donnant des résultats sensiblement moins bons.

3.3.3.3 Examen linguistique des patrons appris

D'un point de vue linguistique, le but n'est pas seulement de trouver de bons couples en relation qualia mais aussi d'identifier les patrons qui expriment ces relations. En conséquence, la question est de savoir ce que les clauses inférées nous apprennent sur les structures linguistiques portant les relations qualia entre un nom et un verbe. Nous savons de travaux antérieurs (voir section 1.3.2.2) se focalisant sur d'autres types de relations sémantiques, qu'une relation donnée peut être instanciée dans un large éventail de structures, et que ces structures sont susceptibles de varier énormément d'un corpus à un autre. Ces recherches se concentrent généralement sur l'hyperonymie (ou relation

est-un) et la méronymie (*partie-de*), qui sont à la base des structures ontologiques. Notre but est similaire, avec la difficulté supplémentaire que les relations qui nous intéressent n'ont jamais été étudiées extensivement dans des corpus et sont plus difficiles à identifier que des relations sémantiques plus traditionnelles.

Nous sommes donc face à un ensemble de neuf clauses que nous essayons d'interpréter dans une perspective linguistique :

1. is_qualia(A,B) :- precedes(B,A), near_verb(A,B), infinitive(B), action_verb(B).
2. is_qualia(A,B) :- contiguous(A,B).
3. is_qualia(A,B) :- precedes(B,A), near_word(A,B), near_ verb(A,B), suc(B,C), preposition(C).
4. is_qualia(A,B) :- near_word(A,B), sentence_beginning(A).
5. is_qualia(A,B) :- precedes(B,A), singular_common_noun(A), suc(B,C), colon(C), pred(A,D), punctuation(D).
6. is_qualia(A,B) :- near_word(A,B), suc(B,C), suc(C,D), action_verb(D).
7. is_qualia(A,B) :- precedes(A,B), near_word(A,B), pred(A,C), punctuation(C).
8. is_qualia(A,B) :- near_verb(A,B), pred(B,C), pred(C,D), pred(D,E), preposition(E), sentence_beginning(A).
9. is_qualia(A,B) :- precedes(A,B), near_verb(A,B), pred(A,C), subordinating_conjunction(C).

Dans ces règles, le prédicat precedes(X,Y) signifie que X apparaît dans la phrase avant Y ; pred(X,Y) indique que Y est le mot précédant immédiatement X et inversement suc(Y,X) précise que X suit immédiatement Y dans la phrase; near_word(X,Y) montre X et Y doivent être séparés par au moins un mot et au plus deux mots, et near_verb(X,Y) qu'il n'y a aucun verbe entre X et Y. Les variables A et B représentent respectivement le nom N et le verbe V étant en relation qualia. Le premier patron correspond par exemple au schéma : V d'action à l'infinitif + (tout sauf un verbe)* + N.

Ce qui est tout d'abord frappant est le fait qu'à ce niveau de généralisation, peu d'indices linguistiques traditionnels sont retenus. Les clauses semblent fournir des indications très générales et ne nous indiquent que peu de choses sur les verbes (les verbes d'action et les infinitifs sont privilégiés), noms et les autres catégories de mot intervenant dans les structures décrites. D'autres informations sont néanmoins exploitées, propres à des niveaux de description moins communs, telles que :

- la proximité : c'est un critère majeur. La plupart des clauses indiquent que le nom et le verbe doivent être soit contigus (clause 2) soit séparés par au plus un élément (clauses 3, 4, 6, 7) et qu'aucun verbe ne doit les séparer (clauses 1, 3, 8, 9).
- la position : les clauses 4, 7 et 8 montrent que l'un des deux éléments du couple apparaît au début de la phrase ou après une marque de ponctuation, tandis que la position relative de N et V (precede/2) est donnée dans les clauses 1, 3, 5, 7 et 9.
- la ponctuation : les marques de ponctuation, et plus spécifiquement les deux-points, sont mentionnées dans les clauses 5 et 7.
- la catégorisation morphosyntaxique : la première clause détecte une très importante structure de notre corpus, correspondant aux verbes d'action à l'infinitif.

Ces éléments de description mettent au jour des patrons linguistiques à la fois simples mais très spécifiques à notre corpus, un texte relevant du genre instructionnel. On trouve ainsi dans le corpus de nombreux exemples où un verbe à l'infinitif se trouve en début de proposition et est suivi d'un syntagme nominal (couverts par la clause 1) :

- *débrancher la prise*
- *enclencher le disjoncteur*
- *déposer les obturateurs*

De même, le patron 5, équivalent au schéma V + : + (tout token)* + [:, ;] + N au singulier, met en évidence les structures de listes très nombreuses dans le corpus, comme par exemple :

- *Ouvrir : le capot coulissant, le capot droit et ...*
- *Poser : le bouchon, la porte d'accès ...*
- *... déclenche : l'allumage du voyant 1, l'allumage du voyant alarme ...*

Pour évaluer plus avant ces découvertes, nous avons comparé ces résultats, obtenus par apprentissage artificiel, aux observations linguistiques effectuées manuellement sur le même corpus.

É. Galy (2000) a listé un ensemble de structures verbales canoniques traduisant la relation télique :

- V infinitif + déterminant + N (*visser le bouchon*)
- V + déterminant + N (*ferment le circuit*)
- N + V participe passé (*bouchon maintenu*)
- N + être + V participe passé (*circuits sont raccordés*)
- N + V (*un bouchon obture*)
- être + V participe passé + par + déterminant + N (*sont obturées par les bouchons*)

Ces deux ensembles de patrons montrent quelques recoupements : les deux expériences démontrent la pertinence des structures infinitives et présentent des patrons dans lesquels le verbe et le nom sont très proches l'un de l'autre. Cependant, les schémas obtenus sont assez différents puisque la méthode d'apprentissage propose une généralisation des structures trouvées par E. Galy. En particulier, l'opposition entre les constructions actives et passives sont réunies dans la clause 2 par l'indication de la contiguïté (V peut apparaître avant ou après N). Certains schémas valides ne sont cependant pas retrouvés, en particulier quand les marqueurs sont des expressions polylexicales telles que *avoir pour but de*, *avoir pour fonction de*, qui ne sont en fait pas étiquetée, et donc pas exploitée par la suite, comme une unité. *A contrario*, quelques indices n'ont pas été observés par l'analyse manuelle parce qu'ils relèvent de niveaux d'informations linguistiques généralement délaissés lors de telles analyses (telles que les marques de ponctuation et la position dans la phrase).

En conséquence, lorsque nous examinons les résultats du processus d'apprentissage d'un point de vue linguistique, il apparaît que les clauses apprises donnent des indices surfaciques très généraux sur les structures portant les relations qualia dans notre corpus. Ces indices sont cependant suffisants pour mettre au jour des patrons particulièrement spécifiques au corpus, ce qui est un résultat très intéressant pour l'étude

comparée de telles structures.

Le système ASARES donne donc des résultats satisfaisants, à la fois en termes de qualité des éléments acquis sur corpus et d'interprétabilité des patrons inférés, notamment grâce à l'utilisation de la PLI dont nous avons décrit l'emploi dans ce chapitre. Il répond ainsi à deux des critères du triple objectif que nous nous sommes fixé pour nos travaux d'acquisition. L'approche d'apprentissage adoptée autorise par ailleurs une souplesse d'utilisation permettant d'appliquer cette technique à l'extraction d'autres éléments sémantiques. Cela permet une certaine généricité de notre outil mais ne suffit pas à satisfaire complètement notre dernier critère : la portabilité et l'automaticité. Nous étudions dans le chapitre suivant les deux obstacles à cette automaticité présents dans la version d'ASARES que nous venons de décrire, et examinons les modifications qu'il est possible d'apporter à notre système pour les contourner et ainsi répondre au mieux à l'ensemble de nos objectifs.

Chapitre 4

Amélioration de la portabilité

L'approche symbolique d'acquisition que nous utilisons au sein d'ASARES consistant à inférer des patrons d'extraction donnent, comme nous venons de le voir, de bons résultats, à la fois en termes d'éléments sémantiques extraits et d'interprétabilité des patrons. Le processus d'inférence détaillé au chapitre précédent a été développé de manière à être le plus efficace possible. Deux « goulots d'étranglement » subsistent cependant dans notre processus d'acquisition et nuisent à l'automaticité que nous souhaitons pour notre outil d'extraction ; il s'agit de la phase d'annotation sémantique et de la phase de supervision dans laquelle un expert doit fournir des exemples en corpus des éléments sémantiques que l'on souhaite acquérir. Ces deux phases requièrent en effet une intervention humaine importante et coûteuse, empêchant ainsi une portabilité totale de notre outil puisqu'elles sont nécessaires pour chaque nouveau corpus.

Ce chapitre est dédié à l'étude successive de ces deux sources de coûts et des moyens utilisés pour les réduire voire les supprimer. Nous montrons que cette amélioration de la portabilité peut se faire sans pour autant diminuer l'interprétabilité et la qualité des résultats présentés précédemment, et peut ainsi permettre à ASARES de répondre au mieux au triple objectif que nous nous sommes fixé. Nous présentons dans la première section l'origine du coût dû à l'étiquetage sémantique du corpus. Nous y examinons l'influence de ces informations sémantiques sur la qualité des résultats à travers deux expériences permettant soit de supprimer totalement ce coût, soit de le réduire significativement sans nuire à la qualité des résultats d'extraction. La seconde source de coût, la phase supervision nécessaire à l'inférence des patrons, est l'objet de la section 4.2. Comme nous l'avons vu, celle-ci consiste en la construction d'ensembles d'exemples et de contre-exemples des éléments sémantiques que l'on souhaite acquérir. Cette phase est inhérente à la PLI, mais nous montrons qu'il est possible de l'automatiser grâce à l'emploi de méthodes d'acquisition statistiques. La combinaison de ces méthodes avec notre technique symbolique permet de rendre ASARES totalement automatique.

4.1 Annotation sémantique

La technique utilisée pour réaliser l'annotation sémantique de notre corpus, c'est-à-dire l'assignation à chaque mot d'une étiquette décrivant sa classe sémantique, obtient de bonnes performances en termes de qualité d'étiquetage. En revanche, elle nécessite une phase préalable de constitution de classes sémantiques qui est à ce jour essentiellement manuelle. Cette phase est donc une source de coût restreignant la portabilité de notre outil. Dans cette partie, nous examinons l'influence de cet étiquetage sur la qualité de nos résultats. Plus précisément, après une description du processus d'annotation, nous présentons en section 4.1.2 deux expériences permettant de mesurer l'impact de l'utilisation partielle ou nulle des informations sémantiques au sein d'ASARES.

4.1.1 Étiquetage sémantique de corpus

L'étiquetage sémantique que nous avons réalisé sur le corpus MATRA-CCR repose sur une approche originale initialement proposée par P. Bouillon *et al.* (2000a). Nous en exposons ci-dessous les principes, puis nous détaillons en sous-sections 4.1.1.2 et 4.1.1.3 la phase coûteuse de construction des classes sémantiques. Nous présentons en sous-section 4.1.1.4 la technique adoptée pour mener la phase de désambiguïsation sémantique de cet étiquetage.

4.1.1.1 Principe de l'étiquetage

L'étiquetage sémantique de notre corpus a donc été réalisé selon la méthode exposée dans (Bouillon *et al.*, 2000a). Cette méthode d'annotation sémantique repose sur deux hypothèses majeures :

1. les informations catégorielles peuvent aider à distinguer les sens des mots polyfonctionnels (Wilks & Stevenson, 1996),
2. les ambiguïtés sémantiques restantes peuvent être résolues (comme pour l'étiquetage catégoriel) par un étiqueteur probabiliste.

Elle est donc effectuée sur le corpus étiqueté catégoriellement et bénéficie ainsi de la désambiguïsation catégorielle, obtenue par un étiqueteur probabiliste, des mots polyfonctionnels tels que *règle* qui peut être à la fois un verbe à l'indicatif et un nom.

L'étiquetage sémantique du texte nécessite dans un premier temps de construire manuellement, pour chaque catégorie de mot, une sorte de lexique contenant pour chaque entrée les différentes étiquettes qu'elle peut porter au sein du corpus. Cela implique de choisir pour chaque catégorie un jeu d'étiquettes sémantiques adapté.

4.1.1.2 Classes sémantiques

Pour classifier les noms du corpus de manière systématique, nous avons utilisé, comme point de départ, les classes les plus génériques de WORDNET (Fellbaum, 1998). Cependant, certaines de ces classes, inusitées dans notre corpus, ont été supprimées, alors que d'autres, très présentes, ont été raffinées en sous-classes plus précises (c'est le

cas en particulier de la classe des objets concrets). Nous avons ainsi obtenu 33 classes, organisées en une hiérarchie représentée en figure 4.1 dans laquelle les classes initiales de WordNet non usitées sont en italiques et les étiquettes sémantiques choisies apparaissent entre parenthèses. Le tableau 4.1 donne, quant à lui, pour chaque classe, son

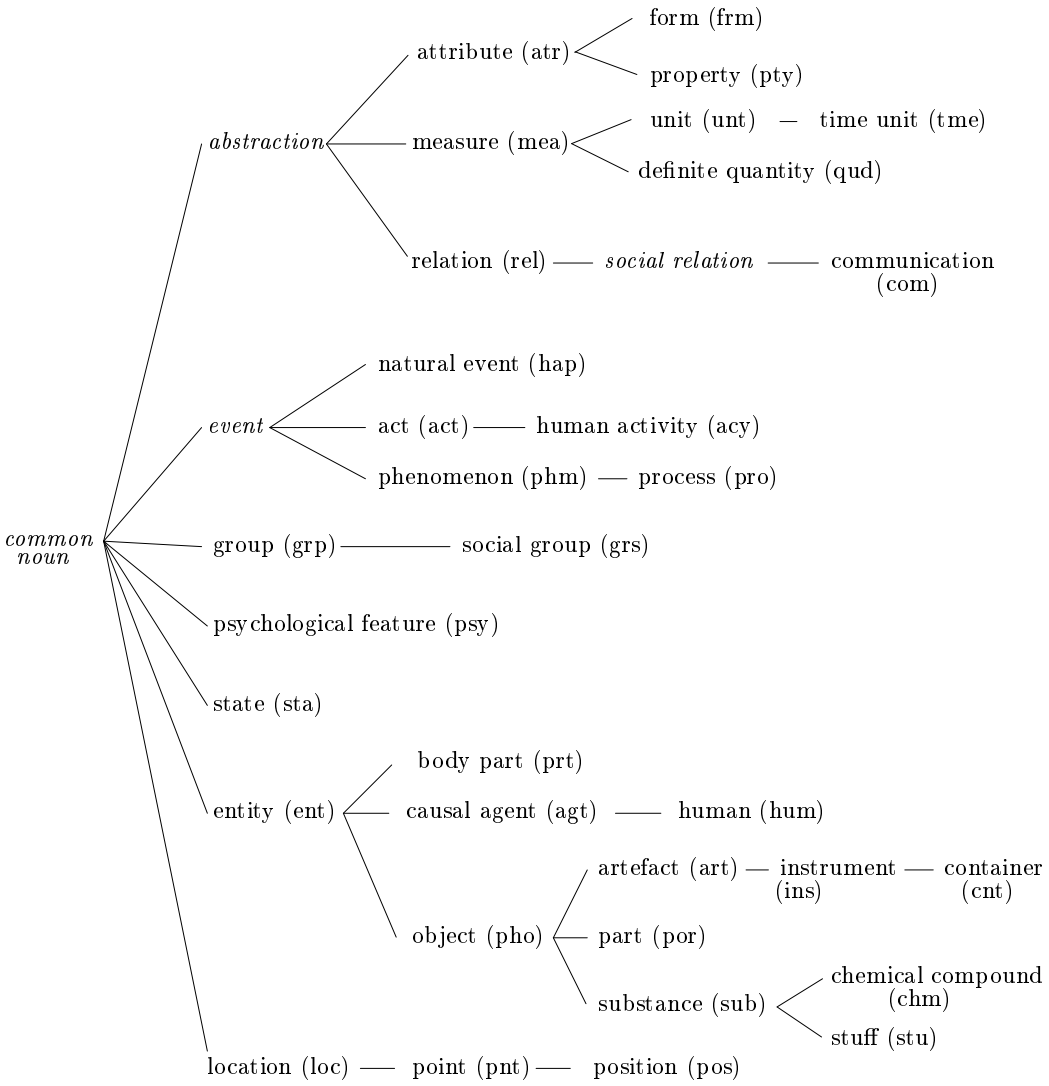


Fig. 4.1 – Hiérarchie de classes pour l'étiquetage sémantique des noms

effectif et quelques représentants.

Environ 8,7 % des entrées du lexique des noms constitué sont ambiguës. Ces ambiguïtés correspondent le plus souvent à des phénomènes de polysémie logique : par exemple, *enfoncement* peut indiquer à la fois un processus et son résultat ; il est donc classifié en pro et sta).

Code	Définition	Effectif	Exemples
act	activité, nom d'action	13	*description, maintenance*
acy	activité humaine	15	*réparation, visite*
agt	cause, agent	5	*contrainte, risque*
art	artefact	358	*filtre, piège*
atr	attribut	17	*aspect, interchangeabilité*
chm	composé chimique	14	*acétone, azote*
cnt	conteneur	21	*bol, bocal*
com	communication	54	*alarme, code*
ent	entité	4	*champignon, oiseau*
frm	forme	35	*arrondi, boursouflure*
grp	groupe	18	*alignement, gamme*
grs	groupe social	3	*personnel, équipage*
hap	événement	64	*anomalie, avarie*
hum	humain	9	*copilote, motoriste*
ins	instrument	266	*anémomètre, peson*
loc	localisation	27	*bord, dehors*
mea	mesure, quantité	31	*minimum, surplus*
phm	phénomène naturel	31	*chaleur, givre*
pho	objet physique	24	*corps, dépôt*
pnt	point de localisation	19	*angle, aplomb*
por	partie, portion	26	*branche, bras*
pos	position d'un objet physique	14	*emplacement, endroit*
pro	processus	274	*accumulation, décapage*
prt	partie du corps	2	*genou, main*
psy	trait ou activité psychologique	30	*besoin, connaissance*
pty	propriété	52	*irrégularité, longueur*
qud	quantité définie	10	*coefficient, double*
rel	relation entre objet	34	*altitude, dépassement*
sta	état	45	*ambiance, besoin*
stu	matériau, matière	21	*alliage, amiante*
sub	substance	37	*carburant, colle*
tme	indication de temps	18	*heure, an*
unt	unité	24	*ampère, bar*
total		1615	

TAB. 4.1 – Définition, effectif et exemples des classes sémantiques des noms

En ce qui concerne les verbes, la classification de WORDNET a été jugée inadaptée du fait d'un trop grand éparpillement des classes. Nous avons donc adopté une partition minimale en sept classes, donnée dans le tableau 4.2, dans laquelle très peu de verbes sont ambigus. Les pronoms ont été classées de manière relativement classique (voir

code	définition	effectif	exemples
acc	activité cognitive	97	*agréer, analyser*
acp	activité physique	426	*abaisser, absorber, accoler*
eta	état	35	*appartenir, devenir, diminuer*
mod	modalité	5	*devoir, pouvoir*
tem	temporalité	9	*dérouler, persister*
posv	possession	5	*appartenir, avoir*
aux	auxiliaire	2	*être, avoir*

TAB. 4.2 – Classification sémantique des verbes

tableau 4.3) ; les prépositions trop ambigües, telles que *de* ou *à*, ont reçu une étiquette propre à elles seules. Les autres catégories de mots du corpus (conjonctions, pronoms,

code	définition	effectif	exemples
rspat	préposition spatiale	27	*dans, sur, au travers de*
rpour	préposition de but	3	*pour, dans le but de*
rtemp	préposition temporelle	12	*après, pendant*
rman	préposition de manière	7	*à l'aide de, par, avec*
rrel	préposition relationnelle	12	*par rapport à, selon*
rneg	préposition de négation	5	*aucun, sans, sauf*
ren	préposition en	1	*en*
rsous	préposition sous	1	*sous*
ra	préposition à	1	*à*
rde	préposition de	1	*de*

TAB. 4.3 – Classification sémantique des prépositions

etc.) ont aussi été organisées en classes et rangées dans un lexique ; là encore, peu d'entrées pour ces catégories sont ambiguës.

4.1.1.3 Synthèse chiffrée des classes sémantiques

Le lexique obtenu en concaténant ce qui a été fait pour les noms, les verbes et toutes les autres catégories contient 1 489 noms dont 129 sont ambigus (ils ont donc au moins deux étiquettes sémantiques). L'ambiguïté la plus fréquente (environ un sixième des ambiguïtés) est entre art (artefact) et pro (processus) ; on la retrouve par exemple dans le nom *ouverture*. C'est un cas très classique de polysémie logique qu'il n'est pas surprenant de retrouver dans notre corpus très technique. Le lexique contient aussi 567 verbes dont seulement 6 ambigus, 8 acronymes, aucun n'étant ambigu, 68 adjectifs dont 4 sont ambigus, 53 prépositions dont 9 sont ambiguës, une quinzaine de déterminants, aucun n'étant ambigu, et environ trente pronoms et pronoms relatifs.

Les faibles taux d'ambiguïté constatés tiennent principalement à deux raisons. D'une part, comme nous l'avons déjà souligné, l'établissement de ce lexique d'étiquettes sémantiques bénéficie de la désambiguïsation catégorielle déjà effectuée (Yarowsky, 1992 ; Ceusters *et al.*, 1996) qui élimine les cas d'homographie par exemple. D'autre part, les catégories sémantiques et leur assignation aux lemmes ont été réalisées en s'appuyant sur le corpus. Elles sont donc, « par construction », particulièrement adaptées pour décrire de manière univoque les classes sémantiques des mots.

4.1.1.4 Désambiguïsation sémantique

L'étiquetage sémantique consiste alors à projeter sur chaque mot du corpus (déjà étiqueté catégoriellement) le contenu de l'entrée correspondante du lexique dont nous venons de décrire le mode de constitution. Les ambiguïtés sont ensuite résolues en utilisant, comme pour l'étiquetage catégoriel, le désambiguïsateur à chaînes de Markov cachées TATOO de l'ISSCO.

Comme mentionné ci-dessus, les ambiguïtés à résoudre sont principalement des problèmes de polysémie logique, puisque les mots ont déjà subi une désambiguïsation morphosyntaxique qui limite la polysémie contrastive. Une portion du texte de près de 6 000 mots a servi à mesurer la précision de notre étiquetage sémantique. Dans cet extrait, 7.78 % des mots étaient initialement ambigus, et l'étiquetage a permis de résoudre correctement 85 % de ces ambiguïtés, soit une précision totale de 98.82%.

4.1.2 Influence de l'étiquetage

Comme nous venons de le voir, la technique d'étiquetage sémantique que nous utilisons requiert la construction d'un lexique contenant les étiquettes possibles de chaque mot. Cette construction est jusqu'à maintenant manuelle et très coûteuse. Par ailleurs, nous avons vu en section 3.3.3.3 que les patrons d'extraction appris exploitent très peu de ces informations sémantiques. Il semble donc naturel de s'interroger sur l'influence de ces étiquettes sur nos résultats au regard du coût qu'elles imposent. C'est l'objet de cette section qui présente pour ce faire deux expériences d'acquisition avec ASARES. La première, décrite dans la sous-section suivante, n'utilise aucune information sémantique sur les mots ; les patrons d'extraction inférés sont donc purement morphosyntaxiques. La seconde, exposée en sous-section 4.1.2.2, est une approche hybride dans laquelle les informations sémantiques sont prises de nouveau en compte sauf celles portant sur les noms, les plus difficiles à construire.

4.1.2.1 Absence d'informations sémantiques

Nous proposons donc dans l'expérience décrite ci-dessous d'examiner les résultats obtenus en effectuant de nouveau un apprentissage dans lequel on interdit à ALEPH d'exploiter les informations sémantiques.

Cette approche est donc dictée par notre volonté d'obtenir un processus le plus automatique possible pour le rendre aisément portable d'un corpus à un autre. En effet, alors que notre technique d'étiquetage sémantique requiert une importante intervention

humaine, l'étiquetage catégoriel est devenu un processus très courant et bien maîtrisé. En particulier, de nombreux outils d'annotation catégorielle sont désormais aisément disponibles et fournissent des résultats de bonne qualité.

Phase d'apprentissage

Pour effectuer ce nouvel apprentissage, nous utilisons le même protocole que celui décrit au chapitre précédent : les exemples positifs et négatifs sont les mêmes et sont codés de la même manière que précédemment ; l'opérateur de raffinement et les stratégies d'élagage sont identiques. Nous ôtons simplement du langage d'hypothèses $\mathscr{L}_H$ les prédicats correspondant aux étiquettes sémantiques. Ces prédicats ne pouvant plus être utilisés pour générer des patrons, le processus d'apprentissage va donc devoir explorer un espace des hypothèses ne contenant plus que des règles exploitant des informations catégorielles sur les mots.

Le tableau 4.4 contient les moyennes et écarts-types, évalués lors de la validation croisée, des différentes grandeurs caractérisant la phase d'apprentissage. Comme attendu, on remarque que le temps d'apprentissage moyen est plus faible que précédemment. En effet, la restriction du langage d'hypothèses a pour effet de réduire la taille de l'espace de recherche et par conséquent la complexité de l'apprentissage, même si ce n'est pas le seul critère intervenant (le taux de bruit autorisé influe également fortement sur cette complexité). Cela se traduit par une diminution de 30% du temps nécessaire pour l'inférence des patrons d'extraction.

	Temps (secondes)	Précision (%)	Rappel (%)	Coefficient Φ
Moyenne	1 757	92.67	80.47	0.75
Écart-type	418	2.24	3.1	0.03

TAB. 4.4 – Résultats de la validation croisée

Le taux de précision obtenu est plus important mais le taux de rappel plus faible, dans des proportions similaires, que ceux obtenus par l'apprentissage avec les informations sémantiques (*cf.* tableau 3.2 page 113). Cela s'explique par le fait que le taux de bruit autorisé maximisant le coefficient Φ est différent de celui retenu précédemment. Celui-ci, automatiquement paramétré selon la procédure décrite en section 3.3.1.2, est plus élevé et a donc privilégié la précision au détriment du rappel. La qualité globale de cet apprentissage, mesurée par Φ, est finalement légèrement supérieure à celle obtenue avec l'ensemble des informations sémantiques.

Toutes ces informations indiquent donc que l'apprentissage est satisfaisant. Les patrons morphosyntaxiques ainsi inférés peuvent donc être utilisés pour l'extraction de couples qualia.

Validation empirique

Comme nous l'avons fait en section 3.3.2.2, nous évaluons les performances d'extraction des patrons inférés à l'aide de notre jeu de test. Nous mesurons ainsi les taux de rappel et de précision obtenus en comparant les couples qualia extraits à ceux retenus par les experts.

La figure 4.2 présente le graphe rappel-précision obtenu en faisant varier le nombre de détections nécessaires pour considérer un couple qualia. Nous y comparons le résultat obtenu au graphe rappel-précision du système ASARES manipulant l'ensemble des informations sémantiques. Ces courbes montrent que l'absence des informations sé-

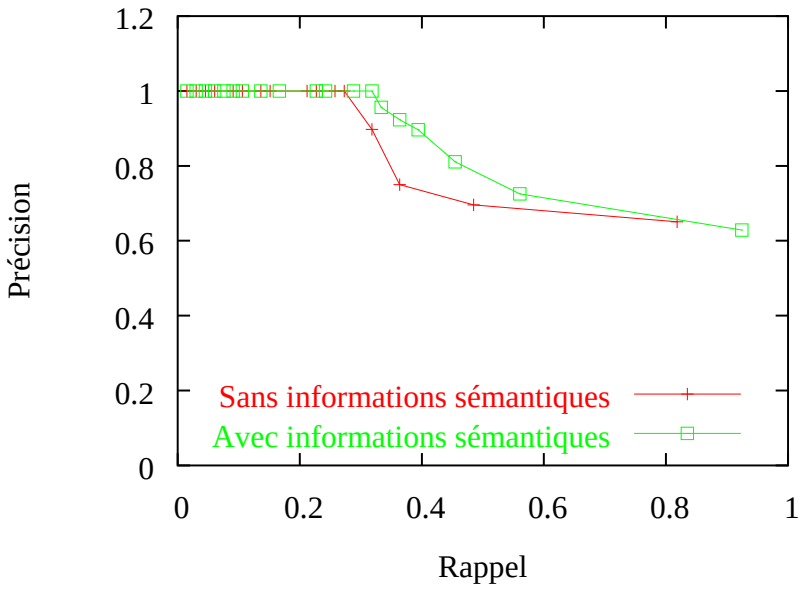


FIG. 4.2 – Courbes rappel-précision du système ASARES avec et sans informations sémantiques

mantiques pénalise sensiblement le système d'extraction. Pour tout rappel, les couples extraits comportent moins de couples qualia ; la précision est donc moins bonne. Les résultats sont néanmoins de bonne qualité. En particulier, le taux de précision est toujours supérieur à 60%.

On recherche également comme précédemment le seuil maximisant le coefficient Φ. La table 4.5 donne les performances des règles inférées sur le jeu de test pour ce seuil. Les résultats confirment nos observations effectuées sur les courbes rappel-précision. Les mesures globales de qualité (la F-mesure et le coefficient Φ) indiquent que le système ASARES obtient de moins bons résultats empiriques en l'absence d'informations sémantiques. Plus précisément, pour des taux de précision comparables, le taux de rappel est

	rappel (%)	précision (%)	F-mesure	Coefficient Φ
	81.8	65.1	0.725	0.637
Gain relatif	−11.5%	+4.7%	−2.5%	−5.1%

TAB. 4.5 – Performances d'ASARES sans informations sémantiques

nettement plus faible que précédemment.

Patrons obtenus

Examinons les patrons morphosyntaxiques qui ont été inférés ; ce sont les neuf suivants :

1. is_qualia(A,B) :- precedes(B,A), near_verb(A,B), suc(A,C), suc(C,D), common_noun(D), preposition(C), infinitive(B).
2. is_qualia(A,B) :- contiguous(A,B).
3. is_qualia(A,B) :- near_word(A,B), near_verb(A,B), suc(A,C), common_noun(C), singular_common_noun(A).
4. is_qualia(A,B) :- precedes(B,A), near_verb(A,B), pred(A,C), ponctuation(C), infinitive(B).
5. is_qualia(A,B) :- precedes(B,A), near_verb(A,B), pred(A,C), ponctuation(C), conjugated(B).
6. is_qualia(A,B) :- near_word(A,B), suc(B,C), suc(C,D), verb(D), infinitive(B).
7. is_qualia(A,B) :- near_word(A,B), suc(B,C), preposition(C), plural_common_noun(A).
8. is_qualia(A,B) :- precedes(B,A), near_verb(A,B), pred(A,C), coordinating_conjunction(C), infinitive(B).
9. is_qualia(A,B) :- precedes(B,A), near_word(A,B), near_verb(A,B), suc(B,C), preposition(C), singular_common_noun(A).

Comme précédemment, tous expriment l'importance d'une relation de proximité entre le nom et le verbe à travers les prédicats near_word/2, near_verb/2 ou contiguous/2. On retrouve également les indices surfaciques telles que les ponctuations dans les clauses 4 et 5.

On remarque que le schéma V + ponctuation + N, permettant de couvrir les structures de listes très spécifiques à notre corpus, est ici exprimé à l'aide des clauses 4 et 5 qui détaillent le cas où le verbe est l'infinitif ou conjugué. La clause 8, portant le patron V infinitif + (tout sauf un verbe)* + conjonction de coordination + N, permet également de trouver les structures impératives propres à notre corpus telles que ... *obturer les raccords et tuyauteries lors du montage* ....

En revanche, l'absence d'informations sémantiques rend impossible de distinguer comme précédemment les verbes d'action. Cette condition semble remplacée par des considérations morphosyntaxiques. Par exemple, les clauses imposant des verbes à l'infinitif sont plus nombreuses et une clause précise que le verbe doit être conjugué.

4.1.2.2 Informations sémantiques partielles

Dans cette expérience, nous nous intéressons à une approche hybride dictée par un souci de portabilité, d'efficacité et de qualité des résultats. Les exemples sont codés ici de la même manière qu'en section 3.1.1.2 et l'opérateur de raffinement est identique. Nous ôtons désormais du langage d'hypothèses les seuls prédicats correspondant aux étiquettes sémantiques des noms. Ceci signifie que la généralisation des exemples ne peut plus se faire sur les catégories sémantiques des noms apparaissant dans le contexte de couples N-V, ni sur la catégorie sémantique du N. Les informations de nature sémantique sur les verbes, les prépositions et les autres catégories de mots du corpus sont en revanche conservées.

Ce choix s'explique par le fait que le coût de l'étiquetage sémantique repose principalement sur la construction du lexique des noms qui est la catégorie apportant le plus d'ambiguïtés (voir section 4.1.1.2). Nous voulons donc ici confronter notre méthode d'apprentissage à un corpus qui ne comporte que des informations pouvant être ajoutées de manière quasi automatique et peu coûteuse.

Phase d'apprentissage

	Temps (secondes)	Précision (%)	Rappel (%)	Coefficient Φ
Moyenne	1 430	82.91	88.79	0.71
Écart-type	2 470	3.54	2.63	0.03

TAB. 4.6 – Résultats de la validation croisée

L'apprentissage réalisé avec ALEPH dans ces conditions donne de bons résultats, avec des chiffres très similaires à ceux de l'expérience décrite au chapitre 3 utilisant l'ensemble des données sémantiques. La qualité intrinsèque de l'apprentissage, mesurée par Φ, est donc de nouveau comparable aux deux premières expériences. Comme pour l'expérience précédente, la taille de l'espace de recherche est plus petite que lors de l'expérience initiale du fait de l'absence des informations sémantiques sur les noms. Cela se traduit sur le temps d'apprentissage qui est en moyenne plus faible ; il est même inférieur de quelques minutes à celui de l'expérience précédente mais avec une variabilité très importante comme le montre l'écart-type.

Validation empirique

Les courbes rappel-précision présentées en figure 4.3 indiquent les performances de ce système comparées à celles du système utilisant toutes les informations sémantiques. La proximité des deux courbes est éloquente. Les deux systèmes obtiennent sur le jeu de test des performances extrêmement proches : quel que soit le rappel, les taux de précision sont quasiment identiques.

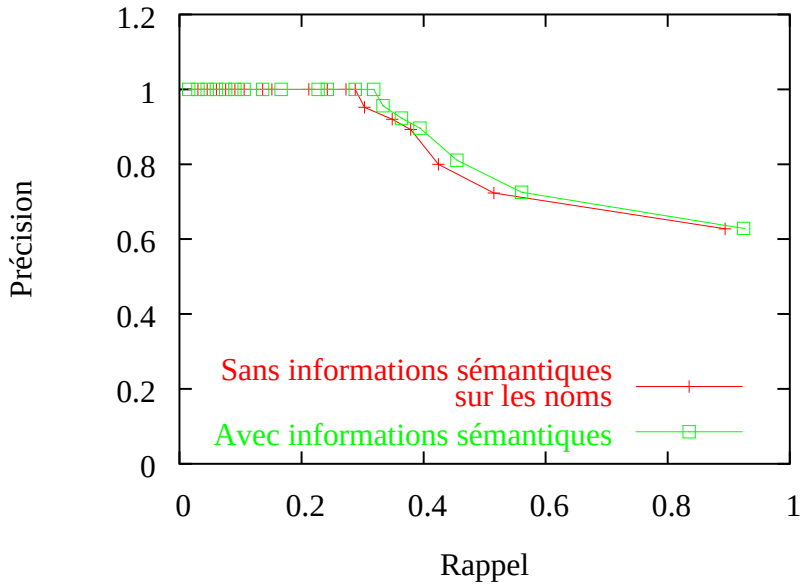


FIG. 4.3 – Courbes rappel-précision du système ASARES avec et sans informations sémantiques sur les noms

Ces constations sont confirmées par le tableau 4.7. Celui-ci contient les valeurs avec un seuil de détections maximisant le coefficient Φ. Elles sont très proches de celles obtenues en conservant toutes les informations sémantiques. Les informations

	rappel (%)	précision (%)	F-mesure	Coefficient Φ
	89.4	62.8	0.738	0.659
Gain relatif	−3.2%	+1%	−0.8%	−1.8%

TAB. 4.7 – Performances d'ASARES sans informations sémantiques sur les noms

sémantiques sur les noms ne semblent donc pas essentielles, ce que l'on avait déjà constaté à l'examen des clauses présentées en section 3.3.3.3.

Patrons obtenus

Le classifieur obtenu lors de cette expérience est composé des 6 clauses suivantes :

1. is_qualia(A,B) :- precedes(B,A), near_verb(A,B), infinitive(B), action_verb(B).
2. is_qualia(A,B) :- contiguous(A,B).
3. is_qualia(A,B) :- precedes(A,B), near_word(A,B), action_verb(B), plural_common_noun(A).

4. is_qualia(A,B) :- precedes(B,A), suc(A,C), de_preposition(C), pred(A,D), ponctuation(D), conjugated(B).
5. is_qualia(A,B) :- near_word(A,B), far_verb(A,B), action_verb(B).
6. is_qualia(A,B) :- precedes(B,A), near_word(A,B), near_verb(A,B), suc(B,C), preposition(C).

Ces clauses inférées en n'utilisant donc que les informations catégorielles des mots et les informations sémantiques sur les catégories de mots autres que les noms reprennent pour la plupart des schémas linguistiques donnés en section 3.3.3.3 ou en section 4.1.2.1. Une fois de plus, nous notons l'importance du critère de proximité (clauses 1, 2, 3, 5 et 6), de la présence des verbes d'action (clauses 1, 3 et 5) et de la ponctuation (clause 4). Certaines des clauses sont même exactement identiques à celles de l'expérience initiale (clauses 1 et 2). La seule information sémantique exploitée dans ces patrons est donc celle marquant les verbes d'action. Celle-ci semble néanmoins importante car elle permet d'obtenir de meilleurs résultats expérimentaux que dans l'expérience précédente.

Ces clauses, bien que proches des clauses obtenues lors de l'expérience rapportée en section 3.3.3, n'y sont pas parfaitement identiques. Cela peut sembler contre-intuitif puisque tous les littéraux nécessaires à l'obtention des mêmes règles, et donc des mêmes résultats, sont disponibles dans cette expérience. Ce fait est cependant naturel puisque la phase 1 de l'algorithme d'ALEPH (*cf.* page 95) amène une dimension aléatoire dans la production des patrons. Cependant, malgré ces différences, on constate que la plupart des patrons produits traduisent des schémas linguistiques similaires, et des résultats expérimentaux quasiment identiques.

Cette dernière expérience donne donc, comme les deux précédentes, de très bons résultats, à la fois en termes de production de règles linguistiquement pertinentes et en termes de construction de lexiques de couples qualia. De plus, elle reste relativement portable d'un corpus à l'autre puisque les informations apportées au corpus peuvent l'être de manière quasi automatique.

4.2 Approches semi-supervisées

Nous nous attaquons dans cette section à une autre source de coût diminuant la portabilité de notre technique d'acquisition concernant cette fois-ci la méthode d'apprentissage que nous employons. Il s'agit du coût humain induit par la phase de supervision nécessaire à l'inférence par PLI des patrons d'extraction. Telle que nous l'avons présentée dans le chapitre précédent, cette phase, qui consiste à construire des ensembles d'exemples et de contre-exemples d'occurrences des informations sémantiques que l'on cherche à acquérir, est effectuée par un expert de manière entièrement manuelle.

Beaucoup de travaux cherchent à améliorer les performances et les coûts des algorithmes d'apprentissage supervisé, non pas en travaillant directement sur ces algorithmes mais en les utilisant, ainsi que les classifieurs qu'ils produisent, de manière particulière. Les techniques de *boosting* (Freund & Schapire, 1999) ou de *bagging* (Breiman, 1996) permettent par exemple d'améliorer la précision des classifieurs. D'autres travaux, rejoignant en cela notre problématique, visent à utiliser le moins possible d'exemples

annotés ; ce sont ces techniques d'apprentissage que l'on dit semi-supervisées. La plupart de ces méthodes reposent sur des variantes de l'amorçage (*bootstrapping*) (Jones *et al.*, 1999) : un petit nombre d'exemples annotés est utilisé pour produire une première version du classifieur ; cette dernière sert alors à annoter des exemples supplémentaires qui aident à générer une deuxième version du classifieur et ainsi de suite.

C'est cette méthode, le *bootstrapping*, que nous proposons d'utiliser pour réduire le coût de supervision. Plus précisément, nous montrons qu'il est possible de combiner notre technique symbolique avec des techniques plus classiques d'acquisition statistique servant d'amorce. Nous présentons ci-dessous ces techniques statistiques et leurs performances puis, dans un second temps, deux façons de les adjoindre à ASARES pour éviter d'avoir à lui fournir des exemples.

4.2.1 Extraction statistique de couples qualia

L'acquisition d'informations sémantiques sur corpus est un domaine de recherche dans lequel les techniques statistiques sont largement employées (voir les sections 1.2.2.1 et 1.3.2.1). Nous montrons ci-dessous qu'il est possible d'utiliser de telles approches numériques pour tenter de mener notre tâche d'acquisition de couples qualia. La sous-section suivante présente le principe de quelques techniques simples que nous mettons en œuvre pour ce faire. Les résultats d'extraction obtenus par celles-ci sont exposés et discutés en sous-section 4.2.1.2.

4.2.1.1 Principe de l'extraction statistique

De nombreux travaux d'acquisition d'informations à partir de textes, et plus particulièrement d'extraction de cooccurrences, ont été menés via des approches statistiques (Manning & Schütze, 1999 ; Pearce, 2002). L'extraction de couples N-V qualia, vus comme une forme spéciale de cooccurrences, peut entrer ce cadre ; il nous est ainsi possible d'utiliser les diverses méthodes statistiques développées pour ce type de tâche.

Beaucoup de ces méthodes se prêtent à une formalisation à l'aide de tables de contingence, dans lesquelles on reporte les nombres d'occurrences conjointes ou non des éléments d'un couple trouvées dans le corpus dans une certaine fenêtre. Le tableau 4.8 en est un exemple adapté à notre cas où les couples sont des paires N-V ; les cooccurrences indiquées sont calculées à partir des lemmes des mots dans une fenêtre d'une phrase de notre corpus MATRA-CCR.

	V_j	$V_k,\ k \neq j$
N_i	a	b
$N_l,\ l \neq i$	c	d

TAB. 4.8 – Table de contingence du couple N_i-V_j

En utilisant les notations de la table de contingence 4.8 et en notant $S = a + b + c + d$, de nombreux coefficients d'association statistique entre le nom N_i et le verbe V_j

s'écrivent simplement :

- le nombre d'occurrences : $occ = a$
- le coefficient de Dice (Smadja, 1993a) : $Dice = \frac{2a}{(a+b)+(a+c)}$
- le coefficient de Kulczinsky : $Kul = \frac{a}{2}\left(\frac{1}{a+b} + \frac{1}{a+c}\right)$
- le coefficient d'Ochiai : $Ochiai = \frac{a}{\sqrt{(a+b)(a+c)}}$
- le coefficient d'Information mutuelle : $IM = log_2 \frac{a}{(a+b)(a+c)}$
- le coefficient d'Information mutuelle au cube (Daille, 1994) : $IM^3 = log_2 \frac{a^3}{(a+b)(a+c)}$
- le coefficient McConnoughy : $McC = \frac{a^2-bc}{(a+b)(a+c)}$
- le coefficient du loglike (Dunning, 1993) : $loglike = a.\log a + b.\log b + c.\log c + d.\log d - (a+b).log(a+b) - (a+c).log(a+c) - (b+d).log(b+d) - (c+d).log(c+d) + S.\log S$
- le coefficient *simple matching* : $SMC = \frac{a+d}{S}$
- le coefficient d'Yule : $Yule = \frac{ad-bc}{ad+bc}$
- le test du Φ^2 (Church & Gale, 1991) : $\Phi^2 = \frac{(ad-bc)^2}{(a+b)(a+c)(b+c)(b+d)}$
- le cosinus dans le cas binaire : $cos = \frac{a}{\sqrt{bc}}$
- le coefficient de Jaccard dans le cas binaire : $Jac = \frac{a}{a+b+c}$

4.2.1.2 Évaluation des techniques statistiques

Toutes ces techniques d'extraction statistiques ont été évaluées, de la même manière que notre technique d'extraction basée sur la PLI, à l'aide du jeu de test décrit en section 3.3.2.1. Tout comme pour notre technique symbolique, les systèmes statistiques associent un score à chaque couple; il faut donc choisir un score-seuil à partir duquel un couple sera considéré comme qualia. Les courbes rappel-précision données en figures 4.4 à 4.16 résument pour chaque seuil possible les performances de tous les systèmes d'extraction basés sur les différents indices statistiques présentés précédemment.

Nous recherchons, comme nous le faisons pour notre méthode symbolique, le seuil optimal pour lequel le coefficient Φ atteint son maximum. La table 4.9 rassemble les résultats obtenus pour chacun des différents indices. Au regard de ce tableau, on remarque que certains de ces indices obtiennent des résultats très proches les uns des autres, ce qui se visualise très bien sur les courbes rappel-précision. Cela s'explique par le fait que l'on peut souvent passer de l'une à l'autre de ces mesures par des transformations monotones (Lerman, 1970).

Peu d'indices statistiques obtiennent d'assez bons résultats pour être utilisés directement pour l'extraction de couples qualia (seuls les systèmes basés sur les scores d'association *Ochiai*, IM^3 et *loglike* ont un coefficient Φ supérieur à 0.5), et aucun

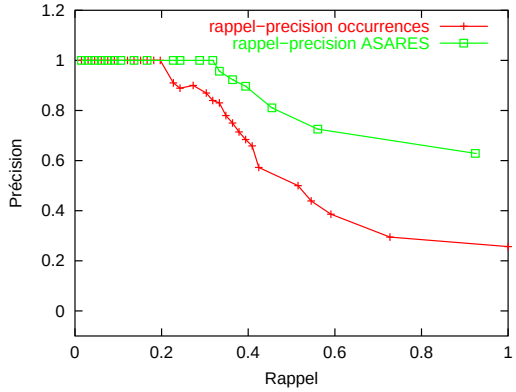


FIG. 4.4 – Courbe rappel-précision du système statistique occurrences

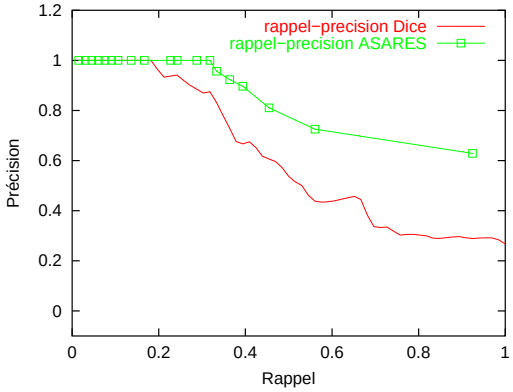


FIG. 4.5 – Courbe rappel-précision du système statistique Dice

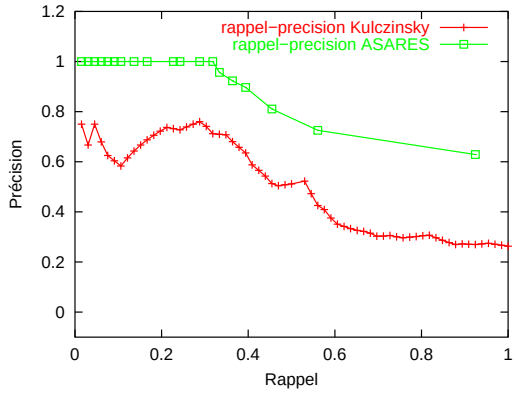


FIG. 4.6 – Courbe rappel-précision du système statistique Kulczinsky

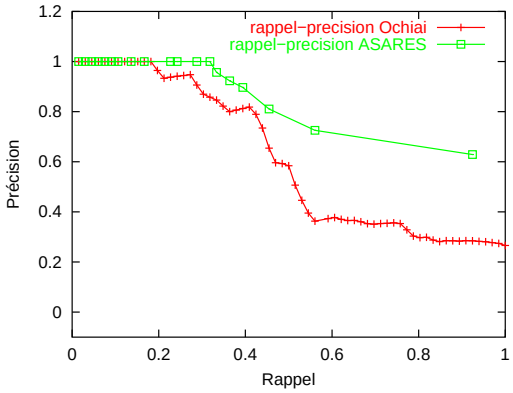


FIG. 4.7 – Courbe rappel-précision du système statistique Ochiai

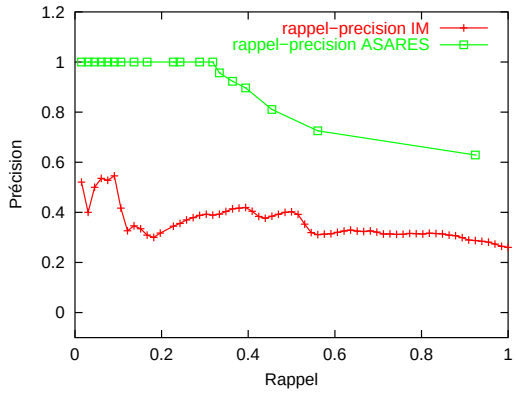


FIG. 4.8 – Courbe rappel-précision du système statistique IM

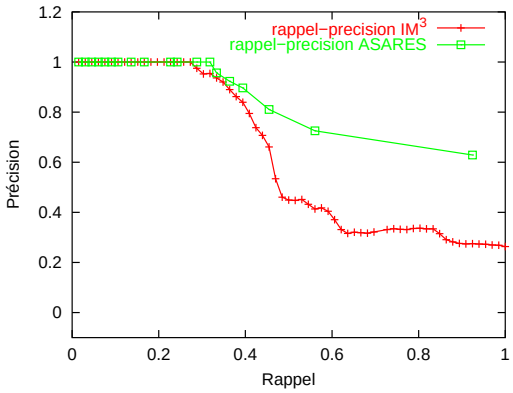


FIG. 4.9 – Courbe rappel-précision du système statistique IM^3

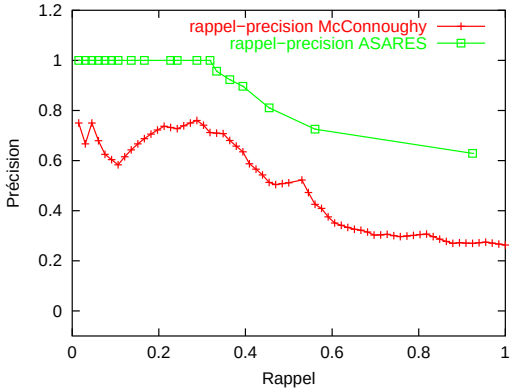


FIG. 4.10 – Courbe rappel-précision du système statistique McC

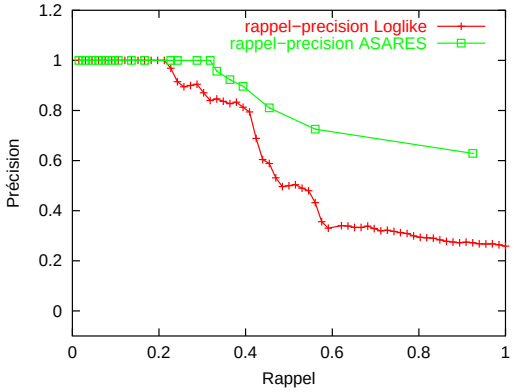


FIG. 4.11 – Courbe rappel-précision du système statistique loglike

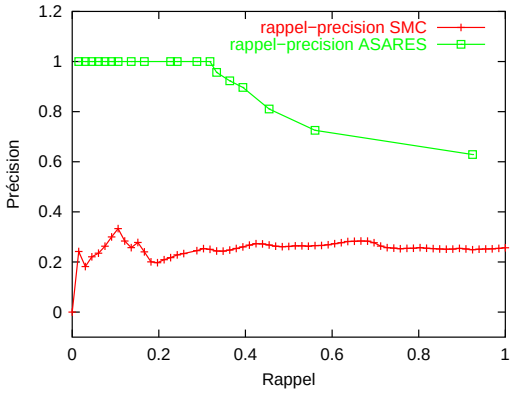


FIG. 4.12 – Courbe rappel-précision du système statistique SMC

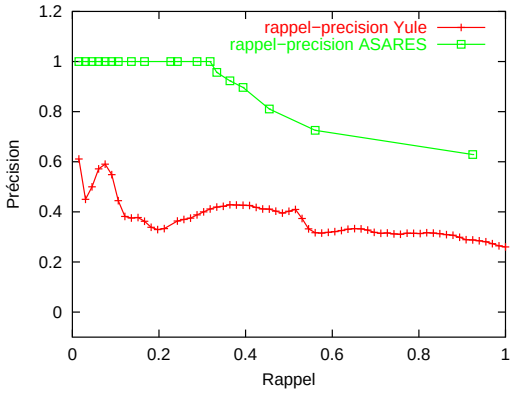


FIG. 4.13 – Courbe rappel-précision du système statistique Yule

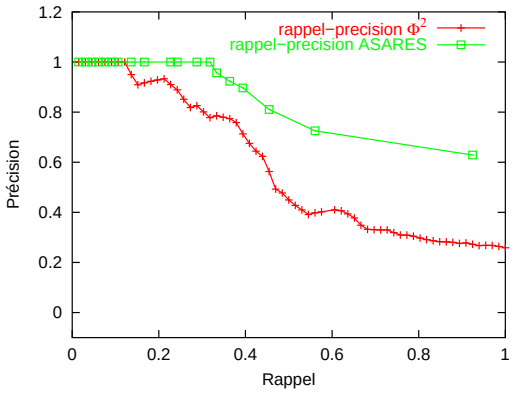


FIG. 4.14 – Courbe rappel-précision du système statistique Φ^2

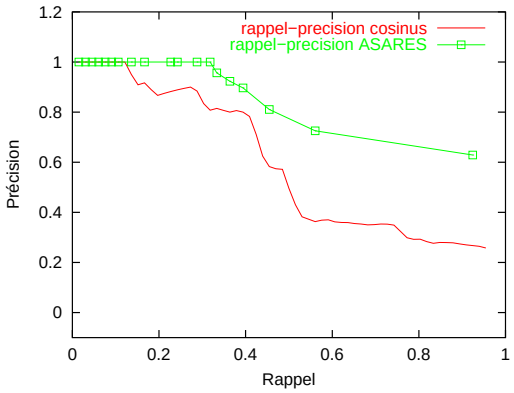


FIG. 4.15 – Courbe rappel-précision du système statistique cosinus

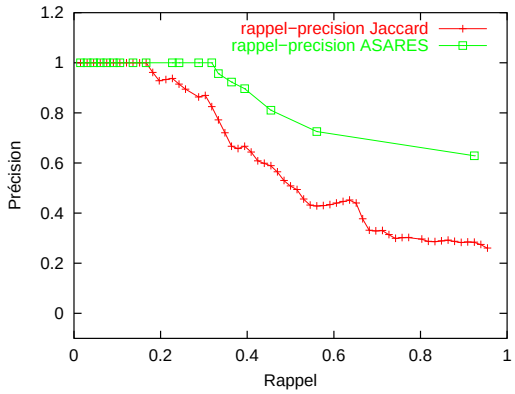


FIG. 4.16 – Courbe rappel-précision du système statistique Jaccard

	rappel (%)	précision (%)	F-mesure	coefficient Φ
occ	33.3	84.6	0.48	0.462
$Dice$	33.3	88	0.48	0.477
Kul	36.4	70.6	0.48	0.414
$Ochiai$	42.4	82.4	0.56	0.517
IM	51.5	40	0.45	0.261
IM^3	36.4	92.3	0.522	0.52
McC	36.4	70.6	0.48	0.414
$loglike$	42.4	80	0.554	0.505
SMC	100	25.3	0.385	0.17
$Yule$	53	41.2	0.464	0.279
Φ^2	37.9	78.1	0.51	0.464
cos	42.4	77.8	0.549	0.493
Jac	31.8	87.5	0.467	0.467

TAB. 4.9 – Résultats des techniques statistiques

d'entre eux n'atteint les performances de notre système d'extraction symbolique. Bien sûr, les différences entre notre approche basée sur l'apprentissage de patrons d'extraction par PLI et l'approche statistique peuvent être aisément expliquées par les disparités de niveau et de quantité des connaissances exploitées par ces deux approches. En effet, alors que les modèles statistiques n'utilisent que les occurrences des lemmes des mots, notre technique d'apprentissage tire parti des étiquettes catégorielles et sémantiques et nécessite des exemples (positifs et négatifs), ce qui est une façon d'incorporer implicitement de la connaissance linguistique dans le processus d'extraction. Par ailleurs, outre ces résultats de faible qualité, ce type de technique ne remplit pas notre condition d'interprétabilité des résultats puisqu'aucun indice n'explique pourquoi un couple est

considéré comme qualia ou non. Cependant, l'approche statistique possède des avantages intéressants : elle est totalement automatique (aucune intervention humaine n'est requise), facile d'utilisation et donc tout à fait portable d'un corpus à un autre.

4.2.2 Combinaison des approches statistiques et symboliques

Nous présentons dans les deux sections suivantes deux approches permettant de faire bénéficier ASARES des qualités d'automaticité des techniques d'acquisition statistiques. Les deux systèmes résultant combinent ainsi les avantages de chacune des approches symbolique et statistique. Grâce à cette combinaison, notre système ASARES, bien que reposant toujours sur l'apprentissage symbolique supervisé, ne nécessite plus qu'un expert lui fournisse des exemples puisque le système d'acquisition statistique qui lui est adjoint (nous avons choisi de prendre le système basé sur la mesure IM^3) fonctionne de manière non-supervisée. C'est en cela que la combinaison de cette technique et de notre approche symbolique relève d'un mode de fonctionnement dit semi-supervisé.

La section suivante présente une première combinaison d'ASARES avec une méthode d'extraction statistique reposant sur un échange séquentiel des résultats entre chacune des méthodes. La section 4.2.2.2 expose quant à elle un second moyen de combiner les deux techniques d'acquisition en incluant plus directement les résultats de l'approche statistique au sein de la phase d'inférence des patrons d'extraction d'ASARES.

4.2.2.1 Combinaison séquentielle

La technique d'extraction mixte présentée dans cette partie repose sur une combinaison séquentielle des systèmes symbolique et statistique présentés précédemment. Comme il est indiqué dans l'algorithme 2, chaque système utilise itérativement en entrée les données de sortie de l'autre système. Plus précisément, la liste de paires N-V générée par un système (L_{PLI} pour le symbolique, L_{IM^3} pour le statistique) est utilisée par l'autre pour construire sa propre liste de couples. La seule contrainte est de débuter cette itération avec la méthode statistique puisqu'elle ne nécessite aucune donnée autre que le corpus.

À l'initialisation, tous les couples N-V apparaissant au sein d'une phrase sont considérés comme potentiellement qualia; cela est indiqué grâce à la règle is_qualia(N,V). donnée dans la liste de patrons d'extraction L_R. L'itération s'arrête lorsque le même ensemble de règles est obtenu lors de deux tours successifs. Lors de nos expériences, le nombre n_1 de couples retenus pour former l'ensemble des exemples positifs a été choisi (à chaque itération) tel que les n_1 premiers couples de L_{IM^3} soient tous ceux ayant un score d'association positif; le nombre n_2 de couples permettant de former l'ensemble des exemples négatifs a quant à lui été choisi tel que $n_2 = n_1$. Le système d'extraction résultant est appelé par la suite système mixte séquentiel.

4.2.2.2 Combinaison intégrée

Contrairement au système présenté ci-dessus dans lequel les techniques statistique et symbolique sont utilisées sans modifications majeures, le second système mixte que

Algorithme 2 Système mixte séquentiel

Initialisation

- $L_R = \{\mathsf{is\_qualia(N,V).}\}$
- application des règles de L_R au corpus ; les couples N-V extraits et leur nombre d'occurrences détectées sont insérés dans L_{PLI}

Itération

1. pour tout couple $N_i - V_j$ de L_{PLI}
   - construction de la table de contingence de $N_i - V_j$ avec les nombres d'occurrences indiqués dans L_{PLI}
   - calcul du score de $N_i - V_j$ selon IM^3
   - insertion, suivant son score, du couple dans la liste triée décroissante L_{IM^3}
2. constitution de l'ensemble E^+ (respectivement E^-) à partir de toutes les occurrences dans le corpus des n_1 (resp. n_2) premiers (resp. derniers) couples de L_{IM^3}
3. apprentissage par PLI avec E^+ et E^- ; les règles obtenues sont regroupées dans L_R
4. application des règles de L_R au corpus, les couples N-V extraits et leur nombre d'occurrences détectées sont réunis dans L_{PLI}

nous proposons combine ces deux approches plus étroitement et nécessite quelques changements dans l'algorithme de PLI.

Comme nous l'avons mentionné dans le chapitre 3, lors de la troisième étape d'un apprentissage par PLI, une règle h est choisie parmi un espace d'hypothèses $\mathscr{E}_H$ si elle maximise une fonction de score Sc. Cette fonction dépend entre autres du nombre d'exemples positifs ($|E_h^+|$) et négatifs ($|E_h^-|$) que h couvre (voir section 3.2.2.2) ; ainsi, on a :

$$h = \underset{h \in \mathscr{E}_H}{\operatorname{argmax}} \; Sc\left(|E_h^+|, |E_h^-|\right)$$

Le principe de notre seconde méthode mixte est de pondérer les exemples selon leur score statistique. Les hypothèses sont donc désormais évaluées à partir des poids des exemples (que nous définissons ci-dessous) qu'elles couvrent. Les ensembles d'exemples et contre-exemples sont donc issus des résultats de la méthode d'acquisition IM^3 : toutes les occurrences dans le corpus des couples ayant les plus hauts scores sont codées dans E^+, et inversement, celles ayant les scores les plus faibles sont placées dans E^-. Un poids w, calculé à partir des scores IM^3, est assigné à chacun de ces exemples. Plus précisément, le poids d'un exemple est le score IM^3 du couple normalisé selon le nombre d'occurrences de ce couple et de manière à ce que la somme des poids des exemples positifs soit égale à la somme des poids des exemples négatifs.

Ainsi, plus un exemple est considéré comme pertinent (c'est-à-dire ayant un score

important) par la méthode statistique, plus il influencera le choix des hypothèses. Finalement, les règles choisies sont celles maximisant $Sc(h)$ redéfinie par :

$$h = \underset{h \in \mathscr{E}_H}{\text{argmax}}\ Sc\left(\sum_{e^+ \in E_h^+} w(e^+), \sum_{e^- \in E_h^-} w(e^-)\right)$$

Avec ces paramétrages et les ensembles E^+ et E^- générés automatiquement, l'algorithme de PLI modifié se déroule comme indiqué dans le chapitre 3 et produit ainsi des règles utilisées ensuite comme patrons d'extraction. Cette technique est appelée par la suite système mixte intégré.

4.2.3 Évaluation de l'approche semi-supervisée

Nous évaluons comme précédemment les résultats obtenus par nos deux techniques semi-supervisées. Nous examinons donc dans un premier temps les performances d'extraction qu'elles obtiennent sur notre jeu de test puis, dans un second temps, les schémas linguistiques portés par les patrons qu'elles inferent.

Performances d'extraction des systèmes mixtes

La figure 4.17 présente les courbes rappel-précision pour nos deux systèmes d'extraction symbolique semi-supervisés ; les systèmes IM^3 et supervisé décrit au chapitre 3 servent de référence. On remarque que les performances de nos systèmes symboliques semi-supervisés sont très proches de la version supervisée et donc nettement supérieures à celles du système statistique, notamment lorsque le rappel est élevé. Plus précisément, on constate que la version semi-supervisée mixte intégrée obtient une meilleure précision pour de faibles rappels alors qu'à l'inverse la technique mixte séquentielle affiche une précision sensiblement supérieure pour des rappels élevés.

Évaluation linguistique

La validation des systèmes symboliques présentés ci-avant passe comme précédemment par une évaluation de l'intérêt linguistique des règles générées. À ce titre, on note tout d'abord de très grandes similarités entre les règles produites par nos deux systèmes semi-supervisés et la version originale d'ASARES. Cela explique bien entendu la similitude, constatée ci-dessus, de leurs performances pour la tâche d'extraction. On retrouve donc dans ces règles des schémas très généraux de proximité entre les constituants des couples, de position, ainsi que l'exploitation des indices surfaciques comme les ponctuations dans des schémas plus spécifiques à notre corpus. Peu d'informations sémantiques sont également utilisées à ce niveau de généralisation, à l'exception notable des verbes puisque comme précédemment les verbes d'action sont privilégiés.

Les deux systèmes semi-supervisés répondent à nos toutes attentes. Ils combinent en cela les avantages des deux approches sur lesquelles ils reposent : l'approche statistique

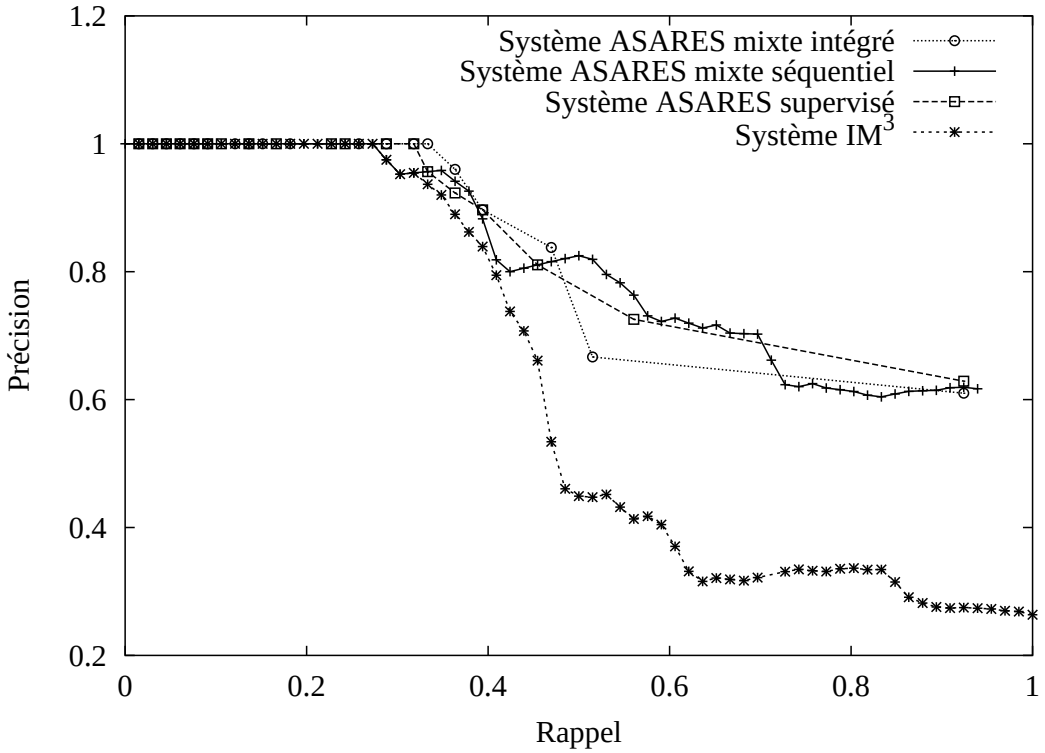


FIG. 4.17 – Courbes rappel-précision des systèmes IM^3, ASARES supervisé, mixte séquentiel et mixte intégré

permettant l'automatisation du processus et l'approche symbolique garantissant une bonne qualité des résultats et des règles d'extraction produites.

4.3 Conclusion

Les aménagements apportés à notre méthode d'acquisition de couples qualia par inférence de patrons d'extraction présentés dans ce chapitre permettent de répondre à nos exigences de qualité des couples extraits, d'interprétabilité linguistique et de portabilité.

Ces techniques de réduction des coûts humains intervenant dans le processus d'acquisition d'ASARES soulèvent toutefois quelques commentaires. Tout d'abord, l'emploi partiel des étiquettes sémantiques et plus généralement le choix des attributs d'apprentissage est discuté dans une perspective d'apprentissage artificiel dans la section suivante. L'utilisation de techniques statistiques de *bootstrapping*, comme nous le faisons dans les deux versions semi-supervisées d'ASARES, est placée dans le contexte d'autres approches semi-supervisées en section 4.3.2.

4.3.1 Choix des attributs

Les expériences présentées en section 4.1 posent la question de l'intérêt de certaines des informations dont nous disposons sur les exemples pour la phase d'apprentissage. Ce problème est bien connu en apprentissage artificiel et fait l'objet de nombreux travaux que l'on regroupe sous le terme de choix des attributs (*feature selection*). Les techniques développées dans le cadre de ces travaux ont pour but de détecter, dans une phase préalable à un apprentissage, les informations qui seront utiles à l'inférence. Cela permet d'améliorer l'efficacité, notamment en termes de complexité et de temps, de l'apprentissage, et est de ce fait très apprécié dans les problèmes comportant de nombreuses informations décrivant les exemples d'apprentissage. L'emploi de telles technique dans notre outil ASARES pourrait permettre d'identifier plus finement les attributs nécessaires à l'apprentissage. Cette connaissance pourrait ainsi être utilisée pour diminuer les coûts de préparation des données, comme les étiquetages, en n'effectuant que les pré-traitements du corpus effectivement exploités par la suite.

4.3.2 Validité de l'approche semi-supervisée

Nos versions semi-supervisées d'ASARES s'appuient sur la collaboration des approches numériques et symboliques par une utilisation relativement simple du *bootstrapping*. Des versions évoluées de *bootstrapping* telles que le *co-training* (Blum & Mitchell, 1998) ou celle proposée dans (Yarowsky, 1995) assurent des propriétés théoriques intéressantes, mais au prix de conditions fortes sur les données. Le *co-training* impose par exemple que les données d'apprentissage puissent être représentées selon deux *vues* conditionnellement indépendantes, deux algorithmes d'apprentissage travaillant ensuite chacun sur une vue des données. Dans le domaine du repérage d'entités nommées, ces deux vues des données peuvent être, par exemple, l'ensemble des mots composant l'entité nommée et l'ensemble des mots composant son contexte.

Cette très forte condition d'indépendance est en fait rarement atteinte dans les données réelles (Abney, 2002) et empêche l'exploitation des résultats théoriques de ces algorithmes semi-supervisés bien que leur efficacité soit empiriquement avérée. C'est néanmoins une condition analogue qui sous-tend intuitivement nos systèmes mixtes. Il faut en effet éviter que la phase d'apprentissage par PLI ne soit biaisée et ne produise des patrons identiques à ceux ayant directement servis à extraire les exemples. Pour permettre l'introduction de nouveaux schémas contextuels dans le processus, les couples extraits par les patrons appris par PLI sont donc filtrés selon un critère indépendant de ces patrons (la mesure IM^3) et toutes les occurrences de ces couples dans le corpus servent ensuite d'exemples. Cette indépendance entre les informations utilisées par les approches symbolique (contexte sémantique et morphosyntaxique) et statistique (occurrences) est mise à mal en pratique car notre corpus comporte de nombreuses instructions répétées à l'identique. Cependant, la ressemblance des patrons produits finalement par nos systèmes mixtes et le système ASARES supervisé semble montrer une bonne tolérance de nos algorithmes mixtes à ce propos.

Les différentes techniques présentées dans ce chapitre permettent de contourner en partie ou complètement les coûts dus à l'étiquetage sémantique et à la phase de supervision de notre approche par apprentissage symbolique. Ces techniques assurent de ce fait une grande portabilité à ASARES, tout en en conservant l'interprétabilité et la qualité des résultats. Notre outil d'acquisition répond donc à notre triple objectif et peut désormais être appliqué relativement aisément à tout nouveau texte. L'application d'ASARES à l'acquisition de relations qualia sur corpus nous a permis d'en vérifier le bon fonctionnement et d'en évaluer les performances d'extraction mais aussi d'appréhender plus facilement, grâce aux patrons, les propriétés linguistiques de ces relations. Outre cette perspective linguistique, l'acquisition de couples qualia avait une autre motivation relevant d'un cadre plus applicatif. C'est sur celle-ci, à savoir, l'étude de l'intérêt des relations qualia en recherche d'information, que nous nous focalisons dans le chapitre suivant.

Chapitre 5

Recherche d'information et Lexique génératif

Le but de la recherche d'information (RI) est de développer des systèmes capables de fournir à un utilisateur les documents d'une base répondant au mieux à sa requête. Dans le cas de documents textuels, on parle de base documentaire et de recherche documentaire. Pour ce faire, il est nécessaire de construire une représentation du contenu du document et de la requête afin de procéder à un appariement entre eux plus pertinent. Une approche communément utilisée est d'associer à chaque document (ou à chaque requête) un index, c'est-à-dire un ensemble de mots, appelés termes d'indexation. Dans le cadre de l'indexation automatique, ces termes sont le plus souvent pris au sein même du document qu'ils doivent décrire. Par exemple, ces termes peuvent être les noms communs (simples ou composés), les verbes et les adjectifs les plus fréquents (d'autres possibilités sont présentées dans (Salton, 1989 ; Sparck-Jones, 1999 ; Strzalkowski, 1995)).

Le résultat présenté à l'utilisateur en réponse à sa requête est un ensemble de documents dont les termes d'indexation sont les plus proches de ceux de la requête. La qualité des systèmes de recherche documentaire dépend donc en grande partie du choix du langage d'indexation. Une façon d'accroître leurs performances est d'améliorer les possibilités de recoupement entre les termes de la requête et ceux des documents. Cela peut s'effectuer en opérant un enrichissement (ou extension) des index, c'est-à-dire en ajoutant aux termes d'indexation d'autres termes proches. Ces termes supplémentaires doivent ainsi permettre des possibilités étendues d'appariement entre documents et requêtes.

À ce titre, les extensions de type morphologique sont assez usuelles bien qu'encore peu utilisées dans les moteurs de recherche grand public. Elles permettent de retrouver dans les documents des mots de la requête quelles que soient leurs flexions (par exemple, *cheval* et *chevaux* peuvent être appariés) voire leurs dérivations. Les systèmes disposant de bases de données linguistiques peuvent également produire une extension cette fois-ci sémantique ; il s'agit par exemple d'enrichir les termes d'indexation par des synonymes ou des mots en relation d'hyponymie-hyperonymie. Dans la réalité, ces systèmes se

limitent le plus souvent à des similarités intra-catégorielles (généralement de nom à nom), et sont par exemple capables de faire correspondre le terme d'indexation *voiture* avec le mot *véhicule*.

Nous avons choisi dans ce chapitre de nous intéresser à un autre type d'extension dont l'importance a été soulignée dans plusieurs travaux d'interrogation de bases de données textuelles (Grefenstette, 1997 ; Fabre & Sébillot, 1999). Il s'agit d'exploiter des liens entre des noms communs et des verbes pour permettre une correspondance entre des formulations nominales et verbales sémantiquement proches. Par exemple, nous souhaitons rendre possible l'appariement entre une requête *magasin de disques* et un texte contenant *vendre des disques* en exploitant l'affinité sémantique entre une entité (*magasin*) et sa fonction typique (*vendre*). Une telle extension nécessite cependant un contrôle précis du lien sémantique entre le nom et le verbe considéré. Ce contrôle est assuré dans notre cas par notre positionnement dans le cadre du modèle du Lexique génératif (Pustejovsky, 1995 ; Bouillon & Busa, 2001), et plus particulièrement par les spécificités de la structure des qualia des noms (voir section 2.3.1). Notre objectif est donc d'évaluer si l'exploitation des relations N-V qualia, inusitées en RI mais dites pertinentes (du moins de manière théorique), permet effectivement d'améliorer les résultats d'un système de recherche d'information.

Les méthodes d'acquisition symbolique des couples qualia présentées au chapitre précédent peuvent être utilisées à ce titre pour construire une collection de couples N-V qualia. Celle-ci peut ensuite permettre d'étendre et de reformuler les requêtes d'un système de recherche d'information (SRI) interrogeant des documents portant sur le même domaine que le corpus ayant permis l'extraction des couples.

Nous présentons dans un premier temps les différents modèles sur lesquels s'appuient les SRI — et plus particulièrement celui utilisé lors de nos expérimentations, le modèle vectoriel — ainsi que les méthodes d'évaluation de ces systèmes. Nous examinons ensuite l'intérêt de l'apport d'informations sémantiques en recherche d'information et nous revenons plus spécifiquement sur l'emploi des relations qualia dans ce cadre. Nous terminons en exposant les résultats des expériences que nous avons menées portant sur l'extension de requêtes par des relations qualia acquises par ASARES.

5.1 Recherche d'information

La recherche d'information, ou plus précisément la recherche documentaire, a pour but de trouver les documents d'une base répondant précisément à une requête posée par un utilisateur. Cette tâche est connue depuis l'antiquité; C. de Loupy (2000) rapporte par exemple que des tablettes datant du troisième millénaire av. J.C. portaient sur leur tranche des indications devant permettre d'en connaître le contenu et donc de retrouver plus facilement l'information cherchée. Cette pratique met en lumière l'intérêt de l'indexation, à savoir, construire une structure permettant de retrouver efficacement les documents répondant à une requête. Bien sûr, l'explosion du nombre de documents disponibles depuis l'avènement d'Internet rend ce domaine de recherche de plus en

plus important. En particulier, pour faciliter la recherche on cherche aujourd'hui des procédures d'indexation qui soient les plus automatiques possible. Ces procédures reposent en grande partie sur une représentation du document selon certains modèles qui doit être calculée automatiquement, souvent à partir de son contenu. Après une description des différentes familles de ces modèles de représentation, nous présentons plus complètement l'un d'eux, le modèle vectoriel, que nous utilisons ensuite pour nos expérimentations. Nous terminons en examinant les différentes mesures permettant de juger de la qualité des SRI à partir de jeux de test.

5.1.1 Modèles de représentation

Il existe différentes façons de représenter documents et requêtes au sein d'un système de recherche d'information. Bien entendu, à chacune de ces représentations sont associées différentes opérations permettant de retrouver le ou les documents répondant aux requêtes posées. Ces différents modèles peuvent être regroupés en plusieurs familles (Piwowarsky, 2003) que nous présentons succinctement ci-dessous.

5.1.1.1 Modèles ensemblistes

Dans les modèles ensemblistes, les documents réponses s'obtiennent par une succession d'opérations sur des ensembles de mots contenus dans les documents. Ce sont donc des modèles très simples dans lesquels les documents sont représentés par un sous-ensemble (voire l'intégralité) des mots qu'ils contiennent.

Modèles booléens

Les modèles booléens sont certainement les plus simples en recherche documentaire, et également parmi les plus anciens. Dans ce type de modèle, les mots représentant un document sont considérés comme en conjonction. Une requête est quant à elle représentée par une formule logique propositionnelle portant sur la présence ou l'absence de mots reliés par des connecteurs (le *ou* $\vee$, le *et* $\wedge$ et le *non* $\neg$). Par exemple, la requête $(\textit{magasin} \vee \textit{marchand}) \wedge (\textit{CD} \vee \textit{disques}) \wedge \neg \textit{vinyls}$ doit renvoyer tous les documents contenant le mot *magasin* ou le mot *marchand* et le mot *CD* ou le mot *disques* mais ne contenant pas le mot *vinyls*.

Le système booléen fait donc correspondre à chaque connecteur $\vee \wedge \neg$ une opération ensembliste portant sur les documents de sa base. Si l'on note Ω la base documentaire, et $\mathcal{D}_q$ l'ensemble des documents de la base correspondant à la formule (la requête) q, on définit récursivement :

requête	ensemble réponse
$q = t$ avec t un terme	$\mathcal{D}_q = \mathcal{D}_t$ l'ensemble des documents contenant t
$q = q_1 \wedge q_2$	$\mathcal{D}_q = \mathcal{D}_{q_1} \cap \mathcal{D}_{q_2}$
$q = q_1 \vee q_2$	$\mathcal{D}_q = \mathcal{D}_{q_1} \cup \mathcal{D}_{q_2}$
$q = \neg q_1$	$\mathcal{D}_q = \Omega \setminus \mathcal{D}_{q_1}$

Les limites de ce modèle découlent directement de la représentation choisie. En effet, la requête, exprimée sous forme logique, est soit vraie soit fausse étant donné un document. En termes ensemblistes, cela va donc se traduire par la discrimination d'un document à partir de la présence ou l'absence d'un seul mot dans ce dernier.

Modèles à ensembles flous

Pour répondre au reproche de la décision binaire d'appartenance d'un terme à un document formulé ci-dessus, une adaptation du modèle booléen a été faite. Elle exploite la théorie des ensembles flous dans laquelle un élément appartient à un ensemble selon un certain degré. L'appartenance d'un terme à un document s'échelonne donc sur l'intervalle réel $[0, 1] \in \mathbb{R}$ et non plus sur l'intervalle entier $[\![0, 1]\!] \in \mathbb{N}$. Cette valeur d'appartenance, généralement notée $\mu_{\mathcal{D}}(\cdot)$ pour l'ensemble de documents $\mathcal{D}$, peut également être interprétée comme une probabilité d'appartenance d'un terme à un ensemble de documents.

Les calculs ensemblistes se font de la même manière que précédemment à ceci près que les opérateurs ensemblistes utilisés sont étendus au cas flou et manipulent donc les degrés d'appartenance des termes. Les réponses proposées à l'utilisateur sont donc les documents pour lesquels le degré d'appartenance des termes de la formule requête est le plus proche de 1.

5.1.1.2 Modèles algébriques

Un autre modèle de représentation très utilisé en recherche documentaire est le modèle algébrique. Dans celui-ci, les documents et les requêtes sont considérés comme faisant partie d'un même espace vectoriel, et leur appariement est fait suivant une mesure algébrique de similarité. Parmi les différentes variantes de ce type de modèle, le plus connu est certainement le modèle vectoriel. Il en existe cependant d'autres, comme les modèles à base de réseaux de neurones (Kwok & Grunfeld, 1993 ; Kwok, 1995). Nous ne présentons ci-dessous que le modèle vectoriel, qui est celui dont nous nous servons lors de nos expérimentations, ainsi que ses variantes les plus courantes.

Modèle vectoriel

Le modèle vectoriel (VSM pour *vector space model*) a été proposé par G. Salton (Salton, 1975 ; Salton, 1989) dans les années soixante-dix mais reste encore très utilisé. Dans ce modèle, les documents et les requêtes sont représentés par des vecteurs et considérés dans le même espace vectoriel. Les documents proposés à l'utilisateur en réponse à sa requête sont ceux dont le vecteur est le plus proche, selon une certaine mesure de similarité choisie, de celui de la requête.

En pratique, les vecteurs définissant l'espace de représentation sont composés de mots apparaissant dans les documents de la collection. Chaque coordonnée de l'espace représente donc un mot. Les vecteurs des documents et des requêtes indiquent les coordonnées de ces derniers dans cet espace à partir des mots qu'ils contiennent (la coordon-

née est nulle si le terme correspondant n'apparaît pas dans le document), coordonnées éventuellement pondérées suivant certains critères (voir section 5.1.2.2). Ainsi, le vecteur colonne D_i représentant le document d_i peut être noté $D_i = (w_{i,1}, w_{i,2}, \cdots, w_{i,n})^t$ où $w_{i,j}$ est la j^{e} coordonnée (*i.e.* le poids) du document d_i dans l'espace vectoriel, n le nombre de termes d'indexation et V^t est la transposée d'un vecteur V.

Finalement, tous les vecteurs des documents de la collection peuvent être rassemblés dans une matrice dont les lignes représentent les documents et les colonnes les termes d'indexation, c'est-à-dire les dimensions de l'espace. Cette matrice est appelée matrice d'occurrences. En notant $m = |\Omega|$ le nombre de documents de la collection et n le nombre de termes d'indexation, la matrice d'occurrences $(m \times n)$ s'écrit :

$$\mathcal{M} = \begin{pmatrix} D_1^t \\ D_2^t \\ \vdots \\ D_m^t \end{pmatrix} = \begin{pmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,n} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{m,1} & w_{m,2} & \cdots & w_{m,n} \end{pmatrix}$$

Les matrices d'occurrences sont le plus souvent très creuses puisque les termes d'indexation formant les bases de l'espace vectoriel apparaissent rarement en totalité dans l'ensemble des documents. Cette propriété est souvent mal supportée par les algorithmes opérant sur cette matrice (tels que ceux calculant les distances entre vecteurs) ; $\mathcal{M}$ est donc souvent transformée en matrice équivalente mais plus dense. De plus, la taille de $\mathcal{M}$ est proportionnelle à la taille de l'espace vectoriel et au nombre de documents de la collection, et est généralement très grande. Cela a pour conséquence de rendre les coûts calculatoires des opérations effectuées sur cette matrice parfois prohibitifs pour des traitements « en-ligne ».

Une autre conséquence liée à la taille de l'espace est connue sous le nom de problème des hautes dimensionnalités (*curse of dimensionality*). Ce phénomène se traduit par le fait que plus un espace est grand (c'est-à-dire plus il compte de dimensions), plus la distance entre les deux objets les plus éloignés est voisine de la distance séparant les objets les plus proches de cet espace (Beyer *et al.*, 1999 ; Pestov, 2000). Autrement dit, dans le cas de la recherche documentaire, tous les documents de la base ont tendance à être proches les uns des autres lorsque l'espace vectoriel est de haute dimensionnalité. On cherche donc souvent à réduire la dimension de l'espace de représentation pour éviter ce phénomène, même si en pratique il ne semble pas trop affecter les systèmes de recherche d'information existants.

Modèle GVSM

Une extension du modèle vectoriel classique, appelée *Generalized Vector Space Model* (GVSM), a été proposée par S. Wong *et al.* (1985). Elle a été développée pour répondre aux critiques selon lesquelles les termes ne sont pas de bonnes bases pour l'espace vectoriel classique puisqu'ils ne sont pas indépendants. L'idée à la base des GSVM est de se placer dans un espace de représentation dual où les documents servent

à décrire les termes et non plus l'inverse. L'avantage de ce simple changement de représentation est que les documents, formant les bases de l'espace, sont plus facilement considérés comme indépendants les uns des autres (Carbonell *et al.*, 1997).

Plus concrètement, sur la matrice $\mathcal{M}$ définie précédemment, on va considérer les colonnes comme représentant le profil d'un terme sur la collection. Un vecteur requête Q dont chaque composant est un terme est simplement transformé en son dual Q' à l'aide de $\mathcal{M}$: $Q' = \mathcal{M}Q$. La dimension des composants de ce vecteur est donc celle des documents. Le critère de similarité entre une requête et un document (ou entre deux documents) est défini par : $sim_{GSVM}(Q, D) = sim_{SVM}(\mathcal{M}Q, \mathcal{M}D)$ où $sim_{SVM}(\cdot)$ est une des mesures utilisées dans le cadre du modèle vectoriel standard (voir section 5.1.2.3).

Modèle LSI

Le modèle LSI (pour *Latent Semantic Indexing*) est une variante du modèle vectoriel dans laquelle on cherche à grouper les mots sémantiquement proches (Deerwester *et al.*, 1990). Il diffère donc principalement du modèle vectoriel standard par les choix des axes de l'espace de représentation qu'il propose : ce ne sont plus les mots présents dans le document qui représentent les dimensions de l'espace mais une combinaison linéaire de ces mots. Ces combinaisons de mots permettent ainsi de mettre au jour des affinités sémantiques latentes entre les mots et offre une meilleure gestion des formulations différentes d'un même concept.

En pratique, cette combinaison linéaire regroupant les mots est obtenue par une décomposition de la matrice d'occurrences $\mathcal{M}$ en valeurs singulières (SVD). Ensuite, seul un nombre restreint (noté k par la suite) de vecteurs propres est conservé. Ces k vecteurs définissent l'espace de représentation et sont orthogonaux.

Plus formellement, on a par SVD $\mathcal{M}^t = U\Sigma V^t$, avec U et V orthogonales ($UU^t = VV^t = 1$) et Σ est une matrice diagonale qui contient les valeurs propres. Chacune de ces dernières indique l'importance du vecteur propre qui lui est associé. Pour réduire la dimension de l'espace, on construit la matrice Σ' en ne retenant de Σ que les k plus fortes valeurs. Cela sélectionne dans U et V les k vecteurs propres réunis dans U' et V'. La mesure de similarité dans cet espace correspond donc à : $sim_{LSI}(Q, D) = sim_{VSM}(U'Q, U'D)$.

Comme nous l'avons dit, cette technique permet, par cette méthode de *clustering* de mots, une représentation plus sémantique des documents. Un autre avantage évident repose sur le fait que l'on a l'assurance, grâce à la SVD et la troncation de la matrice résultante, d'obtenir un espace de dimension faible sans perdre trop d'information. En contrepartie, la SVD est très coûteuse calculatoirement pour de grosses matrices comme peuvent l'être les matrices d'occurrences. Par ailleurs, une critique souvent formulée porte sur le fait que les axes de l'espace soient des combinaisons de mots et rendent donc cette représentation peu appréhendable et difficilement interprétable.

Répondant à la première remarque, Papadimitriou *et al.* (1998) ont proposé une méthode permettant de diminuer la complexité due à la décomposition en valeurs singulières. Pour cela, ils effectuent une première réduction de dimension de $\mathcal{M}$ et calculent

ensuite les vecteurs propres sur cette matrice réduite.

Une variante récente du modèle LSI est le modèle PLSI (le P supplémentaire signifie *Probabilistic*) proposé par T. Hofmann (1999b ; 1999a). L'hypothèse à la base de ce modèle est de considérer, comme précédemment que des classes sémantiques latentes existent, que les termes en sont les indicateurs, et que chaque document est associé à certaines de ces classes latentes.

La technique de représentation vectorielle de documents et ses variantes a été également appliquée avec succès à d'autres domaines connexes à la recherche documentaire comme le filtrage (Foltz, 1990), le routage d'information (Shütze *et al.*, 1995) et la recherche documentaire interlingue (Dumais *et al.*, 1996).

5.1.1.3 Modèles probabilistes

Les modèles probabilistes, dont une présentation très complète est faite par K. Sparck-Jones *et al.* (2000a ; 2000b), tentent quant à eux de modéliser la notion de pertinence. Plus précisément, ils essaient tous d'apporter une réponse à la question (Sparck-Jones *et al.*, 1998) : étant donnés une requête q et un document d, quelle est la probabilité que d soit pertinent pour q, c'est-à-dire qu'il réponde à la requête ?

Contrairement au modèle vectoriel, les modèles probabilistes utilisent une représentation différenciée pour la requête et les documents. Ils peuvent cependant être vus comme des variantes du modèle vectoriel utilisant des pondérations spécifiques (Robertson & Spark-Jones, 1997). En effet, les documents sont représentés par un vecteur dont chaque composante est un terme pondéré, mais les pondérations utilisées pour les documents dépendent de la requête et sont calculées sur des bases probabilistes. Cette technique permet d'ailleurs de ce fait d'éviter les problèmes de haute dimensionnalité évoqués précédemment puisque lors d'une recherche tout se passe comme si l'espace était réduit aux termes de la requête.

Ces approches reposent sur le principe d'ordre des probabilités (*Probability Ranking Principle*) énoncé par S. Robertson (1977). Soient d un document et q une requête ; on note $P(R|d,q)$ la probabilité que d soit pertinent pour la requête q. Robertson montre que présenter les documents d_i à l'utilisateur dans l'ordre décroissant des probabilités $P(R|d_i,q)$ est optimal pour les critères suivants :

- minimisation du nombre de documents consultés si on considère qu'un utilisateur s'arrête au premier document pertinent rencontré ;
- maximisation de l'espérance de la précision pour le même rappel ;
- minimisation de l'espérance des coûts lorsque l'on associe un coût aux faux positifs (documents non pertinents retournés) et aux faux négatifs (documents pertinents non retournés).

Puisque seul l'ordre des probabilités et non leur valeur précise est important, on cherche généralement à estimer une transformation monotone de $P(R|d_i,q)$. Pour pouvoir estimer les probabilités résultant de cette transformation, il est néanmoins nécessaire de

faire certaines hypothèses dont celle d'indépendance des occurrences de termes. Cette dernière étant très forte et peu réaliste, certains auteurs proposent donc des hypothèses plus faibles sur les occurrences de termes (van Rijsbergen, 1977 ; Cooper, 1991 ; Cooper *et al.*, 1994)

Un célèbre représentant des SRI utilisant cette représentation probabiliste est le système OKAPI (Robertson *et al.*, 1981 ; Robertson & Walker, 1994). Ce dernier peut être vu comme une extension du cadre probabiliste prenant en compte non plus simplement la présence ou l'absence d'un terme dans un document mais sa fréquence. Ce système a été utilisé avec succès lors des campagnes d'évaluation TREC et en reste l'une des références.

5.1.2 Détails du modèle vectoriel

Nous revenons dans cette partie sur le modèle vectoriel, modèle que nous utilisons pour nos expérimentations. Nous en détaillons certains principes, éléments et notations nécessaires à la compréhension des expériences d'extension de requêtes par relations qualia que nous présentons dans la dernière section de ce chapitre. Nous présentons le fonctionnement du modèle vectoriel plus précisément, en mettant l'accent sur ses divers paramètres — parfois également valables dans d'autres modèles de représentation — tels que les choix des termes d'indexation, les schémas de pondération possibles des composantes vectorielles et les différentes mesures de similarité existantes.

5.1.2.1 Termes d'indexation

Le choix des termes d'indexation est important car ces derniers forment la structure de l'espace dans lequel seront représentés les documents. Ce choix se fait en plusieurs étapes que nous présentons successivement ici, mais qui sont en pratique souvent imbriquées.

Tokenisation

La *tokenisation* est l'étape qui transforme le texte en une représentation dite « sac de mots » , c'est-à-dire sans information sur leur ordre (on parle aussi de déséquentialisation). Le texte, vu initialement comme une séquence de mots, est alors un multi-ensemble d'unités linguistiques. Ces unités peuvent être soit simplement les mots tels qu'ils sont obtenus après une segmentation, c'est-à-dire des formes fléchies, soit, si le corpus a subi une phase préalable de lemmatisation, les lemmes des mots.

Cette représentation « sac de mots », jugée trop pauvre pour rendre compte correctement du contenu d'un texte, a été remise en question dans certains travaux. Ces derniers proposent par exemple de grouper les mots par syntagmes, ou bien de conserver une information sur leur ordre et proximité, ou encore sur les liens syntaxiques existants au sein des phrases. Ces différents niveaux de représentation peuvent ensuite être combinés pour tenter d'améliorer les performances des systèmes de recherche d'information (Strzalkowski *et al.*, 1999a ; Strzalkowski *et al.*, 1999b).

Choix des termes d'indexation

Le choix des termes d'indexation est une phase très importante lors de l'indexation d'une collection de documents, parfois liée à la phase de pondération que nous exposons ci-après. Ce sont en effet ces termes qui vont représenter le document ou la requête dans l'espace vectoriel. Ils doivent donc être le plus discriminant et univoque possible. Il convient également que ces termes d'indexation ne soient pas trop nombreux car ce sont eux qui vont déterminer la taille de l'espace vectoriel et donc la complexité des calculs de similarité.

Le choix le plus simple est de considérer que tous les éléments obtenus après la phase de tokenisation sont éligibles. Cependant, beaucoup de mots (ou d'ensembles de mots) sont clairement non pertinents pour décrire un texte car porteurs de peu de sens. Il s'agit par exemple de mots dits grammaticaux (comme les prépositions *à*, *de* ou les auxiliaires *être* et *avoir*), communs à tous les textes et donc peu discriminants. Ces mots sont généralement éliminés de la représentation d'un document par l'emploi d'anti-dictionnaires qui contiennent des ensembles de mots qui ne sont pas de bons candidats pour indexer les documents. Ces listes formées de mots très courants sont généralement les mêmes quelle que soit la collection de documents traitée (Savoy, 1999).

Il existe différentes techniques de sélection des termes d'indexation. Celles-ci cherchent à déterminer le pouvoir discriminant des termes en s'appuyant généralement sur la fréquence de ces mots dans le document étudié ou dans la collection, voire également sur leur position dans le document (les mots d'un titre sont favorisés par exemple). G. Salton *et al.* (1975) proposent par exemple de choisir les unités linguistiques ayant une fréquence en documents entre $|\Omega|/100$ et $|\Omega|/10$. D'autres fonctions de sélection, plus complexes, ont également été étudiées. Elles s'appuient par exemple sur des calculs de gain d'information ou d'information mutuelle pour retenir les termes ayant une distribution singulière dans l'ensemble des documents (voir (Yang & Pedersen, 1997) pour une présentation et une comparaison de ces approches). On peut d'ailleurs rapprocher ces travaux de ceux portant sur la sélection des attributs (*feature selection*), particulièrement importants dans le domaine de l'apprentissage artificiel.

5.1.2.2 Pondération

Une fois choisis les termes d'indexation, il peut être intéressant de signifier que tel terme est plus important que tel autre pour décrire un document. Cela est fait en assignant un poids à chaque terme devant refléter son importance. Ce poids est généralement calculé automatiquement à partir de trois critères :

1. l'importance du terme dans le document ;
2. l'importance du terme dans la collection complète de documents ;
3. la taille du document.

Ces trois critères correspondent à trois facteurs de pondération : la pondération locale, la pondération globale et la normalisation (Salton & Buckley, 1988 ; Singhal,

1997). Nous présentons ci-dessous quelques formules mettant en pratique ces trois niveaux de pondération, et plus particulièrement celles utilisées dans le SRI SMART que nous utilisons lors de nos expérimentations.

Pondération locale

La pondération locale d'un terme cherche à mesurer l'importance d'un terme au sein d'un document. Son calcul ne fait donc intervenir que des informations concernant le document considéré.

Généralement, cette pondération locale est fonction de la fréquence du terme dans le document que l'on note *tf* pour *term frequency*. L'idée est en effet que si un terme apparaît souvent dans un document, il est plus pertinent pour décrire le contenu du document qu'un terme n'apparaissant que rarement. Le tableau 5.1 présente quelques mesures classiques utilisées, notamment dans le logiciel SMART, comme schémas de pondération locale ; le document considéré est d et t_i est le terme dont on cherche le poids (c'est-à-dire la i^{e} composante du vecteur représentant d).

Dans ce tableau, outre la fréquence simple du terme (codée n dans SMART), on trouve une pondération binaire (codée b). Cette dernière assigne 1 au terme t_i s'il apparaît dans d, quelle que soit sa fréquence, et 0 sinon. En utilisant cette pondération, on se ramène donc au cas ensembliste.

La pondération normalisée permet quant à elle de prendre en compte non seulement la fréquence du terme t_i dans d, mais aussi de mesurer son importance relativement aux autres termes de d. Le poids d'un terme est ainsi nécessairement compris entre 0 et 1. Une variante de cette pondération locale est le facteur augmenté (codé a) dans lequel la fréquence du terme est normalisée comme précédemment en considérant la fréquence maximale, mais où un poids minimal (de 0.5) est assigné à t_i. On peut donc considérer ce poids comme un mélange d'une pondération binaire avec la pondération normalisée.

La pondération logarithmique (Buckley *et al.*, 1992) a pour but de diminuer l'influence des grandes valeurs. Cela est pertinent lorsque l'on souhaite qu'un document possédant un grand nombre de fois un seul des termes de la requête ne soit pas privilégié face à un document possédant plus de termes de la requête mais avec des fréquences moindres. Si l'on veut au contraire donner plus de poids aux termes très fréquents, il est possible d'utiliser des pondérations telles que la fréquence au carré.

Pondération globale

Alors que la pondération locale a tendance à favoriser les termes très présents, et donc le rappel (Salton & Buckley, 1988), il est également important de privilégier les termes discriminants, c'est-à-dire apparaissant dans peu de documents, pour espérer obtenir une bonne précision.

Contrairement à la pondération locale, le calcul d'une pondération globale doit donc nécessairement exploiter l'ensemble de la base documentaire. Il est notamment basé sur le nombre de documents de la collection dans lesquels le terme considéré apparaît. L'une

code	signification	formule
b	binaire	$l_i = \begin{cases} 1 & \text{si } tf(t_i) > 0 \\ 0 & \text{sinon} \end{cases}$
n	fréquence	$l_i = tf(t_i)$
m	fréquence normalisée	$l_i = \frac{tf(t_i)}{\max\limits_{t \in d} tf(t)}$
a	fréquence augmentée normalisée	$l_i = \begin{cases} 0.5 + 0.5\frac{tf(t_i)}{\max\limits_{t \in d} tf(t)} & \text{si } tf(t_i) > 0 \\ 0 & \text{sinon} \end{cases}$
l	logarithme de fréquence	$l_i = \begin{cases} 1 + \log(tf(t_i)) & \text{si } tf(t_i) > 0 \\ 0 & \text{sinon} \end{cases}$
s	fréquence au carré	$l_i = (tf(t_i))^2$

TAB. 5.1 – Formules de pondération locale

des mesures les plus usitées est la fréquence documentaire inverse, notée *idf* pour *inverse document frequency*, et présentée dans le tableau 5.2.

Les autres mesures apparaissant dans ce tableau sont deux variantes d'*idf* : la fréquence documentaire inverse au carré et fréquence documentaire inverse probabiliste. La première sert simplement à donner plus de poids à un terme discriminant, et ainsi améliorer la précision des recherches. La seconde est dérivée de considérations sur la probabilité de pertinence d'un document contenant le terme t_i (Croft & Harper, 1979 ; Wu & Salton, 1981).

Normalisation

Les pondérations globales et locales permettent, lorsqu'elles sont combinées de faire ressortir (en leur assignant un poids important) les termes à la fois très présents dans le document, et très discriminants. Néanmoins, la comparaison des documents, et donc de

code	signification	formule
n	aucune pondération globale	$g_i = 1$
t	fréquence documentaire inverse	$g_i = \log \frac{\|\Omega\|}{\|\mathcal{D}_{t_i}\|}$
p	fréquence documentaire inverse probabiliste	$g_i = \log \frac{\|\Omega\| - \|\mathcal{D}_{t_i}\|}{\|\mathcal{D}_{t_i}\|}$
s	fréquence documentaire inverse au carré	$g_i = (\log \frac{\|\Omega\|}{\|\mathcal{D}_{t_i}\|})^2$

TAB. 5.2 – Formules de pondération globale

leur vecteur, peut être peu pertinente si les documents sont de tailles très différentes. En effet, un petit document, pouvant être très pertinent pour une requête donnée, contiendra peu de mots et donc *a priori* peu de termes de poids importants. Au contraire, un document long contiendra potentiellement des occurrences plus nombreuses de termes, et sera donc privilégié si une requête porte sur l'un de ces termes. Il est donc parfois nécessaire de normaliser les vecteurs dans le but de rendre les poids des termes peu sensibles à la taille des documents. Le tableau 5.3 présente quelques schémas de normalisation parmi les plus utilisés, où l_k et g_k représentent respectivement la pondération locale et globale du k^{e} terme.

Combinaison des pondérations

Le poids d'un terme est finalement déterminé par la combinaison des pondérations locale, globale et de la normalisation, c'est-à-dire $w_{d,i} = l_i \times g_i \times n_i$.

Les schémas sont notés selon les codes donnés dans les tableaux précédents. Par exemple, le schéma *atc* correspond à la combinaison dans laquelle la pondération locale est la fréquence augmentée normalisée, la pondération globale est la fréquence documentaire inverse et la normalisation celle du cosinus.

Dans certain cas, on peut décider d'adopter des types de pondération différents pour les documents de la collection et pour les requêtes. En effet, ces dernières ont des particularités spécifiques (longueur, absence de répétitions...) dont on peut tirer parti pour choisir un schéma de pondération plus adapté. On note donc par exemple *atc.ltc* le schéma retenu dans une expérience quelconque pour laquelle la pondération des documents est *atc* et celle des requêtes est *ltc*.

code	signification	formule
n	aucune normalisation	$n_i = 1$
s	normalisation L1	$n_i = \frac{1}{\sum_{k=1}^{n}(l_k.g_k)}$
c	normalisation L2 (cosinus)	$n_i = \frac{1}{\sqrt{\sum_{k=1}^{n}(l_k.g_k)^2}}$
f	normalisation en puissance 4	$n_i = \frac{1}{\sum_{k=1}^{n}(l_k.g_k)^4}$
m	normalisation au maximum	$n_i = \frac{1}{\max_{k\in[\![1,n]\!]}(l_k.g_k)}$

TAB. 5.3 – Formules de normalisation

5.1.2.3 Mesures de similarité

Dans le modèle vectoriel et les modèles qui en sont dérivés, les documents et les requêtes, représentés dans un même espace par l'intermédiaire de vecteurs, sont *comparés* à l'aide de mesures dites de similarité. Ces dernières permettent d'assigner une grandeur à un couple de vecteurs quelconques. Cette grandeur doit idéalement refléter la proximité sémantique, voire conceptuelle, de deux documents[1]. Elle doit également induire un ordre permettant, étant donné un document, de classer les autres documents de la collection du plus proche au plus éloigné.

En pratique, ces mesures sont soit effectivement des mesures de similarité (assignant un très haut score aux documents semblables) soit des mesures de dissimilarité (assignant un faible score aux documents semblables). En nous appuyant sur (van Rijsbergen, 1979 ; Besançon, 2001), nous rappelons ci-dessous quelques-unes des mesures les plus utilisées dans ce contexte. On notera par la suite D et Q deux vecteurs (pondé-

[1] On utilise dans la suite le terme générique de document pour désigner tout document de la colection ou toute requête.

rés), représentant deux documents (ou dans le cadre de la recherche documentaire un document et une requête) d et q tels que :

$$D = \begin{pmatrix} w_{d,1} \\ w_{d,2} \\ \vdots \\ w_{d,n} \end{pmatrix} \quad \text{et} \quad Q = \begin{pmatrix} w_{q,1} \\ w_{q,2} \\ \vdots \\ w_{q,n} \end{pmatrix}$$

Mesures ensemblistes

Certaines mesures, dites ensemblistes, comparent la proximité des deux vecteurs en considérant les ensembles de termes d'indexation qu'ils partagent ou non. Elles ne tiennent aucun compte des poids potentiellement associés aux termes et sont donc le plus souvent utilisées avec un modèle de représentation booléen.

On retrouve sans surprise dans ce cadre certaines mesures présentées en section 4.2.1.1 comme scores d'association ; celles-ci sont pour la plupart dérivées du simple coefficient d'occurrences $|D \cap Q|$ qu'elles tentent de normaliser (van Rijsbergen, 1979). Ainsi, la distance issue du coefficient de Dice s'écrit :

$$\delta_{Dice}(D, Q) = 2 \times \frac{|D \cap Q|}{|D| + |Q|}$$

On a de la même façon la mesure du Jaccard :

$$\delta_{Jaccard}(D, Q) = \frac{|D \cap Q|}{|D \cup Q|}$$

La différence symétrique normalisée, définie ci-dessous, a en plus la propriété de vérifier l'inégalité triangulaire, ce qui en fait une distance au sens mathématique du terme :

$$\delta_{\Delta}(D, Q) = \frac{|D \Delta Q|}{|D| + |Q|} = \frac{|(D \cup Q) \setminus (D \cap Q)|}{|D| + |Q|}$$

Comme nous l'avons dit précédemment, ces mesures sont pour la plupart monotones les unes par rapport aux autres (Lerman, 1970). Pour les mesures précédentes, on a par exemple les égalités suivantes : $1 - \delta_{Jaccard}(D, Q) = \frac{2}{2 - \delta_{Dice}(D,Q)}(1 - \delta_{Dice}(D, Q))$ et $\delta_{\Delta}(D, Q) = 1 - \delta_{Dice}(D, Q)$.

Mesures géométriques

Les mesures géométriques peuvent être vues comme des extensions des mesures ensemblistes dans le cas où l'on souhaite prendre en compte les pondérations des termes d'indexation. Ainsi aux opérations ensemblistes correspondent les opérations vectorielles telles que le produit scalaire (noté par $\cdot$) ou le calcul de norme.

Dans ce cadre, la mesure la plus connue et la plus utilisée est le cosinus. Son principe est simple : dans l'espace de représentation, deux vecteurs seront très proches si l'angle

qu'ils forment est très petit, ce qui est le cas si son cosinus est proche de 0. On peut également interpréter le cosinus comme un simple produit vectoriel normé ; cela se traduit sous une norme L2 par la formule suivante :

$$\delta_{cos}(D,Q) = \frac{D \cdot Q}{\|D\|_{L2}\ \|Q\|_{L2}} = \frac{\sum_{i=1}^{n} w_{d,i}\ w_{q,i}}{\sqrt{\sum_{i=1}^{n} w_{d,i}^2 \sum_{i=1}^{n} w_{q,i}^2}}$$

L'avantage du cosinus est donc que l'on a une normalisation des vecteurs par la longueur des documents.

D'autres mesures géométriques classiques sont également utilisées, comme la distance L2 (c'est-à-dire la distance euclidienne) :

$$\delta_{L2}(D,Q) = \|D - Q\|_{L2} = \sqrt{\sum_{i=1}^{n}(w_{d,i} - w_{q,i})^2}$$

et de la même manière la distance L1 :

$$\delta_{L1}(D,Q) = \|D - Q\|_{L1} = \sum_{i=1}^{n}(w_{d,i} - w_{q,i})$$

Là encore, diverses mesures sont liées les unes aux autres. Par exemple, si D et Q sont normalisés par la norme euclidienne, la distance euclidienne est monotone par rapport à la mesure du cosinus :

$$\frac{\delta_{L2}(D,Q)^2}{2} = 1 - \delta_{cos}(D,Q)$$

Cela implique que les documents, bien qu'obtenant des scores différents, seront classés dans le même ordre, et le même ensemble de documents sera donc retourné à l'utilisateur en réponse à sa requête.

Mesures distributionnelles

Si les vecteurs D et Q sont normalisés suivant la norme L1, on a alors $\sum_{i=1}^{n} w_{x,i} = 1$. On peut dans ce cas interpréter les vecteurs comme des distributions de probabilités (Lee, 1997). Des mesures existantes de dissimilarité entre distribution de probabilités sont donc directement transposables et utilisables dans ce contexte de mesures de similarité entre vecteurs.

On peut par exemple définir une distance du χ^2, dont quelques propriétés intéressantes en recherche documentaire sont données par M. Rajman et L. Lebart (1998) :

$$\delta_{\chi^2}(D,Q) = \sqrt{\sum_{i=1}^{n} \rho_i (w_{d,i} - w_{q,i})^2} \quad \text{et} \quad \rho_i = \frac{|\Omega|}{\sum_{d \in \Omega} w_{d,i}}$$

On peut également utiliser dans ce cadre la divergence de Kullback-Liebler (ou entropie relative), qu'il est possible de symétriser pour servir de distance, ou bien encore la divergence de Jensen-Shannon (ou divergence totale à la moyenne).

Une présentation plus développée de ces différentes mesures, ainsi que des liens et des relations qu'elles entretiennent les unes avec les autres, peut être trouvée dans (Lee, 1997 ; Rajman & Lebart, 1998 ; Besançon, 2001).

5.1.3 Évaluation des performances des SRI

La notion de pertinence, dont les différentes approches proposées depuis les débuts de la recherche documentaire automatique sont présentées dans l'état de l'art très complet de S. Mizzaro (1997), est au cœur de l'évaluation des SRI. Malheureusement, c'est une notion hautement subjective et dépendant de nombreux paramètres externes au système testé.

Pour rendre compte de cette pertinence, de nombreuses mesures ont été proposées. Elles s'appuient sur des hypothèses fortes permettant d'écarter certaines causes de subjectivité dans l'analyse des résultats. Nous posons dans la partie suivante les hypothèses sur lesquelles nous nous appuierons pour évaluer les performances de notre technique d'extension de requêtes. La section 5.1.3.2 définit les notions de taux de rappel et de précision et leurs variantes utilisées lors des campagnes d'évaluation des SRI. La dernière partie présente les tests statistiques que nous utilisons lors de nos expériences pour vérifier la portée réelle des résultats observés.

5.1.3.1 Hypothèses

Lors de l'évaluation des SRI comme cela est fait pendant les campagnes d'évaluation telles que TREC, un certain nombre d'hypothèses sont faites. Avant de détailler les mesures d'évaluation et les performances de l'extension de requêtes, il convient de rappeler ces hypothèses dans lesquelles nous plaçons l'évaluation de notre propre technique d'extension de requêtes.

Jugement total

Une des hypothèses les plus importantes est de supposer qu'il est possible de juger de la pertinence, étant donnée une requête quelconque, de tous les documents de la collection. Cette hypothèse permet ainsi de construire une liste des « bonnes » réponses pour chaque requête, à laquelle les systèmes de recherche se rapportent lors de l'évaluation pour mesurer leurs performances. Ainsi, dans un cadre binaire (voir ci-après), cela signifie que n'importe quel document peut être classé soit pertinent, soit non pertinent. Le jugement de pertinence émane donc d'une source dont l'autorité n'est pas remise en question.

Jugement binaire

Pour l'évaluation des systèmes, on se place souvent dans un cadre binaire dans lequel un document ne peut être que soit pertinent, soit non pertinent. Aucune graduation n'est prise en compte, un document pertinent a donc nécessairement égale importance qu'un autre document pertinent.

Le jugement binaire est hérité des systèmes fonctionnant selon un modèle ensembliste. Il a été conservé car il permet l'utilisation de mesures d'évaluation simples telles que le rappel et la précision (voir ci-dessous), bien que les systèmes actuels utilisent désormais d'autres modèles de représentation (vectoriels par exemple) dans lesquels un score est assigné à un document et non une décision de pertinence ou de non pertinence.

Absence d'additivité

Lors de l'analyse des documents retournés à l'utilisateur, on considère que ceux-ci sont examinés de manière indépendante. Cela signifie que la pertinence d'un document ne dépend pas des autres documents répondant à la requête. En particulier, deux documents jugés indépendamment non pertinents restent non pertinents même s'ils se complètent et forment ensemble une information répondant à la requête.

Absence de mémoire

Une autre hypothèse liée à l'indépendance des documents présentés en réponse à une requête est l'absence de mémoire. Elle se traduit par le fait qu'un document reste pertinent même si un autre document au contenu similaire a déjà été présenté à l'utilisateur. Cette notion s'applique également pour les requêtes puisqu'une seule requête est considérée à la fois, indépendamment de celles déjà posées par l'utilisateur. L'interrogation se fait donc en une fois et les résultats ne sont pas conservés pour les requêtes suivantes.

Ces hypothèses sont bien sûr discutables — on leur reproche notamment de ne pas tenir compte du contexte de la recherche, du sujet (aspect conceptuel) de la recherche et de la tâche (utilisation des documents retournés) — et certains travaux s'en affranchissent en partie (Mizzaro, 1998). De nombreuses recherches montrent notamment l'irréalisme de la première hypothèse qui est souvent démentie par les difficultés à obtenir un accord complet entre les personnes chargées de construire ces jugements (Salton & Lesk, 1968 ; Burgin, 1992 ; Schamber, 1994). Cependant, certaines de ces études soulignent que les différences entre annotateurs semblent ne pas influencer excessivement le résultat de la comparaison des systèmes de recherche (Salton & Lesk, 1968 ; Voorhees, 1998 ; Burgin, 1992).

Dans la suite de ce chapitre, nous considérons néanmoins que l'ensemble des hypothèses précédentes est avéré. Cela nous permettra de nous placer dans le même cadre d'évaluation que les campagnes Amaryllis dont nous utilisons les données.

5.1.3.2 Rappel, précision et variantes

Les taux de rappel et de précision, déjà présentés en section 3.3, sont à la base d'un ensemble de mesures très utilisées en recherche d'information pour évaluer les systèmes. Ils sont issus d'une interprétation ensembliste de la pertinence dans laquelle un document appartient soit à l'ensemble des documents pertinents soit à son complémentaire, et fait soit partie de l'ensemble de documents retournés à l'utilisateur en réponse à sa requête, soit pas. Il est possible d'illustrer ces notions à partir du schéma 5.1, dans lequel, étant donné une requête, Ω est l'ensemble des documents contenus dans la base documentaire, Σ l'ensemble des documents pertinents et Δ l'ensemble des documents présentés à l'utilisateur.

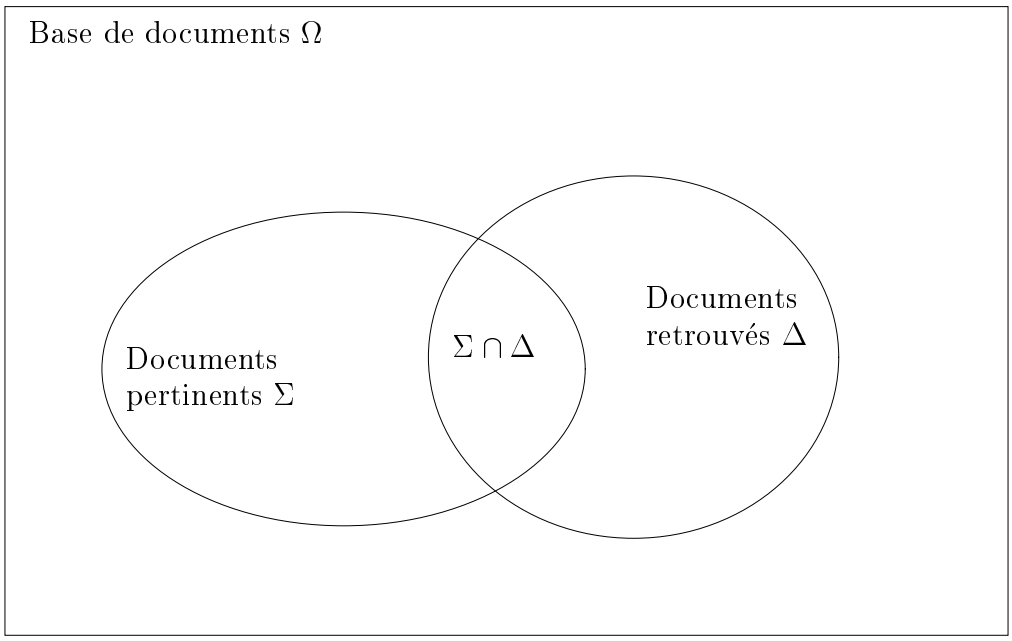


FIG. 5.1 – Interprétation ensembliste du rappel et de la précision

Ainsi, les taux de rappel et de précision peuvent s'exprimer sous forme ensembliste par :

$$R = \frac{|\Delta \cap \Sigma|}{|\Sigma|} \quad \text{et} \quad P = \frac{|\Delta \cap \Sigma|}{|\Delta|}.$$

On définit également de manière duale à l'aide des mêmes opérations sur les ensembles les taux de bruit B (*noise rate*), de silence S (*silence rate*) et de résidu ou chute C (*fallout rate*) :

$$B = 1 - P = \frac{|\Delta \setminus \Sigma|}{|\Delta|}, \quad S = 1 - R = \frac{|\Sigma \setminus \Delta|}{|\Sigma|} \quad \text{et} \quad C = \frac{|\Delta \setminus \Sigma|}{|\overline{\Sigma}|}$$

où $\overline{\Sigma}$ est le complémentaire de l'ensemble Σ dans Ω. On peut enfin définir les notions de densité D et de généralité G qui représentent en quelque sorte la rareté des documents

pertinents dans la base, et donc la difficulté qu'aura un système pour les retrouver :

$$D = \frac{|\Sigma|}{|\Omega|} \quad \text{et} \quad G = 1 - D = \frac{|\overline{\Sigma}|}{|\Omega|}.$$

Les *document cut off values*

Dans le modèle vectoriel, étant donnée une requête, ce sont des scores qui sont assignés aux documents et non pas une décision de pertinence ou de non pertinence. Pour calculer les taux de rappel et de précision, il est donc nécessaire de convertir ces scores en jugement binaire. Cela se fait usuellement en choisissant un seuil de score tel que les documents obtenant un score supérieur au seuil sont considérés comme pertinents et ceux obtenant un score inférieur comme non pertinents.

En pratique, le choix du seuil se fait de manière indirecte en fixant le nombre de documents qui doivent être considérés comme pertinents. Ce nombre de documents est généralement noté DCV pour *document cut off value*. Les mesures présentées précédemment, en particulier le rappel et la précision, sont donc calculées en fonction du DCV. Les valeurs les plus communes sont 5, 10, 15, 20, 30, 50, 100, 200, 500, 1000, 2000, 5000 documents. Par exemple, si le DCV est fixé à 50, la précision, que l'on note alors P(50) est le taux de bons documents dans les 50 documents obtenant les plus hauts scores pour une requête donnée, et le rappel R(50) est le nombre de bons documents dans ces mêmes 50 documents divisé par le nombre total de bons documents dans Ω.

La R-précision (R-Prec) est un cas spécial des mesures évaluées par seuil. En effet, cette dernière représente la précision du système, étant donnée une requête, avec comme DCV le nombre de documents pertinents dans la base, c'est-à-dire avec les notations précédentes R-prec = P($|\Sigma|$).

Mesures globales

La précision moyenne interpolée (IAP) est une mesure décrivant la précision globale du système évalué sur une requête. Pour ce faire, la précision des résultats est calculée sur onze points, correspondant aux DCV pour lesquels le rappel vaut 0, 10, 20, 30, 40, 50, 60, 70, 80, 90 et 100%. Si ces points ne sont pas effectivement atteints en fixant un DCV, les mesures sont alors interpolées. La moyenne de ces 11 précisions forment la précision moyenne interpolée.

De manière duale, on définit la précision moyenne non interpolée (NIAP) comme étant la moyenne des précisions obtenues pour tous les DCV correspondant au rang d'un bon document dans la liste des réponses. Cela se traduit donc, avec les notations précédentes, par la formule suivante :

$$NIAP = \frac{\sum_{d \in \Sigma \cap \Delta} P(rang(d))}{|\Sigma|}$$

D'autres mesures globales utilisées en recherche d'information existent (par exemple la F-mesure définie en section 3.3.2.2). Elles ne seront pas utilisées pour les évaluations décrites ci-après ; nous ne les décrivons donc pas.

5.1.3.3 Tests statistiques

Les mesures d'évaluation sont pour certaines d'entre elles très sensibles (Buckley & Voorhees, 2000) ; un seul changement de paramètre peut les faire varier avec une grande amplitude. Or, on veut être sûr, lors des expérimentations menées, que les améliorations (ou dégradations) constatées ne sont pas le fait du hasard. On veut en particulier qu'elles ne soient pas dues au choix du jeu de requêtes (Nelson, 1995) et qu'elles peuvent donc se généraliser à tout le système. Il est donc important de vérifier que chacune des mesures prises est statistiquement significative.

L'évaluation des SRI se ramène souvent à une comparaison de deux systèmes dont l'un sert d'étalon (*baseline*) et l'autre est le SRI à évaluer. C'est le cas notamment lorsque l'on veut mettre en évidence l'influence d'une modification plus ou moins importante d'un système : les résultats de la version modifiée sont comparés à ceux de la version originale. Si l'on dispose d'une collection de test avec un jeu de requêtes, les performances des deux systèmes (évaluées selon une des mesures présentées précédemment, notée m) seront mises en correspondance deux à deux comme illustré ci-dessous :

Numéro de la requête	Mesure du système original	Mesure du système modifié
1	m_1^1	m_1^2
2	m_2^1	m_2^2
...	...	...
n	m_n^1	m_n^2

Il est alors possible d'étudier la variation des résultats suivant les requêtes et de juger si la performance globale (traduite par la moyenne des performances sur chaque requête) est statistiquement significative en observant les différences de performances requête par requête. On note dans la suite δ_i la différence entre la performance du système modifié et celle du système original pour la requête i (c'est-à-dire $\delta_i = m_i^2 - m_i^1$) et $\bar{\delta}$ la moyenne de ces différences.

Le fonctionnement de ces tests est similaire à celui d'une démonstration par l'absurde : on fait dans un premier temps l'hypothèse que les différences observées sont dues au hasard, et donc non significatives. Cette hypothèse est appelée l'hypothèse nulle et notée H_0. Les tests que nous allons utiliser cherchent à évaluer le risque de première espèce, c'est-à-dire la probabilité que H_0 soit vraie étant données les différences constatées. Plus cette probabilité est petite, plus on peux rejeter H_0 au profit d'une hypothèse alternative statuant que les différences ne sont pas dues au hasard mais reflètent une réelle différence de performances entre les deux systèmes.

Parmi les tests statistiques existants, plusieurs se prêtent à l'étude de telles données en paires et sont effectivement utilisés, par exemple dans SMART (Salton, 1971). Le

plus célèbre dans ce cadre est dérivé du *t-test* de Student : il s'agit du *paired t-test*. Dans ce dernier, on calcule la probabilité p_S que H_0 soit vraie sachant les données expérimentales grâce à la grandeur t (où n est le nombre de requêtes) :

$$t = \frac{\bar{\delta}}{s(\delta_i)/\sqrt{n}} \quad \text{avec} \quad \bar{\delta} = \frac{1}{n}\sum_{i=1}^{n}\delta_i \quad \text{et} \quad s(\delta_i) = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(\delta_i - \bar{\delta})^2}.$$

Sous H_0, t suit une distribution de Student avec $n-1$ degrés de liberté.

Ce test est très performant, mais c'est un test paramétrique qui suppose que la distribution des δ_i suive une loi normale. C'est une hypothèse très forte qui est souvent violée en pratique par les données (Savoy, 1997). Cependant, ce test est relativement tolérant de ce point de vue et se comporte bien pour des violations légères de l'hypothèse (Hull, 1993).

Le test non paramétrique de Wilcoxon (*paired Wilcoxon test*), jugé plus robuste, est souvent préféré en recherche d'information (van Rijsbergen, 1979). La seule assomption faite pour valider son fonctionnement est que les δ_i soient issues d'une distribution continue, symétrique et centrée sur 0. La probabilité p_W que H_0 soit vraie sachant les données expérimentales est calculée grâce à la grandeur t qui doit suivre une distribution normale et est définie par :

$$t = \frac{\sum_{i=1}^{n} R_i}{\sqrt{\sum_{i=1}^{n} R_i^2}} \quad \text{avec} \quad R_i = signe(\delta_i) * rang(|\delta_i|)$$

où $signe(\delta_i)$ renvoie 1 si la différence δ_i est positive et -1 sinon, et $rang(|\delta_i|)$ indique le rang occupé par $|\delta_i|$ dans le classement de la plus petite à la plus grande valeur absolue des différences.

Ce test de Wilcoxon requiert moins de conditions sur les données que le *paired t-test* (cependant, tous deux imposent que les requêtes soient indépendantes, bien que là encore, de légères violations à cette règle n'influencent pas trop les résultats). Mais il est également moins sensible : il ne réagit pas assez vite à des différences significatives. Nous utilisons donc lors de nos expérimentations ces deux tests en indiquant les ordres de grandeur des valeurs respectives de p_S et p_W. Nous considérons que l'hypothèse H_0 peut être rejetée si sa probabilité est inférieure à 0.1.

5.2 Apport de ressources sémantiques

Dans leur grande majorité, les systèmes de recherche d'information n'exploitent pas d'informations de nature linguistique. Lorsqu'ils y ont néanmoins recours, c'est généralement pour prendre en compte des connaissances de bas niveau (morphologiques,

syntaxiques). Or, les problèmes de la recherche d'information — comment décrire le contenu des textes, comment identifier le contenu d'une requête dans un texte ? — sont essentiellement d'ordre sémantique, voire conceptuel.

Le recours à des informations linguistiques dans les applications de recherche d'information vise principalement deux objectifs : définir des descripteurs de contenu correctement discriminants et non ambigus (index complexes, index structurés syntaxiquement (Sparck-Jones, 1999)) ; mettre en rapport des formulations différentes mais sémantiquement proches afin d'élargir la portée des index contenus dans les textes et augmenter ainsi les chances d'apparier la requête et les textes de la base (Debili *et al.*, 1989 ; Smeaton, 1999).

Dans le premier cas, les informations linguistiques mobilisées sont de nature essentiellement morphosyntaxique (segmentation de groupes — *chunking* —, identification de dépendances). Plusieurs travaux ont été effectués dans ce domaine (voir par exemple (Jacquemin *et al.*, 1997 ; Strzalkowski, 1999)) que nous ne présentons pas dans le cadre de ce mémoire. Le second objectif suppose en revanche la prise en compte de ressources linguistiques de plus haut niveau. Ces ressources peuvent être préexistantes à la base — ce sont des bases généralistes comme WordNet — ou directement issues des textes composant la collection. Quelle que soit leur origine, elles peuvent être intégrées dans le système de recherche de deux façons différentes : soit lors de la construction de l'index (voir par exemple les travaux de (Schütze & Perdersen, 1994) et (Besançon, 2001 ; Rajman *et al.*, 2000)), soit lors de l'interrogation de la base par reformulation de requêtes.

C'est cette dernière approche, plus simple à mettre en œuvre, que nous avons choisie pour vérifier l'intérêt des relations qualia en RI. Nous présentons dans la section suivante le cadre général des travaux portant sur l'extension de requêtes par ressources sémantiques. Puis nous examinons l'intérêt des liens sémantiques nom-verbe, et plus particulièrement des relations qualia dans ce cadre.

5.2.1 Extension de requêtes par ressources sémantiques

L'augmentation des performances des systèmes de recherche d'information, principalement en termes de rappel, passe notamment par le traitement du phénomène d'équivalence sémantique. Un même contenu peut être exprimé de manière différente, dans différentes configurations syntaxiques, avec différents mots. Le diagnostic de paraphrase, dès lors qu'il dépasse le cadre strict de la transformation syntaxique, est extrêmement difficile à contrôler et requiert des informations linguistiques riches.

5.2.1.1 Utilisation de ressources externes

La démarche la plus souvent adoptée consiste à recourir à une base de connaissances linguistiques regroupant les mots sémantiquement proches, et structurée selon des relations hyperonymiques ou synonymiques. Les index de la requête peuvent par exemple être automatiquement propagés en suivant les liens exprimés dans la base lexicale,

de manière à disposer d'une description étendue de cette requête (*query expansion*). C'est l'option choisie par exemple à FT-R&D pour la consultation du Minitel en français (Gilloux *et al.*, 1993). On connaît le coût de construction de telles ressources, qui amène généralement ceux qui adoptent cette approche à plaider pour l'utilisation de ressources générales, mutualisables, dont WordNet (Fellbaum, 1998) constitue le modèle (Smeaton, 1999). Le gain apporté par le recours à de telles ressources n'a toutefois pas été démontré jusqu'à présent. Certaines expériences tendent même à invalider cette approche (Voorhees, 1994).

Deux aspects de cette démarche expliquent essentiellement ses limites : tout d'abord, elle fait l'hypothèse d'une ressource lexicale générale valable hors contexte. Or, nous avons vu en section 1.1.2.1 les limites de l'utilisation de telles bases généralistes sur des domaines particuliers. On ne peut en effet pas savoir dans quelle mesure un modèle sémantique conçu *a priori* s'avère adéquat pour représenter le fonctionnement de domaines particuliers. Or, étendre la requête consiste précisément à tenter de la rapprocher des documents qu'elle cherche à explorer, en d'autres termes, à l'ancrer dans les mots réellement utilisés dans le corpus. En second lieu, il manque à l'approche par thésaurus une réflexion linguistique préalable concernant le fonctionnement sémantique des descripteurs. Elle mobilise en effet exclusivement les relations lexicales traditionnelles (hyperonymie, synonymie). Cette option témoigne d'une vision très cloisonnée du lexique. Ainsi A. Smeaton (1999) déclare n'exploiter de WordNet que les noms, ceux-ci étant les principaux détenteurs du contenu des textes. Or, s'il est prouvé que les groupes nominaux constituent le principal mode d'expression des descripteurs, l'apport sémantique d'autres catégories de mots tels que les verbes ne doit pas être négligé pour réaliser l'enrichissement et la reformulation des index.

5.2.1.2 Utilisation de ressources internes

En alternative à l'utilisation de ressources généralistes, certains travaux dérivent directement de la collection de documents les connaissances sémantiques ensuite exploitées dans le processus de recherche.

L'utilisation de cooccurrences a notamment fait l'objet très tôt de plusieurs travaux (Lesk, 1969 ; van Rijsbergen, 1977). L'idée sous-jacente sur laquelle ils s'appuient est que tout terme étroitement lié à un terme d'indexation peut lui-même être utilisé comme terme d'indexation. En pratique, ces termes « étroitement liés » sont calculés à partir des cooccurrences fréquentes des mots, par des méthodes essentiellement numériques (voir section 1.2). Un thésaurus est ainsi construit ; lors d'une interrogation, aux termes de la requête sont alors ajoutés les éléments du thésaurus qui leur sont proches, soit en considérant chaque mot de la requête indépendamment, soit en considérant l'ensemble de la requête (Qiu & Frei, 1993 ; Qiu & Frei, 1995).

L'efficacité de ces approches d'extension de requêtes par cooccurrents est variable selon les travaux mais aucune amélioration franche des résultats ne semble se dégager quelle que soit la collection de documents. H. Peat & P. Wilett (1991) expliquent ce phénomène par le fait que les méthodes utilisées pour l'extraction des cooccurrences favorisent l'acquisition de termes approximativement de même fréquence. Or les termes

de la requête étant eux-mêmes très fréquents, les termes ajoutés sont eux aussi trop fréquents pour être discriminants.

Notons enfin que les travaux se plaçant dans le domaine du retour de pertinence (*relevance feeback*), initiés par J. Rocchio (1971), peuvent également s'interpréter comme une utilisation de ressources sémantiques internes pour l'expansion de requêtes. En effet, le principe de ces travaux (Salton & Buckley, 1990) est d'exploiter les documents retournés en réponse à une requête pour améliorer dans un second temps le résultat de la recherche. En particulier, cela peut se faire en extrayant des documents ramenés par la requête originale de nouveaux termes utilisés à leur tour pour interroger la base. Les techniques utilisées pour l'extraction de ces nouveaux termes sont variées mais peuvent se rapprocher de celles utilisées pour la construction automatique des thésaurus citées précédemment.

5.2.2 Relations qualia et recherche d'information

Dans notre contexte d'extension par verbes sémantiquement liés à des noms, le formalisme du Lexique génératif a été choisi pour le cadre formel qu'il propose pour ce type de lien à travers les relations qualia. L'acquisition de ces relations sur la collection de textes constituant la base documentaire doit ainsi mettre au jour des liens sémantiques pertinents et attestés entre noms et verbes. Nous plaçons donc notre approche dans la famille vue ci-dessus des travaux effectués en expansion de requête utilisant des ressources internes à la base documentaire traitée. Nous nous en distinguons cependant par les relations sémantiques originales sur lesquelles nous nous concentrons. Plus précisément, nous cherchons à valider les hypothèses suivantes :

- les ressources lexicales sémantiques permettent d'améliorer les performances d'un système de recherche d'information ;
- les relations sémantiques entre noms et verbes, jugées intéressantes pour la RI par certains auteurs, le sont effectivement ;
- ces relations doivent être acquises en contexte à partir de corpus du domaine.

Ce dernier item est porté par deux raisons majeures. Outre le fait que, comme nous l'avons déjà souligné, de telles connaissances linguistiques d'ordre sémantique dépendent fortement du domaine des textes traités, les relations sémantiques dont nous voulons tirer parti, les liens N-V, sont en grande partie absentes des ressources lexicales existantes. Ces ressources, même lorsqu'elles sont spécialisées, exploitent en effet massivement des relations lexicales traditionnelles (hyperonymie, *etc.*) et n'intègrent pas le type de lien auquel nous nous intéressons. Nous proposons d'évaluer si l'enrichissement des index peut passer par des liens de nom à verbe (*disque dur - stocker* ; *lettre - communiquer*) et non pas seulement par des rapports tels que la synonymie ou l'hyperonymie, exprimant des liens de nom à nom (*e.g. disque dur - mémoire* ou *message - lettre*). Enfin, notre technique d'acquisition, présentée dans les chapitres précédents, s'appuyant sur une approche essentiellement symbolique, doit permettre de dépasser les limites des techniques numériques évoquées par H. Peat et P. Wilett (1991).

Nous examinons dans la sous-section suivante les travaux considérant le lien N-V dans une perspective de recherche d'information. Nous nous intéressons ensuite plus spécifiquement aux relations de type qualia et motivons par des considérations théoriques et pratiques l'intérêt de ces relations pour la reformulation de requêtes.

5.2.2.1 Exploitation du lien nomino-verbal

Si l'on souhaite définir le plus complètement possible la sémantique d'un nom, on ne peut pas se limiter aux liens intra-catégoriels qu'il entretient ; il faut aussi explorer les liens N-V qui dévoilent d'autres facettes de son sens. Ces liens inter-catégoriels permettent en particulier d'accéder, à l'instar des liens N-N, à des formulations différentes mais sémantiquement équivalentes du concept exprimé par le nom N. L'importance des relations N-V pour la définition de la sémantique des noms a d'ailleurs été soulignée par plusieurs travaux menés en terminologie, analyse et typologie des textes (Bourigault & Condamines, 1999 ; Klavans & Kan, 1998).

Une expérience de C. Fabre & C. Jacquemin (2000) vise à prendre en compte la variation verbo-nominale des termes afin d'exploiter ce type de lien entre des termes nominaux (ex : *méthode d'obtention*) et des formulations verbales proches (ex : *obtenues par d'autres méthodes*). Ce travail constitue une première étape vers la prise en compte de critères de reformulation sémantique pour exploiter la relation nom-verbe. Le but de cette expérience est d'augmenter l'ensemble des catégories de variation terminologique traitées par FASTR (Jacquemin *et al.*, 1997). Dans cette expérience, seule la relation nom-verbe validée par un lien morphologique est prise en compte. Nous proposons d'étendre le traitement de cette relation inter-catégorielle au cas d'associations nom-verbe sans lien morphologique et de tester l'influence de ce lien en RI.

Cette hypothèse selon laquelle les liens sémantiques entre noms et verbes peuvent être exploités en RI n'est certes pas nouvelle. G. Grefenstette, par exemple, souligne l'importance de tels liens syntagmatiques pour aider à préciser et à désambiguïser les noms contenus dans des requêtes courtes (Grefenstette, 1997). Il montre ainsi qu'un moyen de caractériser sémantiquement un nom comme *research* est d'extraire l'ensemble des verbes utilisés avec ce nom, de manière à recenser ce que la recherche permet de faire (*research show*, *research reveal*, *etc.*) et ce qui est fait en direction de la recherche (*do research*, *support research*, *etc.*). Nous proposons, pour notre part, un moyen de systématiser cette approche utilisant les liens N-V en RI et de nous donner un critère pour définir les paires pertinentes : nous ne retenons, parmi les paires N-V possibles pour la reformulation sémantique, que celles qui sont décrites dans la structure des qualia du Lexique génératif, reprennant en cela l'idée développée dans P. Bouillon (2000b).

5.2.2.2 Pertinence du lien N-V qualia

Jusqu'ici, l'objectif principal de J. Pustejovsky a surtout été de montrer que les structures des qualia, telles qu'elles sont définies dans le Lexique génératif (Pustejovsky, 1995 ; Bouillon & Busa, 2001), permettent de représenter adéquatement la polysémie lo-

gique des expressions linguistiques, entre autres les différences dans la forme syntaxique d'un argument (exemple 5.1).

Exemple 5.1 (Différences dans la forme syntaxique)

a. je commence le livre (+NP)

b. je commence à lire le livre (+VP)

Nous proposons d'évaluer l'intérêt de l'utilisation des représentations proposées pour gérer ce type d'expressions métonymiques pour prédire les prédicats pertinents du point de vue de la recherche d'information. Par exemple, l'exploitation de ces représentations riches permet de dépasser certaines différences de formulation; il est ainsi possible d'apparier les phrases *a* et *b* de l'exemple 5.1.

Sur le plan pratique, la validité de l'approche consistant à exploiter les relations qualia en RI a déjà été partiellement testée. Tout d'abord, C. Fabre (1996) a montré que les couples qualia pouvaient être utilisés pour calculer la représentation sémantique de séquences binomiales (voir section 2.3.2.1), et offrent ainsi des possibilités étendues de reformulations des index composés. C. Fabre & P. Sébillot (1999) ont exploité ces relations sémantiques au sein d'un service télématique (annuaire du Minitel).

Une autre expérience a également été menée dans un service de documentation d'une banque belge (Vandenbroucke, 2000). Les documentalistes de ce service utilisent traditionnellement des requêtes booléennes avec des termes nominaux. Il leur a été demandé d'évaluer, de manière complètement qualitative, la pertinence des verbes qualia pour leur tâche de recherche. Pour ce faire, à chaque requête, les verbes qualia associés aux noms présents dans leur question leur étaient proposés pour spécifier leur recherche. Il a été montré que cela leur permettait d'accéder à certains documents auxquels ils n'auraient pas penser et pas pu obtenir avec la requête originale. Bien que purement qualitative et effectuée sur un faible nombre de documentalistes ces résultats sont encourageants et abondent dans le sens de l'utilité d'associer des verbes qualia à des requêtes au sein d'un système de recherche documentaire.

D'autres travaux, relevant plus de l'indexation et de l'aide à la recherche abondent également dans ce sens. Ainsi, il a été montré (Pustejovsky *et al.*, 1997) que les structures des qualia peuvent aussi servir à alimenter une *toile lexicale* (*lexical web*), c'est-à-dire un réseau de termes pertinents et de relations qui, ensemble, définissent le sujet d'un texte, à la manière d'un index traditionnel. Ce réseau, mis à plat et présenté comme l'index d'un livre, permet à l'utilisateur de naviguer dans le texte. Pour *disk*, par exemple, J. Pustejovsky *et al.* (1997) proposent l'ensemble des relations N-V représentée en figure 5.2 qui, selon eux, définissent extensionnellement le sens du mot dans le domaine traité (la documentation technique de Macintosh, *Macintosh Reference*).

5.3 Extension de requêtes par couples qualia

Comme nous venons de le voir, les liens sémantiques entre noms et verbes, et plus spécialement les liens N-V qualia, semblent particulièrement intéressants dans

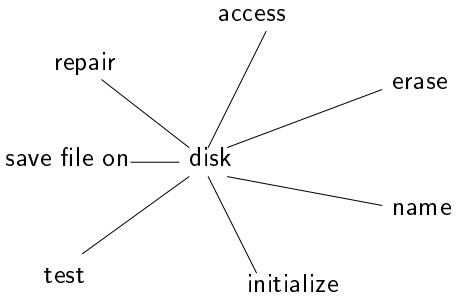


FIG. 5.2 – Toile lexicale de disk selon J. Pustejovsky *et al.* (1997)

un contexte de recherche d'information. Nous nous proposons dans cette partie d'en vérifier expérimentalement l'intérêt en se servant de couples N-V en relation qualia pour étendre des requêtes et ainsi mesurer l'impact sur la pertinence des documents retournés.

Nous présentons dans la section suivante le protocole selon lequel se déroule ces expérimentations. Nous décrivons ensuite plus spécifiquement en section 5.3.2 la mise-en-œuvre de l'extension de requêtes à l'aide des relations qualia acquises automatiquement par ASARES. Enfin, nous examinons et discutons les résultats obtenus par cette technique d'extension sur notre collection de test dans la section 5.3.3.

5.3.1 Protocole expérimental

Le protocole expérimental que retenons pour nos expérimentations se veut le plus usuel possible. La première sous-section présente le cadre technique adopté, et plus particulièrement le SRI utilisé et ses différents paramétrages. Nous détaillons ensuite les caractéristiques de la collection de test sur laquelle nous évaluons l'intérêt de l'extension de requêtes par couples qualia. Nous exposons enfin le mode de constitution des couples qualia devant être exploités pour cette extension.

5.3.1.1 Système de recherche

Le système de recherche d'information que nous utilisons pour nos expérimentations est SMART, le système développé par G. Salton (1971) pour mettre en œuvre ses idées concernant le modèle vectoriel. Il est encore très utilisé de nos jours et sert de référence pour évaluer les systèmes plus récents. Ce SRI est donc particulièrement indiqué dans notre cas puisque nous voulons montrer l'intérêt de l'extension de requêtes par couples qualia dans les conditions les plus usuelles possibles.

Parmi les différents schémas de pondération qu'offre ce système (voir section 5.1.2.2), nous utilisons le schéma de pondération *ltc.ltc*. Ce schéma est l'un des plus communément utilisé; il correspond à une pondération pour les requêtes et les documents conjuguant une pondération locale en logarithme de la fréquence, à une

pondération globale équivalente à la fréquence documentaire inverse et à la normalisation du cosinus.

5.3.1.2 Collection de test

Les données utilisées lors de nos expérimentations sont issues de la campagne d'évaluation des systèmes de recherche d'information Amaryllis (Coret *et al.*, 1997 ; Landi *et al.*, 1998). Les données de cette campagne se décomposent en deux parties contenant chacune un corpus, un jeu de requêtes et leurs réponses : l'une sert pour entraîner les systèmes — ce sont les données d'entraînement — et l'autre sert au test (nouvelles requêtes sur un corpus connu, requêtes connues sur de nouveaux documents, nouvelles requêtes sur de nouveaux documents). Notre expérience d'extension de requêtes et les paramétrages de notre SRI ne reposant pas sur une méthode d'apprentissage, les données que nous utilisons effectivement sont celles correspondant à la seule phase d'entraînement de la campagne Amaryllis.

Nous travaillons donc sur un corpus fournis par l'OFIL (Observatoire français des langues) regroupant des articles du journal Le Monde sur une période de trois mois, d'un jeu de requêtes et de leurs réponses. Ainsi, ce sont 11 016 articles (chacun représente un document et est identifié par un numéro unique) qui composent notre base documentaire Ω (voir les exemples donnés en annexe C.1). Pour chacune des 26 requêtes du jeu de test, une liste non nulle des identificateurs des documents pertinents est fournie.

Les requêtes ont été construites à partir de questions réelles d'utilisateurs et sont structurées en un domaine général, un titre, une question détaillée en langage naturel, une explication des documents attendus en réponse, et d'une liste de concepts (sous forme de mots-clés) proches (voir un extrait de ces requêtes en annexe C.2). Les réponses aux requêtes servant de référence pour l'évaluation ont été établies par un groupe de juges humains.

5.3.1.3 Constitution de la collection de couples qualia

Une collection de couples qualia nécessaires à l'extension de requêtes est construit à partir du corpus constitué de tous les documents de Ω. Pour cela, nous utilisons le système ASARES dans sa version semi-supervisée intégrée décrite en section 4.2.2.2. L'ensemble des articles a donc été étiqueté ; à la différence du corpus MATRA-CCR, nous avons utilisé cette fois-ci l'outil CORDIAL ANALYSEUR[2] et aucun étiquetage sémantique n'a été réalisé. Une première phase d'extraction statistique, utilisant cette fois-ci le coefficient du loglike, a ensuite été menée et a servi à définir deux ensembles pondérés d'exemples positifs et négatifs. La phase d'apprentissage sur ces deux ensembles a permis d'inférer des patrons morphosyntaxiques qui ont ensuite été appliqués aux articles pour en extraire les couples qualia. Les couples extraits sont rassemblés dans une base lexicale avec leur nombre de détections (nombre d'occurrences trouvées par les patrons inférés).

[2] CORDIAL ANALYSEUR est un produit de la société Synapse Développement ; http:www.synapse-fr.com.

5.3.2 Description de la méthode d'extension des requêtes

Nous présentons dans cette partie la mise en œuvre de la phase d'extension de requêtes adoptée dans nos expériences. Nous examinons en premier lieu diverses considérations nous amenant à faire certains choix quant à la taille des requêtes que nous utilisons. Nous détaillons ensuite les différents paramètres intervenant dans l'ajout des verbes qualia aux requêtes originales.

5.3.2.1 Besoins réels et taille des requêtes

Les systèmes de recherche documentaires sont très largement utilisés par le grand public, principalement à travers les moteurs de recherche, généralistes ou spécialisés, sur le Web. Beaucoup de ces utilisateurs sont novices et possèdent peu de connaissances sur les principes de la recherche d'information, ce qui se traduit en pratique par une utilisation très particulière des systèmes de recherche.

En ce qui concerne l'interrogation, l'ensemble des études effectuées à partir des *logs* de moteurs de recherche du Web montre une utilisation extrêmement pauvre des fonctionnalités proposées par les systèmes. Cela se manifeste par des requêtes très courtes, en moyenne inférieure à deux mots (Jensen *et al.*, 1998 ; Jensen *et al.*, 2000 ; Silverstein *et al.*, 1998), et une quasi absence d'opérateurs booléens ou de pondérations lorsque ceux-ci existent.

La phase d'examen des résultats est elle aussi symptomatique d'une utilisation minimaliste des moteurs par la majorité de leurs utilisateurs. En effet, selon ces mêmes études, parmi les documents retournés par le système en réponse à une requête, un utilisateur ne consulterait en moyenne que les 10 à 20 premiers.

Sans se placer explicitement dans un cadre de recherche documentaire de type Web ou grand public, il est important de considérer ces faits lors de la construction et de l'évaluation des systèmes de recherche. Cela a notamment été fait dans les campagnes de type TREC par la création des évaluations *short query*. Pour notre part, certains des paramètres de nos expériences sont fixés pour correspondre au mieux à des utilisations réelles d'un SRI. Ainsi, rejoignant les conclusions de C. de Loupy & M. El-Bèze (2002), les requêtes que nous utilisons sont uniquement composées des titres des requêtes Amaryllis. Par exemple, la requête 1 (*cf.* annexe C.2), non étendue, utilisée lors des expérimentations décrites ci-dessous, est : La séparation de la Tchécoslovaquie. Le nombre de mots pleins utilisés est ainsi plus proche d'une utilisation du système de recherche dans des conditions réelles et ouvertes.

5.3.2.2 Extension des requêtes

Choix des éléments d'extension

Pour tester l'apport des verbes qualia à la recherche documentaire, nous proposons d'étendre chaque requête avec les verbes qualia correspondant aux noms communs

présents dans cette requête. La stratégie utilisée en pratique pour réaliser cette extension est très simple :

- le nombre maximum de verbes ajoutés par nom, noté Nb_V, est fixé (nous étudions l'effet de Nb_V sur les performances du système en section 5.3.3.2) ;
- tous les noms présents dans la requête sont candidats à l'extension ;
- pour un nom fixé, les Nb_V verbes qualia choisis dans la collection de couples sont ceux ayant été détectés le plus de fois par ASARES.

Cette extension de requête se fait donc en considérant les noms de la requête de manière disjointe, mais prend en compte, à travers le choix des verbes détectés le plus souvent, une sorte de degré de certitude fourni par notre système d'acquisition.

Composition de la requête étendue

La requête étendue se compose donc des termes de la requêtes originale et des verbes qualia. Cette requête étendue ne remplace pas l'ancienne requête ; cette dernière est en effet également utilisée lors de la recherche grâce aux mécanismes de sous-vecteurs proposés dans l'extension du modèle vectoriel de E. Fox (1983). Dans ce modèle, les requêtes sont composées de sous-vecteurs, chacun de ces vecteurs pouvant représenter un type d'information différent appelé *ctype* pour *concept type*.

La similarité d'un document et d'une requête est la somme pondérée des similarités selon chaque sous-vecteur. En notant C l'ensemble des *ctypes* et Q_i le sous-vecteur de Q correspondant au *ctype* i, cela se traduit par :

$$sim(Q, D) = \sum_{i \in C} \alpha_i * sim(Q_i, D)$$

où les α_i sont les poids associés à chaque sous-vecteur, représentant l'influence que l'on souhaite donner au sous-vecteur dans le calcul similarité.

Cette extension du modèle vectoriel standard est particulièrement adaptée à la manipulation d'extensions ou de reformulations. E. Voorhees (1994) l'utilise par exemple dans ses expériences consistant à étendre des requêtes avec des termes appartenant au mêmes *synsets* de WORDNET que les termes de la requête. Une telle technique permet de différencier les types d'extension effectuée et de conserver la requête originale lors de la recherche des documents.

Dans notre cas, nous considérons deux *ctypes*, c'est-à-dire deux sous-vecteurs composant le vecteur requête : le *ctype* 1 représente la requête originale et le *ctype* 2 la requête étendue. Le rapport entre les poids α_1 et α_2 est appelé par la suite taux de mixité (noté Tx). Par exemple un taux de mixité de 1/3 indique que la requête étendue a 3 fois plus de poids que la requête originale. L'influence de ce taux sur les performances du système de recherche est étudiée en section 5.3.3.3.

5.3.3 Évaluation des performances de l'extension de requêtes

Comme nous l'avons vu, beaucoup de paramètres interviennent de manière non négligeable dans le système de recherche utilisé et dans la technique réalisant l'extension

de requêtes avec les verbes qualia. L'examen de l'influence exacte de chacun de ces paramètres, en fonction de tous les autres paramètres intervenants, serait complexe à mettre en œuvre, et l'analyse et la visualisation des résultats seraient pour le moins difficile.

Nous proposons de nous attacher à une étude de quelques-uns de ces paramètres en fixant les autres à des valeurs arbitraires pour faire ressortir leur rôle dans les variations de performances. On qualifiera d'extension de référence le système de recherche utilisant des requêtes étendues et paramétré avec l'ensemble de ces valeurs par défaut.

Nous étudions ci-dessous les performances de ce système de référence, puis nous étudions dans un second temps l'influence de la taille de l'extension sur ces performances. Nous analysons ensuite de la même manière l'influence du taux de mixité, c'est-à-dire du poids relatif donné aux extensions par rapport aux termes originaux de la requête. Les performances de ces différents paramétrages sont comparées à celles du système de recherche n'utilisant que les requêtes originales. Il faut noter que l'ensemble de ces expériences est réalisé sur la collection OFIL lemmatisée.

5.3.3.1 Performances de l'extension de référence

Le système servant de référence par la suite a les caractéristiques suivantes :

- le schéma de pondération utilisé pour l'indexation est *ltc.ltc* (voir les sections 5.1.2.2 et 5.3.1) ;
- 5 verbes par noms sont ajoutés en extension à la requête ($Nb_V = 5$) ;
- le taux de mixité (le poids relatif du sous-vecteur requête originale/requête étendue) est de 1/3 ;

En pratique, le choix de ces différents paramètres a été fait par sondage de façon à ce que le système résultant soit celui maximisant la précision moyenne non interpolée (NIAP).

Résultats

La figure 5.3 présente les courbes rappel-précision du système de référence comparé au même système mais utilisant uniquement les requêtes originales (sans extension). Deux jeux de courbes sont fournis : l'un décrit les performances de ces deux systèmes en fixant le nombre de documents examinés (DCV) à 20, l'autre considère tous les documents pertinents. Il en ressort que l'extension semble avoir un effet bénéfique notable sur la précision lorsque le rappel est inférieur à 30% ; au-delà de cette limite les systèmes avec extension et sans extension se comportent de manière identique. On remarque également qu'à rappel identique, l'extension de requête améliore plus nettement les performances lorsque le DCV est fixé à 20 documents.

Cette dernière remarque est par ailleurs confirmée par la figure 5.4 représentant la précision du système calculée sur les 5, 10, 15, 20, 30, 50, 100, 200, 500, 1 000, 2 000 et 5 000 premiers documents retournés. L'amélioration amenée par l'extension de requête (notée ici par $Nb_V = 5$) par rapport au même système utilisant les requêtes

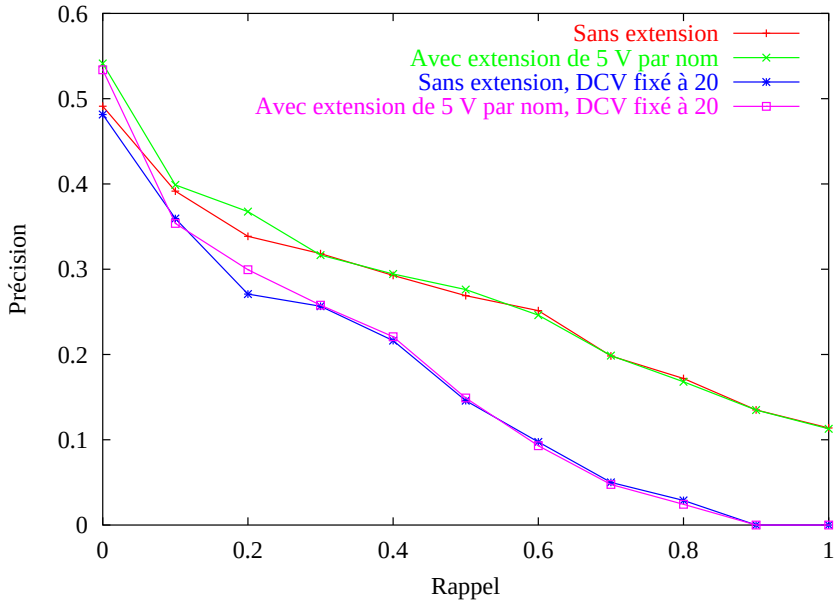


FIG. 5.3 – Courbes rappel-précision du système de référence avec et sans extensions

non étendues (notée par $Nb_V = 0$) est flagrante pour des DCV < 20. Au-delà de ce seuil, aucune modification n'est apportée par l'ajout de verbes qualia.

Ces remarques sont également confirmées par les données recueillies dans le tableau 5.4. Dans ce dernier, seules les mesures jugées statistiquement significatives par l'un des deux tests employés sont indiquées. Les probabilités non indiquées sont supérieures au seuil fixé de 0.1. Toutes les différences statistiquement significatives sont positives, c'est-à-dire au bénéfice de l'extension de requêtes. Le résultat principal que l'on observe est une nette amélioration des performances, à la fois en termes de rappel et de précision, pour des faibles DCV (5, 10 et 30 documents), grâce à l'ajout des verbes qualia à la requête. La précision globale du système, mesurée par la précision moyenne interpolée et non interpolée, en bénéficie également avec une augmentation légère mais significative. Les taux de rappel et de précision mesurés sur les 5 000 premiers documents sont eux aussi en très légère hausse, ainsi que la R-précision, avec l'utilisation des requêtes étendues.

Discussion des résultats

D'après les résultats précédents, il semble que l'utilisation des verbes qualia permette de retrouver plus rapidement des documents qui auraient finalement été proposés à l'utilisateur, mais à des rangs prohibitifs. Ainsi, l'extension concentre en tête de liste les documents pertinents, plus qu'il n'agit sur la précision au détriment du rappel comme cela est souvent dit en recherche documentaire.

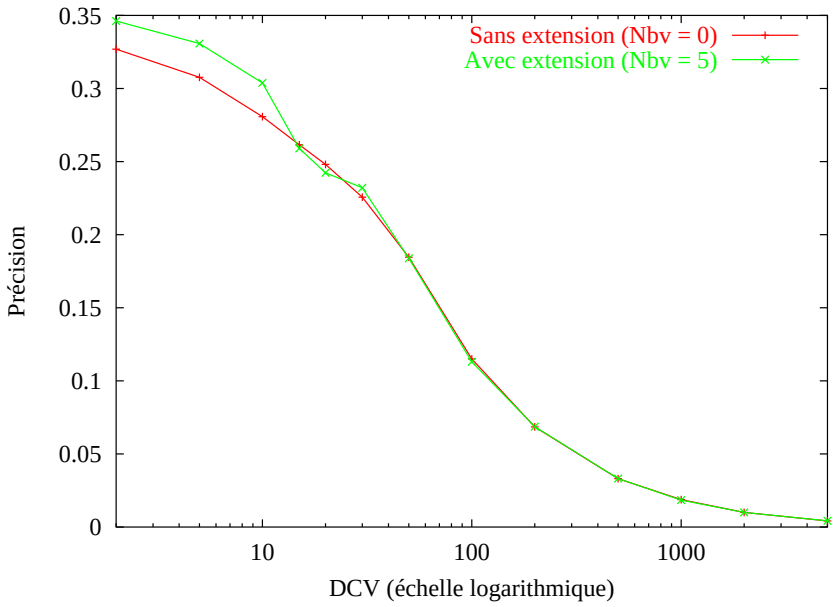


FIG. 5.4 – Précisions du système de référence selon différents DCV

	Sans extension (%)	Avec extension (%)	Amélioration (%)	Probabilité de H_0
IAP	27.01	27.77	+2.81%	$p_S < 0.05$, $p_W < 0.1$
NIAP	25.32	25.91	+2.33%	$p_S < 0.05$
P(5)	30.77	33.08	+7.5%	$p_S < 0.1$
P(10)	28.08	30.38	+8.22%	$p_S < 0.025$, $p_W < 0.06$
P(30)	22.56	23.21	+2.84%	$p_S < 0.05$, $p_W < 0.08$
P(5000)	0.41	0.42	+1.11%	$p_S < 0.05$
R(5)	8.14	8.61	+5.8%	$p_S < 0.1$
R(10)	15.35	16.86	+9.79%	$p_S < 0.025$, $p_W < 0.1$
R(30)	34.91	35.53	+1.78%	$p_S < 0.05$, $p_W < 0.08$
R(5000)	92.77	94.52	+1.89%	$p_S < 0.025$
R-Prec	29.24	29.49	+0.86%	$p_S < 0.1$

TAB. 5.4 – Performances de l'extension de requête

D'un point de vue pratique, cette extension de requêtes par ressources sémantiques — à savoir des verbes qualia liés aux noms contenus dans la requête — permet donc d'améliorer légèrement les performances globales d'un système de recherche documentaire standard. Mais cette extension est particulièrement performante, et donc intéressante à mettre en œuvre, pour les systèmes nécessitant une bonne précision et un bon rappel dès les premiers documents retournés à l'utilisateur. Un tel processus d'exten-

sion serait par exemple particulièrement profitable à des systèmes grand public pour lesquels on sait qu'en moyenne seuls les 20 premiers documents retournés sont consultés par les utilisateurs (voir section 5.3.2.1).

Par ailleurs, l'extension à l'aide de verbes est intéressante à un autre titre. Les utilisateurs de SRI, même avertis, ont tendance à spécifier naturellement une requête en y ajoutant de nouveaux noms. L'extension par verbes qualia permet donc de faire émerger des documents qui n'auraient pas nécessairement été trouvés par une extension manuelle.

5.3.3.2 Influence de la taille de l'extension

Nous étudions dans cette section l'influence de la taille de l'extension sur les performances de notre système de recherche standard. Pour ce faire, nous faisons varier le seul paramètre Nb_V (le nombre de verbes qualia ajoutés pour chaque nom de la requête) du système de référence et comparons les mesures recueillies à celles du système n'utilisant aucune extension de ses requêtes.

Variation de la précision

Les figures 5.5, 5.6 et 5.7 présentent l'effet de la taille de l'extension sur la précision mesurée à différents DCV. Seules les précisions pour des seuils de documents (DCV) allant de 0 à 100 sont présentées, aucune influence de l'extension n'étant constatée au-delà. La taille de l'extension, c'est-à-dire le nombre Nb_V de verbes qualia ajoutés

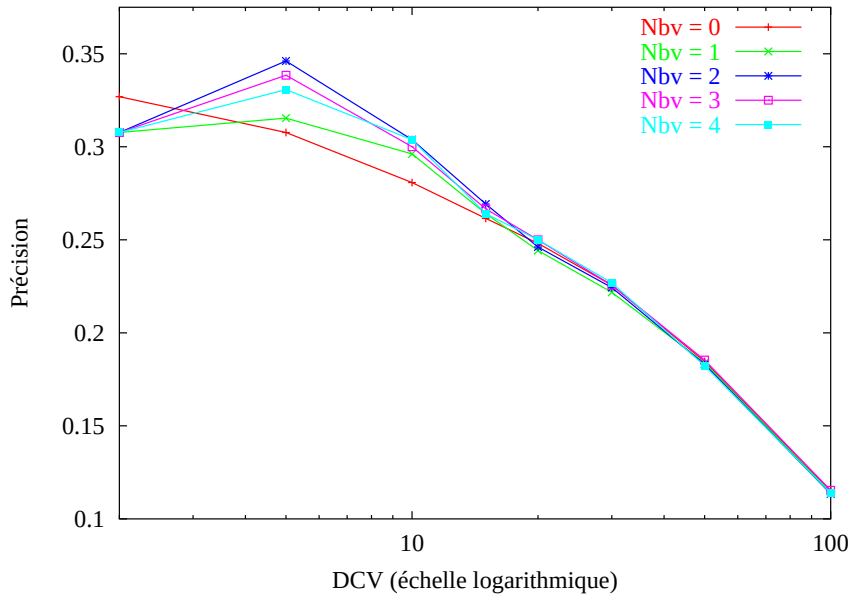


FIG. 5.5 – Variation de la précision selon différents DCV et $1 \leq Nb_V \leq 4$

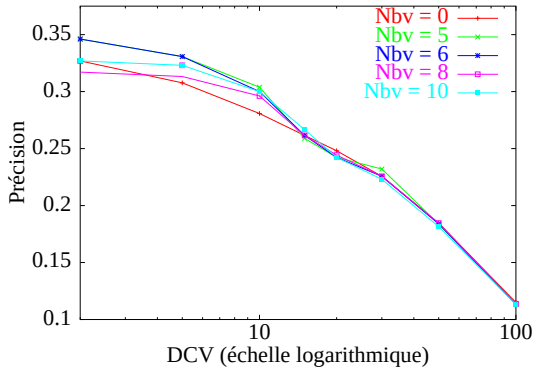


FIG. 5.6 – Variation de la précision selon différents DCV et $5 \leq Nb_V \leq 10$

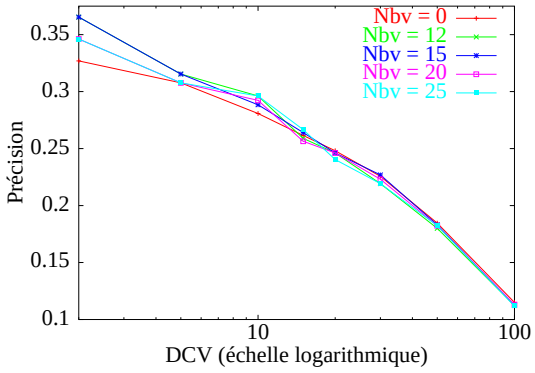


FIG. 5.7 – Variation de la précision selon différents DCV et $10 \leq Nb_V \leq 25$

	Sans extension (%)	Extension $Nb_V = 1$ (%)		Extension $Nb_V = 2$ (%)		Extension $Nb_V = 3$ (%)		Extension $Nb_V = 4$ (%)	
IAP	27.01	(27.19	0.65%)	27.46	1.63%	27.68	2.47%	27.77	2.8%
P(5)	30.77	(31.54	2.5%)	34.62	12.5%	33.84	10%	33.08	7.5%
P(10)	28.08	29.62	5.48%	30.38	8.22%	30	6.85%	30.38	8.22%
P(15)	26.15	(26.41	0.98%)	26.92	2.94%	26.67	1.96%	(26.41	0.98%)
R(5)	8.14	(8.24	1.3%)	8.89	9.29%	8.73	7.33%	8.61	5.8%
R(10)	15.35	16.35	6.46%	16.98	10.58%	16.86	9.8%	16.86	9.8%
R(15)	21.14	21.52	1.81%	21.82	3.19%	21.52	1.8%	(21.48	1.61%)

TAB. 5.5 – Performances de l'extension de requête pour de faibles DCV

par nom, semble influer de manière relativement modérée sur les résultats, du moins pour des extensions de taille faible ou moyenne ($Nb_V < 15$). Au-delà de ce seuil, l'amélioration pour des DCV petits est plus anecdotique. Le tableau 5.5, dans lequel apparaissent entre parenthèses les valeurs n'ayant pas été détectées significatives par les tests de Student ou de Wilcoxon, détaille les résultats obtenus pour de faibles DCV (5, 10, 15 documents). Il permet de noter que le pic est atteint pour une valeur de Nb_V de 2 verbes par nom. On a alors de très intéressantes améliorations des performances du système, à la fois en termes de rappel et de précision pour ces faibles DCV. Ces améliorations, très localisées, permettent d'améliorer la précision globale du système, mesurée ici par IAP.

Pour toutes les valeurs de Nb_V testées, c'est-à-dire pour toutes les tailles d'extension, la précision mesurée sur les 10 premiers documents est améliorée de façon statistiquement significative. En revanche, la taille de l'extension elle-même ne semble pas avoir d'importance, la précision variant légèrement d'une valeur de Nb_V à une autre sans

suivre de schéma particulier. Cela est attesté en figure 5.8.

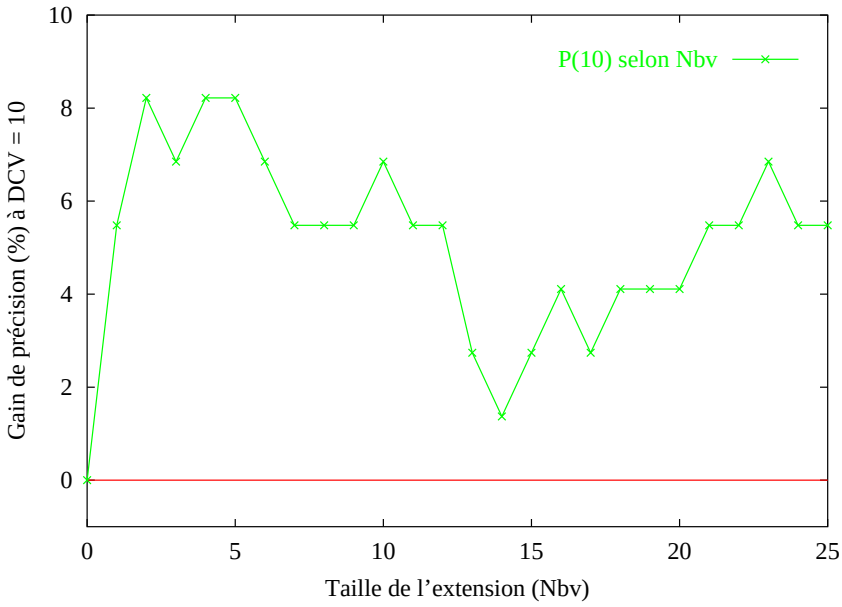


FIG. 5.8 – Variation de la précision à 10 documents selon différents Nb_V

Variation du rappel

Comme nous le constatons en figure 5.9, le taux de rappel est lui aussi influencé par la taille de la requête. Selon cette figure, l'extension de requêtes semble agir différemment selon les DCV. Pour de faibles DCV, on retrouve les améliorations constatées précédemment, dans des amplitudes assez variables selon la taille de l'extension. Pour des DCV compris entre 20 et 1 000 documents, le rappel est identique, voire légèrement inférieur à celui obtenu par la requête sans extension. Au-delà de ce seuil, on constate de nouveau une amélioration du taux de rappel, atteignant un maximum pour un DCV de 5000 documents, assez faible mais vérifiée quel que soit le nombre de verbes qualia ajoutés aux requêtes.

Cette amélioration du rappel pour un DCV fixé à 5000 documents est détaillée en figure 5.10. On constate que l'amélioration du taux de rappel croît régulièrement avec la taille de l'extension. Cela confirme l'intuition selon laquelle l'ajout de verbes qualia permet de trouver plus de documents pertinents dans la base documentaire.

5.3.3.3 Influence du taux de mixité

Nous étudions ci-dessous l'influence du taux de mixité, c'est-à-dire du poids relatif de la requête originale et de la requête étendue, sur les performances de notre système de recherche. Nous fixons donc cette fois-ci Nb_V à 5 verbes par nom et faisons varier

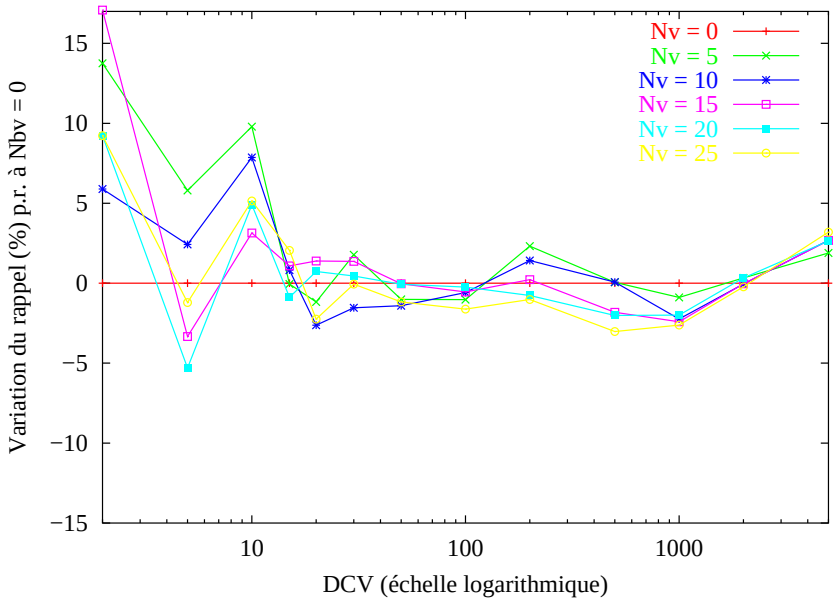


FIG. 5.9 – Variation du rappel (en pourcentage) pour différents DCV

le seul paramètre Tx. Comme précédemment, nous comparons les mesures recueillies à celles du système n'utilisant aucune extension de ses requêtes.

Taux de mixité

Les figures 5.11 et 5.12 présentent l'effet de la taille de l'extension sur la précision mesurée pour des seuils de documents allant de 0 à 100, aucune influence de l'extension n'étant constatée pour des DCV supérieurs. On observe que quelle que soit la valeur de ce taux, l'extension se traduit par une amélioration plus ou moins importante de la précision pour des DCV inférieurs à 15 documents. Par ailleurs, plus ce taux donne de l'importance à l'extension, plus l'amélioration est importante pour ces faibles DCV. Cela se constate en particulier pour les taux de rappel et de précision à DCV = 10, dont on indique les gains pour différents taux de mixité possibles en figures 5.13 et 5.14.

Dans le cas où le taux de mixité donne plus de poids à la requête originale qu'à l'extension, les résultats sont plus faibles, mais plus nombreux à être statistiquement significatifs, comme cela peut se vérifier dans le tableau 5.6. Au contraire, comme on le constate dans le tableau 5.7, un poids important donné à l'extension produit des résultats plus tranchés, avec des améliorations de l'ordre de 10% du rappel et de la précision pour un DCV fixé à 10 documents, mais aussi des dégradations de performances significatives pour des DCV entre 50 et 1 000 documents.

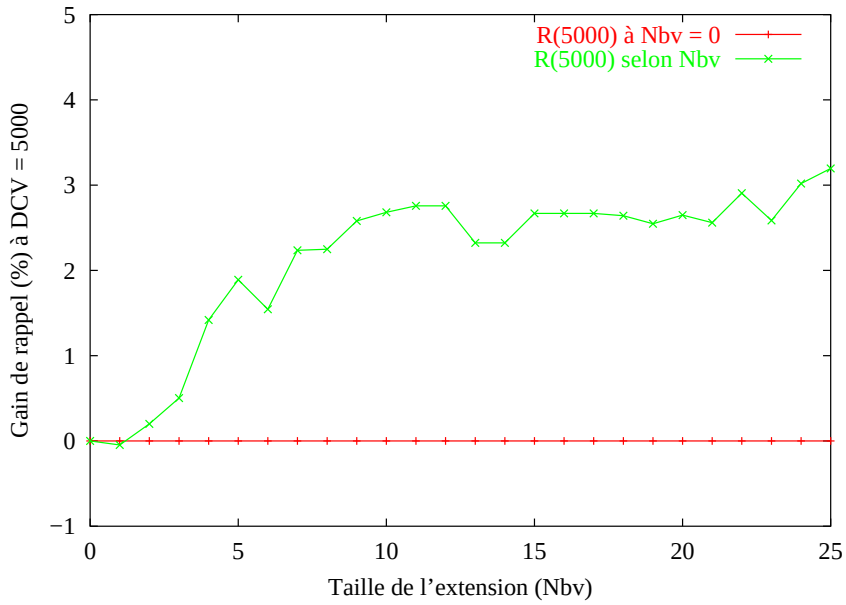


FIG. 5.10 – Variation du rappel (en pourcentage) à DCV fixé à 5 000 documents

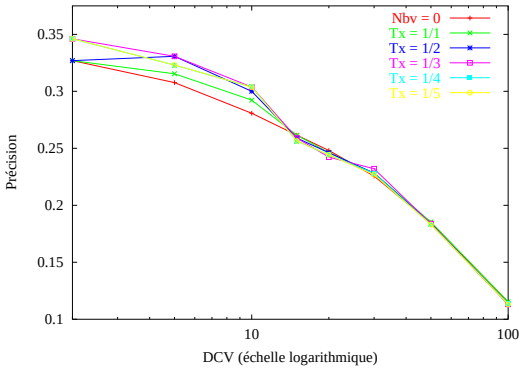


FIG. 5.11 – Variation de la précision selon différents DCV et $1/5 \leq Tx \leq 1/1$

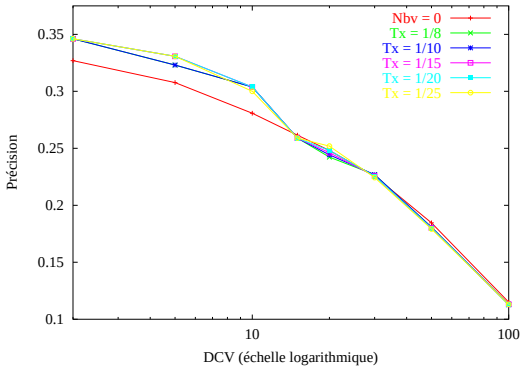


FIG. 5.12 – Variation de la précision selon différents DCV et $1/25 \leq Tx \leq 1/8$

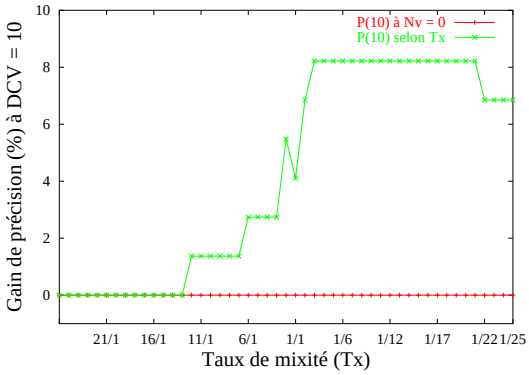


FIG. 5.13 – Variation de la précision selon différents Tx à DCV = 10

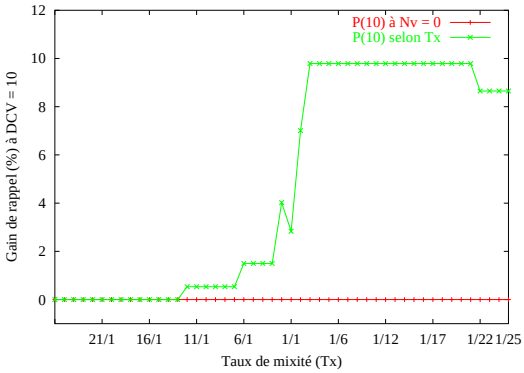


FIG. 5.14 – Variation du rappel selon différents Tx à DCV = 10

	Sans extension (%)	Extension Tx = 2/1 (%)	
NIAP	25.32	25.59	1.06%
P(10)	28.08	29.62	5.48%
P(30)	22.56	22.82	1.13%
P(50)	18.46	18.61	0.83%
P(200)	6.84	6.94	1.40%
P(1000)	1.87	1.88	0.82%
P(2000)	0.99	1.00	0.39%
P(5000)	0.41	0.42	0.92%
R(10)	15.35	15.97	4.02%
P(30)	34.91	35.09	0.53%
R(50)	47.37	47.58	0.46%
R(200)	66.76	68.72	2.93%
R(1000)	86.22	86.58	0.42%
R(2000)	90.20	90.62	0.46%
R(5000)	92.77	94.20	1.54%

TAB. 5.6 – Performances de l'extension de requête à Tx = 2/1

	Sans extension (%)	Extension Tx = 1/15 (%)	
P(10)	28.08	30.38	8.22%
P(50)	18.46	18.00	−2.50%
P(1000)	1.87	1.80	−3.70%
P(5000)	0.41	0.42	1.11%
R(10)	15.35	16.86	9.79%
R(50)	47.37	45.58	−3.78%
P(1000)	86.22	83.94	−2.65%
P(2000)	90.20	90.42	0.24%
P(5000)	92.77	94.52	1.89%
R-Prec	29.24	28.13	−3.79%

TAB. 5.7 – Performances de l'extension de requête à Tx = 1/15

Bilan et discussion

Synthèse

Le but premier des travaux présentés dans ce mémoire était de développer un outil d'acquisition d'informations lexicales sémantiques sur corpus devant répondre à un objectif triple :

1. Il devait tout d'abord bien entendu fournir des résultats de bonne qualité.
2. Ensuite, le fonctionnement de ce système d'acquisition devait être interprétable, c'est-à-dire que son processus et ses résultats soient appréhendables par un utilisateur.
3. Enfin, partant du constat que ces informations sont nécessaires pour de nombreuses applications du TAL, mais spécifiques à chacune d'elles et au domaine traité, le dernier critère de ce triple objectif était que l'outil développé soit générique et aisément portable.

Pour atteindre cet objectif, nous nous sommes placé dans le cadre formel de l'apprentissage symbolique supervisé, relativement peu employé pour ce type de tâche. Plus précisément, notre outil, ASARES, exploite la programmation logique inductive pour inférer des patrons d'extraction morphosyntaxiques et sémantiques à partir d'exemples d'occurrences en contexte des informations recherchées. Ces patrons sont ensuite utilisés pour acquérir de nouvelles instances de ces informations au sein du corpus. Cette approche permet à notre outil d'obtenir de bons résultats, et les patrons inférés fournissent une définition opérationnelle et interprétable par des linguistes du concept même de l'information recherchée. Cet interprétabilité, ainsi que l'efficacité de la phase d'inférence, sont assurées en pratique par les adaptations de l'algorithme de PLI à nos besoins spécifiques que nous avons effectuées. De plus, la souplesse d'utilisation de la PLI permet, à travers l'utilisation d'exemples idoines, de s'intéresser à l'acquisition de tout type d'information sémantique lexicale, ce qui confère à notre outil une bonne généricité. Ce dernier critère est par ailleurs renforcé par la combinaison originale que nous proposons de cette technique symbolique avec une méthode d'acquisition numérique. L'approche semi-supervisée résultante permet ainsi à ASARES d'être plus automatique et donc plus facilement portable. Notre outil répond ainsi à l'ensemble de notre triple objectif, même si quelques problèmes de coûts, notamment dus à l'étiquetage sémantique, n'ont reçu qu'une réponse partielle.

Pour vérifier la validité de notre approche, nous avons choisi d'appliquer ASARES à l'acquisition de relations sémantiques très peu étudiées : les relations nom-verbe qualia, proposées dans le modèle du Lexique génératif de J. Pustejovsky. Ce cadre applicatif revêtait deux intérêts. D'une part, les relations qualia sont définies dans le Lexique génératif de manière purement théorique ; l'interprétabilité linguistique des patrons produits par ASARES nous a permis à ce titre d'en étudier et d'en formaliser les réalisations réelles en contexte, contribuant ainsi à une meilleure connaissance de ces objets linguistiques. Il ressort de cette étude que les relations N-V qualia sont portées par des schémas simples mais inhabituels, à la fois en termes d'éléments manipulés et de niveau de généralisation, et relativement propres au corpus. D'autre part, ces relations nomino-verbales, bien qu'assez peu étudiées, semblent intéressantes dans un contexte de recherche d'information pour donner accès à des formulations différentes mais sémantiquement équivalentes d'une même idée. Pour vérifier la portée réelle de cette affirmation, nous avons évalué expérimentalement l'intérêt des relations qualia pour étendre des requêtes au sein du système de recherche d'information (SRI) SMART. Les résultats que nous avons obtenus montrent une amélioration globale des performances du SRI, significative mais de faible amplitude. Cette amélioration est plus particulièrement sensible sur certains points localisés : la pertinence des dix premiers documents retournés bénéficie notamment nettement de l'extension des requêtes par les relations qualia, même si ces résultats demandent à être confirmés par d'autres tests. Cela est d'autant plus intéressant que ces améliorations localisées correspondent à l'usage standard qui est fait des moteurs de recherche grand public. L'utilisation des relations qualia pour l'extension de requêtes est avantageuse à un autre titre. En effet, comme nous l'avons déjà signalé, elle correspond à une précision sémantique verbale qui n'est pas celle que fait intuitivement un humain, ce dernier ayant plutôt tendance à ajouter des noms à la requête s'il veut plus de pertinence dans les documents retournés. Cette extension par des relations qualia lui permet donc d'avoir accès à des variantes auxquelles il ne serait certainement pas parvenu par lui-même. Enfin, ces résultats sont encourageants pour l'utilisation de ressources sémantiques acquises automatiquement sur corpus dans les SRI. En effet, dans les travaux présentés dans ce mémoire, nous nous sommes focalisé sur une sorte de relations sémantiques très précise et très restreinte. D'autres types d'informations sémantiques pourraient être utilisés de la même manière et ainsi, même si quelques travaux de ce genre font état de résultats mitigés, multiplier les effets bénéfiques sur les performances d'une recherche documentaire.

Perspectives

De nombreuses perspectives sont ouvertes sur différents aspects de notre travail, aussi bien techniques, à travers le fonctionnement d'ASARES, qu'applicatifs, à travers l'acquisition d'éléments du Lexique génératif et leur utilisation en RI.

L'approche utilisée dans la phase d'apprentissage d'ASARES, consistant à inférer des patrons manipulant des informations issues d'étiquetages, peut être qualifiée de *knowledge-poor*. En effet, aucune connaissance autre que celle apportée par ces éti-

quetages, morphosyntaxiques ou sémantiques, n'est exploitée. En conjonction de ces informations, on peut imaginer en utiliser d'autres, par exemple morphologiques ou bien encore syntaxiques. Dans ce dernier cas, les relations de dépendances syntaxiques telles que celles ajoutées par des outils comme SYNTEX (Bourigault & Fabre, 2000) pourraient constituer d'excellents prédicats relationnels. De même, les prédicats structurels (pred/2 ou suc/2) utilisés dans ASARES forment de la manière la plus simple la structure des patrons. On pourrait à ce titre enrichir notre langage d'hypothèses par des relations plus complexes permettant une expressivité de nos patrons plus proche de celle des expressions régulières par exemple. Ce type d'enrichissement, s'il est possible, doit cependant répondre à deux critères importants. Il doit d'une part conserver l'interprétabilité linguistique des patrons produits et, d'autre part, ne pas rendre le processus d'inférence, simplifié jusqu'ici par l'emploi de prédicats déterministes et de l'identité objet, d'une complexité calculatoire trop importante.

On peut plus généralement se demander si n'importe quelle relation sémantique est modélisable à l'aide de schémas contextuels tels que les patrons d'extraction que nous inférons. À cette question, J. Pearson (Pearson, 1998) apporte en partie une réponse : il ressort de son étude sur les éléments définitoires que l'utilisation de patrons lexico-syntaxiques se révèle inadaptée pour capturer la notion de synonymie entre termes alors même qu'elle donne de bons résultats pour d'autres relations paradigmatiques telles que l'hyperonymie (voir (Morin, 1999)). À l'issue de nos travaux présentés dans ce mémoire, cette question reste ouverte et sa réponse ne peut être trouvée que par la multiplication d'expériences d'acquisition d'informations sémantiques variées sur des données diverses.

Notre outil d'acquisition peut également bénéficier des travaux portant sur le développement, aussi bien théorique que pratique, des méthodes de programmation logique inductive. Tout d'abord, la PLI, contrairement à beaucoup d'autres méthodes d'apprentissage, ne peut se ramener dans le cas général au modèle d'apprenabilité PAC de Valiant (Valiant, 1984), même si certains travaux ont montré que des sous-ensembles de formules étaient PAC-apprenables (De Raedt & Džeroski, 1994). Un cadre plus adapté à la PLI a donc été proposé : il s'agit de l'apprenabilité U (*U-learnability*) (Muggleton & De Raedt, 1994). Une des principales différences de ce cadre avec le modèle PAC est de considérer les complexités et les temps dans le cas moyen et non pas dans le pire cas. Ces travaux peuvent permettre de mieux modéliser le fonctionnement d'ASARES, et plus particulièrement de ses deux versions semi-supervisées présentées au chapitre 4, notamment en ce qui concerne le nombre d'exemples nécessaires à l'inférence de patrons et la gestion des données bruitées. D'un point de vue plus pratique, parmi les différentes pistes de recherche en PLI identifiées dans (Page & Srinivasan, 2003), celles portant sur l'utilisation de nouvelles méthodes de recherche dans $\mathscr{E}_H$, la parallélisation des exécutions ou l'incorporation de probabilités explicites peuvent également améliorer l'efficacité de notre processus d'inférence. Ce dernier point, l'utilisation de probabilités au sein de la phase d'inférence, serait d'ailleurs d'un intérêt indéniable pour mettre en œuvre une approche semi-supervisée d'ASARES similaire aux deux que nous avons proposées mais bénéficiant d'un cadre formel sûr. Les travaux effectués en invention de prédicats (Kramer, 1995) ou, dans une optique relativement proche, en complétion

de théorie (Muggleton & Bryant, 2000), sont aussi des axes de recherche pouvant être exploités dans ASARES pour accroître la pertinence des patrons produits en découvrant dans les données des relations inconnues ou manquantes. Les travaux plus généraux en apprentissage artificiel, comme ceux que nous avons déjà évoqués portant sur le *bootstrapping* ou la *feature selection*, peuvent de la même manière profiter à notre outil d'acquisition.

Plusieurs perspectives intéressantes de nos travaux concernent notre cadre applicatif. L'un de nos buts était de développer une technique d'acquisition d'informations sémantiques sur corpus aisément portable et adaptable à des besoins spécifiques. À ce titre, on peut envisager l'utilisation d'ASARES pour l'extraction d'autres informations que les relations nom-verbe qualia. Nous projetons notamment de nous intéresser, toujours dans le cadre du Lexique génératif, aux liens N-N tels que ceux définis par le rôle constitutif de la structure des qualia et, dans un autre cadre linguistique (en collaboration avec M.-C. L'Homme, OLST, université de Montréal), aux fonctions lexicales (Mel'čuk, 1998 ; Kahane & Polguère, 2001) définies dans la théorie Sens-Texte d'I. Mel'čuk.

À l'issue des expériences présentées dans le dernier chapitre de ce mémoire concernant l'utilisation de ressources lexicales sémantiques acquises sur corpus en RI, de nombreuses perspectives restent également ouvertes. Tout d'abord, l'exploitation de ressources d'autres types que les relations qualia peut être envisagée. Les différents types d'informations lexicales peuvent être combinés et ainsi permettre des reformulations plus riches et plus variées. Par exemple, les autres éléments du Lexique génératif mentionnés ci-dessus peuvent être utilisés dans ce but, de même que des informations plus habituelles dans ce contexte, comme les relations de synonymie ou hyperonymie-hyponymie. L'apport de telles ressources doit être cependant considéré avec attention puisque, comme nous l'avons déjà dit, quelques travaux exploitant ce type d'informations sémantiques mentionnent des performances parfois peu convaincantes (Voorhees, 1994).

De plus, l'utilisation que nous faisons des ressources sémantiques acquises par ASARES au sein du SRI décrite au chapitre 5 reste relativement simple puisqu'il s'agit d'ajouter indistinctement des verbes en relation qualia aux noms de la requête. On peut imaginer des stratégies d'extension plus subtiles, s'appuyant par exemple sur une analyse de la requête et une reformulation plus contrôlée (d'une manière proche de celle de C. Fabre & C. Jacquemin (2000)). Celles-ci nous assureraient des extensions plus pertinentes en imposant des contraintes linguistiques sur les noms de la requête à considérer (par exemple n'étendre que les têtes de syntagme) et sur les verbes qualia les plus adaptés.

Outre l'extension de requêtes, d'autres approches d'inclusion de telles ressources lexicales sémantiques sont également possibles, comme cela est fait par exemple dans le système DSIR (Besançon, 2001). Dans celles-ci, les descripteurs des documents ne seraient plus simplement des sacs de mots, mais des structures complexes, ce qui nécessite de redéfinir un modèle de représentation, une notion de distance entre ces descripteurs...

De telles représentations, qui relèvent encore largement de la recherche, seraient plus à même d'intégrer les informations sémantiques lexicales, et permettraient donc leur manipulation plus naturellement au sein des SRI. Des approches plus simples sont néanmoins étudiées (Strzalkowski *et al.*, 1999b) et s'appuient par exemple sur une représentation composée de sous-représentations des documents en niveaux (celui des mots, des noms, des syntagmes, *etc.*). Ces sous-représentations, prises individuellement, sont manipulables de manière standard par les SRI et donc plus faciles à mettre en œuvre. Les ressources lexicales sémantiques acquises pourrait dans ce type d'approche constituer l'un des niveaux de description des documents.

Annexe A

Éléments de logique

Nous rappelons dans cette annexe quelques définitions et notations des éléments de logique utilisés dans les chapitres 2 et 3 concernant la PLI. De ce fait, seuls ces éléments et quelques-unes de leurs propriétés exploitées au sein du mémoire sont introduits ici. Pour une présentation plus complète de ces notions logiques, le lecteur intéressé peut se reporter aux ouvrages de référence tels que (Lloyd, 1987) et (Chang & Lee, 1973).

A.1 Définitions et notations

Cette partie présente les éléments de base de la logique des prédicats, logique à la base de la plupart des algorithmes de PLI. Nous reprenons à cet effet certaines définitions données au chapitre PLI de l'ouvrage de A. Cornuéjols et L. Miclet (2002). Après un rappel du vocabulaire élémentaire, nous présentons successivement le langage des prédicats logiques puis celui des clauses.

A.1.1 Vocabulaire élémentaire

En logique du premier ordre, une constante réfère à un objet ; on la notera en syntaxe Prolog par des mots débutant par des minuscules (par exemple, a, b, *socrate*, 1, $m_4 12$, *etc.*). Les variables servent quant à elles à dénommer un objet sans pour autant l'identifier. Elles seront notées par des mots commençant par des majuscules (par exemple, X, Y, $Word$). Les variables peuvent prendre leurs valeurs sur un ensemble de constantes appelé domaine. Deux quantificateurs permettent de manipuler les variables : le quantificateur universel, représenté par le symbole $\forall$, et le quantificateur existentiel, représenté par $\exists$.

Nous définissons également des symboles servant de connecteurs entre notions logiques :

- $\neg$ représente la négation ; c'est le seul connecteur unaire (*i.e.* à un seul argument) ;
- $\wedge$ représente la conjonction (le *et*) ;
- $\vee$ représente la disjonction (le *ou*) ;
- $\rightarrow$ représente l'implication ; $a \rightarrow b$ se lit *a implique b* ;

– $\leftrightarrow$ représente l'équivalence (la double-implication).

L'implication, ici de gauche à droite, est parfois utilisée dans le sens inverse (représentée par le symbole $\leftarrow$), notamment en logique des prédicats pour représenter des clauses (voir section A.1.3). Enfin, des parenthèses peuvent être utilisées pour éliminer les ambiguïtés éventuelles.

A.1.2 Logique des prédicats

La logique des prédicats est une logique du premier ordre et manipule donc des variables, mais également des fonctions et bien sûr des constantes. Ces trois objets sont qualifiés du nom unique de terme, qui forme l'élément de base des fonctions, prédicats et littéraux définis ci-dessous, ces derniers formant eux-mêmes formules et sous-formules.

Définition 8 (Terme) *Un terme se définit récursivement comme étant :*

- *soit une constante (par exemple* $a1$*, toto) ;*
- *soit une variable (par exemple* Var*) ;*
- *soit une fonction appliquée à des termes (par exemple* $g(a1)$*,* $f(a1, g(Var), toto)$*).*

□

Définition 9 (Fonction) *Une fonction est représentée par un symbole, généralement un mot en minuscule et une arité, représentant son nombre d'arguments attendus (par exemple f/3, age/2). Une fonction d'arité 0 est considérée comme une constante. Le domaine de résultat d'une fonction est quelconque.* □

Définition 10 (Prédicat) *Un prédicat est une fonction dont le domaine est {vrai, faux} ; il est donc également représenté par un symbole et une arité (par exemple, est_pere/2, nom_commun/1).* □

Définition 11 (Atome) *Un atome est un prédicat appliqué à des termes.* □

Définition 12 (Littéral) *Un littéral est un atome, éventuellement précédé du symbole de négation* $\neg$ *(par exemple,* $est\_pere(Toto)$*,* $\neg nom\_commun(m527)$*).* □

Définition 13 (Formule) *Une formule se définit récursivement ainsi :*

- *un littéral est une formule ;*
- *si* ϕ *et* ψ *sont des formules alors* $(\phi \vee \psi)$*,* $(\phi \wedge \psi)$*,* $(\neg\phi)$*,* $(\neg\psi)$*,* $(\phi \rightarrow \psi)$ *et* $(\phi \leftrightarrow \psi)$ *sont des formules ;*
- *si* X *est une variable et* ϕ *une formule, alors* $((\forall X)\phi)$ *et* $((\exists X)\phi)$ *sont des formules.* □

Définition 14 (Sous-formule) *Une sous-formule est une suite de symboles d'une formule qui est elle-même une formule ; par exemple,* $g(b1) \vee \neg h$ *est une sous-formule de* $(\exists X) f(X) \vee g(b1) \vee \neg h$ □

Définition 15 (Variable libre, variable liée) *Une variable X est dite liée dans une formule Φ s'il existe dans cette formule une sous-formule commençant par $((\forall X)$ ou $((\exists X)$. Dans le cas contraire, elle est dite libre.* □

Un atome, un littéral ou une formule ne contenant aucune variable est dit clos (*ground* en anglais).

A.1.3 Langage des clauses et programmes logiques

Le langage des clauses, et plus particulièrement celui des clauses de Horn et des clauses définies (voir ci-dessous), permet de ne manipuler que des formules ayant une certaine forme. Les programmes logiques sont composés d'une conjonction de clauses.

Définition 16 (Clause) *Une clause est une formule composée d'une disjonction finie de littéraux dans laquelle toutes les variables sont quantifiées universellement.* □

Par souci de lisibilité, on simplifie généralement l'écriture de ces clauses en omettant de faire apparaître les quantificateurs universels. Ainsi, $\forall X \forall Y (f(X) \vee g(Y))$ s'écrira simplement $f(X) \vee g(Y)$.

Définition 17 (Clause de Horn) *Une clause de Horn est une clause contenant au plus un littéral positif.* □

Soit une clause de Horn $C = l_t \vee \neg l_1 \vee ... \vee \neg l_n$. C peut également s'écrire $l_1 \wedge ... \wedge l_n \rightarrow l_t$, soit encore en inversant le sens de l'implication $l_t \leftarrow l_1 \wedge ... \wedge l_n$. Finalement, en remplaçant la conjonction par une virgule et l'implication par le symbole :- on obtient la syntaxe utilisée en Prolog : l_t :- $l_1, ..., l_n$.

Définition 18 (Clause définie) *Une clause définie est une clause de Horn contenant exactement un littéral positif.* □

On appelle tête d'une clause définie C le seul littéral positif d'une clause (l_t dans l'exemple précédent) et on appelle corps de C l'ensemble des littéraux négatifs $(l_1, ..., l_n)$. Une clause unitaire est une clause dont le corps est vide, notée $l_t \leftarrow$ ou plus simplement l_t en syntaxe Prolog. Inversement, une clause n'ayant pas de littéral tête est appelé but et est notée $\leftarrow l_1, ..., l_n$ (en Prolog, cela correspond à la requête $l_1, ..., l_n$). Intuitivement, le corps d'une clause donne une définition du concept dénoté par la tête de la clause; ainsi, `t(X) :- a(X), b(X), c(X)` signifie $(\forall X a(X) \wedge b(X) \wedge c(X) \rightarrow t(X))$ mais aussi que X vérifie la propriété t à la condition (suffisante) que X vérifie les propriétés a, b et c.

Définition 19 (Programme logique) *Un programme logique défini, ou plus simplement programme logique, est un ensemble fini de clauses définies, équivalent à leur conjonction.* □

Par exemple, le programme `p(X) :- a(X). a(1). a(2).` correspond à la formule $(\forall X p(X) \rightarrow a(X)) \wedge a(1) \wedge a(2)$; ce petit programme logique permet d'ailleurs de démontrer $p(1)$ et $p(2)$ en réponse à la requête `p(X)`.

A.1.4 Skolemisation

Définition 20 (Skolemisation) *La skolemisation d'une formule ϕ purement existentiellement quantifiée (i.e. dont les variables sont toutes quantifiées existentiellement) est obtenue en ôtant les quantificateurs existentiels et en remplaçant systématiquement les variables par de nouvelles (i.e. n'apparaissant pas ailleurs) constantes.* □

Ainsi, si ϕ est la formule $\exists X \exists Y (pr1(X, Y) \wedge pr2(X, b))$, alors la formule obtenue par skolemisation sera $pr1(c, d) \wedge pr2(c, b)$.

L'intérêt de la skolemisation réside dans le théorème suivant (Chang & Lee, 1973).

Théorème 1 *Un ensemble de formules sans fonction est satisfiable si et seulement si sa skolemisation est satisfiable.* □

La skolemisation est utilisée en PLI dans des systèmes comme PROGOL ou ALEPH pour construire la clause la plus spécifique couvrant un exemple, la *bottom clause* (voir section 2.2.2.4).

A.2 Implication entre ensemble de formules

Cette section est dévolue à la présentation des notions d'implication sémantique entre formules et des règles d'inférence.

A.2.1 Satisfaction de formules

Définition 21 (Interprétation) *Une interprétation est une fonction v de l'ensemble des formules dans $\{0, 1\}$ (représentant $\{faux, vrai\}$) qui respecte la sémantique des connecteurs. Soit ϕ une formule ; on a ainsi :*

$$
\begin{array}{l}
v(\neg\phi) = 1 - v(\phi) \\
v(\phi \wedge \psi) = v(\phi) \cdot v(\psi) \text{ où } \cdot \text{ marque la multiplication} \\
v(\phi \vee \psi) = min(v(\phi) + v(\psi), 1) \\
v(\phi \rightarrow \psi) = \left\{ \begin{array}{ll} 0 & si\ v(\phi) = 1\ et\ v(\psi) = 0 \\ 1 & sinon \end{array} \right. \\
v(\phi \leftrightarrow \psi) = \left\{ \begin{array}{ll} 1 & si\ v(\phi) = v(\psi) \\ 0 & sinon \end{array} \right.
\end{array}
$$
□

Définition 22 (Satisfaction de formules) *On dit qu'une interprétation v satisfait (ou vérifie) une formule ϕ si $v(\phi) = 1$; on note cela $v \models \phi$. Si $v(\phi) = 0$, alors l'interprétation v falsifie ϕ et on note $v \not\models \phi$.* □

On appelle tautologie une formule vraie pour toutes les interprétations possibles.

Définition 23 (Équivalence entre formules) *On dit que deux formules ϕ et ψ sont équivalentes si pour toute interprétation v on a l'égalité $v(\phi) = v(\psi)$.* □

A.2.2 Implication logique

Définition 24 (Satisfaction d'ensembles de formules) *On dit qu'une interprétation v satisfait un ensemble de formules Φ, ou est modèle de Φ, si pour chaque formule ϕ de Φ, v satisfait ϕ ; on emploie la même notation que pour les formules, soit $v \models \Phi$.* □

On dit ainsi que Φ est satisfiable si Φ a au moins un modèle et est insatisfiable sinon.

Définition 25 (Implication) *Soit Φ un ensemble de formules et ψ une formule ; on dit que Φ implique ψ si chaque modèle de Φ satisfait ψ ; on note alors $\Phi \models \psi$.* □

Plus concrètement, cela signifie que toute interprétation satisfaisant toutes les formules de Φ satisfait également la formule ψ. Il faut noter que cette implication est donc uniquement basée sur les valeurs de vérité des formules, qui en représente en quelque sorte le sens, et non sur les formules elles-mêmes ; c'est pourquoi on appelle parfois cette relation « implication sémantique ». On remarque également que si $\Phi \models \psi$ alors l'ensemble de formules $\Phi \cup \{\neg\psi\}$ est insatisfiable. En effet, supposons que $\Phi \models \psi$ et soit v une interprétation. Soit v est un modèle de Φ, et donc $v \models \psi$ et alors $v \not\models \neg\psi$, ce qui implique que v ne peut pas être modèle de $\Phi \cup \{\neg\psi\}$. Soit v n'est pas un modèle de Φ, ce qui signifie qu'elle ne peut pas être modèle non plus de $\Phi \cup \{\neg\psi\}$. $\Phi \cup \{\neg\psi\}$ est donc insatisfiable sous ces conditions.

A.2.3 Règles d'inférence

Les règles d'inférence sont les règles permettant de manipuler les formules pour trouver, ou dériver, à partir d'un ensemble Φ de formules un autre ensemble Ψ. On dit alors que Φ prouve Ψ par l'ensemble de règles I, et on note $\Phi \vdash_I \Psi$. Contrairement à l'implication logique, où l'on raisonnait sur les modèles satisfaisant les deux ensembles de formules dans une approche *sémantique*, il s'agit plutôt de raisonner de manière *syntaxique* sur les formules. Les règles d'inférence peuvent se voir alors comme des règles de réécriture.

Une règle d'inférence bien connue est le *modus ponens* : $(\phi \wedge (\phi \rightarrow \psi)) \vdash \psi$. Elle permet de déduire la formule mortel(socrate) à partir des formules mortel(X):-homme(X) et homme(socrate).

Les règles d'inférence peuvent être variées, mais on cherche à ce qu'elles vérifient au mieux les deux propriétés suivantes :

- la correction ; un ensemble I de règles d'inférence est dit correct si tout ce qu'il permet de dériver est valide, *i.e.* si $\Phi \vdash_I \Psi$ alors on a $\Phi \models \Psi$;
- la complétude ; un ensemble I de règles d'inférence est dit complet si tout ce qui est valide peut en être dérivé, *i.e.* si $\Phi \models \Psi$ alors on a $\Phi \vdash_I \Psi$.

A.2.4 Modèle de Herbrand

Comme nous venons de le voir, vérifier l'implication entre deux ensembles de formules nécessite de s'assurer que toutes les modèles de l'un soient modèles de l'autre.

Dans le cas des logiques d'ordre supérieur ou égal à 1, le nombre de modèles est potentiellement infini. Il est donc impossible de vérifier l'implication logique directement. Heureusement, un sous-ensemble spécial d'interprétations suffit, sous certaines conditions, à déterminer l'implication ; ce sont les interprétations de Herbrand.

Définition 26 (Univers de Herbrand) *L'univers de Herbrand d'un ensemble de formules (avec au moins une constante) est l'ensemble de toutes les constantes utilisées dans les formules. Si aucune constante n'est utilisée, on en ajoute arbitrairement une.* □

Pour illustrer cette définition, considérons l'ensemble de formules Φ composé de $\forall X(pr(a, X) \rightarrow pr(X, b))$ et $\forall X \forall Y \forall Z(pr(X, Y) \wedge pr(Y, Z) \rightarrow pr(X, Z))$. L'univers de Herbrand de Φ est a, b.

Définition 27 (Base de Herbrand) *Une base de Herbrand d'un ensemble de formules est l'ensemble des atomes clos qui peuvent être formés en utilisant les constantes de l'univers de Herbrand.* □

À partir de l'exemple précédent, la base de Herbrand de Φ est l'ensemble :
$\{pr(a, a), pr(a, b), pr(b, a), pr(b, b)\}$.

Définition 28 (Interprétation de Herbrand) *Une interprétation de Herbrand d'un ensemble de formules est n'importe quel sous-ensemble de sa base de Herbrand.* □

Toujours sur le même exemple, les interprétations de Herbrand de Φ sont les 16 ensembles (dont les éléments sont valués à 1) :

$\{\}$	$\{pr(a, b), pr(b, a)\}$
$\{pr(a, a)\}$	$\{pr(a, b), pr(b, b)\}$
$\{pr(a, b)\}$	$\{pr(b, a), pr(b, b)\}$
$\{pr(b, a)\}$	$\{pr(a, a), pr(a, b), pr(b, a)\}$
$\{pr(b, b)\}$	$\{pr(a, a), pr(a, b), pr(b, b)\}$
$\{pr(a, a), pr(a, b)\}$	$\{pr(a, a), pr(b, a), pr(b, b)\}$
$\{pr(a, a), pr(b, a)\}$	$\{pr(a, b), pr(b, a), pr(b, b)\}$
$\{pr(a, a), pr(b, b)\}$	$\{pr(a, a), pr(a, b), pr(b, a), pr(b, b)\}$

Définition 29 (Modèle de Herbrand) *Un modèle de Herbrand d'un ensemble de formules Φ est une interprétation de Herbrand de Φ qui est modèle de Φ.* □

Ces modèles de Herbrand sont très utiles puisqu'il permettent, grâce au théorème suivant, de décider plus simplement de la satifiabilité des formules.

Théorème 2 (Théorème de Herbrand) *Un ensemble de formules est satisfiable si et seulement si un de ses modèles de Herbrand le satisfait.* □

Considérons par exemple qu'un ensemble Φ contienne les formules $pr(a, X)$ et $pr(X, Y) \vee pr(Y, X)$. La base de Herbrand de Φ est $\{pr(a, a)\}$ et un modèle de Herbrand satisfaisant Φ est $\{pr(a, a)\}$.

L'intérêt de ce théorème est bien sûr le fait qu'il permet de se limiter à un nombre restreint de modèles lorsque l'on veut vérifier la satisfiabilité d'une formule.

A.3 SLD-résolution

La SLD-résolution est une technique largement utilisée, notamment au sein de Prolog, pour prouver des formules logiques exprimées sous forme de clauses. Nous en présentons dans cette section les opérations basiques, telles que l'unification et la construction de la résolvante, puis terminons en décrivant son principe. On considère dans la suite une clause comme l'ensemble des littéraux (en disjonction) qui la composent ; on peut ainsi les manipuler avec les opérateurs habituels d'union ($\cup$), de soustraction ensembliste ($\setminus$)...

A.3.1 Unification et substitution

Définition 30 (Substitution) *Une substitution est une application de l'ensemble des variables dans l'ensemble des termes. On la note souvent sous forme de liste, de la manière suivante : $\theta = [X_1/t_1, ..., X_n/t_n]$. Elle est utilisée pour remplacer dans un littéral l les variables X_i par les termes t_i ; le littéral obtenu est $l\theta$.* □

On note généralement une substitution θ, ou s'il y en a plusieurs par des minuscules grecques.

Une substitution θ est un *unificateur* de deux littéraux l_1 et l_2 si $l_1\theta = l_2\theta$. Elle sera le *plus grand unificateur* (pgu) si en plus, pour tout unificateur σ de l_1 et l_2, il existe une substitution τ tel que $\sigma = \theta\tau$. Si θ est un pgu de l_1 et l_2, alors le littéral $l = l_1\theta = l_2\theta$ est appelé *plus grande instance* de l_1 et de l_2. Considérons par exemple les deux littéraux $l_1 = f(a, X, Y)$ et $l_2 = f(Z, b, T)$; ceux-ci peuvent s'unifier en $l_1\sigma = l_2\sigma = f(a, b, Y)$ avec $\sigma = [Z/a, X/b, Y/c, T/c]$ ou avec la combinaison du pgu $\theta = [Z/a, X/b, Y/T]$ et de la substitution $\tau = [T/a]$.

A.3.2 Résolvante

Définition 31 (Résolvante) *Soient C_1 et C_2 deux clauses n'ayant pas de variable en commun. Une clause C est une résolvante de C_1 et C_2 si et seulement si les deux conditions suivantes sont remplies :*

1. *il y a des littéraux $l_1 \in C_1$ et $l_2 \in C_2$ tels qu'il existe un pgu θ de l_1 et $\neg l_2$, i.e. $l_1\theta = (\neg l_2)\theta$;*
2. *C s'écrit $(C_1 \setminus \{l_1\})\theta_1 \cup (C_2 \setminus \{l_2\})\theta_2$ avec $\theta = \theta_1\theta_2$.* □

Plus concrètement, dans le cas de clauses définies, le littéral l_1 est un des littéraux du corps de C_1, et la clause C_2, dont l_1 est la tête, représente une définition de la

propriété notée par le prédicat de l_1 (et de l_2). La résolvante est donc la clause C_1 dans laquelle on remplace un littéral par sa définition, c'est-à-dire par le corps de C_2. Ainsi, cette méthode de construction permet d'inférer la résolvante de deux clauses, soit encore, avec les mêmes notations que précédemment $C_1, C_2 \vdash C$.

A.3.3 Résolution de Robinson

L'idée de base du principe de résolution de Robinson (Robinson, 1965) est que l'inconsistance d'un programme logique peut être établie si l'on peut en dériver une contradiction $\square$ ($\square$ représente un ensemble vide de formules ; dériver $\square$ revient donc à dériver faux). S'appuyant sur ce fait, la négation $\neg\psi$ de ce que l'on cherche à démontrer (le but ou la requête en Prolog) est ajoutée à l'ensemble des formules Φ. Si l'on peut montrer $\square$ à partir de $\Phi \cup \{\neg\psi\}$, c'est-à-dire si l'on peut réfuter $\neg\psi$, alors cela signifie que ψ peut être prouvée par Φ.

Ce principe de résolution est celui appliqué dans Prolog. Il s'applique donc à des clauses définies (ayant exactement un littéral dans leurs têtes). Pour choisir la clause aidant à résoudre le but, une fonction de sélection est utilisée : elle consiste simplement à choisir la première clause dont la tête correspond (à une substitution près, c'est-à-dire par unification) au but. Enfin, la clause résultant de l'étape précédente de résolution est utilisée à l'étape suivante (c'est donc une stratégie en profondeur d'abord). Ce sont ces trois caractéristiques qui ont donné son nom à cette technique de résolution, la SLD-résolution (S pour fonction de Sélection, L pour résolution Linéaire et D pour clauses Définies). Il faut noter que cette technique n'utilise qu'une seule et unique règle, appelée $\mathcal{R}$ ci-dessous, pour conduire l'inférence.

Cette technique répond bien à une de deux propriétés attendues des règles d'inférence puisqu'elle est correcte ($\Phi \vdash_{\mathcal{R}} \psi \Rightarrow \Phi \models \psi$). Elle n'est en revanche pas complète (les tautologies ne peuvent pas être dérivées via ce principe de résolution), mais elle est complète du point de vue de la réfutation, *i.e.* si $\Phi \cup \{\neg\psi\} \models \square$ alors $\Phi \cup \{\neg\psi\} \vdash_{\mathcal{R}} \square$.

Annexe B

Algorithme de PLI

Cet annexe présente certains détails techniques et théoriques concernant les expériences rapportées aux chapitres 3 et 4 de ce mémoire. Nous présentons dans un premier temps à travers un extrait du *background knowledge* les définitions des divers prédicats hiérarchisés servant lors de la phase d'inférence des patrons d'extraction. Nous revenons ensuite sur la structure de l'espace de recherche exploré lors de cette phase d'inférence et montrons qu'il s'agit d'un treillis.

B.1 *Background Knowledge*

Voici le *listing* de la partie du *background knowledge* décrivant les informations linguistiques utilisées dans les expériences présentées aux chapitres 3 et 4.

```
%%%%%%%%%%%%%%%%%%%%%%%%%
% background knowledge

% common noun %%%%%%%%%%%%
common_noun( W ) :- plural_common_noun( W ).
common_noun( W ) :- singular_common_noun( W ).
common_noun( W ) :- abstraction( W ).
common_noun( W ) :- event( W ).
common_noun( W ) :- group( W ).
common_noun( W ) :- psychological_feature( W ).
common_noun( W ) :- state( W ).
common_noun( W ) :- entity( W ).
common_noun( W ) :- location( W ).
plural_common_noun(W):- tags(W,tc_noun_pl, _ ).
singular_common_noun(W):- tags(W,tc_noun_sg, _ ).
abstraction( W ) :- attribute( W ).
abstraction( W ) :- measure( W ).
abstraction( W ) :- relation( W ).
event( W ) :- natural_event( W ).
event( W ) :- act(W).
```

```
event( W ) :- phenomenon(W ).
natural_event( W ) :- tags(W, _, ts_hap ).
phenomenon( W ) :- tags(W, _, ts_phm ).
phenomenon( W ) :- process( W ).
process( W ) :- tags(W, _, ts_pro ).
act( W ) :- tags(W, _, ts_act ).
act( W ) :- human_activity( W ).
human_activity( W ) :- tags(W, _, ts_acy ).
group( W ) :- tags(W, _, ts_grp ).
group( W ) :- social_group( W ).
social_group( W ) :- tags(W, _, ts_grs ).
psychological_feature( W ) :- tags(W, _, ts_psy ).
state( W ) :- tags(W, _, ts_sta ).
entity( W ) :- tags(W, _, ts_ent ).
entity( W ) :- body_part( W ).
entity( W ) :- causal_agent( W ).
entity( W ) :- object( W ).
body_part( W ) :- tags(W, _, ts_prt ).
object( W ) :- tags(W, _, ts_pho ).
object( W ) :- artefact( W ).
object( W ) :- part( W ).
object( W ) :- substance( W ).
part( W ) :- tags(W, _, ts_por ).
location( W ) :- tags(W, _, ts_loc ).
location( W ) :- point(W).
point( W ) :- tags(W, _, ts_pnt ).
point( W ) :- position( W ).
position( W ) :- tags(W, _, ts_pos ).
attribute( W ) :- tags(W, _, ts_atr ).
attribute( W ) :- form( W ).
attribute( W ) :- property( W ).
form( W ) :- tags(W, _, ts_frm ).
property( W ) :- tags(W, _, ts_pty ).
measure( W ) :- tags(W, _, ts_mea ).
measure( W ) :- definite_quantity( W ).
measure( W ) :- unit( W ).
time_unit( W ) :- tags(W, _, ts_tme ).
definite_quantity( W ) :- tags(W, _, ts_qud ).
unit( W ) :- tags(W, _, ts_unt ).
unit( W ) :- time_unit( W ).
relation( W ) :- tags(W, _, ts_rel ).
relation( W ) :- communication( W ).
communication( W ) :- tags(W, _, ts_com ).
causal_agent( W ) :- tags(W, _, ts_agt ).
causal_agent( W ) :- human( W ).
human( W ) :- tags(W, _, ts_hum ).
artefact( W ) :- tags(W, _, ts_art ).
artefact( W ) :- instrument(W).
instrument( W ) :- tags(W, _, ts_ins ).
```

```
instrument( W ) :- container( W ).
container( W ) :- tags(W, _, ts_cnt ).
substance( W ) :- tags(W, _, ts_sub ).
substance( W ) :- chemical_compound( W ).
substance( W ) :- stuff( W ).
chemical_compound( W ) :- tags(W, _, ts_chm ).
stuff( W ) :- tags(W, _, ts_stu ).

% verb %%%%%%%%%%%%%%%%%
verb( W ) :- infinitive( W ).
verb( W ) :- participle( W ).
verb( W ) :- conjugated( W ).
verb( W ) :- action_verb( W ).
verb( W ) :- state_verb( W ).
verb( W ) :- modal_verb( W ).
verb( W ) :- temporality_verb( W ).
verb( W ) :- possesion_verb( W ).
verb( W ) :- auxiliary( W ).
infinitive( W ) :- tags(W, tc_verb_inf, _ ).
participle( W ) :- present_participle( W ).
participle( W ) :- past_participle( W ).
present_participle( W ) :- tags(W, tc_verb_prp, _ ).
past_participle( W ) :- tags(W, tc_verb_pap, _ ).
conjugated( W ) :- conjugated_plural(W).
conjugated( W ) :- conjugated_singular(W).
conjugated_plural( W ) :- tags(W, tc_verb_pl, _ ).
conjugated_singular( W ) :- tags(W, tc_verb_sg, _ ).
action_verb( W ) :- cognitive_action_verb( W ).
action_verb( W ) :- physical_action_verb( W ).
cognitive_action_verb( W ) :- tags(W, _, ts_acc ).
physical_action_verb( W ) :- tags(W, _, ts_acp ).
state_verb( W ) :- tags(W, _, ts_eta ).
modal_verb( W ) :- tags(W, _, ts_mod ).
temporality_verb( W ) :- tags(W, _, ts_tem ).
possesion_verb( W ) :- tags(W, _, ts_posv ).
auxiliary( W ) :- tags(W, _, ts_aux ).

% preposition %%%%%%%%
preposition( W ) :- tags(W, tc_prep, _ ).
preposition( W ) :- spat_preposition( W ).
preposition( W ) :- goal_preposition( W ).
preposition( W ) :- temp_preposition( W ).
preposition( W ) :- manner_preposition( W ).
preposition( W ) :- rel_preposition( W ).
preposition( W ) :- caus_preposition( W ).
preposition( W ) :- neg_preposition( W ).
preposition( W ) :- en_preposition( W ).
preposition( W ) :- sous_preposition( W ).
preposition( W ) :- a_preposition( W ).
```

```
preposition( W ) :- de_preposition( W ).
spat_preposition( W ) :- tags(W, _, ts_rspat ).
goal_preposition( W ) :- tags(W, _, ts_rpour ).
temp_preposition( W ) :- tags(W, _, ts_rtemp ).
manner_preposition( W ) :- tags(W, _, ts_rman ).
rel_preposition( W ) :- tags(W, _, ts_rrel ).
caus_preposition( W ) :- tags(W, _, ts_rcaus ).
neg_preposition( W ) :- tags(W, _, ts_rneg ).
en_preposition( W ) :- tags(W, _, ts_ren ).
sous_preposition( W ) :- tags(W, _, ts_rsous ).
a_preposition( W ) :- tags(W, _, ts_ra ).
de_preposition( W ) :- tags(W, _, ts_rde ).

% adjective %%%%%%%%%%%
adjective( W ) :- singular_adjective( W ).
adjective( W ) :- plural_adjective( W ).
adjective( W ) :- verbal_adjective( W ).
adjective( W ) :- comparison_adjective( W ).
adjective( W ) :- concrete_prop_adjective( W ).
adjective( W ) :- abstract_prop_adjective( W ).
adjective( W ) :- nominal_adjective( W ).
singular_adjective( W ) :- tags(W, tc_adj_sg, _ ).
plural_adjective( W ) :- tags(W, tc_adj_pl, _ ).
verbal_adjective( W ) :- tags(W, tc_verb_adj, _ ).
comparison_adjective( W ) :- tags(W, _, ts_acomp ).
concrete_prop_adjective( W ) :- tags(W, _, ts_apty ).
abstract_prop_adjective( W ) :- tags(W, _, ts_apa ).
nominal_adjective( W ) :- tags(W, _, ts_anom ).

% pronoun %%%%%%%%%%%%%
pronoun( W ):- rel_pronoun( W ).
pronoun( W ):- non_rel_pronoun( W ).
rel_pronoun(W) :- tags(W, tc_pron_rel, _ ).
non_rel_pronoun(W) :- tags(W, tc_pron, _ ).

% others %%%%%%%%%%%%%
proper_noun( W ) :- tags(W, _, ts_nompropre ).
proper_noun( W ) :- tags(W, _, ts_numero ).
coordinating_conjunction(W) :- tags(W, _, ts_rconj ).
subordinating_conjunction(W) :- tags(W, _, ts_subconj ).
bracket( W ) :- tags(W, _, ts_paro ).
bracket( W ) :- tags(W, _, ts_parf ).
ponctuation( W ) :- comma( W ).
ponctuation( W ) :- colon( W ).
ponctuation( W ) :- dot( W ).
ponctuation( W ) :- tags(W, tc_wpunct, _ ).
comma( W ) :- tags(W, _, ts_virg ).
colon( W ) :- tags(W, _, ts_ponct ).
dot( W ) :- tags(W, _, ts_punct ).
```

```
figures( W ) :- tags(W, _, ts_quant ).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% order
%
precedes(V, N) :- distances(N, V, X, _ ), 0<X.
precedes(N, V) :- distances(N, V, X, _ ), 0>X.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% distances in verbs
%
near_verb(N, V) :- distances(N, V, _, 1 ).
near_verb(N, V) :- distances(N, V, _, -1 ).
far_verb(N, V) :- distances(N, V, _, X ), -1>X , -3<X.
far_verb(N, V) :- distances(N, V, _, X ), 1<X , X<3.
very_far_verb(N, V) :- distances(N, V, _, X ), -2>X.
very_far_verb(N, V) :- distances(N, V, _, X ), X>2.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% distances in words
%
contiguous(N, V) :- distances(N, V, 1, _ ).
contiguous(N, V) :- distances(N, V, -1, _ ).
near_word(N, V) :- distances(N, V, X, _ ), -1>X , -4<X.
near_word(N, V) :- distances(N, V, X, _ ), 1<X , X<4.
far_word(N, V) :- distances(N, V, X, _ ), -3>X, -8<X.
far_word(N, V) :- distances(N, V, X, _ ), X>3, X<8.
very_far_word(N, V) :- distances(N, V, X, _ ), -7>X.
very_far_word(N, V) :- distances(N, V, X, _ ), X>7.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% other predicates
suc(X, Y) :- pred(Y, X).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% information about examples
tags(m15278_1_deb, tc_vide, ts_vide ).
tags(m15278_1, tc_verb_inf, ts_tem ).
pred(m15278_1, m15278_1_deb ).
...
```

B.2 Espace de recherche des hypothèses

L'espace des clauses ordonnées par θ_{OI}-subsomption n'est pas en général un treillis alors que c'est le cas sous θ-subsomption (Semeraro *et al.*, 1994). Cependant, nous montrons dans la section suivante qu'un tel espace est bien un treillis si certaines conditions sur les clauses sont réunies. Nous étendons ce résultat à l'espace de recherche

qui est exploré pour les besoins de notre application lorsque celui-ci est ordonné par la θ_{NV}-subsomption.

B.2.1 Treillis des hypothèses sous θ_{OI}-subsomption

Un ensemble quasi-ordonné par θ_{OI}-subsomption (voir définition 5 en page 97) n'est en général pas un treillis car les infima et suprema de deux clauses ne sont pas uniques. Cependant, considérons le cas des clauses liées et déterministes (voir section 3.1.2.1) et un espace borné en bas par la clause *bottom* ($\bot$). Dans un tel espace, nous montrons que l'infimum et le supremum de deux clauses quelconques sont uniques. Dans cette première section, pour alléger la notation, $C \succeq D$ (respectivement $C \sim D$) signifie que C est plus générale que (resp. équivalente à) D au regard de l'ordre induit par la θ_{OI}-subsomption.

Propriété 8 *Pour toutes clauses C et D de l'espace des clauses liées et déterministes ordonné par la θ_{OI}-subsomption, si $C \succeq D$ alors la substitution injective θ telle que $C\theta \subseteq D$ est unique.* □

Preuve

Reductio ad absurdum.
Supposons qu'il existe deux substitutions injectives θ_1 et θ_2 telles que $C\theta_1 \subseteq D$ et $C\theta_2 \subseteq D$. Puisque θ_1 et θ_2 sont injectives, $C\theta_1$ et $C\theta_2$ diffèrent seulement par le nommage des variables. C et D sont des clauses liées, ce qui signifie qu'il existe un littéral $l \in C$ tel que $l\theta_1 \in D$, $l\theta_2 \in D$ et $l\theta_1 \neq l\theta_2$ où les variables d'entrée de l sont identiques entre $l\theta_1$ et $l\theta_2$ et les variables de sortie différentes. Cela contredit le fait que tous les littéraux sont déterministes. □

Propriété 9 *Dans l'espace des clauses liées et déterministes ordonné par θ_{OI}-subsomption et borné en bas par la clause $\bot$, le supremum de deux clauses est unique.* □

Preuve

Reductio ad absurdum.
Supposons que A_1 et A_2 soient deux suprema différents de C_1 et C_2. A_1, A_2, C_1 et C_2 sont plus générales que $\bot$; il existe donc une unique $\theta_\bot^{A_1}$ telle que $A_1\theta_\bot^{A_1} \subseteq \bot$ (prop. 8). De la même façon, nous avons $\theta_\bot^{A_2}$, $\theta_\bot^{C_1}$, et $\theta_\bot^{C_2}$ uniques telles que $A_2\theta_\bot^{A_2} \subseteq \bot$, $C_1\theta_\bot^{C_1} \subseteq \bot$ et $C_2\theta_\bot^{C_2} \subseteq \bot$.

A_1 est supremum pour C_1 donc $A_1 \succeq C_1\theta_\bot^{C_1}$ puisque $C_1 \sim C_1\theta_\bot^{C_1}$. Donc, il existe θ_1 telle que $A_1\theta_1 \subseteq C_1\theta_\bot^{C_1}$. Par ailleurs, $C_1\theta_\bot^{C_1} \subseteq \bot$ et donc $A_1\theta_1 \subseteq \bot$, ce qui signifie que $\theta_1 = \theta_\bot^{A_1}$ (prop. 8). Nous avons donc $A_1\theta_\bot^{A_1} \subseteq C_1\theta_\bot^{C_1}$ et de manière similaire, $A_1\theta_\bot^{A_1} \subseteq C_2\theta_\bot^{C_2}$, $A_2\theta_\bot^{A_2} \subseteq C_1\theta_\bot^{C_1}$ et $A_2\theta_\bot^{A_2} \subseteq C_2\theta_\bot^{C_2}$.

Posons $S = A_1\theta_\bot^{A_1} \cup A_2\theta_\bot^{A_2}$. Ainsi, $S \subseteq C_1\theta_\bot^{C_1}$ et $S \subseteq C_2\theta_\bot^{C_2}$. Cela signifie que $S \succeq C_1$ et $S \succeq C_2$ puisque $C_1\theta_\bot^{C_1} \sim C1$ et $C_2\theta_\bot^{C_2} \sim C2$. Par ailleurs, $A_1 \succeq S$, $A_2 \succeq S$ et

$S \nsim A_1$, $S \nsim A_2$ car $A_1 \nsim A_2$. Cela contredit le fait que A_1 et A_2 sont suprema pour C_1 et C_2. □

Propriété 10 *Dans l'espace des clauses liées et déterministes ordonné par la θ_{OI}-subsomption et borné en bas la clause bottom $\perp$, l'infimum de deux clauses quelconques est unique.* □

Preuve

Même démonstration que pour le supremum avec C_1 et C_2 deux infima pour A_1 et A_2, et en considérant $I = C_1\theta_{\perp}^{C_1} \cap C_2\theta_{\perp}^{C_2}$. □

D'après les propositions 9 et 10, nous pouvons conclure que l'espace des clauses liées déterministes ordonné par la θ_{OI}-subsomption et borné en bas par une clause *bottom* $\perp$ est un treillis.

B.2.2 Treillis des hypothèses sous θ_{NV}-subsomption

Comme pour la θ_{OI}-subsomption, nous montrons que, dans le cadre particulier de notre application, l'espace de recherche des hypothèses ordonné par la θ_{NV}-subsomption est un treillis. Dans la suite, B représente le *background knowledge* utilisé pour notre tâche d'apprentissage, $\succeq$ et $\sim$ dénote la θ_{NV}-subsomption telle que définie en page 99.

Propriété 11 *Dans l'espace des clauses bien formées ordonné par la θ_{NV}-subsomption, pour toutes clauses C et D, si $C \succeq D$ alors la substitution injective θ telle que $f(C)\theta \subseteq D$ (avec f telle que $\forall l \in C,\ \ B, f(l) \models l$) est unique.* □

Preuve

Même preuve que pour la proposition 8 en considérant C^{chain}, le sous-ensemble de C contenant son littéral tête et ses pred/2, et en notant que C^{chain} contient toutes les variables de C et qu'au regard de notre *background knowledge* particulier, pour toute f telle que $f(C)\theta \subseteq D$ avec f telle que $\forall l \in C,\ \ B, f(l) \models l$, nécessairement $\forall l \in C^{chain},\ f(l) = l$. □

Propriété 12 *Dans l'espace des clauses bien formées ordonné par θ_{NV}-subsomption, le supremum de deux clauses quelconques est unique.* □

Preuve

Reductio ad absurdum.
Supposons que A_1 et A_2 soient deux suprema différents pour C_1 et C_2. A_1 est plus général que $\perp$; ainsi $\exists\theta_{\perp}^{A_1}$ injective et $f_{\perp}^{A_1}$ telle que $f_{\perp}^{A_1}(A_1)\theta_{\perp}^{A_1} \subseteq \perp$ et $\theta_{\perp}^{A_1}$ est unique (prop. 11). De la même manière, nous avons $\theta_{\perp}^{A_2}$, $\theta_{\perp}^{C_1}$, et $\theta_{\perp}^{C_2}$ uniques.

A_1 est supremum pour C_1, donc il existe θ_1 et f_1 telles que $f_1(A_1)\theta_1 \subseteq C_1\theta_\perp^{C_1}$. Ainsi, avec A_1^{chain} défini comme ci-dessus, $A_1^{chain}\theta_1 \subseteq C_1^{chain}\theta_\perp^{C_1}$ puisque $f(A_1^{chain}) = A_1^{chain}$. De la même manière, on a $C_1^{chain}\theta_\perp^{C_1} \subseteq \perp$. Nous avons donc $A_1^{chain}\theta_1 \subseteq C_1^{chain}\theta_\perp^{C_1} \subseteq \perp$ et alors, d'après la prop. 11, $\theta_1 = \theta_\perp^{A_1}$. Finalement, nous avons $f_1(A_1)\theta_\perp^{A_1} \subseteq C_1\theta_\perp^{C_1}$ et de la même manière, il existe f_2, f_3 et f_4 telles que $f_2(A_1)\theta_\perp^{A_1} \subseteq C_2\theta_\perp^{C_2}$, $f_3(A_2)\theta_\perp^{A_2} \subseteq C_1\theta_\perp^{C_1}$ et $f_4(A_2)\theta_\perp^{A_2} \subseteq C_2\theta_\perp^{C_2}$.

Notons $S = A_1\theta_\perp^{A_1} \cup A_2\theta_\perp^{A_2} \setminus \{l_1 \mid l_1, l_2 \in (A_1\theta_\perp^{A_1} \cup A_2\theta_\perp^{A_2}), l_1 \neq l_2, \text{ et } B, l_2 \models l_1\}$. S est une hypothèse bien formée et par construction $S \preceq A_1\theta_\perp^{A_1}$ et $S \preceq A_2\theta_\perp^{A_2}$. De plus, puisque $A_1\theta_\perp^{A_1} \sim A_1$ et $A_2\theta_\perp^{A_2} \sim A_2$, alors $S \preceq A_1$ et $S \preceq A_2$. Nous définissons f_5 telle que $\forall l_i^S \in S, f_5(l_i^S) = f_1(l_i^S)$ si $l_i^S \in A_1\theta_\perp^{A_1}$ et $f_5(l_i^S) = f_3(l_i^S)$ sinon. Similairement, nous définissons f_6 telle que $\forall l_i^S \in S, f_6(l_i^S) = f_2(l_i^S)$ si $l_i^S \in A_1\theta_\perp^{A_1}$ et $f_6(l_i^S) = f_4(l_i^S)$ sinon. Ainsi, $f_5(S) \subseteq C_1\theta_\perp^{C_1}$ et $f_6(S) \subseteq C_2\theta_\perp^{C_2}$, ce qui signifie que $S \succeq C_1\theta_\perp^{C_1}$ et $S \succeq C_2\theta_\perp^{C_2}$. Donc, $S \succeq C_1$ et $S \succeq C_2$. Cela contredit le fait que A_1 et A_2 sont suprema pour C_1 et C_2. □

Propriété 13 *Dans l'espace des clauses bien formées ordonné par θ_{NV}-subsomption et au regard de notre background knowledge, l'infimum de deux clauses quelconques est unique.* □

Preuve

Même preuve que pour l'unicité du supremum avec C_1 et C_2 deux infima de A_1 et A_2 et considérer $I = (C_1\theta_\perp^{C_1} \cap C_2\theta_\perp^{C_2}) \cup \{l_1 \mid l_1, l_2 \in C_1\theta_\perp^{C_1} \cup C_2\theta_\perp^{C_2}, l_1 \neq l_2 \text{ et } B, l_2 \models l_1\}$. □

D'après les propositions 12 et 13, nous pouvons conclure que notre espace d'hypothèses ordonné par θ_{NV}-subsomption est un treillis.

Annexe C

Données de la campagne Amaryllis

Afin de mesurer l'intérêt des relations qualia pour l'extension de requêtes, nous utilisons un des jeux de données de la campagne d'évaluation des systèmes de recherche d'information Amaryllis. Ce jeu de test se compose d'une collection de documents, d'un ensemble de requêtes et de leurs réponses attendues. La section suivante présente le corpus de textes constituant la collection de documents et en donne un extrait ; la section C.2 décrit les requêtes et en fournit également quelques exemples.

C.1 Extraits du corpus OFIL

Le corpus utilisé lors de la campagne d'évaluation Amaryllis est une collection d'articles du Monde. Chaque article représente un document, et est renseigné par des balises SGML selon les recommandations de la *Text Encoding Initiative* (TEI) délimitant le numéro de l'article (son identifiant), son titre, et le corps.

Le corpus d'entraînement sur lequel tous nos tests ont été effectués se compose ainsi de 11 016 articles. Nous en présentons ci-dessous quelques exemples.

<TEI.2><text> <body> <title> Le président du patronat allemand redoute une forte augmentation du chômage. </title> M. Klaus Murmann, président du patronat allemand, a déclaré, mardi 29 décembre, que 5,5 millions d'Allemands seraient au chômage complet ou partiel à la fin de 1993, si la politique salariale ne changeait pas. De nouveaux postes de travail ne pourront être créés, a déclaré le patron des patrons allemands, si une rupture ne se produit pas " avec le cours actuel du confort, de la mentalité d'un État prospère et de l'esprit du chacun pour soi dans les négociations " . M. Murmann a fermement soutenu le projet du chancelier Helmut Kohl d'un pacte de solidarité, impliquant notamment des augmentations salariales modérées. (AFP.) </body></text></TEI.2>

<TEI.2><text> <body> <title> Indicateurs. </title> États-Unis : activité : + 0,8 % pour l'indice composite en novembre. L'indice composite, qui

regroupe onze indicateurs de l'économie américaine, a augmenté de 0,8 % en novembre, a annoncé le département du commerce mercredi 30 décembre. Il s'agit de la seconde hausse mensuelle consécutive (+ 0,5 % en octobre) et de la plus forte progression de cet indice - un bon indicateur de l'évolution à court terme - depuis le mois de janvier 1992. En novembre, ce sont les commandes aux entreprises, les ventes au détail et surtout la confiance des consommateurs, qui ont joué. Grande-Bretagne : faillites : + 31 % en 1992. Les faillites ont augmenté de près d'un tiers en 1992 en Grande-Brtagne, après une augmentation de deux tiers l'année précédente, mais la situation devrait s'améliorer sensiblement en 1993, selon deux études publiées mercredi 30 décembre. Près de 63 000 entreprises ont sombré pendant l'année écoulée (+ 31 % par rapport à 1991), pour la plupart de petite taille, a rapporté le cabinet d'information financière Dun and Bradstreet. Pour les sociétés de taille plus importante, les faillites n'ont augmenté que de 11 %.</**p**></**div**> </**body**></**text**></**TEI.2**>

<**TEI.2**><**text**> <**body**> <**div type='article' id=2271456**> <**title**> Vie des entreprises : chiffres et mouvements. </**title**> <**p**>Résultat : Renault : un dernier trimestre " beaucoup moins bon ", selon son PDG. M. Louis Schweitzer, PDG de Renault, a déclaré, mardi 29 décembre à Rennes, que si en 1992 " l'entreprise fera des bénéfices honorables ", " le dernier trimestre sera beaucoup, beaucoup moins bon et favorable " . M. Schweitzer s'est cependant refusé à donner des chiffres. Il a également affirmé que " les temps qui sont devant nous seront plus difficiles " . Regroupement : Accor regroupe ses hôtels 2 étoiles. Le groupe Accor a rassemblé dans une même structure juridique ses hôtels deux étoiles (enseignes Ibis et Arcade), poursuivant ainsi le mouvement de restructuration de ses activités, qui avait été rendu nécessaire à la suite de l'acquisition du groupe des Wagons-Lits. Après l'absorption de la société Sephi (Arcade), détenue à 50 % par la Compagnie des Wagons-Lits, la société Sphère (Ibis) rassemble, désormais, 391 hôtels dans 16 pays. En contre-partie, les Wagons-Lits, filiale à 70 % d'Accor, ont obtenu une participation de 6 % dans Sphère. Ces restructurations ont été ratifiées, mardi 29 décembre, par une assemblée générale extraordinaire des actionnaires de Sphère. Renfort : recapitalisation et allégement de bilan pour la banque Duménil-Leblé. Afin de permettre à la Banque Duménil Leblé de respecter ses ratios prudentiels, Cerus et la Société financière de Genève réalisent une avance d'actionnaires d'un montant global de 430 millions de francs. Cette avance d'actionnaires, actuellement versée sur un compte bloqué, sera transformée en augmentation de capital effective, dès l'approbation des comptes de la Banque Duménil-Leblé pour l'exercice 1992. Cerus (Compagnies européennes réunies, holding européen de l'homme d'affaires italien Carlo De Benedetti) contribue à cette avance à hauteur de 221 millions de francs, la Société financière de Genève apportant, de son côté, 209 millions de francs. Licenciement : Sopalin (Kimberly-Clark) : la justice ordonne la suspension des licenciements. Le tribunal de grande instance de Rouen a ordonné la suspension de la procédure de licenciement engagée à l'encontre de 312 salariés de l'usine Sopalin de Rouen (groupe Kimberly-Clark). Annoncés le 19 novembre, ces licenciements s'inscrivent dans un plan de restructuration de ce groupe américain spécialisé dans la transformation du papier. Il prévoit notamment le transfert vers d'autres sites européens de la plupart des fabrications de l'usine de Rouen (serviettes périodiques, essuie-tout, papier hygiènique...), qui conserverait uniquement les mouchoirs jetables de marque Kleenex. Dans ses attendus, le tribunal a constaté " le défaut de communication " aux représentants du personnel d'une étude concernant les coûts comparés des fabrications dans les différentes usines du groupe. Rachats : BSN rachète Verdome à Perrier pour 200 millions de francs. Le groupe agroalimentaire BSN a conclu un accord avec Perrier, filiale du Suisse Nestlé, pour lui racheter sa participation de 95,8 % dans le fabricant de bouteilles en verre

Verdome, au prix d'environ 200 millions de francs. La transaction sera réalisée pour un prix de 556 francs par action et sera suivie d'une garantie de cours au même prix. En 1991, l'entreprise a produit environ 200 000 tonnes de bouteilles en verre dans son usine de Puy-Guillaume (Puy-de-Dôme) et réalisé un chiffre d'affaires de 505 millions de francs. Son bénéfice a été de 10 millions. Le groupe Marland Distribution racheté par un fonds d'investissement. Le groupe Marland Distribution, rebaptisé Kléber 55 SA après la démission récente de M. François Marland pour raisons de santé, va être racheté par un fonds d'investissement, a annoncé la société mercredi 23 décembre. Ce fonds, dont l'identité n'a pas été révélée, a été conseillé par la banque Colbert, mais ni cette banque ni Altus Finances (maison-mère de la banque Colbert) ne font partie du nouveau montage financier. Ce fonds a affecté à l'opération 1,25 milliard de franc et a confié à M. Jean-Pierre Andrevon, ancien directeur général de Pinault puis de Point P, la présidence du groupe. Le plan de reprise prévoit la création de quatre holdings regroupant les six sociétés acquises, qui vont de la distribution grossiste alimentaire (enseigne Disco) à la fabrication de panneaux publicitaires (Ma'Pub) et à la fabrication et vente de vêtements (Financière du cuir). Notation : Philips : Moody's réduit la note financière de la multinationale néerlandaise. La firme de notation financière Moody's a réduit, mardi 22 décembre, la note financière de la dette à long terme du groupe électronique néerlandais Philips, qui passe de A3 à Baa1. Cette décision touche environ 1,5 milliard de dollars de dettes du groupe. Moody's a justifié sa décision par le caractère incertain du niveau de cash-flow de Philips.</**p**></**div**> </**body**></**text**></**TEI.2**>

C.2 Requêtes du corpus OFIL

Lors de la campagne Amaryllis, 26 requêtes ont été utilisées dans la phase d'entraînement. Ce sont ces requêtes qui sont utilisées lors de nos expérimentations. Nous en présentons ci-dessous quelques exemples. Le champ **record** contient l'ensemble de la question dont le numéro est donné par le champ **num**. Le champ **dom** indique le thème général de la question posée, **suj** précise le sujet sur laquelle elle porte, et **que** est la question proprement dite. Le texte repéré par la balise **cinf** explicite quels types de documents sont attendus en réponses et **ccept** annonce une liste de concepts proches de la question dont chaque item est balisé par **c**. Dans notre cas, seul le contenu des champs **suj** est utilisé pour interroger la base documentaire.

<**record**>
<**num**>1</**num**>
<**dom**>International</**dom**>
<**suj**>La séparation de la Tchécoslovaquie</**suj**>
<**que**>Pourquoi et comment avoir divisé la Tchécoslovaquie et quelles ont été les répercussions économiques et sociales?</**que**>
<**cinf**>Prendre en compte les différentes versions présentées</**cinf**>
<**ccept**>
  <**c**>Partition de la Tchécoslovaquie</**c**>
  <**c**>Causes et modalités de la partition</**c**>
  <**c**>Création de la Slovaquie et de la République Tchèque</**c**>

<**c**>Points de vue</**c**>
<**c**>Économie</**c**>
</**ccept**>
</**record**>

<**record**>
<**num**>2</**num**>
<**dom**>International</**dom**>
<**suj**>Le conflit yougoslave</**suj**>
<**que**>Comment ont été traités les civils pendant le conflit ?</**que**>
<**cinf**>Les documents pertinents devront préciser les conditions de vie et les sévices subis par les populations civiles, de même que l'aide qui leur a été apportée</**cinf**>
<**ccept**>
<**c**>Guerre en ex-Yougoslavie</**c**>
<**c**>Conditions des civils</**c**>
<**c**>Viols systématiques en Bosnie</**c**>
<**c**>Serbes, Musulmans, Croates</**c**>
<**c**>Victimes</**c**>
</**ccept**>
</**record**>

<**record**>
<**num**>3</**num**>
<**dom**>Justice</**dom**>
<**suj**>Le sang contaminé</**suj**>
<**que**>Quel a été le résultat de l'enquête dans l'affaire du sang contaminé et comment a réagi l'opinion publique ?</**que**>
<**cinf**>Les documents pertinents devront prendre en compte que l'affaire du sang contaminé est un drame national et que des centaines de personnes ont été infectées par le virus du SIDA. Ils doivent également préciser que le droit pénal français permet d'oublier les délits après un délai de trois ans.</**cinf**>
<**ccept**>
<**c**>Sang contaminé</**c**>
<**c**>France</**c**>
<**c**>Membre du gouvernement</**c**>
<**c**>SIDA</**c**>
<**c**>Victimes</**c**>
</**ccept**>
</**record**>

...

<**record**>
<**num**>26</**num**>

<**dom**>Société française</**dom**>
<**suj**>La drogue en France</**suj**>
<**que**>La consommation de la drogue en France est-elle dans une phase de croissance ?</**que**>
<**cinf**>Les documents recherchés concerneront à la fois les saisies de drogue importée et les arrestations de trafiquants</**cinf**>
<**ccept**>
    <**c**>Drogue</**c**>
    <**c**>Cocaïne</**c**>
    <**c**>Marijuana</**c**>
    <**c**>Trafiquants</**c**>
    <**c**>Saisie</**c**>
    <**c**>Dealer</**c**>
    <**c**>Haschich</**c**>
</**ccept**>
</**record**>

Index

Références

Abney S. (1990). Rapid Incremental Parsing with Repair. In *Proceedings of the 6th New OED Conference: Electronic Text Research*, Waterloo, Ontario, Canada.

Abney S. (2002). Bootstrapping. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, ACL'02*, Philadelphie, Pennsylvanie, États-Unis.

Agarwal R. (1995). *Semantic Feature Extraction from Technical Texts with Limited Human Intervention.* PhD thesis, Mississippi State University, États-Unis.

and G. S. (1975). A Theory of Term Importance in Automatic Text Analysis. *Journal of the American Society for Information Science*, **26**(1), 33–44.

Armstrong S. (1996). Multext: Multilingual Text Tools and Corpora. In H. Feldweg & W. Hinrichs, Eds., *Lexikon und Text*, p. 107–119. Tübingen: Niemeyer.

Assadi H. (1998). *Construction d'ontologies à partir de textes techniques - Applications aux systèmes documentaires.* Thèse de doctorat, Université de Paris VI, France.

Badea L. (2000). Perfect Refinement Operators can be Flexible. In *Proceedings of the 14th European Conference on Artificial Intelligence, EACL'00*, Berlin, Allemagne.

Badea L. & Stanciu M. (1999). Refinement Operator can be (Weakly) Perfect. In *Proceedings of the 9th International Conference on Inductive Logic Programming, ILP-99*, Bled, Slovénie.

Benveniste É. (1974). *Problèmes de linguistique générale*, volume 2. Gallimard.

Bergadano F., Giordana A. & Saitta L. (1988). Automated Concept Acquisition in Noisy Environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **10**(4), 555–578.

Besançon R. (2001). *Intégration de connaissances syntaxiques et sémantiques dans les représentations vectorielles des textes. Application au calcul de similarité sémantique dans le cadre du modèle DSIR.* Thèse de doctorat, École polytechnique fédérale de Lausanne, Suisse.

Beyer K. S., Goldstein J., Ramakrishnan R. & Shaft U. (1999). When Is "Nearest Neighbor" Meaningful ? In *Proceedings of the 7th International Conference on Database Theory, ICDT'99*, Jérusalem, Israël.

Blockeel H. (1998). *Top-down Induction of First-Order Logical Decision Trees.* PhD thesis, Université Catholique de Louvain, Belgique.

Blockeel H. & De Raedt L. (1998). Top-down Induction of First-Order Logical Decision Trees. *Artificial Intelligence*, **101**(1-2), 285–297.

Blum A. & Mitchell T. (1998). Combining Labeled and Unlabeled Data with Co-training. In *COLT: Proceedings of the Workshop on Computational Learning Theory*, Madisson, Wisconsin, États-Unis.

Bouaud J., Habert B., Nazarenko A. & Zweigenbaum P. (1997). Regroupements issus de dépendances syntaxiques en corpus : catégorisation et confrontation avec deux modélisations conceptuelles. In *Actes de Ingénierie de la Connaissance*, Roscoff, France.

Bouillon P., Baud R. H., Robert G. & Ruch P. (2000a). Indexing by Statistical Tagging. In *Actes des 5es Journées internationales d'analyse statistique des données textuelles, JADT'00*, Lausanne, Suisse.

P. Bouillon & F. Busa, Eds. (2001). *Generativity in the Lexicon.* CUP:Cambridge.

Bouillon P., Fabre C., Sébillot P. & Jacqmin L. (2000b). Apprentissage de ressources lexicales pour l'extension de requêtes. *TAL (Traitement automatique des langues), numéro spécial Traitement automatique des langues pour la recherche d'information*, **41**(2), 367–393.

P. Bouillon & K. Kanzaki, Eds. (2001). *Proceedings of the 1st International Workshop on the Generative Approaches to the Lexicon, GL'01*, Genève, Suisse.

P. Bouillon & K. Kanzaki, Eds. (2003). *Proceedings of the 2nd International Workshop on the Generative Approaches to the Lexicon, GL'03*, Genève, Suisse.

Bouillon P. & Pustejovsky J. (1995). Aspectual Coercion and Logical Polysemy. *Journal of Semantics*, **12**, 133–162.

Bourigault D. (1992). *Lexter*, un logiciel d'extraction de terminologie. In *Actes du 2ème Colloque International de TermNet*, Avignon, France.

Bourigault D. (1994). *Acquisition de terminologie.* Thèse de doctorat, École des Hautes Études en Sciences Sociales, Paris, France.

Bourigault D. (2002). Analyse distributionnelle étendue pour la construction d'ontologies à partir de corpus. In *Actes de la conférence Traitement automatique des langues naturelles, TALN'02*, Nancy, France.

Bourigault D. & Condamines A. (1999). Alternance nom/verbe : explorations en corpus spécialisés. In B. Victorri & J. François, Eds., *Sémantique du lexique verbal*, Cahiers de l'Elsap, Caen, France.

Bourigault D. & Fabre C. (2000). Approche linguistique pour l'analyse syntaxique de corpus. *Cahiers de Grammaires*, **25**, 131–151.

Bourigault D. & Jacquemin C. (2000). Construction de ressources terminologiques. In J.-M. Pierrel, Ed., *Ingénierie des langues*, chapitre 9, p. 215–223. Hermès.

Bourigault D. & Slodzian M. (1999). Pour une terminologie textuelle. *Terminologies nouvelles*, **19**, 29–32.

Breiman L. (1996). Bagging Predictors. *Machine Learning*, **24**(2), 123–140.

Brill E. (1992). A Simple Rule-Based Part of Speech Tagger. In *Proceedings of the 3rd Conference on Applied Natural Language Processing, ANLP92*, Trento, Italie.

Brill E. (1994). Some Advances in Transformational-Based Part of Speech Tagging. In *Proceedings of the 12th National Conference on Artificial Intelligence, AAAI'94*, Seattle, Washington, États-Unis.

Brill E. (2000). A Closer Look at the Automatic Induction of Linguistic Knowledge. In J. Cussens & S. Džeroski, Eds., *Learning Language in Logic*, volume 1925 de *Lecture Notes in Computer Science*, p. 49–56. Springer-Verlag.

Brown P. F., della Pietra V. J., deSouza P. V., Lai J. C. & Mercer R. L. (1992). Class-based n-gram Models of Natural Language. *Computational Linguistics*, **18**, 467–479.

Buckley C., Salton G. & Allan J. (1992). Automatic Retrieval with Locality Information using smart. In *Proceedings of the 1st Text Retrieval Conference, TREC'92*, Gaithersburg, Maryland, États-Unis.

Buckley C. & Voorhees E. M. (2000). Evaluating Evaluation Measure Stability. In *Proceedings of the 23rd Annual ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'00*, Athènes, Grèce.

Buntine W. L. (1988). Generalized Subsumption and its Application to Induction and Redundancy. *Artificial Intelligence*, **36**, 375–399.

Burgin R. (1992). Variations in Relevance Judgements and the Evaluation of Retrieval Performance. *Information Processing and Management: an International Journal*, **28**(5), 619–627.

Califf E. & Mooney R. J. (1997). Applying ILP-Based Techniques to Natural Language Information Extraction: An Experiment in Relational Learning. In *Procee-*

dings of the IJCAI'97 Workshop on Frontiers in Inductive Logic Programming, Nagoya, Japon.

Carbonell J. G., Yang Y., Frederking R. E., Brown R. D., Geng Y. & Lee D. (1997). Translingual Information Retrieval: A Comparative Evaluation. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence, IJCAI'97*, Nagoya, Japon.

Ceusters W., Spyns P., DeMoor G. & Martin W. (1996). *Tagging of Medical Texts: The Multi-TALE Project*. Amsterdam : IOS Press.

Champesme M., Brézellec P. & Soldano H. (1995a). Empirically Conservative Search Space Reductions. In *Proceedings of the 5th International Workshop on Inductive Logic Programming, ILP'95*, Louvain, Belgique. Poster publié.

Champesme M., Brézellec P. & Soldano H. (1995b). Réductions de l'espace de recherche : résultats théoriques et expérimentaux. In *Actes des Journées Acquisition Validation Apprentissage, JAVA'95*, Grenoble, France.

Chang C.-L. & Lee R. C.-T. (1973). *Symbolic Logic and Mechanical Theorem Proving*. New-York : Academic Press.

Church K. W. (1988). A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text. In *Proceedings of the 2nd Conference on Applied Natural Language Processing, ANLP'88*, Austin, Texas, États-Unis.

Church K. W., Gale W., Hanks P. & Hindle D. (1991). Using Statistics in Lexical Analysis. In U. Zernick, Ed., *Lexical Acquisition: Using On-line Resources to Build a Lexicon*, p. 115–164. Laurence Erlbaum.

Church K. W. & Gale W. A. (1991). Concordances for Parallel Texts. In *Proceedings of the 7th Annual Conference of the UW Centre for the New OED and Text Research*, University of Waterloo, Ontario, Canada.

Church K. W. & Hanks P. (1989). Word Association Norms, Mutual Information, and Lexicography. In *Proceeding of the 27th Annual Meeting of the Association for Computational Linguistics, ACL'89*, Vancouver, Canada.

Church K. W. & Hanks P. (1990). Word Association Norms, Mutual Information, and Lexicography. *Computational Linguistics*, **16**(1), 22–29.

Cohen W. W. (1994). Grammatically Biased Learning: Learning Logic Programs Using an Explicit Antecedent Description Language. *Artificial Intelligence*, **68**, 303–366.

Cohen W. W. (1995). Fast Effective Rule Induction. In *Machine Learning: Proceedings of the 12th International Conference*, Lac Tahoe, Californie, États-Unis.

Condamines A. & Amsili P. (1993). Terminology between Language and Knowledge: an Example of Terminological Knowledge Base. In *Proceedings of the 3rd International Congress on Terminology and Knowledge Engineering*, Cologne, Allemagne : Indeks Verlag.

Cooper W. S. (1991). Some Inconsistencies and Misnomers in Probabilistic Information Retrieval. In *Proceedings of the 13th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'91*, Chicago, Illinois, États-Unis.

Cooper W. S., Chen A. & Gey F. C. (1994). Experiments in the Probabilistic Retrieval of Full Text Documents. In *Proceedings of the 3rd Text Retrieval Conference, TREC'94*, Gaithersburg, Maryland, États-Unis.

Copestake A. (2001). The Semi-Generative Lexicon: Limits on Lexical Productivity. In P. Bouillon & K. Kanzaki, Eds., *Proceedings of the 1st International Workshop on the Generative Approaches to the Lexicon, GL'01*, Genève, Suisse.

Coret A., Kremer P., Landi B. & Schibler D. (1997). Towards a Methodology for Evaluating Information Retrieval Systems Adapted to Textual Documents in the French Language: the Amaryllis exploratory cycle. In *Proceedings of the SALT Workshop on Evaluation in Speech and Language Technology*, Sheffield, Royaume-Uni.

Cornuéjols A. & Miclet L. (2002). *Apprentissage artificiel*. Eyrolles.

Croft W. B. & Harper D. J. (1979). Using Probabilistic Models of Information Retrieval without Relevance Information. *Journal of Documentation*, **35**(4), 285–295.

Cruse D. A. (1986). *Lexical Semantics*. Textbooks in Linguistics. Cambridge University Press.

Cussens J. (1996). *Part-of-Speech Disambiguation using ILP*. Rapport interne, Oxford University Computing Laboratory.

Cussens J. (1998). Notes on Inductive Logic Programming Methods in Natural Language Processing (European work).

J. Cussens & S. Džeroski, Eds. (2000). *Learning Language in Logic*. Lecture Notes in Artificial Intelligence. Springer Verlag.

Cussens J. & Pulman S. (2000). Experiments in Inductive Chart Parsing. In J. Cussens & S. Džeroski, Eds., *Learning Language in Logic*, volume 1925 de *Lecture Notes in Computer Science*, p. 143–156. Springer-Verlag.

H. Czap & W. Nedobity, Eds. (1990). *Proceedings of the 2nd International Congress on Terminology and Knowledge Engineering*, Trier, Allemagne. Indeks Verlag.

Dagan I. & Church K. (1997). *Termight*: Coordinating Man and Machine in

Bilingual Terminology Acquisition. *Machine Translation*, **12**(1-2), 89–107.

DAILLE B. (1994). *Approche mixte pour l'extraction automatique de terminologie : statistique lexicale et filtres linguistiques.* Thèse de doctorat, Université Paris VII, France.

DAILLE B. (1999). Identification des adjectifs relationnels en corpus. In *Actes de la conférence Traitement Automatique des Langues Naturelles, TALN'99*, Cargèse, France.

DAILLE B. (2001). Identification des adjectifs relationnels. *TAL (Traitement automatique des langues)*, **42**(3), 815–832.

DAILLE B. (2002). *Découvertes linguistiques en corpus.* Habilitation à diriger des recherches, Université de Nantes, France.

DAVID S. & PLANTE P. (1990). De la nécessité d'une approche morpho-syntaxique dans l'analyse de textes. *Intelligence Artificielle et Sciences Cognitives au Québec*, **3**(3), 140–154.

DAVID S. & PLANTE P. (1991). Le progiciel *TERMINO* : de la nécessité d'une approche morpho-syntaxique pour le dépouillement terminologique de textes. In *Actes du Colloque sur les industries de la langue*, Québec, Canada.

DE CHALENDAR G. (2001). *SVETLAN', un système de structuration du lexique guidé par la détermination automatique du contexte thématique.* Thèse de doctorat, Université de Paris XI, France.

DE CHALENDAR G. & GRAU B. (2000). SVETLAN' ou comment classer les mots en fonction de leur contexte. In *Actes de la conférence Traitement Automatique des Langues Naturelles, TALN'00*, Lausanne, Suisse.

DE LOUPY C. (2000). *Évaluation de l'apport de connaissances linguistiques en désambiguïsation sémantique et recherche documentaire.* Thèse de doctorat, Université d'Avignon et des Pays de Vaucluse.

DE LOUPY C. & EL-BÈZE M. (2002). Managing Synonymy and Polysemy in a Document Retrieval System Using WordNet. In *Proceedings of the LREC'02 Workshop on Using Semantics for Information Retrieval and Filtering*, Las Palmas de Gran Canaria, Espagne.

DE RAEDT L. & BRUYNOOGHE M. (1993). A Theory of Clausal Discovery. In *Proceedings of the 13th International Joint Conferrence on Artificial Intelligence, IJCAI'93*, Chambéry, France.

DE RAEDT L. & DEHASPE L. (1996). *Clausal Discovery.* Rapport interne CW 238, Department of Computing Science, Université Catholique de Louvain, Belgique.

DE RAEDT L. & DŽEROSKI S. (1994). First Order *jk*-Clausal Theories are PAC-

Learnable. *Artificial Intelligence*, **70**, 375–392.

De Raedt L. & van Lear W. (1995). Inductive Constraint Logic. In *Proceedings of the 6th Workshop on Algorithmic Learning Theory, ALT'95*, volume 997 de *Lecture Notes in Artificial Intelligence*, Fukuoka, Japon : Springer-Verlag.

de Saussure F. (1916). *Cours de linguistique générale.* Éditions Payot et Rivages, édition de 1996.

Debili F., Radasoa P. & Fluhr C. (1989). About Reformulation in Full-Text IRS. *Information Processing and Management*, **25**(6), 647–657.

Deerwester S., Dumais S. T., Furnas G. W., Landauer T. K. & Harshman R. (1990). Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*, **41**(6), 391–407.

Dehaspe L. & De Raedt L. (1995). *A Declarative Language Bias for Concept Learning and Knowledge Discovery Engines.* Rapport interne CW 214, Department of Computing Science, Université Catholique de Louvain, Belgique.

Dehaspe L., van Lear W. & De Raedt L. (1994). Applications of a Logical Discovery Engine. In *Proceedings of the 4th International Workshop on Inductive Logic Programing, ILP'94*, Bonn, Allemagne.

Dumais S. T., Landauer T. K. & Littman M. (1996). Automatic Cross-Linguistic Information Retrieval using Latent Semantic Indexing. In *Proceedings of the SIGIR Workshop on Cross-Linguistic Information Retrieval*, Zurich, Suisse.

Dunning T. E. (1993). Accurate Methods for the Statistics of Surprise and Coincidence. *Computational Linguistics*, **19**(1), 61–74.

Džeroski S. & Bratko I. (1992). Handling Noise in Inductive Logic Programming. In *Proceedings of the 2nd International Workshop on Inductive Logic Programming, ILP'92*, Tokyo, Japon.

Džeroski S., Cussens J. & Manandhar S. (2000). An Introduction to Inductive Logic Programming and Learning Language in Logic. In J. Cussens & S. Džeroski, Eds., *Learning Language in Logic*, volume 1925 de *Lecture Notes in Computer Science*, p. 3–35. Springer-Verlag.

Eineborg M. & Lindberg N. (2000). ILP in Part-of-Speech Tagging - An Overview. In J. Cussens & S. Džeroski, Eds., *Learning Language in Logic*, volume 1925 de *Lecture Notes in Computer Science*, p. 157–169. Springer-Verlag.

Enguehard C. (1992). *Acquisition naturelle automatique d'un réseau sémantique.* Thèse de doctorat, Université de Technologie de Compiègne, France.

Enguehard C. & Pantera L. (1995). Automatic Natural Acquisition of a termino-

logy. *Journal of Quantitative Linguistics*, **2**(1), 27–32.

ESPOSITO F., LATERZA A., MALERBA D. & SEMERARO G. (1996). Refinement of Datalog Programs. In *Proceedings of the MLnet Familiarization Workshop on Data Mining with Inductive Logic Programming*, Bari, Italie.

EVANS D. A. & ZHAI C. (1996). Noun-Phrase Analysis in Unrestricted Text for Information Retrieval. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics, ACL'96*, Santa-Cruz, États-Unis.

FABRE C. (1996). *Interprétation automatique des séquences binominales en anglais et en français. Application à la recherche d'informations.* Thèse de doctorat, Université de Rennes 1, France.

FABRE C. & JACQUEMIN C. (2000). Boosting Variant Recognition with Light Semantics. In *Proceedings of the 18th International Conference on Computational Linguistics, COLING'00*, Saarbrücken, Allemagne.

FABRE C. & SÉBILLOT P. (1999). Semantic Interpretation of Binominal Sequences and Information Retrieval. In *Proceedings of the International ICSC Congress on Computational Intelligence: Methods and Applications, CIMA'99, Symposium on Advances in Intelligent Data Analysis, AIDA'99*, Rochester, États-Unis.

FAURE D. (2000). *Conception de méthode d'apprentissage symbolique et automatique pour l'acquisition de cadres de sous-catégorisation de verbes et de connaissances sémantiques à partir de textes : le système* ASIUM. Thèse de doctorat, Université de Paris XI Orsay, France.

FAURE D. & NÉDELLEC C. (1999). Knowledge Acquisition of Predicate Argument Structures from Technical Texts using Machine Learning: the System *ASIUM*. In D. F. R. STUDER, Ed., *Proceedings of the 11th European Workshop EKAW'99*, Dagstuhl, Allemagne : Springer-Verlag.

C. FELLBAUM, Ed. (1998). *WordNet: An Electronic Lexical Database*. Cambridge, Massachusetts, États-Unis : The MIT Press.

FELLBAUM C., GRABOWSKY J., LANDES S. & BAUMANN A. (1996). Matching Words to Senses in WordNet: Naive vs. Expert Differentiation of Senses. In C. FELLBAUM, Ed., *WordNet: an Electronic Lexical Database and Some of its Applications*. MIT Press, Cambrige, Massachusetts, États-Unis.

FLACH P. (1992). A Framework for Inductive Logic Programming. In S. MUGGLETON, Ed., *Inductive Logic Programming*. Academic Press.

FOLTZ P. W. (1990). Using Latent Semantic Indexing for Information Retrieval. In *Proceedings of the Conference on Office Information Systems*, Cambridge, Massaschusetts, États-Unis.

Fox E. A. (1983). *Extending the Boolean and Vector Space Models of Information Retrieval with P-norm Queries and Multiple Concept Types.* PhD thesis, Cornell University, New-York, États-Unis.

Frantzi K. & Ananiadou S. (1996). Extracting Nested Collocations. In *Proceedings of the International Conference on Computational Linguistics, COLING'96*, Copenhague, Danemark.

Frantzi K., Ananiadou S. & Mima H. (2000). Automatic Recognition of Multi-Word Terms: the C-Value/NC-Value Method. *Journal of Digital Library*, **3**(2), 115–130.

Freund Y. & Schapire R. E. (1999). A Short Introduction to Boosting. *Japanese Society for Artificial Intelligence*, **14**(5), 771–780.

Galy É. (2000). Repérer en corpus les associations sémantiques privilégiées entre le nom et le verbe : le cas de la fonction dénotée par le nom. Mémoire de Maîtrise, Université de Toulouse - Le Mirail, France.

Gamberger D. & Lavrač N. (1996). Noise Detection and Elimination Applied to Noise Handling in KRK Chess Endgame. In *Proceedings of the 6th International Workshop on Inductive Logic Programming, ILP'96*, Stockholm, Suède.

Garcia D. (1998). Exploitation, pour l'élaboration de requêtes de filtrage de textes, des connaissances causales détectées par coatis. In *Actes de la Rencontre Internationale sur le Filtrage et le Résumé automatique, RIFRA'98*, Sfax, Tunisie.

Garcia D., Aussenac-Gilles N. & Courcelle A. (2000). Exploitation, pour la modélisation, des connaissances causales repérées par coatis dans les textes. In J. Charlet, M. Zacklad, G. Kassel & D. Bourigault, Eds., *Ingénierie des connaissances : évolutions récentes et nouveaux défis.* Eyrolles.

Gilloux M., Lassalle E. & Ombrouck J.-M. (1993). Interrogation en langage naturel du Minitel guide des services. *Écho des recherches*, **146**, 1–20.

Gonzalo J., Verdejo F., Peters C. & Calzolari N. (1998). Applying EuroWordNet to Cross-Language Text Retrieval. *Computers and the Humanities, Special issue on EuroWordNet*, **32**(2), 185–207.

Gottlob G. (1987). Subsumption and Implication. *Information Processing Letters*, **24**(2), 109–111.

Grefenstette G. (1992). *Sextant*: Exploring Unexplored Contexts for Semantic Extraction from Syntactic Analysis. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics, ACL'92*, Newark, Delaware, États-Unis.

Grefenstette G. (1994a). Corpus-Derived First, Second and Third-Order Word Affinities. In *Proceedings of EURALEX'94*, Amsterdam, Pays-Bas.

GREFENSTETTE G. (1994b). *Explorations in Automatic Thesaurus Discovery.* Dordrecht: Kluwer Academic Publishers.

GREFENSTETTE G. (1997). *SQLET*: Short Query Linguistic Expansion Techniques, Palliating One-Word Queries by Providing Intermediate Structure to Text. In *Actes de la Conférence Recherche d'Informations Assistée par Ordinateur, RIAO'97*, Montréal, Québec, Canada.

HABERT B. & FABRE C. (1999). Elementary Dependency Trees for Identifying Corpus-Specific Semantic Classes. *Computer and the Humanities*, **33**(3), 207–219.

HABERT B. & JACQUEMIN C. (1993). Noms composés, termes, dénominations complexes : problématiques linguistiques et traitements automatiques. *TAL (Traitement automatique des langues)*, **34**(2).

HABERT B., NAZARENKO A. & SALEM A. (1997). *Les linguistiques de corpus.* Armand Collin/Masson, Paris.

HARRIS Z., GOTTFRIED M., RYCKMAN T., MATTICK(JR) P., DALADIER A., HARRIS T. N. & HARRIS S. (1989). The Form of Information in Science, Analysis of Immunology Sublanguage. *Boston Studies in the Philosophy of Science*, **104**.

HEARST M. A. (1992). Automatic Acquisition of Hyponyms from Large Text Corpora. In *Proceedings of the 14th International Conference on Computational Linguistics, COLING'92*, Nantes, France.

HEARST M. A. (1998). Automated Discovery of WordNet Relations. In C. FELLBAUM, Ed., *WordNet: an Electronic Lexical Database*, chapitre 5, p. 131–151. Cambridge, Massachusetts, États-Unis : MIT Press.

HEID U. (2000). A Linguistic Bootstrapping Approach to the Extraction of Term Candidates from German Text. *Terminology*, **5**(2), 161–182.

HEID U., JAUSS S., KRÜGER K. & HOHMANN A. (1996). Term Extraction with Standard Tools for Corpus Extraction. Experience from German. In *Proceedings of the 4th International Congress on Terminology and Knowledge Engineering, TKE'96*, Vienne, Autriche.

HELFT N. (1987). Inductive Generalization: A Logical Framework. In *Proceedings of the 2nd European Working Session on Learning, EWSL*, Bled, Slovénie.

HELFT N. (1989). Induction as Non-Monotonic Inference. In *Proceedings of the 1st International Conference on Principles of Knowledge Representation and Reasoning*, Toronto, Ontario, Canada.

HIRATA K. (1999). Flattening and Implication. In *Proceedings of the 10th International Conference on Algorithmic Learning Theory, ALT'99*, Tokyo, Japon.

HOFMANN T. (1999a). Probabilistic Latent Semantic Indexing. In *Proceedings of the 22nd SIGIR Conference on Research and Development in Information Retrieval*, Berkeley, Californie, États-Unis.

HOFMANN T. (1999b). Probabilistic Semantic Analysis. In *Proceedings of the 15th Annual Conference on Uncertainty in Artificial Intelligence, UAI'99*, Stockholm, Suède.

HULL D. (1993). Using Statistical Testing in the Evaluation of Retrieval Experiments. In *Proceedings of the 16th Annual ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'93*, Pittsburgh, États-Unis.

IDE N. & VÉRONIS J. (1994). *MULTEXT* (Multilingual Tools and Corpora). In *Proceedings of the 15th International Conference on Computational Linguistics, COLING-94*, Kyoto, Japon.

IDESTAM-ALMQUIST P. (1995). Generalization of Clauses under Implication. *Journal of Artificial Intelligence Research*, **3**, 467–489.

JACQUEMIN C. (1996). A Symbolic and Surgical Acquisition of Terms through Variation. In S. WERMTER, E. RILOFF & G. SCHELER, Eds., *Connectionist, Statistical and Symbolic Approaches to Learning for Natural Language Processing*, p. 425–438. Springer, Heidelberg.

JACQUEMIN C. (1997). *Variation terminologique : reconnaissance et acquisition automatique de termes et de leurs variantes en corpus.* Habilitation à diriger des recherches, Université de Nantes, France.

JACQUEMIN C. (2001). *Spotting and Discovering Terms through NLP.* Cambridge, Massachusetts, États-Unis : MIT Press.

JACQUEMIN C., KLAVANS J. L. & TZOUKERMANN E. (1997). Expansion of Multi-Word Terms for Indexing and Retrieval Using Morphology and Syntax. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics, ACL'97*, Madrid, Espagne.

JENSEN B. J., SPINK A., BATEMAN J. & SARACEVIC T. (1998). Real Life Information Retrieval: A Study of User Queries on the Web. *SIGIR Forum*, **32**(1), 5–17.

JENSEN B. J., SPINK A. & SARACEVIC T. (2000). Real Life, Real Users, and Real Needs: A Study and Analysis of User Queries on the Web. *Information Processing and Management*, **36**(2), 207–227.

JONES R., MCCALLUM A., NIGAM K. & RILOFF E. (1999). Bootstrapping for Text Learning Tasks. In *Proceedings of the IJCAI-99 Workshop on Text Mining: Foundations, Techniques and Applications*, Stockholm, Suède.

JOUIS C. (1993). *Contributions à la conceptualisation et à la modélisation des connaissances à partir d'une analyse linguisique de textes.* Thèse de doctorat, Université de

Paris-Sorbonne, France.

JOUIS C. (1995). *SEEK*, un logiciel d'acquisition de connaissances utilisant un savoir linguistique sans employer de connaissances sur le monde externe. In *Actes des 6es Journées Acquisition, Validation, JAVA'95*, Grenoble, France.

JOUIS C., BISKRI I., DESCLES J.-P., PRIO F. L., MEUNIER J.-G., MUSTAPHA W. & NAULT G. (1997). Vers l'intégration d'une approche sémantique linguistique et d'une approche numérique pour un outil d'aide à la construction de bases terminologiques. In *Actes des 1ères Journées scientifiques et techniques du réseau francophone de l'ingénierie de la langue de l'AUPELF-UREF*, Avignon, France.

JUNKER M., SINTEK M. & RINK M. (2000). Learning for Text Categorization and Information Extraction with ILP. In J. CUSSENS & S. DŽEROSKI, Eds., *Learning Language in Logic*, volume 1925 de *Lecture Notes in Computer Science*, p. 247–258. Springer-Verlag.

JUSTESON J. S. & KATZ S. M. (1995). Technical Terminology: some linguistic properties and an algorithm for identification in text. *Natural Language Engineering*, **1**(1), 9–27.

KAHANE S. & POLGUÈRE A. (2001). Formal Foundation of Lexical Functions. In *Proceedings of the Workshop on Collocation: Computational Extraction, Analysis and Exploitation, 39th Annual Meeting and 10th Conference of the European Chapter of the Association for Computational Linguistics, ACL'01-EACL'01*, Toulouse, France.

KAPUR D. & NARENDRAN P. (1986). NP-Completeness of the Set Unification and Matching Problems. In *Proceedings of the 8th Conference on Automated Deduction, CADE'86*, Oxford, Royaume-Uni.

KAZAKOV D. & MANANDHAR S. (2001). Unsupervised Learning of Word Segmentation Rules with Genetic Algorithms and Inductive Logic Programing. *Machine Learning*, **43**, 121–162.

KIETZ J.-U. (1993). A Comparative Study of Structural Most Specific Generalizations Used in Machine Learning. In *Proceedings of the 3rd International Workshop on Inductive Logic Programming, ILP'93*, Bled, Slovénie.

KIETZ J.-U. & WROBEL S. (1992). Controlling the Complexity of Learning in Logic through Syntactic and Task-Oriented Models. In S. MUGGLETON, Ed., *Inductive Logic Programming*, p. 335–359. Academic Press.

KILGARRIFF A. (2001). How Much of the Time does the Generative Lexicon Account for Novel Word Uses ? In P. BOUILLON & K. KANZAKI, Eds., *Proceedings of the 1st International Workshop on Generative Approaches to the Lexicon, GL'01*, Genève, Suisse.

KLAVANS J. & KAN M.-Y. (1998). Role of Verbs in Document Analysis. In *Proceedings of the 17th International Conference on Computational Linguistics and Association for Computational Linguistics, COLING-ACL*, Montréal, Québec, Canada.

KOHAVI R. (1995). A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence, IJCAI 95*, Montréal, Québec, Canada.

KRAMER S. (1995). *Predicate Invention: A Comprehensive View.* Rapport interne OFAI-TR-95-32, Austrian Research Institute for Artificial Intelligence.

KWOK K. L. (1995). A Network Approach for Probabilistic Information Retrieval. *ACM Transactions on Information Systems, TOIS*, **13**(3), 325–354.

KWOK K. L. & GRUNFELD L. (1993). TREC-2 Document Retrieval Experiments using *PIRCS*. In *Proceedings of the 2nd Text Retrieval Conference, TREC'93*, Gaithersburg, Maryland, États-Unis.

LANCASTER F. W. (1968). *Information Retrieval Systems: Characteristics, Testing and Evaluation.* New-York : Wiley.

LANDI B., KREMER P. & SCHMITT L. (1998). Amaryllis: an Evaluation Experiment on Search Engine in a French-Speaking Context. In *Proceedings of the 1st International Conference on Language and Resources Evaluation, LREC'98*, Grenade, Espagne.

LAPATA M. & LASCARIDES A. (2003). A Probabilistic Account of Logical Metonymy. *Computational Linguistics*, **29**(2), 263–317.

LAVRAČ N. & DŽEROSKI S. (1992). Inductive Learning of Relations from Noisy Examples. In S. MUGGLETON, Ed., *Inductive Logic Programming*, volume 38 de *APIC*, p. 495–516. Academic Press.

LAVRAČ N. & DŽEROSKI S. (1994). *Inductive Logic Programming: Techniques and Applications.* New York, États-Unis : Ellis Horwood.

LEBART L. & SALEM A. (1994). *Statistique textuelle.* Dunod.

LEE C. (1967). *A Completeness Theorem and a Computer Program for Finding Theorems Derivable from Given Axioms.* PhD thesis, University of California, Berkeley, États-Unis.

LEE L. J. (1997). *Similarity-Based Approaches to Natural Language Processing.* PhD thesis, Harvard University, États-Unis. Publié en rapport technique TR-11-97.

LERAT P. (1995). *Les langues spécialisées.* PUF.

LERMAN I.-C. (1970). *Les bases de la classification automatique.* Paris : Gauthier-Villars.

Lesk M. E. (1969). Word-Word Association in Document Retrieval Systems. *American documentation*, **20**, 27–38.

Lloyd J. W. (1987). *Foundations of Logic Programming.* Berlin, Allemagne : Springer.

Manning C. D. & Schütze H. (1999). *Foundations of Statistical Natural Language Processing.* Cambridge, Massachusetts, États-Unis : The MIT Press.

Marcinkowski J. & Pacholski L. (1992). Undecidability of the Horn Clause Implication Problem. In *Proceedings of the 33rd Annual IEEE Symposium on Foundations of Computer Science*, Los Alamitos, Californie, États-Unis.

Mayer E. (1999). *Apprentissage inductif de scénarios pour la supervision de réseaux de télécommunications.* Thèse de doctorat, Université de Rennes 1.

Mel'čuk I. (1998). Collocations and lexical functions. In A. P. Cowie, Ed., *Phraseology: Theory, Analysis, and Applications*, chapitre 2, p. 23–54. Oxford : Clarendon Press.

Miller A. G., Fellbaum C. & Gross D. (1989). WordNet: A Lexical Database Organised on Psycholinguistic Principles. In *Proceedings of the 1st International Lexical Acquisition Workshop*, Détroit, États-Unis.

Mitchell T. M. (1980). The Need for Biases in Learning Generalizations. In *Readings in Machine Learning*, p. 184–191. Morgan Kaufmann. Publié en 1991.

Mitchell T. M. (1982). Generalization as Search. *Artificial Intelligence*, **18**(2), 203–226.

Mitchell T. M. (1997). *Machine Learning.* McGraw-Hill.

Mizzaro S. (1997). Relevance: the Whole History. *Journal of the American Society of Information Science*, **48**(9), 810–832.

Mizzaro S. (1998). How Many Relevance in Information Retrieval ? *Interacting with Computers*, **10**(3), 303–320.

Mooney R. J. (1997). Inductive Logic Programming for Natural Language Processing. In S. Muggleton, Ed., *Inductive Logic Programming: Selected Papers from the 6th International Workshop*, Lecture Notes in Computer Science, p. 3–22. Springer Verlag, Berlin.

Mooney R. J. (1999). Learning for Semantic Interpretation: Scaling Up Without Dumbing Down. In *Proceedings of the Learning Language in Logic Workshop, LLL'99*, Bled, Slovénie.

Morin E. (1997). Extraction de liens sémantiques entre termes dans des corpus de textes techniques : application à l'hyponymie. In *Actes de la conférence Traitement*

Automatique des Langues Naturelles, TALN'97, Grenoble, France.

MORIN E. (1998). *PROMÉTHÉE* un outil d'aide à l'acquisition de relations sémantiques entre termes. In *Actes de la conférence de Traitement Automatique des Langues Naturelles, TALN'98*, Paris, France.

MORIN E. (1999). *Extraction de liens sémantiques entre termes à partir de corpus de textes techniques*. Thèse de doctorat, Université de Nantes, France.

MORIN E. & MARTIENNE E. (1999). Raffinement de patrons lexico-syntaxiques par un système d'apprentissage. In *Actes de la conférence Ingénierie des Connaissances, IC'99*, Palaiseau, France.

MUC-7 (1998). *Proceedings of the 7th Message Understanding Conference, MUC-7*, Fairfax, Virginie, États-Unis. Morgan Kaufmann.

MUGGLETON S. (1987). Duce, an Oracle Based Approach to Constructive Induction. In *Proceedings of the 10th International Joint Conference on Artificial Intelligence, IJCAI'87*, Milan, Italie.

MUGGLETON S. (1992). Inverting Implication. In *Proceedings of the 2nd International Workshop on Inductive Logic Programming, ILP'92*, Tokyo, Japon.

MUGGLETON S. (1995). Inverse Entailment and Progol. *New Generation Computing*, **13**(3-4), 245–286.

MUGGLETON S. (1998). Completing Inverse Entailment. In *Proceedings of the 8th International Conference on Inductive Logic Programming, ILP98*, Madison, Wisconsin, États-Unis.

MUGGLETON S. (1999a). Inductive Logic Programming: Issues, Results and the Challenge of Learning Language in Logic. *Artificial Intelligence*, **114**(1-2), 283–296.

MUGGLETON S. (1999b). Scientific Knowledge Discovery Using Inductive Logic Programming. *Communications of the ACM*, **42**(11).

MUGGLETON S. & BRYANT C. H. (2000). Theory Completion Using Inverse Entailment. In *Proceedings of the 10th International Conference on Inductive Logic Programming, ILP'00*, Londres, Royaume-Uni.

MUGGLETON S. & BUNTINE W. L. (1988). Machine Invention of First-Order Predicates by Inverting Resolution. In *Proceedings of the 5th International Conference on Machine Learning*.

MUGGLETON S. & DE RAEDT L. (1994). Inductive Logic Programming: Theory and Methods. *Journal of Logic Programming*, **19-20**, 629–679.

MUGGLETON S. & FENG C. (1990). Efficient Induction of Logic Programs. In *Pro-*

ceedings of the 1st Conference on Algorithmic Learning Theory, Tokyo, Japon.

Muggleton S., Srinivasan A. & Bain M. (1992). Compression, Significance and Accuracy. In *Proceedings of the 9th International Conference on Machine Learning*, Aberdeen, Écosse.

Naulleau É. (1997). *Apprentissage et filtrage syntaxico-sémantique de syntagmes nominaux pertinents pour la recherche documentaire.* Thèse de doctorat, Université de Paris XIII, France.

Naulleau É. (1999). Profile-Guided Terminology Extraction. In *Proceedings of Terminology and Knowledge Extraction, TKE'99*, Innsbrück, Autriche.

Nédellec C. & Rouveirol C. (1994). *Specifications of the HAIKU System.* Rapport interne 928, Laboratoire de Rcherche en Informatique, Université de Paris Sud, France.

Nédellec C., Rouveirol C., Adé H., Bergadano F. & Tausend B. (1996). Declarative Bias in Inductive Logic Programming. In L. De Raedt, Ed., *Advances in Inductive Logic Programming*, p. 82–103. IOS Press.

Nelson M. J. (1995). The Effect of Query Characteristics on Retrieval Results in the TREC Retrieval Tests. In *Proceedings of the Annual Conference of the Canadian Association for Information Science, ACSI'95*, Edmonton, Alberta, Canada.

Nienhuys-Cheng S.-H. & de Wolf R. (1996). Least Generalizations and Greatest Specializations of Sets of Clauses. *Journal of Artificial Intelligence Research*, **4**, 341–363.

Nienhuys-Cheng S.-H. & de Wolf R. (1997). *Foundations of Inductive Logic Programming*, volume 1228 de *Lectures Notes in Artificial Intelligence.* Springer-Verlag.

Oueslati R. (1999). *Aide à l'acquisition de connaissances à partir de corpus.* Thèse de doctorat, Univesité Louis Pasteur, Strasbourg, France.

Page D. & Srinivasan A. (2003). ILP: A Short Look Back and a Longer Look Forward. *Journal of Machine Learning Research*, **4**, 415–430.

Palmer M. (1998). Are WordNet Sense Distinctions Appropriate for Computational Lexicons ? In *Proceedings of the SENSEVAL Workshop*, Herstmonceux Castle, Royaume-Uni.

Papadimitriou C. H., Raghavan P., Tamaki H. & Vempala S. (1998). Latent Semantic Indexing: A Probabilistic Analysis. In *Proceedings of the 17th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, Seattle, Washington, États-Unis.

Pazzani M. & Brunk C. (1991). An Investigation of Noise-Tolerant Relational

Concept Learning Algorithms. In *Proceedings of the 8th International Workshop on Machine Learning*, San Mateo, Californie, États-Unis.

Pazzani M., Brunk C. & Silverstein G. (1991). A Knowledge-Intensive Approach to Learning Relational Concepts. In *Proceedings of the 8th International Workshop on Machine Learning*, San Mateo, Californie, États-Unis.

Pearce D. (2002). A Comparative Evaluation of Collocation Extraction Techniques. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation, LREC 02*, Las Palmas de Gran Canaria, Espagne.

Pearson J. (1998). *Terms in Context.* Studies in Corpus Linguistics. John Benjamins Publishing Company.

Peat H. J. & Willett P. (1991). The Limitations of Term Co-Occurrence Data for Query Expansion in Document Retrieval Systems. *Journal of the American Society for Information Science*, **42**(5), 378–383.

Pestov V. (2000). On the Geometry of Similarity Search: Dimensionality Curse and Concentration of Measure. *Information Processing Letters*, **73**(1-2), 47–51.

Petitpierre D. & Russell G. (1994). *Mmorph - The Multext Morphology Program.* Rapport technique, ISSCO, Genève, Suisse.

Pichon R. & Sébillot P. (1997). *Acquisition automatique d'informations lexicales à partir de corpus : un bilan.* Rapport de recherche n°3321, INRIA, Rennes, France.

Pichon R. & Sébillot P. (1999). Différencier les sens des mots à l'aide du thème et du contexte de leurs occurrences : une expérience. In *Actes de la conférence Traitement automatique des langues naturelles, TALN'99*, Cargèse, France.

Piwowarsky B. (2003). *Techniques d'apprentissage pour le traitement d'informations structurées : application à la recherche d'information.* Thèse de doctorat, Université de Paris VI, France.

Plotkin G. D. (1970). A Note on Inductive Generalization. In B. Meltzer & D. Michie, Eds., *Machine Intelligence*, volume 5, p. 153–163. Edinburgh : Edinburgh University Press.

Plotkin G. D. (1971). A Further Note on Inductive Generalization. In B. Meltzer & D. Michie, Eds., *Machine Intelligence*, volume 6, p. 101–124. Edinburgh : Edinburgh University Press.

Polanco X., François C., Royauté J., Grivel L., Besagni D., Dejean M. & Oto C. (1998). Organisation et gestion des connaissances en veille scientifique et technologique. In *Actes de Veille Stratégique, Scientifique et Technologique, VSST'98*, Toulouse, France.

PUSTEJOVSKY J. (1995). *The Generative Lexicon.* Cambridge, Massachusetts, États-Unis : The MIT Press.

PUSTEJOVSKY J., ANICK P. & BERGLER S. (1993). Lexical Semantic Techniques for Corpus Analysis. *Computational Linguistics, special issue on Using Large Corpora*, **19**(2), 331–358.

PUSTEJOVSKY J., BOGURAEV B., VERHAGEN M., BUITELAAR P. & JOHNSTON M. (1997). Semantic Indexing and Typed Hyperlinking. In *Proceedings of the American Association for Artifical Intelligence Conference, Spring Symposium, NLP for WWW.*, p. 120–128, Stanford University, Californie, États-Unis.

QIU Y. & FREI H.-P. (1993). Concept Based Query Expansion. In *Proceedings of the 16th ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'93*, Pittsburgh, Pennsylvanie, États-Unis.

QIU Y. & FREI H.-P. (1995). *Improving the Retrieval Effectiveness by a Similarity Thesaurus.* Rapport interne 225, ETH Zurich, Department of Computer Science, Zurich, Suisse.

QUINLAN J. R. (1990). Learning Logical Definitions from Relations. *Machine Learning*, **5**(3), 239–266.

QUINLAN J. R. & CAMERON-JONES R. M. (1995). Induction of Logic Programs: FOIL and Related Systems. *New Generation Computing*, **13**, 287–312.

RAJMAN M., BESANÇON R. & CHAPPELIER J.-C. (2000). Le modèle DSIR : une approche à base de sémantique distributionnelle pour la recherche documentaire. *TAL (Traitement automatique des langues), numéro spécial Traitement automatique des langues pour la recherche d'information*, **41**(2), 549–578.

RAJMAN M. & LEBART L. (1998). Similarités pour données textuelles. In *Actes des 4^{es} Journées internationales d'analyse de données textuelles, JADT'98*, Nice, France.

RAMSHAW L. A. & MARCUS M. P. (1995). Text Chunking using Transformation-Based Learning. In *Proceedings of the 3rd Workshop on Very Large Corpora.*

RASTIER F. (1995). Le terme : entre ontologie et linguistique. *La banque des mots*, **7**, 35–65.

RIEGER A. (1996). *Optimizing Chain Datalog Programs and their Inference Procedures.* LS-8 Report 20, Université de Dortmund, Lehrstuhl Informatik VIII, Allemagne.

ROBERTSON S. & SPARK-JONES K. (1997). *Simple Proven Approaches to Text Retrieval.* Rapport interne TR 356, Cambridge University Computer Laboratory, Royaume-Uni.

ROBERTSON S. E. (1977). The Probability Ranking Principle in Information Retrieval.

Journal of Documentation, **33**, 294–304.

ROBERTSON S. E., VAN RIJSBERGEN C. J. & PORTER M. (1981). Probabilistic Models of Indexing and Searching. In R. N. ODDY, S. E. ROBERTSON & C. J. VAN RIJSBERGEN, Eds., *Information Research and Retrieval*, chapitre 4, p. 35–56. Butterworths.

ROBERTSON S. E. & WALKER S. (1994). Some Simple Effective Approximations to the 2-Poisson Model Probabilistic Weighted Retrieval. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'94*, Dublin, Irlande.

ROBINSON J. A. (1965). A Machine-Oriented Logic Based on the Resolution Principle. *Journal of Association for Computing Machinery*, **12**(1), 23–41.

ROBISON H. R. (1970). Computer-Detectable Semantic Structure. *Information Storage and Retrieval*, **6**, 273–288.

ROCCHIO J. J. (1971). Relevance Feedback in Information Retrieval. In G. SALTON, Ed., *The SMART Retrieval System: Experiments in Automatic Document Processing*, p. 313–323. Prentice-Hall, Englewood Cliffs.

ROSSIGNOL M. & SÉBILLOT P. (2002). Automatic Generation of Sets of Keywords for Theme Characterization and Detection. In A. MORIN & P. SÉBILLOT, Eds., *Actes des 6es Journées internationales d'analyse statistique des données textuelles, JADT'02*, Saint-Malo, France.

ROUSSELOT F., FRATH P. & OUESLATI R. (1996). Extracting Concepts and Relations from Corpora. In *Proceedings of the Corpus-Oriented Semantic Analysis ECAI'96 Workshop*, Budapest, Hongrie.

ROUVEIROL C. (1992). Extensions of Inversion of Resolution Applied to Theory Completion. In S. MUGGLETON, Ed., *Inductive Logic Programming*. Londres, Royaume-Uni : Academic Press.

ROUVEIROL C. (1994). Flattening and Saturation: Two Representation Changes for Generalization. *Machine Learning Journal*, **14**, 219–232.

ROUVEIROL C. & PUGET J.-F. (1990). Beyond Inversion of Resolution. In *Proceedings of the 7th International Conference on Machine Learning, ICML'90*, Austin, Texas, États-Unis.

ROUVEIROL C. & SEBAG M. (1997). Tractable Induction and Classification in First-Order Logic Via Stochastic Matching. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence, IJCQI'97*, Nagoya, Japon.

ROYAUTÉ J., SCHMITT L. & OLIVETAN E. (1992). Les expériences d'indexation à

l'INIST. In *Proceedings of the 15th International Conference on Computational Linguistics, COLING'92*, Nantes, France.

G. Salton, Ed. (1971). *The* smart *Retrieval System: Experiments in Automatic Document Processing.* Prentice-Hall, Englewood Cliffs.

Salton G. (1975). *A Theory of Indexing.* Regional Conference Series in Applied Mathematics. Philadelphy : Society for Industrial and Applied Mathematics.

Salton G. (1989). *Automatic Text Processing.* Addison-Wesley.

Salton G. & Buckley C. (1988). Term-Weighting Approaches in Automatic Text Retrieval. *Information Processing and Management*, **24**(5), 513–523.

Salton G. & Buckley C. (1990). Improving Retrieval by Relevance Feedback. *Journal of the American Society fo Information Science*, **41**(4), 288–297.

Salton G. & Lesk M. E. (1968). Computer Evaluation and Indexing of Text Processing. *Journal of the ACM*, **15**(1), 8–36.

Sammut C. & Banerji R. B. (1986). Learning Concepts by Asking Questions. In R. Michalski, J. Carbonnel & T. Mitchell, Eds., *Machine Learning: An Artificial Inteligence Approach*, volume 2, p. 167–192. Morgan Kaufmann.

Savoy J. (1997). Statistical Inference in Retrieval Effectiveness Evaluation. *Information Processing and Management*, **33**(4), 495–512.

Savoy J. (1999). A Stemming Procedure and a Stopword List for General French Corpora. *Journal of the American Society for Information Science*, **50**(10), 944–952.

Schamber L. (1994). Relevance and Information Behavior. In M. E. Williams, Ed., *Annual Review of Information Science and Technology*, volume 29, chapitre 1, p. 3–48. Learned Information Inc. Medford, New Jersey, États-Unis.

Schmidt-Schauss M. (1988). Implication of Clauses is Undecidable. *TCS: Theoretical Computer Science*, **59**(3), 287–296.

Schütze H. & Perdersen J. O. (1994). A Co-Occurrence Based Thesaurus and two Applications to Information Retrieval. In *Actes de la Conférence de Recherche d'information assistée par ordinateur, RIAO'94*, p. 266–274, New-York, États-Unis.

Sebastiani F. (2002). Machine Learning in Automated Text Categorization. *ACM Computing Surveys*, **34**(1), 1–47.

Semeraro G., Esposito F., Malerba D., Brunk C. & Pazzani M. J. (1994). Avoiding Non-Termination when Learning Logic Programs: A Case Study with foil and focl. In L. Fribourg & F. Turini, Eds., *Proceedings of Logic Program Synthesis*

and Transformation - MetaProgramming in Logic, LOPSTR 1994, Lecture Notes in Computer Science 883. Springer-Verlag.

Semeraro G., Esposito F., Malerba D., Fanizzi N. & Ferilli S. (1997). A Logic Framework for the Incremental Inductive Synthesis of Datalog Theories. In *Logic Programming Synthesis and Transformation, Proceedings of 7th International Workshop, LOPSTR'97*, Louvain, Belgique.

Séguéla P. (2001). *Construction de modèles de connaissances par analyse linguistique de relations lexicales dans les documents techniques.* Thèse de doctorat, Université de Toulouse III, France.

Séguéla P. & Aussenac-Gilles N. (1999). Extraction de relations sémantiques et enrichissement de modèles du domaine. In *Actes de la conférence sur l'Ingénierie des Connaissances, IC'99*, Palaiseau, France.

Shapiro E. Y. (1981). *Inductive Inference of Theories from Facts.* Rapport de recherche 624, Department of Computer Science, Yale University, New Haven, États-Unis.

Shütze H., Hull D. & Pedersen J. O. (1995). A Comparison of Classifiers and Document Representations for the Routing Problem. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development on Information Retrieval*, Seattle, Washington, États-Unis.

Silverstein C., Henzinger M., Marais H. & Moricz M. (1998). *Analysis of a Very Large AltaVista Query Log.* Rapport interne 1998-014, Systems Research Center, Digital Equipment Corp., Palo Alto, États-Unis.

Sinclair J. (1991). *Corpus, Concordance, Collocation.* Oxford University Press.

Singhal A. K. (1997). *Term-Weighting Revisited.* PhD thesis, Cornell University, États-Unis.

Skuce D. R. & Meyer I. (1991). Terminology and Knowledge Acquisition: Exploring a Symbiotic Relationship. In *Proceedings of the 6th Knowledge Acquisition for Knowledge Based Systems Workshop*, Banff, Canada.

Smadja F. (1993a). Retrieving Collocations from Text: Xtract. *Computational Linguistics*, **19**(1), 143–178.

Smadja F. (1993b). Xtract: an Overview. *Computer and the Humanities*, **26**, 399–413.

Smeaton A. F. (1999). Using NLP or NLP Resources for Information Retrieval Tasks. In T. Strzalkowski, Ed., *Natural Language Information Retrieval*, p. 99–111. Kluwer Academic Publishers.

Sparck-Jones K. (1999). What is the Role of NLP in Text Retrieval ? In T. Strzalkowski, Ed., *Natural Language Information Retrieval.* Kluwer Academic Publishers.

Sparck-Jones K., Walker S. & Robertson S. E. (1998). *A Probabilistic Model of Information Retrieval: Development and Status.* Rapport interne TR 446, Cambridge University Computer Laboratory, Royaume-Uni.

Sparck-Jones K., Walker S. & Robertson S. E. (2000a). A Probabilistic Model of Information Retrieval: Development and Comparative Experiments - Part 1. *Information Processing and Management*, **36**(6), 779–808.

Sparck-Jones K., Walker S. & Robertson S. E. (2000b). A Probabilistic Model of Information Retrieval: Development and Comparative Experiments - Part 2. *Information Processing and Management*, **36**(6), 809–840.

Srinivasan A. (2001). *The* aleph *manual.*

Strzalkowski T. (1995). Natural Language Information Retrieval. *Information Processing and Management*, **31**(3), 397–417.

Strzalkowski T. (1999). Preface. In T. Strzalkowski, Ed., *Natural Language Information Retrieval*, p. xiii–xxii. Kluwer Academic Publishers.

Strzalkowski T., Lin F., Wang J. & Perez-Carballo J. (1999a). Evaluating Natural Language Processing Techniques in Information Retrieval. In T. Strzalkowski, Ed., *Natural Language Information Retrieval*, p. 113–145. Kluwer Academic Publishers.

Strzalkowski T., Perez-Carballo J., Karlgren J., Hulth A., Tapanainen P. & Lahtinen T. (1999b). Natural Language Information Retrieval: TREC-8 Report. In *Proceedings of the 8th Text Retrieval Conference, TREC-8*, Gaithersburg, Maryland, États-Unis.

Tamaddoni-Nezhad A. & Muggleton S. (2000). Searching the Subsumption Lattice by a Genetic Algorithm. In *Proceedings of the 10th International Conference on Inductive Logic Programming, ILP'00*, Londres, Royaume-Uni.

Tamaddoni-Nezhad A. & Muggleton S. (2002). A Genetic Algorithms Approach to ILP. In *Proceedings of the 12th International Conference on Inductive Logic Programming, ILP'02*, Sydney, Australie.

Thompson C. A. & Califf M. E. (2000). Improving Learning by Choosing Examples Intelligently in Two Natural Language Tasks. In J. Cussens & S. Džeroski, Eds., *Learning Language in Logic*, volume 1925 de *Lecture Notes in Computer Science*, p. 279–299. Springer-Verlag.

Torre F. (2000). *Intégration des biais de langage à l'algorithme générer-et-tester.* Thèse de doctorat, Université de Paris-Sud XI, France.

Torre F. & Rouveirol C. (1997a). Natural Ideal Operators in Inductive Logic Programming. In M. van Someren & W. G., Eds., *Proceedings of the 9th European*

Conference on Machine Learning, ECML'97, volume 1224 de *Lecture Notes in Artificial Intelligence*, Prague, République Tchèque : Springer-Verlag.

TORRE F. & ROUVEIROL C. (1997b). Opérateurs naturels en Programmation Logique Inductive. In *Actes des 12es Journées françaises d'apprentissage, JFA'97*, Roscoff, France.

TORRE F. & ROUVEIROL C. (1997c). *Private Properties and Natural Relations in Inductive Logic Programming.* Rapport technique 1118, Laboratoire de Recherche en Informatique d'Orsay (LRI), France.

TURENNE N. (2000). *Apprentissage statistique pour l'extraction de concepts à partir de textes. Applications au filtrage d'informations textuelles.* Thèse de doctorat, École nationale des arts et industries de Strasbourg, France.

VALIANT L. G. (1984). A Theory of the Learnable. *Communications of the ACM*, **27**, 1134–1142.

VAN DER LAAG P. R. & NIENHUYS-CHENG S.-H. (1994a). Existence and Nonexistence of Complete Refinement Operators. In F. BERGADANO & L. DE RAEDT, Eds., *Proceedings of the 7th European Conference on Machine Learning, ECML94*, volume 784 de *Lecture Notes on Artificial Intelligence*, Bad Honnef, Bonn, Allemagne : Springer-Verlag.

VAN DER LAAG P. R. & NIENHUYS-CHENG S.-H. (1994b). A Note on Ideal Refinement Operators in ILP. In S. WROBEL, Ed., *Proceedings of the 4th International Workshop on Inductive Logic Programming, ILP94*, volume 237 de *GMD-Studien* : Gesellschaft für Mathematik und Datenverarbeitung MBH.

VAN RIJSBERGEN C. J. (1977). A Theoretical Basis for the Use of Co-occurrence Data in Information Retrieval. *Journal of Documentation*, **33**(2), 106–119.

VAN RIJSBERGEN C. J. (1979). *Information Retrieval.* Royaume-Uni : Dept. of Computing Science, University of Glasgow, seconde édition.

VANDENBROUCKE L. (2000). Indexation automatique par couples nom-verbe pertinents, Mémoire de DES en information et documentation. Mémoire de maîtrise, Université Libre de Bruxelles, Belgique.

VOORHEES E. M. (1994). Query Expansion using Lexical-Semantic Relations. In *Proceedings of ACM SIGIR'94*, Dublin, Irlande.

VOORHEES E. M. (1998). Variations in Relevance Judgements and the Evaluation of Retrieval Performance. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Melbourne, Australie.

VOUTILAINEN A. (1993). *NPTOOL*, a Detector of English Noun Phrases. In *Proceedings of the Workshop on Very Large Corpora*, Ohio State University, Ohio, États-Unis.

WILKS Y. & CATIZONE R. (1999). Can we Make Information Extraction more Adaptive? In *Proceedings of the SCIE'99 Workshop*, Rome, Italie.

WILKS Y. & STEVENSON M. (1996). *The Grammar of Sense: is Word-Sense Tagging much more than Part-of-Speech Tagging?* Rapport technique, University of Sheffield, Royaume-Uni.

WIRTH R. (1989). Completing Logic Programs by Inverse Resolution. In *Proceedings of the 4th European Working Session on Learning, EWSL'89*, Montpellier, France.

WONG S. K. M., ZIARKO W. & WONG P. C. N. (1985). Generalized Vector Space Model in Information Retrieval. In *Proceedings of the 8th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'85*, Montréal, Québec, Canada.

WU H. & SALTON G. (1981). A Comparison of Search Term Weighting: Term Relevance vs. Inverse Document Frequency. In *Proceedings of the 4th Annual ACM SIGIR Conference on Research an Development in Information Retrieval*, Oakland, Californie, États-Unis.

YANG Y. & PEDERSEN J. O. (1997). A Comparative Study on Feature Selection in Text Categorization. In *Proceedings of the 14th International Conference on Machine Learning, ICML'97*, Nashville, États-Unis.

YAROWSKY D. (1992). Word-Sense Disambiguation Using Statistical Models of Roget's Categories Trained on Large Corpora. In *Proceedings of the 14th International Conference on Computational Linguistics, COLING-92*, Nantes, France.

YAROWSKY D. (1995). Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics, ACL'95*, Cambridge, Massachusetts, États-Unis.

ZELLE J. M. (1995). *Using Inductive Logic Programming to Automate the Construction of Natural Language Parsers.* PhD thesis, Departement of Computer Sciences, University of Texas, Austin, Texas, États-Unis.

ZELLE J. M., MOONEY R. J. & KONVISSER J. B. (1994). Combining Top-down and Bottom-up Techniques in Inductive Logic Programming. In *Machine learning: Proceedings of the Eleventh International Conference*, New-Brunswick, États-Unis.

Résumé

De nombreuses applications du traitement automatique des langues (recherche d'information, traduction automatique, *etc.*) requièrent des ressources sémantiques spécifiques à leur tâche et à leur domaine. Pour répondre à ces besoins spécifiques, nous avons développé ASARES, un système d'acquisition d'informations sémantiques lexicales sur corpus. Celui-ci répond à un triple objectif : il permet de fournir des résultats de bonne qualité, ses résultats et le processus ayant conduit à leur extraction sont interprétables, et enfin, il est assez générique et automatique pour être aisément portable d'un corpus à un autre. Pour ce faire, ASARES s'appuie sur une technique d'apprentissage artificiel symbolique — la programmation logique inductive — qui lui permet d'inférer des patrons d'extraction morphosyntaxiques et sémantiques à partir d'exemples des éléments lexicaux sémantiques que l'on souhaite acquérir. Ces patrons sont ensuite utilisés pour extraire du corpus de nouveaux éléments. Nous montrons également qu'il est possible de combiner cette approche symbolique avec des techniques d'acquisition statistiques qui confèrent une plus grande automaticité à ASARES.

Pour évaluer la validité de notre méthode, nous l'avons appliquée à l'extraction d'un type de relations sémantiques entre noms et verbes définies au sein du Lexique génératif appelées relations qualia. Cette tâche d'acquisition revêt deux intérêts principaux. D'une part, ces relations ne sont définies que de manière théorique ; l'interprétabilité linguistique des patrons inférés permet donc d'en préciser le fonctionnement et les réalisations en contexte. D'autre part, plusieurs auteurs ont noté l'intérêt de ce type de relations dans le domaine de la recherche d'information pour donner accès à des reformulations sémantiquement équivalentes d'une même idée. Grâce à une expérience d'extension de requêtes, nous vérifions expérimentalement cette affirmation : nous montrons que les résultats d'un système de recherche exploitant ces relations qualia, acquises par ASARES, sont améliorés de manière significative quoique localisée.

Summary

Many applications in the field of Natural Language Processing (information retrieval, machine translation, *etc.*) need semantic resources that are specific to their tasks and domains. To satisfy this need we have developed ASARES, a corpus-based lexical semantic acquisition system. It fulfills three objectives: it has good extraction results; these results and the whole acquisition process are interpretable; and it is generic and automatic enough to be easily portable from a corpus to another. To achieve these goals, ASARES uses a machine learning method —inductive logic programming— which makes possible to infer part-of-speech and semantic patterns from examples of the semantic elements we want to acquire. These patterns are then used to extract new elements from the corpus. We also show that it is possible to combine this symbolic method with statistical acquisition methods to make ASARES more automatic.

To validate our system, we have used it to acquire a kind of semantic relations between nouns and verbs defined in the Generative Lexicon and called qualia relations. This task has two main interests. On one hand, these relations are defined only in a theoretical point of view; the linguistic interpretation of the patterns thus allows to have a deeper understanding of their contextual realizations. On the other hand, several authors have noticed that such relations can be useful in information retrieval tasks because they make semantically equivalent reformulations of ideas accessible. With the help of a query expansion experiment using qualia relations extracted with ASARES, we show that this assumption is true to a certain extend: the performances of an information retrieval system are significantly improved though localized.