GASPAR : un dispositif pour le TALN basé sur la programmation à prototypes

Serge Fleury

UMR 9952 - CNRS, Ecole Normale Supérieure de Fontenay-St Cloud

31 avenue Lombart, F-92260 Fontenay aux Roses

e-mail : fleury@ens-fcl.fr, http://www.ens-fcl.fr/~fleury

Résumé

Nous présentons ici le dispositif GASPAR qui construit des représentations des mots sous la forme d'objets informatiques appelés des prototypes ; GASPAR associe à ces objets les comportements syntaxiques et sémantiques des mots en prenant appui sur des informations extraites à partir d'un corpus. GASPAR a pour première tâche de construire progressivement une représentation informatique des mots, sans présumer de leurs descriptions linguistiques ; il doit ensuite reclasser les mots représentés et mettre au jour, de manière inductive, les classes de mots du sous-langage étudié. Nous montrons comment la programmation à prototypes permet de représenter des mots dynamiquement par apprentissage et par affinements successifs. Elle permet ensuite d'amorcer un début de classement de ces mots sur la base de leurs contraintes syntaxico-sémantiques en construisant des hiérarchies locales de comportements partagés.

1 . Introduction

Cet article présente la mise en œuvre d'un dispositif expérimental de Traitement Automatique du Langage Naturel qui porte le nom de GASPAR (Fleury 1997). Ce dispositif vise à établir une représentation et un classement évolutifs d'unités lexicales représentées sous la forme d'objets informatiques appelés des prototypes. GASPAR a pour but de construire des représentations évolutives pour les mots à partir d'informations extraites sur corpus. Il doit conduire également à la construction de classes de mots de manière inductive. Les classes de mots produites peuvent ensuite être utilisées dans des applications liées à la construction de bases de connaissances sur un domaine de spécialité. Notre travail s'inscrit dans une reprise de l'approche harrisienne (Harris 1970, 88) et vise à automatiser les traitements de représentation des mots et de leur classement et à souligner les limites de cette induction de savoirs. Nous exposons d'abord les problèmes posés par la représentation d'unités de langue dans un dispositif de TALN. Nous examinons ensuite une chaîne de traitements qui permet l'acquisition d'informations à partir de corpus. Puis nous présentons le cadre de représentation choisi pour construire des représentations informatiques des mots et des comportements associés. Nous décrivons également les processus mis en place pour une représentation et un classement automatiques des mots et enfin nous présentons les premiers résultats construits avec GASPAR.

2 Représentation et classement automatiques de mots

2.1. Un problème : représenter et classer les mots

Notre travail vise à prendre appui sur des connaissances très générales qui sont affinées voire remodelées et changées au gré des observations rencontrées. La première tâche de GASPAR consiste à construire des représentations informatiques des mots et de leurs comportements. Il s'agit de repérer des descriptions des mots et de leurs comportements et de les représenter. Dans la mesure où les descriptions linguistiques peuvent évoluer, notre approche vise à ne pas préjuger des informations que l'on peut associer aux mots. Il ne s'agit donc pas d'encoder à la main des informations prédéterminées. La seconde tâche consiste à reclasser les informations représentées en prenant appui sur le fait que la représentation de savoirs évolutifs implique un classement évolutif de ces savoirs. Les processus de classement visent aussi à organiser le matériel lexical dans un domaine de spécialité afin de déterminer les classes sémantiques sous-jacentes aux classes de mots

construites (Habert *& al.* 1997).

2.2. Un corpus

Notre approche de représentation vise à repérer des informations à partir de réalisations rencontrées sur corpus (Habert *& al.* 1997). Le corpus utilisé est celui qui est constitué dans le cadre du projet MENELAS (Zweigenbaum 1994) pour la compréhension de textes médicaux. Ce corpus est utilisé par le Groupe de Travail Terminologique et Intelligence Artificielle (PRC-GDR Intelligence Artificielle, CNRS). L'unité thématique de ce corpus a trait aux maladies coronariennes. La phase d'acquisition de savoirs à partir de corpus prend appui sur la systématicité structurelle et sémantique propre aux sous-langages (Harris 1970, 88) afin de mettre au jour les proximités de cooccurrences entre mots pour dégager les relations sémantiques sous-jacentes. Les informations utilisées par les processus de représentation informatique des mots et des arbres associés sont issues d'une chaîne de traitements composée des logiciels LEXTER, AlethIP et ZELLIG. Le but de ces outils est d'extraire des informations à partir de corpus (LEXTER, AlethIP) et de simplifier ces informations puis de caractériser leurs fonctionnements (ZELLIG). Les informations extraites sont des arbres d'analyse, ces arbres étant ensuite simplifiés dans le but de déterminer les arbres élémentaires de dépendance qu'il est possible d'associer aux mots : sont considérés comme élémentaires les arbres mettant en évidence une relation binaire entre deux mots pleins, nom ou adjectif, dans des schémas comme, par exemple, `N Prep N` ou `N Adj`. Sur la séquence "alteration severe de la fonction ventriculaire gauche", ZELLIG met en évidence les dépendances élémentaires : `(a) alteration severe, (b) alteration de fonction, (c) fonction gauche, (d) fonction ventriculaire`.

2.3. Un outil

L'outil de représentation choisi est la programmation à prototypes (Blascheck 1994) (désormais notée PàP) dont les principaux domaines d'application se situent dans le développement d'interfaces utilisateur (Smith 1995). Notre travail utilise le prototype comme un outil de représentation de faits linguistiques dans la mesure où nous pensons qu'il peut répondre à certains problèmes que posent ces faits de langue. Cet outil conduit à construire des structures de représentation simples et ajustables pour rendre compte justement des problèmes d'ajustements qui sont à l'œuvre dans la construction du sens dans le langage naturel. Avec la PàP, il ne s'agit pas de partir d'une somme d'informations figées et connues par avance mais de construire progressivement les entités informatiques suivant les informations dont on dispose. Si les informations à représenter ne sont pas connues de manière définitive, il est possible de commencer le processus de représentation en utilisant les informations déjà recensées puis d'affiner dynamiquement les objets construits dès que de nouvelles informations sont disponibles. Cette mise à jour des objets peut être réalisée manuellement ou automatiquement (en définissant les opérations idoines). On peut donc envisager des processus de représentation qui se déroulent de manière continue suivant les flux d'informations disponibles. Une fois que l'on a construit un objet particulier, on utilise deux opérations fondamentales pour représenter d'autres éléments sous la forme de prototypes : le clonage et l'ajustement. L'opération de clonage produit une copie conforme de l'objet cloné. On peut ensuite d'ajuster le prototype issu de l'opération de clonage pour représenter adéquatement le nouvel élément. L'ajustement de l'objet cloné n'altère en rien le prototype qui a servi de support pour l'opération de clonage. On peut modifier les propriétés du nouvel objet sans modifier les propriétés du prototype initial. On peut ensuite réitérer les opérations de clonage et d'ajustement pour représenter les objets souhaités. Il est important de souligner que la notion de prototype mise en avant par la PàP ne correspond en rien à la notion de prototype développée par la psychologie cognitive. Self est le langage utilisé pour l'implémentation du système GASPAR. Self est un langage à prototypes qui permet l'héritage multiple et dynamique. Self a été conçu par David Ungar et Randall Smith à l'université de Stanford (Ungar, Smith 1987). Sa 1ère implémentation date elle aussi de 1987. La dernière version, Self-4.0, est disponible depuis juillet 95 (Self Group 1995). Self est désormais développé par Sun Microsystems. Un prototype est un objet composé d'attributs. Self offre des primitives qui permettent d'ajouter ou de supprimer dynamiquement des attributs aux

objets. Les objets dialoguent entre eux via un mécanisme d'envoi de messages. L'héritage en Self se réalise au travers de la notion de délégation qui permet de factoriser localement des comportements partagés. Dans le langage Self, un parent commun à plusieurs prototypes est appelé un objet `traits`.

3. Construction inductive des savoirs avec GASPAR

Un premier objectif est de construire des représentations informatiques évolutives de mots à partir d'informations extraites sur corpus. Le travail de représentation mis en place ne construit pas une représentation prédéterminée du sens attaché aux mots ou aux structures syntaxiques représentées, il propose des amorces d'interprétation qui doivent être affinées par un travail d'interprétation plus fin.

3.1. Représentation dynamique des unités lexicales

GASPAR dispose, au départ, d'informations extraites à partir d'un corpus (sous la forme d'un fichier texte). Pour chaque entrée lexicale, GASPAR dispose d'informations morphologiques et sémantiques décrivant ces mots, d'une liste d'arbres élémentaires et d'une liste d'arbres d'analyse associés aux arbres élémentaires. GASPAR s'appuie uniquement sur ces informations pour construire des prototypes afin de représenter les mots et leurs comportements (les arbres associés). Le processus de génération des mots se déroule de la manière suivante. Pour chaque unité lexicale, GASPAR vérifie s'il existe une représentation prototypique adéquate pour la représenter. Si elle existe, GASPAR conserve l'objet trouvé. Si elle n'existe pas, et s'il n'existe aucune représentation prototypique de la catégorie dont elle relève, GASPAR commence par créer automatiquement une représentation prototypique de cette nouvelle famille catégorielle, puis il construit une représentation prototypique de ce nouveau représentant de cette famille (en tenant compte des informations fournies pour décrire ce nouvel élément) (Figure 1a). Si le mot à représenter ne possède pas de représentation prototypique et s'il existe déjà une représentation prototypique d'un élément de la même famille catégorielle, GASPAR utilise les opérations de clonage et d'ajustement pour représenter ce nouvel élément (Figure 1b).

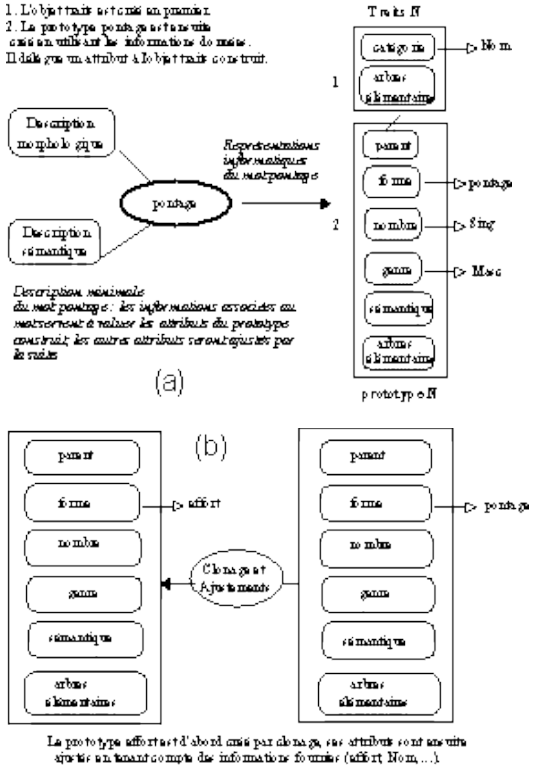

Figure 1. Génération automatique des prototypes de mot " pontage "(a) et " effort "(b).

3.2. Représentation dynamique des contraintes syntaxiques

GASPAR procède de la même manière pour la représentation des arbres. Pour chaque entrée

lexicale lue, on dispose d'une liste d'arbres élémentaires à représenter. Avant de représenter ces arbres élémentaires, GASPAR vérifie si ces arbres disposent déjà d'une représentation prototypique. Si elle n'existe pas, il la créé automatiquement en tenant compte des informations fournies : constituants et contraintes. Pour chaque arbre élémentaire construit, on peut avoir une liste d'arbres d'analyse à représenter. Avant de représenter ces arbres, GASPAR vérifie là encore si ces arbres disposent déjà d'une représentation prototypique. Si elle n'existe pas, il la créé automatiquement en tenant compte des informations fournies : constituants et contraintes. GASPAR affecte ensuite les prototypes d'arbres aux prototypes de mots auxquels ils sont associés. De même, il associe les prototypes d'arbres d'analyse construits aux prototypes d'arbres élémentaires associés. Dans la figure qui suit, le dispositif GASPAR construit les structures pour représenter les arbres associés à `pontage` en tenant compte des informations données pour la description de ces arbres.

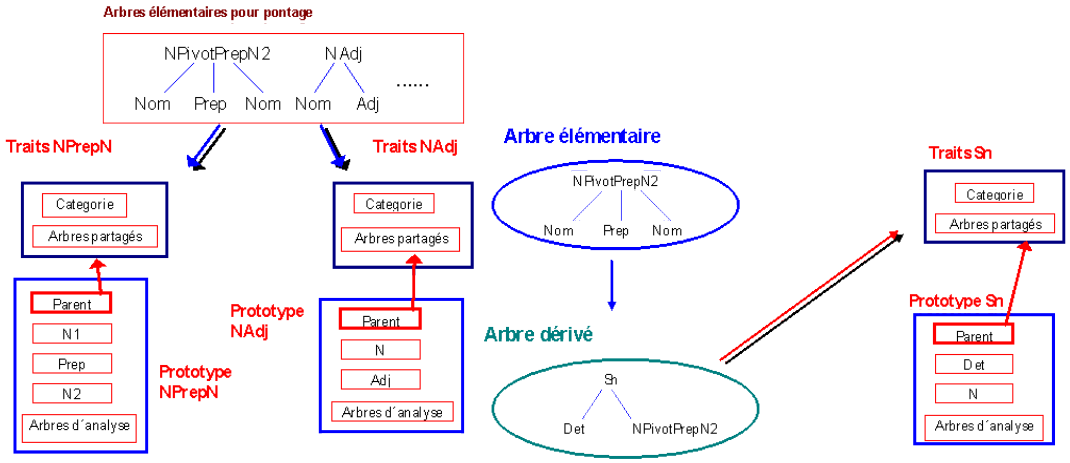

Figure 2. Génération automatique des prototypes d´arbres associés au mot " pontage ".

Dans la figure suivante, on présente, via l´interface graphique de Self, les prototypes construits pour représenter le mot `pontage` et les deux prototypes d'arbre élémentaire associés : il s'agit des prototypes représentant l'arbre `NAdj` et l'arbre `N1PrepNPivot`. Cette figure contient les masques graphiques construits pour représenter les mots et les arbres manipulés par GASPAR. Ces masques graphiques constituent des points d´entrée pour la présentation des informations associées aux prototypes construits.

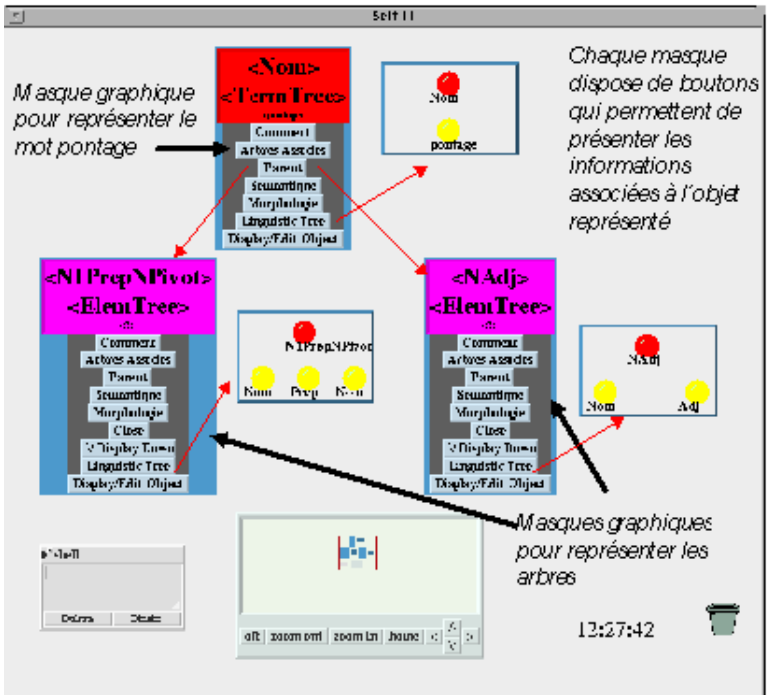

Figure 3. Objets GASPAR pour le nom `pontage`.

4. Classement des prototypes

Le second objectif est de classer les mots représentés et de tendre vers la détermination de classes sémantiques, de manière inductive. Tout d'abord, il faut souligner que le classement automatique présenté ici s'appuie principalement sur des contraintes syntaxiques associés aux mots. Dans notre travail, la syntaxe est utilisée pour dégrossir la représentation et le classement des mots mais ne permet pas à elle seule de classer les mots représentés. A l'inverse des approches harrissiennes et statistiques, notre approche ne conduit pas à la détermination de classes sémantiques satisfaisantes mais elle constitue une méthode d'amorçage pour l'élaboration de l'ontologie du domaine, nous suivons sur ce point la démarche suivie par (Habert, Nazarenko 1996) : la construction de l'ontologie du domaine étudié nécessite un part d'interprétation. Notre approche de classement des mots est conçu en fait pour aider à accéder aux sens (Habert *& al.* 1997).

Les processus de classement utilisent la notion d'héritage définie dans l'implémentation de Self pour réaliser la mise en place d'un réseau de comportements partagés sur l'ensemble des prototypes construits. Ces processus ajoutent de manière dynamique des liens de délégation entre les prototypes de mots ou d'arbres et des pôles de comportements partagés construits automatiquement. Le classement des mots ne s'intéresse qu'aux comportements linguistiques associés à ces mots : on cherche à évaluer les partages possibles de tels comportements. Il s'agit en particulier de chercher les prototypes d'arbres élémentaires communs à un ensemble de prototypes de mots. Si on considère les noms `stenose` et `lesion`, ils partagent des comportements (les arbres `NPivotPrepN2` et `NAdj`). Si on considère maintenant le nom `angioplastie`, celui-ci entre dans des constructions du type "indication de angioplastie" (l'arbre `N1PrepNPivot`). GASPAR construit donc un pôle de comportements partagés qui va porter les arbres élémentaires communs. Il établit un lien de délégation entre ce pôle et les prototypes concernés. Sur notre famille de mots comprenant les noms `stenose`, `lesion` et `angioplastie,` on obtient le mini-réseau suivant :

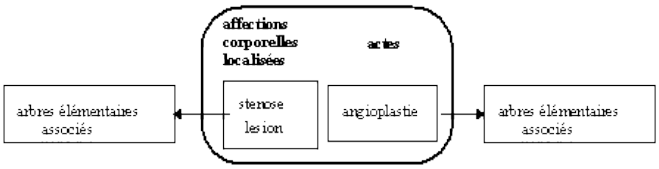

Figure 4. Un mini-réseau de comportements partagés.

Les processus construits permettent en fait d'évaluer plusieurs types de recherches de comportements partagés sur les prototypes construits. (1) GASPAR peut tout d'abord rechercher sur tous les mots d'une même catégorie s'il existe des arbres élémentaires en commun. Si tous les prototypes de mots d'une même catégorie partagent exactement les mêmes comportements (les mêmes prototypes d'arbres élémentaires), l'objet `traits` qui porte les comportements partagés de cette catégorie est mis à jour : il portera ces comportements communs. Dans tous les cas, les prototypes de mots portent, quant à eux, leurs comportements propres. (2) GASPAR peut ensuite rechercher sur les prototypes pris deux à deux s'ils partagent des arbres élémentaires. Si deux prototypes de mots d'une même catégorie partagent un ou plusieurs comportements (un ou plusieurs prototypes d'arbres élémentaires), un objet `traits` est automatiquement construit pour porter ces comportements partagés. Dans ce cas, GASPAR ajoute automatiquement aux prototypes concernés un attribut `parent` qui pointe sur ce nouvel objet porteur de comportements partagés. (3) GASPAR peut aussi évaluer les comportements partagés sur des sous-familles de prototypes de mots de même catégorie. Si plusieurs prototypes de mots d'une même catégorie partagent exactement les mêmes comportements (les mêmes prototypes d'arbres élémentaires), un objet `traits` est automatiquement construit pour porter ces comportements partagés. Dans ce cas, GASPAR ajoute automatiquement aux prototypes concernés un attribut `parent` qui pointe sur ce nouvel objet porteur de comportements partagés. Les comportements propres à chacun des prototypes leur restent attachés. (4) GASPAR permet aussi d'évaluer automatiquement les

différences comportementales des arbres élémentaires. Il est en effet possible d'établir une recherche sur les arbres élémentaires de même catégorie des comportements partagés (arbres d'analyse) par ces arbres élémentaires. Ce classement utilise une démarche similaire à celle qui est utilisée pour classer les mots. Si plusieurs prototypes d'arbres élémentaires d'une même catégorie partagent exactement les mêmes comportements (les mêmes prototypes d'arbres d'analyse), un objet `traits` est automatiquement construit pour porter ces comportements partagés. Là encore, GASPAR ajoute automatiquement aux prototypes concernés un attribut `parent` qui pointe sur ce nouvel objet porteur de comportements partagés. La figure qui suit donne une trace graphique de pôles de comportements partagés construits sur notre corpus de test. Elle présente en particulier la classe de mots regroupant les adjectifs `marginale, circonflexe, coronaire et carotide`. Dans cet exemple, ces adjectifs partagent un arbre élémentaire porté par l'objet `traits` construit. Il est à noter que la classe produite ici est sémantiquement homogène.

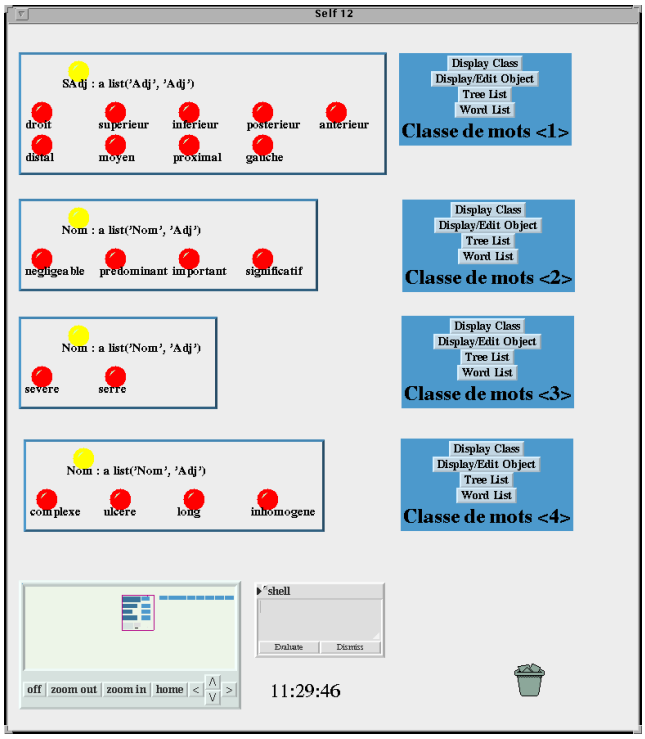

Figure 5. Classes de mots sur corpus de test.

GASPAR permet d'activer des processus de classement qui proposent des regards multiples et croisés sur les savoirs représentés. Ces processus construisent des réseaux de hiérarchies locales entre prototypes de mots et prototypes d'arbres ou entre prototypes d'arbres, ces liens multiples constituent autant de pistes de sens à interpréter. Pour le moment GASPAR construit des pôles de comportements partagés par des ensembles de prototypes. Il reste ensuite à interpréter ces pôles de comportements partagés par une intervention manuelle. En effet le classement opéré s'appuie essentiellement sur des contraintes syntaxiques, et la syntaxe ne permet pas à elle seule de délimiter des classes de noms reflétant une notion. Si GASPAR peut automatiser le classement des prototypes lexicaux sur la base des comportements qui leurs sont associés, les résultats restent à qualifier, à nommer : dans notre dispositif, c'est l'observateur conscient qui donne le sens. C'est en examinant à la main les rapprochements constatés et les classes de mot construites que l'on pourra leur donner un nom c'est à dire nommer les choses. Le réseau mis en place ne construit donc pas une représentation du sens attaché aux unités lexicales ou aux structures syntaxiques représentées, il doit proposer des amorces d'interprétation qui doivent être affinées par un travail d'interprétation plus fin. Ce réseau est conçu en fait pour aider à accéder aux sens (Habert *& al.* 1997). Les pôles obtenus et les classes de mots sous-jacentes sont des ébauches imparfaites qui permettent une organisation du matériel lexical. Ces classes doivent ensuite aider à affiner le travail de description des unités lexicales représentées sous la forme de prototypes.

5. Ajustements dynamiques et interprétations manuelles

Notre approche vise à établir une démarche interprétative contrôlée et progressive. La démarche suivie s'inscrit dans une perspective expérimentale à différents niveaux : (1) construire des représentations des mots à partir de savoirs extraits d'un corpus ; (2) construire des représentations des comportements des mots : les arbres associés ; (3) établir un premier classement. Si les informations attachées aux mots ne sont pas disponibles dès la première phase de génération des prototypes, il sera toujours possible d'ajuster les représentations construites en utilisant un nouveau flux d'informations disponibles ultérieurement. On peut par exemple projeter les résultats intermédiaires construits sur des bases de savoirs établies par ailleurs pour ensuite affiner le travail de description amorçé à la manière de la démarche suivie par A. Mikheev et S. Finch (1994). Il s'agit en fait de tendre vers une cohérence des classes sémantiques issues des processus de classement afin de dégager par affinements successifs des descriptions sémantiques pour les mots représentés. Cette construction progressive de la description des mots passe par un apprentissage manuel de nouveaux savoirs. C'est à l'utilisateur du dispositif d'interpréter et d'évaluer les objets construits et les résultats produits. En l'occurrence, cette intervention peut être réalisée par l'utilisateur linguiste du dispositif ou par un spécialiste du domaine. Ce travail d'interprétation est d'ailleurs un passage obligé de toutes les approches en classification automatique (en analyse de données par exemple, mais aussi dans des traitements syntaxiques à la Grefenstette (1993, 94).

6. GASPAR : un dispositif expérimental

Les informations utilisées sont nettement insuffisantes pour produire des résultats significatifs. Le classement opéré prend appui sur des caractéristiques grossières à la fois en raison de contraintes matérielles et de la difficulté à récupérer et organiser les informations à représenter. Les résultats produits par GASPAR sont en fait limités actuellement par les contraintes matérielles imposées par ce langage expérimental sur les machines que nous utilisons. Il faut en effet beaucoup de mémoire et d'espace disque sur les machines qui portent le système Self pour mettre en œuvre cette représentation de la mouvance. La mise en œuvre de ce dispositif peut être considérée comme une expérience pilote qui offre une image partielle, pour le moment, des traitements réalisés et des résultats à venir. Pour le travail réalisé sur les gros corpus nous avons restreint le nombre de contraintes syntaxiques associées aux mots. Dans un premier temps nous avons travaillé sur des séquences `NAdj` extraites via LEXTER. À partir de 8 754 séquences comportant des groupes nominaux, nous avons extraits 586 mots (des noms) auxquels sont attachés 1 413 arbres élémentaires de type : Sn_1 `-> Nom Adj,` Sn_2 `-> Adj Nom,` Sn_3 `-> Adj XX,` Sn_4 `-> XX Adj.` Cette première sélection a donc consisté à ne garder que les arbres binaires portant les feuilles `Nom/XX` et `Adj`. Le processus de génération conduit à la création des prototypes pour représenter les catégories syntaxiques `Nom, Adj, XX, Sn1`, `Sn2`, `Sn3`, `Sn4`, des objets `traits` associés et de plus de 2 000 prototypes par copie et ajustement. Avec les prototypes créés, GASPAR a ensuite cherché à repérer ceux qui partageaient exactement les mêmes comportements. Ce processus de classement conduit à la création automatique de 54 pôles de comportements partagés. On présente ci-dessous quelques classes de mots obtenues.

Pôles de mots partageant un arbre `SN -> Nom Adj`	**Adj**
(1) occipital bras aisselle epaule	gauche
(2) exces surcharge	ponderal
(3) octobre juillet juin mai mars avril	dernier
(4) besoin tableau	clinique

(5) staff discussion reunion exerese geste reparation resection revascularisation	medico-chirurgical
(6) equipe solution oedeme parenchyme plage coeur tuberculose vascularisation	chirurgical
(7) sommet base	pulmonaire
(8) bloc sillon	auriculo-ventriculaire
(9) expression positivite seringue	electrique
(10) oreillette ventricule retard	droit, gauche

Pôles de mots partageant un arbre `SN -> Adj Nom`	**Adj**
(17) majorite variabilite	grand
(28) ballon extension	petit

Les classes produites sont, dans l'ensemble, cohérentes mais ne produisent pas encore des résultats pertinents sur le domaine étudié : certaines classes évidentes ou prévisibles sont mises au jour. La classe de mot associée au pôle n°3 est homogène dans sa relation avec l'adjectif `dernier`, de même pour la classe n°2 dans sa relation avec l'adjectif `ponderal`. La classe n°1, où la relation de localisation qualifie un membre ou une région du corps, est elle aussi cohérente ; pour cette classe, on note que les noms qualifiés ne le sont que pour l'adjectif localisant `gauche` ; à la différence de la classe n°10, celle-ci étant moins homogène. Les classes n°5, 6, 9 regroupent quant à elles des noms sémantiquement plus éloignés. Pour enrichir ce travail de description du comportement des mots, on doit évidemment pouvoir examiner d'autres types de relation binaire. On doit aussi d'examiner en détail tous les types possibles de regroupements de mots : certains mots partagent individuellement plus de comportements avec d'autres mots. L'absence de critères numériques manque aussi pour comparer les fréquences de réalisation des proximités de cooccurrences rencontrées.

7. Perspectives

Le choix des prototypes semble cohérent avec la volonté de représenter des savoirs évolutifs. Les prototypes sont malléables : ils se construisent contextuellement et leur spécialisation se définit suivant les évolutions contextuelles. Ils peuvent commencer par fixer un savoir minimal - qu'il est possible d'attacher à une entrée lexicale - puis finir par étendre ce noyau de sens dans les directions permises par les configurations interprétatives rencontrées. Les limites de l'automatisation des processus de représentation et de classement marquent le champ de travail qu'il reste à effectuer manuellement pour mener à bien la représentation et le classement des unités lexicales. La nécessité de sous-représenter la description des mots dans un dispositif de TALN conduit de fait à une perte en richesse expressive. La syntaxe est utilisée dans notre travail comme un *" marche-pied pour l'acquisition de connaissances "* (Habert, Nazarenko 1996). La PàP permet ensuite d'établir un compromis intéressant entre formalisation et implémentation : cette démarche de représentation permet en effet de mener un travail d'expérimentation qui doit conduire à une représentation par ajustements successifs. Notre travail a porté, dans l'immédiat, sur la mise en œuvre de processus automatiques pour la représentation et le classement de mots sous la forme de prototypes. Nous n'avons pas pu testé les processus définis dans GASPAR sur des gros corpus. Les résultats actuels restent donc limités. De nombreux prolongements restent à faire. Sur le plan technique, la couverture des gros corpus doit être réalisée ; on pourrait ainsi, sur le plan linguistique, évaluer et analyser les résultats construits. Le développement d'outils pour la représentation et pour le

classement dynamiques des savoirs linguistiques doit être étendu. La réalisation d'un classement automatique général des unités lexicales semble hors d'atteinte tant qu'on ne pourra pas construire de nouvelles connaissances capables d'enrichir les savoirs déjà établis. Il reste malgré tout possible d'affiner les processus de classement mis en place et de fait d'affiner le classement visé tout en continuant à travailler sur les métaconnaissances à l'œuvre dans les faits de langue utilisés.

Remerciements

Je remercie Benoît Habert (ELI - ENS de Fontenay-St Cloud) qui a suivi pas à pas les différentes étapes de ce travail. Je remercie également Didier Bourigault (DER-EDF), Benoît Habert et Adeline Nazarenko pour m'avoir donné accès aux résultats de LEXTER et de ZELLIG. Je n'oublie bien évidemment pas les membres du groupe Self à Stanford et à Sun Microsystems qui ont développé le langage Self et qui ont toujours répondu avec bienveillance à mes sollicitations.

Références

Blascheck G. (1994), *Object-Oriented Programming with Prototypes*, Springer-Verlag, Berlin

Bourigault Didier (1993), " An endogeneous Corpus-Based Method for Structural Noun Phrase Disambiguation ", Actes *EACL*, p. 81-86

Fleury Serge (1997), *La programmation à prototypes, un outil pour une linguistique expérimentale. Mise en oeuvre de représentations évolutives des connaissances pour le traitement automatique du langage naturel*, Thèse de doctorat, Paris 7-Denis Diderot

Grefenstette Gregory (1993), " Automatic Thesaurus Generation from Raw Text using Knowledge-Poor Techniques ", 9th Annual Conference of the university of Waterloo, Centre for the New Oxford English Dictionary and Text Research, Oxford

Grefenstette Gregory (1994), " Corpus-Derived First, Second and Third-Order Word Affinities ", Actes *EURALEX*, Amsterdam

Habert Benoît, Nazarenko Adeline (1996), " La syntaxe comme marche-pied de l'acquisition de connaissances : bilan critique d'une expérience ", *Journées d'Acquisition de Connaissances*

Habert Benoît, Salem André, Nazarenko Adeline (1997), *Les linguistiques de corpus*, Armand Colin, Paris

Harris Zellig (1970), " La structure distributionnelle ", Langages n°20, Larousse, Paris

Harris Zellig (1988), *Language and Information*, Columbia University Press, New York

Mikheev Andrei, Mœns Marc (1994), " Acquiring and Representing Background Knowledge for a natural Language Processing System ", Proceedings *AAAI*

Self Group : Agesen O., Bak L., Chambers C., Chang B.W., Hölzle U., Maloney J., Smith B.R., Ungar D., Wolczko M. (1995), " The Self 4.0 Programmer's Reference Manual ", Sun Microsystems, Inc. and Stanford University

Smith Walter R. (1995), " Using a Prototype-based Language for User Interface : The Newton Project's Experience ", Proceedings *OOPSLA*, p. 61-72, SIGPLAN Notices

Ungar David, Smith Randall B. (1987), " SELF : The power of Simplicity ", Proceedings *OOPSLA*, SIGPLAN Notices

Zweigenbaum Pierre (1994), " MENELAS : an Access System for Medical Records using Natural Language ", *Computer Methods and Programs in Biomedicine*, volume 45, p. 117-120