Représentation évolutive de mots

Serge Fleury

UMR 9952)
Ecole Normale Supérieure de Fontenay-St Cloud
31 avenue Lombart
F-92260 Fontenay aux Roses
fleury@ens-fcl.fr

RESUME. *Cet article présente le dispositif expérimental 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. Cette phase de classement prend appui sur une recherche de clusters d'entités ayant un comportement plus homogène. 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.*

ABSTRACT. *This paper present a NLP system called GASPAR. This system's aims is first to build evolutive representation for words with objects called prototypes, second to classify these objects according to the linguistic information associated to these objects. Our goal is to develop an approach in which word representations can continually evolve according to their behaviors in corpus. This work of progressively refining word representations leans on a method of clustering words. We present the prototype-based paradigm and its applications for the representation and for the classification of words represented as prototypes. We describe how we can use a prototype-based language to generate automatically objects, defined step by step, and how a prototype-based language allows to classify these objects according to the associated constraints*

MOTS CLES : *Linguistique basée sur corpus, langage à prototypes, apprentissage et classification automatique*
KEYWORDS : *Corpus linguistics, prototype-based language, learning and classifying automatically*

Revue d´Intelligence Artificielle. Volume 1-n° 1/1998, pages 1 à x

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 [FLE 97]. Ce dispositif vise à établir une représentation et un classement évolutifs de mots représentés sous la forme d'objets informatiques appelés des prototypes. Notre travail de représentation s'inscrit dans une approche expérimentale qui ne présume pas complètement des choses à représenter. L'hypothèse suivie pour la représentation des informations linguistiques consiste à ne pas prédéterminer de manière figée ni les structures définies pour cette représentation ni leurs classements. Notre démarche coïncide avec une approche ascendante pratiquée dans le cadre du data mining[1] et qui doit conduire à une « découverte de connaissances ». Pour cela, on prend appui sur les données brutes sans hypothèse particulière quant à l'organisation de celles-ci. Dans ce cadre, notre travail vise à repérer des similitudes entre les informations linguistiques traitées pour guider la représentation des mots puis pour organiser le classement de ces mots. Notre approche met en place un processus de représentation évolutif : les structures de représentation construites devront pouvoir être affinées dès que de nouveaux savoirs seront mis à jour. On peut en effet considérer que les comportements des mots ne sont pas tous prédéfinis mais que ceux-ci émergent dans le contexte dans lequel ces mots agissent. Il n'est donc pas raisonnable de les considérer comme acquis par définition. Il s'agit au contraire de les mettre en lumière ainsi que les corrélations multiples qui existent entre les mots dans un flot continu de discours. Notre démarche consiste en quelque sorte à « faire émerger » les comportements des mots puis à les représenter ou à affiner les représentations existantes et enfin à classer les structures de représentation construites. L'enrichissement des connaissances repose dans notre travail sur l'utilisation de corpus dans des domaines de spécialité, ici les comptes rendus d'hospitalisation en médecine coronarienne. Nous ne nous imposons ni une représentation prédéfinie des informations linguistiques ni un classement prédéfini de ces informations, mais nous choisissons une démarche qui tente de construire ces représentations puis d'établir progressivement leur classement en tenant compte des informations délivrées par le travail d'acquisition de savoirs à partir de corpus.

Le dispositif construit a pour but : d'une part, de construire des représentations évolutives pour les mots à partir d'informations extraites sur corpus ; d'autre part, il doit conduire à la construction de classes de mots de manière inductive. Cette phase

1 « Le data mining /./ est l'exploration et l'analyse de grandes quantités de données afin de découvrir des formes et des règles significatives en utilisant de moyens automatiques ou semi-automatiques » [BER 97] (p. 6). « Le data mining n'a de sens que si l'ensemble de données est suffisamment volumineux. De fait, la plupart des algorithmes d'exploration de données exigent de grands volumes afin de construire des modèles d'apprentissage qui seront utilisés pour réaliser la classification, l'estimation et les autres tâches de data mining » [BER 97] (p. 7).

de classement prend appui sur une recherche de clusters[2] d'entités ayant un comportement plus homogène. Les classes de mots produites peuvent ensuite être utilisées pour affiner la représentation des mots ; elles doivent aussi conduire à un étiquetage de mots dans des corpus de spécialité. Ce travail utilise le prototype[3] (dans le cadre de la programmation à prototypes [BLA 94]) 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'oeuvre dans la construction du sens dans le langage naturel. Nous soulignons d'abord les problèmes posés par la représentation d'unité 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 automatique des mots et enfin nous présentons les premiers résultats construits avec GASPAR.

2. Origine et choix

2.1. *Comment représenter la mouvance ?*

Notre travail vise à la mise au point d'outils de modélisation adaptés à des problèmes fondamentaux de représentation des connaissances en sémantique lexicale. L'hypothèse suivie dans ce travail est que ces connaissances ne sont pratiquement jamais complètes et intangibles. Au contraire, il faut gérer leur extension et leur modification. Un examen des comportements des mots révèle des variations[4] qu'il semble difficile de fixer dans des structures de représentations statiques. La langue évolue en permanence et les résultats acquis par la description linguistique sont toujours remis en question par la prise en compte de nouvelles informations. De plus, certaines utilisations peuvent prendre le contre-pied de résultats généraux établis par ailleurs : il peut y avoir en effet des différences considérables dans l'utilisation des mots quand on passe d'un sous-langage à un autre [BIB 93]. Si les descriptions linguistiques se bornent à refléter un savoir partagé sur le monde, en particulier si le lexique est organisé (de manière hiérarchique ou non) sur la base de connaissances (le monde objectif), on perd une grande partie des mécanismes à l'oeuvre dans le langage naturel. L'interprétation

---

[2] *« La classification par clusters est une des rares activités d'exploration de données qui puisse être qualifiée de découverte non dirigée de connaissances ou d'apprentissage sans supervision »* [BER 97].

[3] Notre tâche n'est pas de prendre appui sur la théorie des prototypes des cognitivistes pour théoriser des faits de langue.

[4] Les savoirs généraux que l'on peut associer aux mots ne sont pas toujours pertinents [BIB 93].

permet en effet la mise en oeuvre de processus sémantiques qui créent de nouvelles acceptations et de nouveaux emplois non dérivables de la connaissance encyclopédique. Dans ce cas les parcours (arborescents ou non) n'apprennent rien que l'on ne sache déjà. Figer les savoirs linguistiques dans des taxonomies figées et non évolutives ne convient guère à la structure sémantique des langues et assez mal à celle des discours de spécialité [BIB 93]. Un travail de représentation des faits de langue doit donc s'attacher à rendre les savoirs de base évolutifs en sachant qu'un ajustement est toujours potentiellement à venir sur ces savoirs.

2.2. *Pourquoi est-il utile, nécessaire, intéressant... d'attacher des informations aux mots ?*

La langue présentant trop de variations pour pouvoir être appréhendée globalement, il est illusoire de vouloir traiter globalement tous les problèmes qu´elle présente. Ces variations se retrouvent le plus souvent au niveau des mots où elles sont plus directement accessibles. Notre travail s'inscrit dans une approche lexicaliste et vise à associer aux mots les informations minimales qu'il est possible de leur associer et qui permettent des ajustements ou des affinements potentiels à venir. Le problème à résoudre consiste donc à représenter ces informations initiales en tenant compte de ce que l'on a affaire à des informations évolutives et non figées puis il convient de définir des processus qui permettent d'enrichir ces informations.

2.3. *Quelles informations peut-on associer aux mots ?*

D'un point de vue sémantique, il est clair qu'il est impossible de dresser une liste stable de toutes les significations de chaque mot. Le sens des mots n'est pas une donnée figée, il peut varier dans des contextes particuliers. En médecine coronarienne, `longue` marque le degré dans la séquence nominale "une occlusion longue». Dans d'autres sous-langages, cet adjectif n'est pas typé pour marquer le degré. De plus, la création toujours possible de nouvelles interprétations ne permet pas de prévoir les comportements sémantiques des mots de manière absolue. La représentation informatique des mots doit donc rester ouverte et proposer des pistes de sens. De même si on examine les comportements syntaxiques des mots, les différences se manifestent très clairement. D'une part, les mêmes comportements syntaxiques ne se réalisent sur tous les mots d'une même catégorie, ils se distribuent sur des sous-familles particulières. D'autre part, ces comportements ne se réalisent pas de manière uniforme sur les différentes familles de mots d'une même catégorie. Sur telle famille de mots, un comportement se réalise d'une manière qui est différente de sa réalisation sur une autre famille ; `artere` et `infarctus`, par

exemple[5], entrent dans des relations de localisation et on a deux réalisations distinctes : `artere` est associé aux adjectifs {`coronaire circonflexe diagonal`}, alors qu'`infarctus` est associé aux adjectifs {`anterieur inferieur apical`}.

2.4. *Un problème : représenter et classer les mots*

Notre travail s'inscrit dans une reprise de l'approche harrissienne [HAR 70-88] et vise à automatiser les traitements de représentation des mots et de leur classement, en utilisant des informations extraites à partir de corpus, et à souligner les limites de cette induction de savoirs. Notre travail de représentation et de classement s'appuie sur une recherche initiale au niveau des mots des régularités et des redondances d'utilisation dans des corpus donnés. Dans ce cadre, notre travail vise à repérer des similitudes entre les informations linguistiques traitées pour guider le travail de représentation des mots puis pour organiser le classement de ces mots.

2.4.1. *Représenter les mots*

La première tâche à résoudre consiste à construire des représentations évolutives des mots et de leurs comportements. 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. Notre travail vise à prendre appui sur des connaissances très générales qui sont affinées voire remodelées et changées au gré des observations rencontrées. Le travail de représentation mis en place ne construit donc pas une représentation prédéterminée du sens attaché aux mots ou aux structures syntaxiques représentées, il doit proposer des amorces d'interprétation qui doivent être affinées par un travail d'interprétation plus fin.

2.4.2. *Classer les mots suivant leurs comportements*

La seconde tâche à résoudre va consister à reclasser les informations représentées en prenant appui sur le fait que la représentation d´informations évolutives implique un classement évolutif de ces informations. Nous convenons avec [HAT 91] que *«la construction d'une hiérarchie est un processus incrémental»* et qu'une hiérarchie *«évolue et s'améliore en fonction des résultats obtenus jusqu'à ce qu'une certaine forme de stabilité soit atteinte»*. Cette phase de classement prend appui sur une recherche de clusters d'entités ayant un comportement plus homogène. Les processus de classement visent ainsi à organiser le matériel lexical dans un domaine de spécialité afin de déterminer les classes sémantiques sous-jacentes aux

[5] Les exemples présentés ici sont issus du corpus MENELAS [ZWE 94].

classes de mots construites [HAB 97b]. Il convient de souligner que le classement automatique présenté ici s'appuie principalement sur des contraintes syntaxiques associées 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[6]. A l'inverse des approches harrissiennes[7] et statistiques[8], 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 [HAB 96a] : 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 [HAB 97b]. Il est important de souligner que les résultats construits ne sont pertinents que pour les corpus étudiés. Les relations syntactico-sémantiques mises en avant par le travail d'extraction de savoirs sont propres aux corpus examinés[9], le travail de représentation et de classement des mots sur la base des savoirs reçus est donc lui aussi complètement dépendant de ces corpus. Si on peut envisager d'élaborer une ontologie à partir des résultats construits (un travail d'interprétation sur les résultats construits reste à définir et à faire), celle-ci ne sera elle aussi pertinente que pour le corpus étudié [ZWE 97].

2.5. *Un Corpus*

Notre approche de représentation vise à repérer des informations à partir de réalisations rencontrées sur corpus [HAB 97b]. Le travail d'acquisition de connaissances sur corpus débouche sur la mise au jour d´informations syntaxico-sémantiques qu'il est possible d'attacher aux mots. Le corpus utilisé est celui qui est constitué dans le cadre du projet MENELAS [ZWE 94] 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. On utilise donc le

---

[6] *« La détection automatique de clusters est rarement utilisée seule parce que la recherche des clusters n'est pas une fin en soi. Une fois les clusters détectés, d'autres méthodes doivent être appliquées pour dégager leur signification »* [BER 97].

[7] Dans l'approche suivie par N. Sager, les classes de mots et les formules syntaxiques d'une grammaire de sous-langage sont données comme en correspondance étroite avec les classes d'objets du monde réel et à leurs relations, le travail d'interprétation des résultats construits semble sous-estimé.

[8] Dans les approches statistiques, les classes de mots sémantiquement pertinentes sont obtenues statistiquement et induites du corpus à partir des seules mesures de similarités entre mots : *« En réalité, les listes de mots obtenues ne constituent pas de véritables classes de mots cohérentes et homogènes. Il est souvent difficile de leur attribuer une étiquette globale, et ce d'autant plus que les mesures de similarité constituent des résumés statistiques bruts dont les critères de construction sont effacés »* [HAB 96a].

[9] *« Un corpus de spécialité ne permet pas de repérer toutes les relations, bon nombre d'entre elles font partie de l'implicite partagé par les locuteurs du sous-langage et n'effleurent pas »* [HAB 96a].

corpus MENELAS à partir duquel une chaîne de traitement extrait des informations pour construire des représentations des mots. Les informations extraites sont des arbres d'analyse fournis par des analyseurs robustes, ces arbres étant ensuite simplifiés dans le but de déterminer les arbres minimaux qu'il est possible d'associer aux mots. Les arbres minimaux et les arbres d'analyse associés décrivent une partie de ce que nous appelons ici le comportement de ces mots.

2.6. *Un outil*

L'outil de représentation choisi est la programmation à prototypes (PàP) [BLA 94] dont les principaux domaines d'application se situent dans le développement d'interfaces utilisateur [SMI 95]. 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 sur le domaine visé. 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. La flexibilité du modèle de représentation choisi constitue une propriété fondamentale pour développer des processus de représentation capables de rendre compte des aspects mouvants du domaine décrit. Par la suite nous utilisons les expressions suivantes : on appelle prototype de mot/d'arbre l'objet informatique (le prototype) défini pour représenter un mot/arbre.

3. Acquisition de connaissances en corpus

La phase d'acquisition de connaissances à partir de corpus prend appui sur la systématicité structurelle et sémantique propre aux sous-langages [HAR 70-88] afin de mettre au jour les proximités de cooccurrences entre mots pour dégager les relations sémantiques sous-jacentes. Le point de départ du travail de représentation est donc constitué par le corpus MENELAS. 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[10], et ZELLIG[11]. Le but de ces outils est d'une part d'extraire des informations à partir de corpus (LEXTER) et d'autre part de simplifier ces informations puis de caractériser leurs fonctionnements (ZELLIG).

LEXTER prend en entrée des textes arbitrairement longs et produit des arbres d'analyse de séquences nominales en décomposant ces séquences en Tête et Expansion et ce de manière récursive. Il est important de souligner que la démarche de LEXTER est endogène c'est à dire que le travail d'analyse s'appuie uniquement sur les résultats d'analyse déjà construits pour analyser des séquences ambigües. Si le corpus contient la séquence «angine de poitrine instable», on a deux analyses possibles pour cette séquence : (1) `[angine de poitrine] instable`, (2) `angine de [poitrine instable]`. LEXTER va prendre appui sur des séquences déjà analysées pour produire une analyse de cette séquence. Si on a la séquence «angine de poitrine» et si l'on n'a pas «poitrine instable», LEXTER produira l'analyse (1). L'approche endogène suivie par LEXTER s'inscrit parfaitement dans le cadre de représentation qui est le nôtre. ZELLIG a ensuite pour tâche de simplifier les arbres d'analyse fournis ici par LEXTER et de mettre en évidence les relations élémentaires de dépendance entre mots pleins, nom ou adjectif, dans des schémas comme, par exemple, `N Prep N` ou `N Adj`. Les arbres issus de LEXTER sont d'abord transformés en arbres syntagmatiques via le transducteur FRT (un module de ZELLIG) [HAB 97a]. Le programme Cyclade (un autre module de ZELLIG) est ensuite chargé de déterminer les arbres élémentaires via un filtrage de quasi-arbres [HAB 96b]. Ces dépendances élémentaires associent à un élément gouverneur (nommé `tête`) soit un `argument` soit un `circonstant`. Les dépendances élémentaires constituent en fait un point d´entrée pour amorcer la délimitation de classes sémantiques des mots sur la base de leurs comportements effectifs dans le corpus.

On résume ici de manière synthétique ce que la chaîne de traitements réalise pour produire les arbres minimaux à partir du corpus initial. Le point de départ est constitué de rapports médicaux dont on donne un extrait ci-dessous.

Patient âgé de 70 ans, diabétique, qui a présenté il y a un an une douleur thoracique nocturne probablement en rapport avec un infarctus antéro-septal. Il est toujours symptomatique sous la forme d'un angor d'effort qu'il a lui-même négligé, avec semble-t-il plusieurs épisodes de préchordialgies de repos. La coronographie met en évidence des lésions bitronculaires. L'occlusion de l'IVA est responsable

---

[10] LEXTER [BOU 93] est un outil d'acquisition terminologique conçu et réalisé dans un environnement industriel : la Direction des Etudes et Recherches de EDF, pour aider à la mise au point de terminologie.

[11] Le logiciel ZELLIG [HAB 96a-96b] est une chaîne de recyclage développée par Benoît Habert et Adeline Nazarenko en C sous Linux et Solaris. Elle a bénéficié en outre, au sein de l'ELI à l´ENS de Fontenay Saint Cloud des apports de Cécile Fabre, Helka Folch et Elie Naulleau. ZELLIG se situe en amont d´analyseurs robustes préexistants dont il recycle les résultats.

d'une hypokinésie antérieure. Une sténose serrée, diagonale et circonflexe est responsable de l'angor d'effort.

LEXTER produit ensuite des arbres d'analyse de séquences nominales en décomposant ces séquences et en les structurant. Sur la séquence «alteration severe de la fonction ventriculaire gauche», on obtient l'analyse suivante.

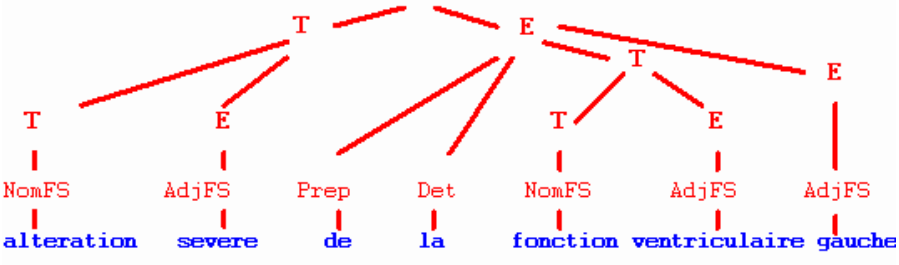

Figure 1. *LEXTER*

Enfin, ZELLIG met en évidence les dépendances élémentaires dans ces arbres d'analyse. À partir de l'arbre précédent, les arbres élémentaires (a, b, c, d) mis au jour sont décrits dans la figure qui suit :

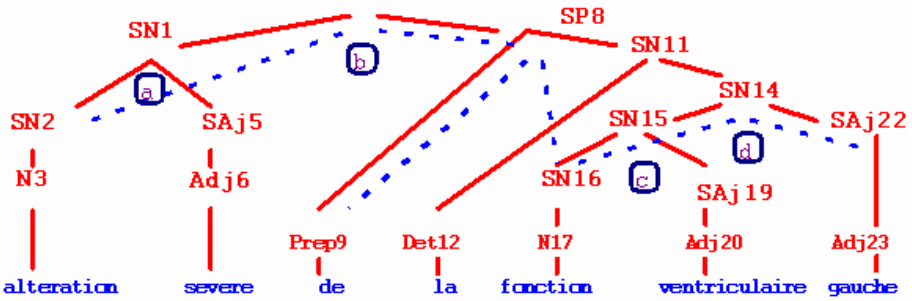

Figure 2. *ZELLIG*

4. Esquisse d'une démarche de représentation

Suite au travail réalisé par LEXTER puis ZELLIG, on dispose d'arbres associés aux mots. GASPAR est connecté à une partie particulière de la chaîne ZELLIG : il n'utilise que les arbres minimaux produits par la chaîne : Analyseur -> FRT -> Cyclade. Ces informations sont dispersées sur l'ensemble des mots et ne contiennent pas de généralisations. C'est à partir de ce savoir éclaté que l'on choisit de donner une représentation informatique des mots et de leurs comportements. La démarche

de représentation que nous allons suivre consiste à utiliser ces informations pour construire des objets informatiques tout en essayant d'utiliser les similarités observées pour élargir le domaine à représenter et réduire cet éclatement des connaissances. Cyclade met en particulier en avant des proximités comportementales entre mots de même catégorie. Si on considère les noms `stenose` et `lesion`, ils appartiennent à la même famille catégorielle et ils partagent des comportements (les arbres `NPivotPrepN2` et `NAdj`). On va donc utiliser la représentation de l'un de ces mots pour représenter l'autre. Si on construit une représentation informatique de `stenose`, on construira celle de `lesion` par copie et ajustement à partir de celle de `stenose`. De même si on construit une représentation informatique des arbres élémentaires associés à `stenose`, on utilisera ces représentations pour construire celles des arbres élémentaires associés à `lesion`. Et enfin, puisque ces deux mots partagent des informations (les arbres élémentaires `NPivotPrepN2` et `NAdj`), on factorisera les comportements communs aux représentations de ces deux mots (ces mêmes arbres élémentaires). La démarche présentée correspond en fait à celle qui est fréquemment utilisée dans une approche de représentation informatique utilisant des prototypes. Ce mode de représentation s'appuie sur des connaissances contextuelles des entités à représenter : on construit des objets en fonction des informations disponibles sur des entités particulières. La PàP met en place une représentation des informations que l'on peut considérer comme peu organisées au départ. La PàP ne rejette pas pour autant toute idée de classification, bien au contraire. Elle met d'ailleurs en place un système simple d'héritage qui permet de factoriser localement des comportements partagés.

5. Un outil de représentation : la Programmation à Prototypes

La PàP [BLA 94] s'est développée à la fin des années 80, à partir des travaux entre autres de Liebermann [LIE 86] et de l'équipe d'informaticiens de Stanford [SEL 93-95] qui ont par la suite développé le langage à prototypes Self.

5.1. *Programmation à Prototypes vs. Programmation à Objets*

Pour introduire la présentation de la PàP, il convient de mettre en avant la différence d'approche qui existe entre la Programmation à Objets (PàO) dans les langages de classe et la Programmation à Prototypes. Dans la PàO, on définit au préalable une classe abstraite dans laquelle sont définies des propriétés. À partir de cette abstraction, de ce moule, on peut créer des objets. On est donc obligé de prédéterminer toutes les propriétés des objets à représenter, ces propriétés étant définies dans le moule que constitue la classe qui sert de support à la définition des futurs instances de cette classe. La PàP utilise une démarche inverse. À partir d'un élément particulier d'un domaine à représenter, on va construire un objet informatique sans passer par un filtre prédéfinitoire : le lien classe-instance de classe n'existe pas dans la PàP. La PàP ne prédétermine donc pas toutes les propriétés des

objets d'un domaine à représenter. 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. Il convient 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 de ce 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.

5.2. *Un mot, un prototype*

Pour illustrer les principes de la PàP, on va donner l'exemple d'une représentation de la catégorie grammaticale des noms. On part donc d'un représentant particulier de cette catégorie, le nom `pontage`, qui va servir de point de départ pour la représentation de cette catégorie avec des prototypes. On construit donc un objet informatique, le prototype `pontage`, à partir d'un savoir (de sens commun) que l'on a sur ce nom : *pontage est un nom masculin singulier*. On crée donc un objet qui porte des attributs qui porte ces valeurs.

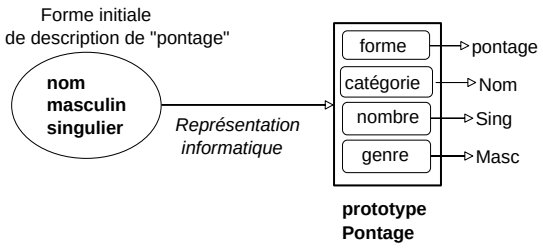

Figure 3. *Un mot, un prototype*

Pour représenter un autre élément de la catégorie des noms, par exemple le nom `lesion`, on utilise le clonage puis l'ajustement. L'opération de clonage appliquée au prototype `pontage` permet l'obtention d'une copie conforme de ce prototype. Pour représenter le nom `lesion`, on ajuste les attributs du prototype résultant en donnant à ces derniers les valeurs adéquates. Après clonage du prototype `pontage`, les attributs `forme` et `genre` sont donc mis à jour.

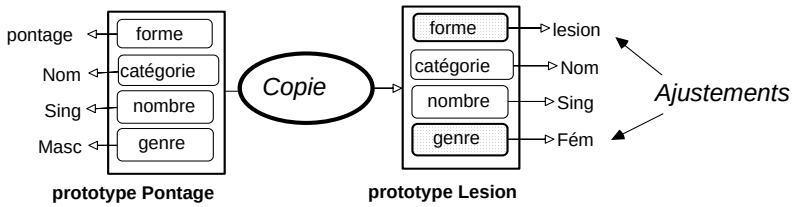

Figure 4. *Copie et ajustements*

On peut ensuite mettre en place du partage local d'information via le mécanisme de délégation. La délégation est en fait l'opération qui consiste à factoriser localement les informations partagées par un ensemble de prototypes.

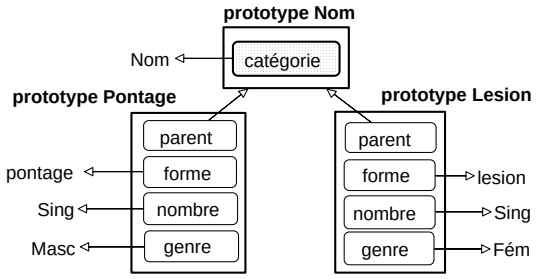

Figure 5. *Partage local d'attribut*

Dans le cas de nos prototypes, ces deux objets possèdent en commun l'attribut `catégorie`, par exemple, qui traduit leur appartenance à la même catégorie grammaticale. On peut donc définir un objet `prototype Nom` qui va porter cet attribut. Et l'attribut `parent` que l'on ajoute aux prototypes `pontage` et `lesion` indique que ces derniers délèguent un attribut au `prototype Nom`.

5.3. *Self : présentation*

Self est le langage à prototypes qui est utilisé ici pour représenter les mots et leurs comportements. Self permet l'héritage multiple et l'héritage dynamique. Il a été conçu en 1986 par David Ungar & Randall Smith [UNG 87]. La première implémentation a été réalisée à Stanford en 1987. La dernière version, Self-4.0[12], est disponible depuis juillet 1995 [SEL 95]. Ce langage est désormais développé par Sun Microsystems.

[12] Self est disponible à l'adresse suivante : http://self.sunlabs.com

5.3.1. *Concrétude, uniformité et flexibilité*

Les propriétés fondamentales du langage Self et de l'interface utilisateur sont la concrétude, l'uniformité et la flexibilité. La concrétude se manifeste clairement par le fait qu'un utilisateur manipule directement les objets du monde Self et peut en créer de nouveaux à partir d'objets existants en les dupliquant et en les ajustant. L'uniformité se traduit par le fait qu'en Self tout est objet et que tous les objets dialoguent entre eux par envoi de messages. Les attributs des objets peuvent porter des données ou des méthodes. De plus la distinction entre l'implémentation d'un programme et son exécution n'existe pas en Self : on peut ajuster dynamiquement les objets. Et enfin, l'environnement graphique de Self est composé d'éléments de base (les `morphs` [MAL 95]) disponibles et réutilisables à volonté. La flexibilité découle directement des aspects de concrétude et d'uniformité de Self.

5.3.2. *Héritage et Self : la délégation, une vision dynamique de l'héritage*

L'héritage en Self se réalise au travers de la notion de délégation. Pour mettre en place l'héritage, on commence par créer des objets qui vont porter les comportements partagés. On établit ensuite un lien entre ces objets et les prototypes qui délèguent les comportements factorisés en inscrivant dans ces prototypes le chemin de délégation adéquat. Pour inscrire un chemin de délégation dans un objet on ajoute un attribut `parent` qui pointe sur l'objet qui porte les comportements délégués. Dans le langage Self, un parent commun à plusieurs prototypes est appelé un objet `traits`.

6. Construction inductive des connaissances

GASPAR est une chaîne de traitements automatiques pour la représentation et le classement automatiques de mots et de leurs comportements syntaxico-sémantiques. GASPAR dispose, au départ, d'informations extraites à partir d'un corpus (sous la forme d'un fichier texte) : pour chaque mot, 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 (des contraintes peuvent être attachées aux composants des arbres, on y reviendra *infra*). Ces descriptions sont généralement peu précises ou en tout cas sous-déterminées. GASPAR utilise uniquement ces informations pour construire des prototypes afin de représenter les mots et leurs comportements (les arbres associés).

6.1. *Représentation dynamique des mots*

Le processus de génération des mots se déroule de la manière suivante. Pour chaque mot, GASPAR vérifie s'il existe une représentation prototypique

équivalente. Si le mot à représenter possède une représentation prototypique, GASPAR conserve l'objet trouvé. Si le mot à représenter ne possède pas de représentation prototypique, et s'il n'existe aucune représentation prototypique de sa famille catégorielle, 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 6).

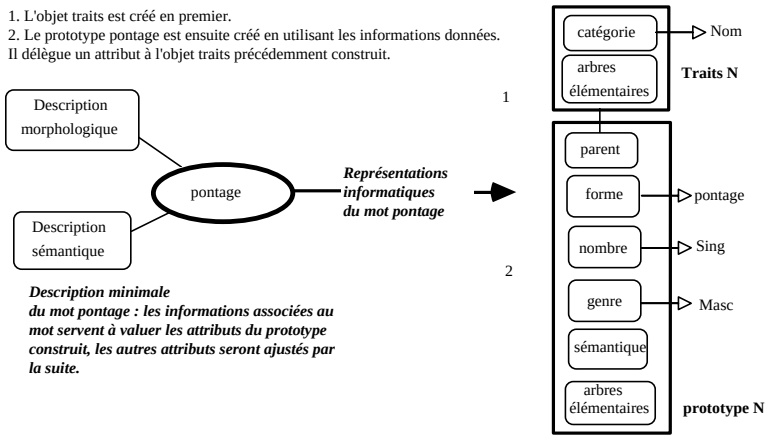

Figure 6. *Génération automatique d'un prototype de mot*

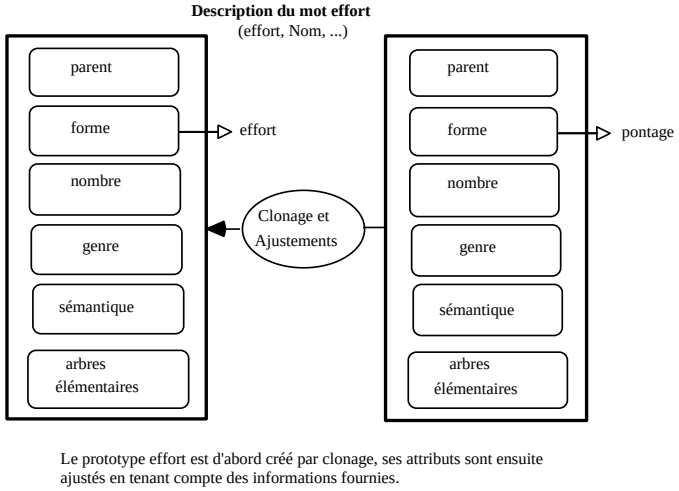

Figure 7. *Génération d'un prototype de mot par clonage et ajustement*

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 (en tenant compte des informations fournies pour décrire ce nouvel élément) (figure 7).

6.2. *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'arbre(s) élémentaire(s) à représenter.

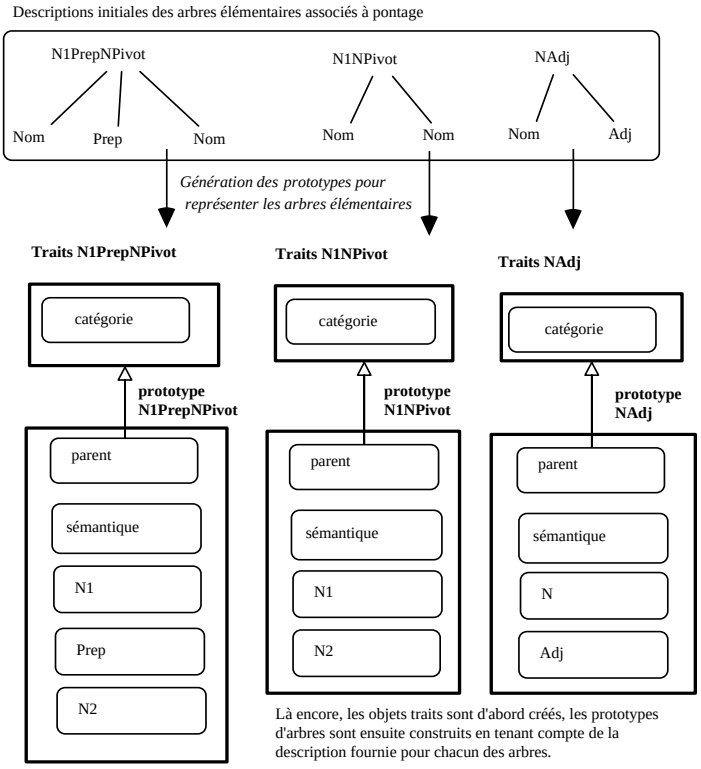

Figure 8. *Génération de prototypes d'arbre*

Avant de représenter ces arbres élémentaires, GASPAR vérifie si ces arbres disposent déjà d'une représentation prototypique. Si elle n'existe pas, il la crée

automatiquement en tenant compte des informations fournies : constituants et contraintes[13]. Dans la figure précédente, GASPAR construit les structures pour représenter les arbres associés à `pontage` en tenant compte des informations données pour la description de ces arbres. Pour chaque arbre élémentaire construit, on peut avoir une liste d'arbre 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 crée automatiquement en tenant compte des informations fournies : constituants et contraintes. GASPAR affecte ensuite aux prototypes de mots construits leurs comportements. Après la création des prototypes d'arbres élémentaires, il affecte 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.

6.3. *Représentation évolutive des mots*

GASPAR utilise des informations issues d'un travail d'extraction à partir de corpus. À chaque entrée lexicale sont associées les informations suivantes : (1) une description morphologique et sémantique du mot ; (2) une suite de contextes : un arbre élémentaire ou une suite d'arbres élémentaires auxquels est associée éventuellement une suite d'arbres d'analyse. Ces informations peuvent être connues avant la phase de génération automatique des objets ou peuvent être utilisées dès qu'elles sont disponibles pour affiner la représentation des objets construits. Pour chaque entrée lexicale lue, GASPAR est donc amené à construire : (1) un prototype de mot (ce prototype porte une représentation morphologique et sémantique sous-déterminée du mot visé) ; (2) une suite de prototypes d'arbres élémentaires ; (3) à chaque prototype d'arbre élémentaire est associée éventuellement une suite de prototypes d'arbres d'analyse. GASPAR construit aussi les liens qui existent entre chacun des objets construits. Pour chaque entrée lexicale, GASPAR a donc construit un mini-réseau de prototypes décrivant la micro-syntaxe associée à ce mot. La représentation des mots reste bien évidemment sous-déterminée. C'est l'examen des contextes (les arbres associés) qui doit permettre de tracer des pistes de sens. Si de nouvelles informations sont disponibles, on peut ensuite affiner la représentation syntaxico-sémantique des mots et des arbres en utilisant le potentiel dynamique de Self (ajustement dynamique des objets). Dès cette phase de représentation, l'utilisateur peut intervenir pour ajuster les objets construits suivant les nouvelles informations disponibles.

7. Classement évolutif des prototypes

[13] Les contraintes associées à un arbre décrivent par exemple des informations qui filtrent ou sélectionnent les constituants possibles pour cet arbre.

Après la phase de représentation des mots et de leurs comportements syntaxiques, GASPAR peut amorcer un début de classement des prototypes en fonction de leurs comportements syntaxiques. Pour réaliser la mise en place de ce réseau de comportements partagés sur les objets construits, GASPAR utilise la notion d'héritage définie dans l'implémentation de Self. Ce dernier permet l'héritage multiple : il n'y pas de restriction dans le nombre de parent possible pour un prototype donné. Self permet aussi l'héritage dynamique : on peut à tout moment modifier un lien de délégation sur un prototype donné. Soit en ajoutant un nouveau lien de délégation, soit en modifiant un lien existant si ce dernier est réalisé via un attribut assignable, soit en supprimant un lien. Les processus de classement utilisent ces deux propriétés fondamentales 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.

7.1. *Nature des classements*

Notre approche distingue deux types de classement. Un premier classement concerne une possible factorisation maximale des comportements partagés par l'ensemble des prototypes disponibles dans le système ou par des sous-ensembles de prototypes. Ces comportements s'appliquent à tous les types possibles de comportements définis dans les objets. Une factorisation de tels comportements[14] permet de ne pas avoir à dupliquer du code dans de nombreux objets : une éventuelle mise à jour de ce code partagé peut ainsi être réalisée globalement pour tous les prototypes déléguant ces comportements décrits par ce code. Le deuxième classement escompté ne concerne que l'examen des prototypes au regard des informations linguistiques qui leurs sont attachées. Le classement des mots en ce sens signifie que l'on s'intéresse aux comportements linguistiques associés à ces mots et que l'on cherche à évaluer les partages possibles de tels comportements. Pour les mots, il s'agit en particulier de chercher les prototypes d'arbres élémentaires communs à un ensemble de prototypes de mots. Les processus présentés *infra* se situent dans cette seconde perspective de classement.

7.2. *Des regards multiples sur les mots*

La mise en œuvre des processus de classement construits confirme les multiplicités de comportements possibles sur les mots. Comme on pouvait s'y attendre, on ne trouve jamais de comportement(s) partagé(s) par tous les membres d'une même famille catégorielle. Il convient donc d'interroger les objets construits

[14] Le travail de Moore [MOO 95, 96] constitue une solution adaptée pour la résolution de ce classement.

de manière plus fine si on veut y découvrir des similarités comportementales [HAB 96a]. Les processus construits permettent en fait d'évaluer plusieurs types de recherches de comportements partagés sur les objets construits.

7.2.1. *Recherche sur tous les mots d'une même catégorie des comportements partagés*

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.

7.2.2. *Recherche des arbres élémentaires communs à deux prototypes*

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.

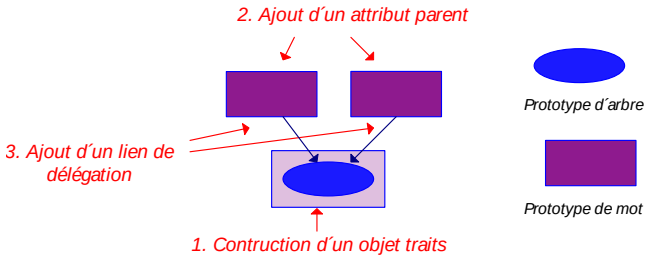

Figure 9. *Classement des mots deux à deux*

7.2.3. *Recherche sur des sous-familles de mots d'une même catégorie des comportements partagés et construction automatique d'un pôle de comportements partagés*

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.

7.2.4. *Recherche sur des sous-familles d'arbres élémentaires d'une même catégorie des comportements partagés et construction d'un pôle de comportements partagés*

Le dispositif 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. Les résultats sur le classement des arbres élémentaires peuvent donc être utilisés pour affiner le classement mis en place pour les mots. C'est en examinant les résultats construits à la suite de ces deux phases de classement qu'il est possible de déterminer si ces délégations de comportements à double niveau enrichissent la description des mots et de leurs comportements.

8. Mise en place de réseaux de prototypes : autant de réseaux à interpréter

Le dispositif construit permet donc d'activer des processus de classement qui proposent des regards multiples et croisés sur les mots représentés.

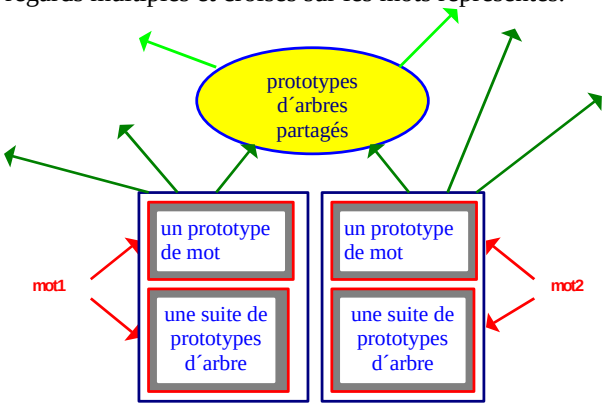

Figure 10. *Un réseau de prototypes*

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 le dispositif construit des pôles de comportements partagés par des ensembles de prototypes. Ce classement[15] s'appuie sur les comportements syntaxiques attachés aux prototypes de mot. Il ne dit rien de plus sur les agrégats de comportements partagés construits. Il convient ensuite d'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. Par exemple, la similitude des syntagmes `pontage coronarien` et `pontage saphene` n'indique pas que la mention du vaisseau joue dans un cas le rôle de localisateur (pontage sur la coronaire) et dans l'autre celui d'un instrument (pontage à l'aide de la veine saphène). L'utilisateur peut ainsi être amené à intervenir manuellement sur les résultats pour affiner le travail de représentation et de classement. Les contraintes associées aux arbres à construire peuvent, par exemple, permettre d'affiner la représentation des informations associées aux mots. L'utilisateur peut, par exemple, contraindre l'arbre `NAdj` associé à `pontage` avant le processus de génération des objets avec des contraintes de type sur les nœuds de l'arbre, si ces informations sont disponibles. Le dispositif construit alors deux prototypes d'arbre élémentaire qui vont tenir compte de la différence de type possible sur l'adjectif dans l'arbre `NAdj`. L'utilisateur peut aussi en tenir compte plus tard, dès que ces informations sont disponibles, en créant un deuxième prototype `NAdj` par clonage et indiquer dans les deux prototypes `NAdj` le type sémantique adéquat sur l'adjectif.

8.1. *Une démarche interprétative contrôlée et en spirale*

Notre approche vise à établir une démarche interprétative contrôlée et progressive. La démarche suivie s'inscrit donc dans une perspective expérimentale à différents niveaux : (1) construire des représentations des mots à partir d´informations extraites 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 [MIK 94]. Le réseau mis en place ne construit donc pas une représentation du sens attaché aux mots ou aux

[15] [RES 93] présente une approche de construction de classes sémantiques de manière inductive dans une optique statistique.

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 [HAB 97b]. Les pôles obtenus et les classes de mots sous-jacentes sont des ébauches imparfaites qui permettent un organisation du matériel lexical. Ces classes doivent ensuite aider à affiner le travail de description des mots représentés sous la forme de prototypes.

8.2. *Le retour du linguiste*

Notre démarche cherche à réaliser une adéquation entre les occurrences linguistiques réalisées et les prédictions de représentations construites. Il ne s'agit pas de produire d'emblée un résultat définif qui réalise cette adéquation de manière parfaite ; mais plutôt de tendre vers cette adéquation, par touches successives, en affinant les prédictions construites.

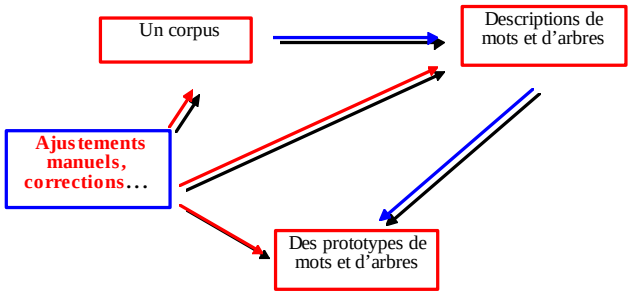

Figure 11. *Une démarche ascendante de réprésentation*

La mise en oeuvre des représentations est donc conçue comme un mécanisme évolutif qui, d'une part, doit tenir compte d'un nombre important de sources de connaissances, et d'autre part, doit être capable d'intégrer de nouvelles informations à chaque étape. GASPAR commence par construire des objets informatiques pour représenter les mots et les arbres associés à partir d'informations extraites sur corpus. GASPAR établit ensuite un classement des mots sur la base des contraintes syntaxiques associées. Ce réseau propose des amorces d'interprétation qui doivent être affinées par un travail d'interprétation plus fin. Ce travail d'interprétation est réalisé en utilisant l'interface graphique et interactive de Self. Cette interface permet de mener les aller-retour continuels entre observation d'occurrences en corpus et structures de représentation contruites pour mener à bien un travail expérimental de représentation.

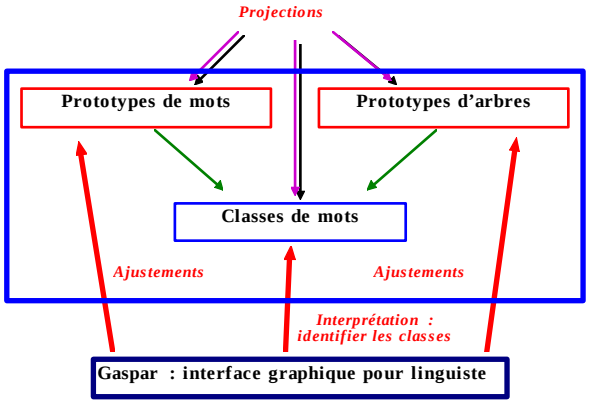

Figure 12. *Représentation contrôlée et en spirale*

Si GASPAR peut automatiser le classement des prototypes de mots 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. 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 nouvelles informations. 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 comme par exemple pour évaluer le rapprochement de `pontage saphene` et `pontage coronarien`. 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 [GRE 93-94]).

9. GASPAR : Ce qui est construit

Nous passons maintenant à une présentation des premiers résultats construit avec GASPAR. Celui-ci est constitué de différents modules [FLE 97] : un module de représentation à partir d'informations extraites sur corpus, un module de classement des prototypes construits, un module d'analyse automatique, un méta-niveau d'analyse. D'une part, GASPAR crée des représentations informatiques de manière automatique. D'autre part, il permet à l'utilisateur d'intervenir à tout moment sur les résultats construits ou sur les données traitées pour les ajuster ou pour les modifier.

Il en va ainsi pour le module de représentation automatique des mots. Ce module est constitué d'une chaîne de traitements capable de construire des objets à partir d'informations linguistiques extraites sur corpus. Il permet parallèlement à l'utilisateur d'ajuster les représentations construites si de nouvelles informations sont disponibles. La construction de GASPAR s´accompagne du développement d´un site WEB, qui lui est dédié, à l´adresse suivante *http://www.ens-fcl.fr/~fleury/*. Ce site est par ailleurs intégré au programmes informatiques mis en œuvre avec GASPAR. Il présente GASPAR et ses composants. Il permet aussi de rendre accessible les données traitées par GASPAR au gré de leurs évolutions. Il permet enfin une présentation des résultats construits par GASPAR. Ces résultats sont mis à jour automatiquement par les programmes développés dans ce dispositif. Les classes de mots construites sur les corpus traités constituent à ce jour les premiers résultats disponibles via cette interface.

9.1. *Un dispositif expérimental*

Les processus mis en place dans 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.

9.2. *Travail sur corpus de test*

Le mini-corpus de test utilisé pour mettre en œuvre les processus de représentation des mots contient une cinquantaine de mots et une centaine d'arbres. On présente ici quelques résultats obtenus avec ce corpus et via l'interface graphique de Self. Les figures qui suivent présentent des traces graphiques des prototypes générés lors du processus de génération et des pôles de comportements partagés construits. À chaque objet est associé un masque graphique constitué de formes graphiques Self appelées des `morphs`. Ces interfaces graphiques constituent des points d'entrée pour la présentation des informations représentées. Suivant le type d'objet associé à ces masques, on a défini des boutons qui permettent la présentation des informations associées à ces objets. Dans le cas des prototypes de mots par exemple, on dispose de boutons qui peuvent activer la présentation des descriptions morphologiques et sémantiques du mot, d'un bouton qui permet la présentation des prototypes d'arbres élémentaires associés au mot, et de l'affichage de l'objet lui-même dans l'interface graphique. Des outils graphiques du même type sont disponibles pour les prototypes d'arbres ou pour les objets `traits`. 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`.

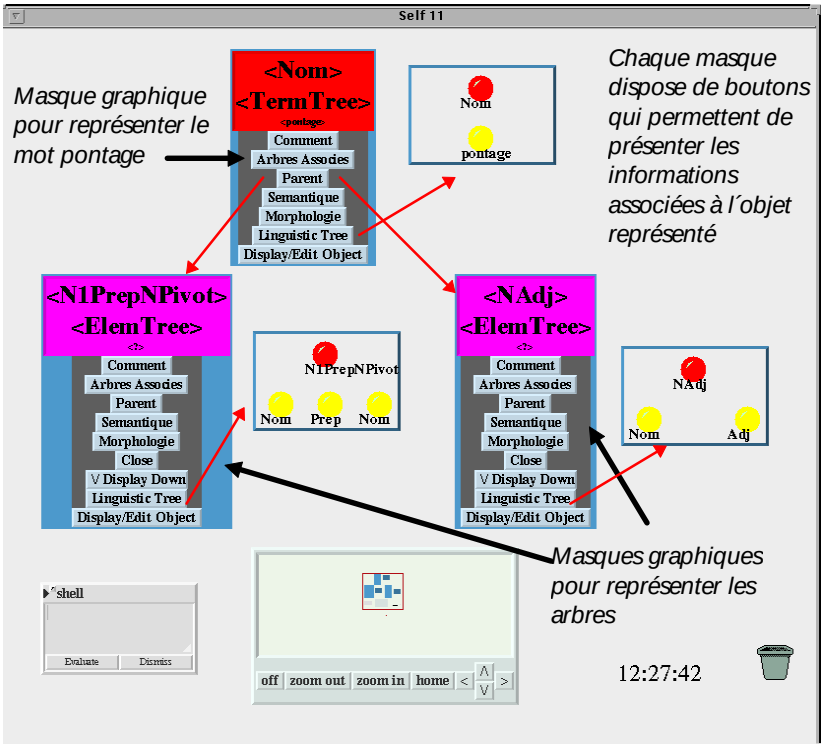

Figure 13. *Objets GASPAR pour le nom* `pontage`

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 `predominant`, `negligeable`, `important` et `significatif`. 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.

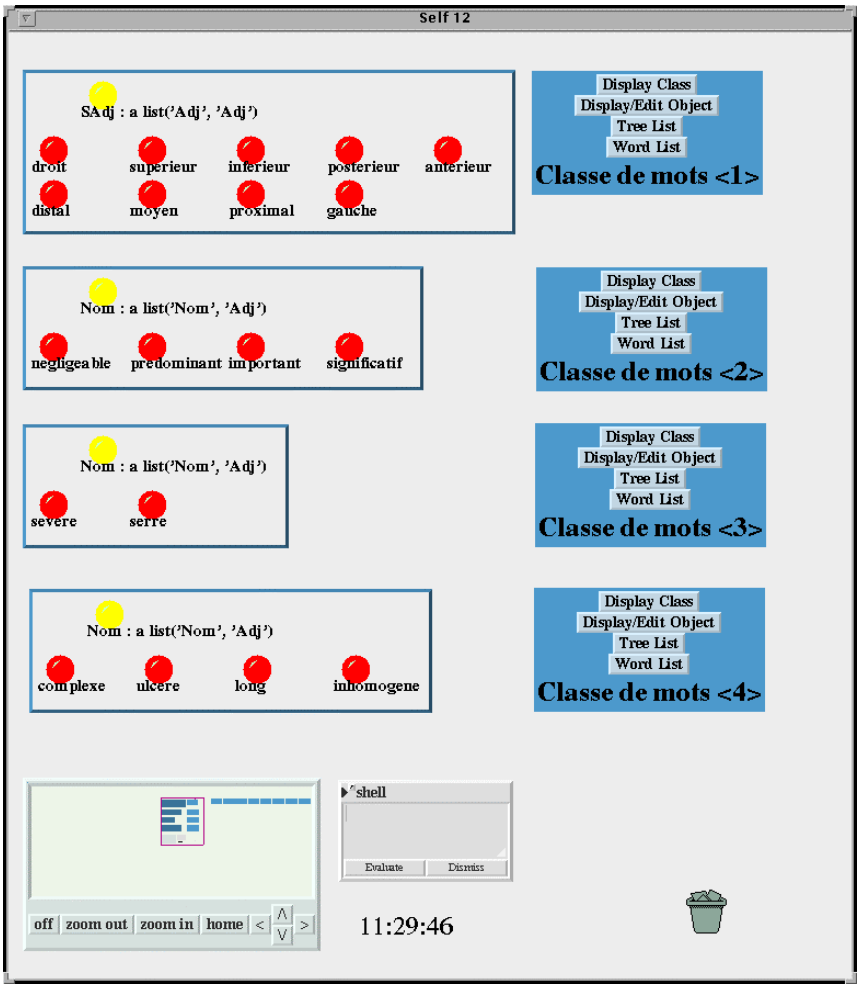

Figure 14. *Classement des prototypes sur corpus de test*

9.3. *Travail sur de « gros corpus »*

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 extrait 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`, Sn_1, Sn_2, Sn_3, Sn_4, des objets `traits` associés et de plus de 2 000 prototypes par copie et ajustement. Avec les prototypes créés, GASPAR a cherché à repérer ceux qui partageaient exactement les mêmes comportements. Ce processus de classement conduit à la création automatique de 55 pôles de comportements partagés.

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
(11) fait amaigrissement etude	recent
(12) sujet individu	asymptomatique
(13) calcification algie	diffus
(14) ouverture prolapsus	mitral
(15) capture fonction conduction	ventriculaire
(16) fistule frere circulation atherosclerose	coronarien
(18) fourche etage	carotidien
(19) foyer fuite	aortique, mitral
(20) pedieux dominance nodule	droit
(21) perforation rupture	septale
(22) frottement reaction epanchement	pericardique
(23) centre soir	meme
(24) gastrectomie alopecie	partiel
(25) cholesterolemie hysterectomie obliteration	total
(26) genou machoire membre siege	inferieur
(27) greffon heterogreffe monopontage	veineux
(29) hematome asthenie	important
(30) triglyceride anatomie ascension	normal
(31) praticien administration milieu	hospitalier
(32) horaire difficulte	particulier
(34) prevention epilepsie epreuve	secondaire
(35) usage decours defibrillation	immediat
(36) vasodilatateur arterite	peripherique
(37) prurit dyspnee	intense
(38) crosse sigmoide	aortique
(39) impasse implication escalade nouvelle	therapeutique

(40) radiographie radio	pulmonaire, thoracique
(41) vitesse marche	rapide
(42) impossibilite raison	technique
(43) acrocyanose nausee	transitoire
(44) reanimation muscle	cardiaque
(45) agregant anti-agregant	plaquettaire
(46) relation lien	etroit
(47) remontee dosage	enzymatique
(48) interpretation interrogatoire analyse	difficile
(49) lacune medecin	present
(51) entrainement sedation medication	adequat
(52) apparition augmentation	brutal
(53) gazometrie ponction hypertension coudure	arteriel
(54) fibrillation sonde pancreatite phase	aigu

pôles de mots partageant un arbre `sn -> adj nom`	**adj**
(17) majorite variabilite	grand
(28) ballon extension	petit
(33) intention symptome	premier
(50) accord repermeabilisation suivi	bon
(55) moyenne natif	circonflexe

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 homogène ; pour cette classe, on note que les régions qualifiées 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, par exemple, regroupent quant à elles des noms sémantiquement plus éloignés.

9.4. *Evaluer les descriptions linguistiques représentées*

La génération automatique des prototypes est largement conditionnée par la somme d'informations issue du travail d'extraction réalisé en amont. Les informations utilisées, à ce stade de notre travail, sont nettement insuffisantes pour produire des résultats significatifs et pour réaliser le nécessaire travail d´évaluation des résultats produits par GASPAR. 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. L'examen de la simple relation binaire `Nom Adj` ne permet pas de décrire complètement le fonctionnement des mots. Pour enrichir ce travail de description du comportement

des mots, il convient évidemment de pouvoir examiner d'autres types de relation binaire puis les relations syntaxiques complexes (lien arbre élémentaire-arbre d'analyse). Il convient 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.

10. Perspectives

Nous avons présenté une approche qui vise à établir une représentation inductive des mots et des comportements syntaxico-sémantiques associés suivie d'une classification progressive des objets construits. Concrètement ce travail a permis de produire un premier dispositif qui doit conduire à la construction d'une grammaire de sous-langage [HAR 70-88]. Il vise aussi à produire des classes de mots sur le domaine de connaissances étudié. Le choix des prototypes semble cohérent avec la volonté de représenter des connaissances évolutives. 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 un mot - 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 mots. 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»* [HAB 96a]. 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. Il reste aussi à mettre en œuvre une interface graphique qui permette à l'utilisateur d'évaluer et d'interpréter les résultats. Ce développement constitue une voie de travail de grande ampleur mais reste en marge des problèmes exposés ici. Self offre d'ailleurs des solutions intéressantes pour réaliser cette interface. Les aspects conceptuels n'ont pas été articulés aux phénomènes traités dans les processus construits ; une étude théorique et technique de ces articulations avec les autres niveaux de connaissances représentés doit conduire à étendre et affiner le

travail de représentation amorcé. Le traitement de connaissances extra-linguistiques peut aussi complèter et affiner les processus mis en place. Le développement d'outils pour la représentation et pour le classement dynamiques des savoirs linguistiques doit être étendu. Ce travail est contraint par une double nécessité. D'une part, il est nécessaire de mettre en place des outils de contrôle capables à tout moment d'évaluer les différents types d´informations manipulées et leurs interrelations existantes ou à venir. D'autre part, la réalisation d'un classement automatique général des mots semble hors d'atteinte tant qu'on ne pourra pas construire de nouvelles connaissances capables d'enrichir les connaissances déjà établies : en sachant de plus que ce classement à atteindre ne sera jamais un résultat définitif. Il reste malgré tout possible d'affiner les processus de classement mis en place et de fait d'affiner le classement visé (en connaissant ses limites) tout en continuant à travailler sur les métaconnaissances à l'œuvre dans les faits de langue utilisés.

11. Bibliographie

Abréviations employées :
AAAI : American Association for Artificial Intelligence
ACL : Association for Computational Linguistics
EACL : European Conference of the Association for Computational Linguistics
OOPSLA : Object-Oriented Programming : Systems, Languages and Applications
TOOLS : Technology of Object-Oriented Programming and Systems
TAL : Traitement Automatique des Langues, revue de l'ATALA, Association pour le Traitement Automatique des LAngues.

[BER 97] BERRY MICHAEL J.A., LINOFF GORDON, *DATA MINING* « Techniques appliquées au marketing, à la vente et aux services clients », InterEditions, 1997.

[BIB 93] BIBER DOUGLAS, « Using register-diversified corpora for general language studies », *Computational Linguistics*, volume 19, numéro 2, p. 219-241, 1993.

[BLA 94] BLASCHECK G., *Object-Oriented Programming with Prototypes*, Springer-Verlag, Berlin, 1994.

[BOU 93] BOURIGAULT DIDIER, « An endogeneous Corpus-Based Method for Structural Noun Phrase Disambiguation », Actes *EACL'93*, 1993.

[FLE 97] FLEURY SERGE, « Mise en œuvre de représentations évolutives des connaissances pour le traitement automatique du langage naturel », Thèse de doctorat, Paris 7-Denis Diderot.

[GRE 93] GREFENSTETTE GREGORY, « 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, Septembre 1993, Oxford.

[GRE 94] GREFENSTETTE GREGORY, « Corpus-Derived First, Second and Third-Order Word Affinities », Actes *EURALEX'94*, Amsterdam 1994.

[HAB 96a] HABERT BENOIT, NAZARENKO ADELINE, « La syntaxe comme marche-pied de l'acquisition de connaissances : bilan critique d'une expérience », *Journées d'Acquisition de Connaissances*, 1996.

[HAB 96b] HABERT BENOIT, FOLCH HELKA, « Les quasi-arbres : un formalisme logique pour exprimer des requêtes en indexation structurée », Actes *Informatique et Langue Naturelle*, Nantes, 1996.

[HAB 97a] HABERT BENOIT, BOURIGAULT DIDIER, « A Frontier to Root Transducer for the Evaluation of Two Noun Phrase Extractors », *Journées Scientifiques et Techniques*, 1997.

[HAB 97b] HABERT BENOIT, SALEM ANDRE, NAZARENKO ADELINE, *Les linguistiques de corpus*, Armand Colin, Paris, 1998.

[HAR 70] HARRIS ZELLIG, « La structure distributionnelle », *Langages n°20*, Paris, Larousse, 1970.

[HAR 88] HARRIS ZELLIG, *Language and Information*, Columbia University Press, New York, 1988.

[HAT 91] HATON JEAN PAUL, NADJET BOUZID, CHARPILLET FRANÇOIS, HATON MARIE CHRISTINE, LAASRI BRIGITTE, LAASRI HASSAN , MAQUIS PIERRE, MONDOT THIERRY, NAPOLI AMADEO, *Le Raisonnement en Intelligence Artificielle : Modèles, Techniques et Architectures pour les sytèmes à base de connaissances*, InterEditions, 1991.

[LIE 86] LIEBERMANN HENRI, « Using Prototypical Objects to implement Shared Behavior in Object Oriented System », Artificial Intelligence Laboratory, MIT, 1986.

[MAL 95] MALONEY JOHN, « Morphic : the Self User Interface Framework », Sun Microsystems, Inc. and Stanford University, 1995.

[MIK 94] MIKHEEV ANDREI, MŒNS MARC, « Acquiring and Representing Background Knowledge for a natural Language Processing System », Actes *AAAI'94*, 1994.

[MOO 95] IVAN MOORE, « GURU - A Tool for Automatic Restructuring of Self Inheritance », *Actes TOOLS USA'95*, Prentice-Hall, 1995.

[MOO 96] IVAN MOORE, TIM CLEMENT, « A Simple and Efficient Algorithm for Inferring Inheritance Hierarchies », Actes *TOOLS Europe'96* , Prentice-Hall, 1996.

[RES 93] RESNIK PHILIP STUART, « Selection and Information : a Class-Based Approach to Lexical relationships », Ph.D Dissertation, University of Pennsylvalia IRCS Report 93-42, 1993.

[SEL 95] SELF GROUP 95 : AGESEN O., BAK L., CHAMBERS C., CHANG B.W., HÖLZLE U., MALONEY J., SMITH B.R., UNGAR D., WOLCZKO M., « The Self 4.0 Programmer's Reference Manual », Sun Microsystems, Inc. and Stanford University, 1995.

[SMI 95] SMITH WALTER R., « Using a Prototype-based Language for User Interface : The Newton Project's Experience », Actes *OOPSLA'95*, p. 61-72, SIGPLAN Notices, 1995.

[UNG 87] UNGAR DAVID, RANDALL B. SMITH, « SELF : The power of Simplicity », Actes *OOPSLA'87*, SIGPLAN Notices, 1987.

[UNG 95] UNGAR DAVID, « Programing as an experience : The Inspiration for SELF », Actes *ECOOP'95*, Aahrus, Denmark, 1995.

[WOL 95] WOLCZKO MARIO, RANDALL B. SMITH, Prototype-Based Application Construction Using Self 4.0, Sun Microsystems Inc, 1995.

[ZWE 94] ZWEIGENBAUM PIERRE, « MENELAS : an Access System for Medical Records using Natural Language », *Computer Methods and Programs in Biomedicine*, volume 45, p. 117-120, 1994.

[ZWE 97] ZWEIGENBAUM PIERRE, BOUAUD JACQUES, HABERT BENOIT, NAZARENKO ADELINE, « Coopération apprentissage en corpus et connaissances du domaine pour la construction d'ontologies », *Journées Scientifiques et Techniques*, FRANCIL, Avignon, 1997.

Annex e I

Evolutive representation of words

Serge Fleury

UMR 9952)
Ecole Normale Supérieure de Fontenay-St Cloud
31 avenue Lombart
F-92260 Fontenay aux Roses
fleury@ens-fcl.fr

Annexe II

Logiciel utilisé : Word 7 sous Windows 95. Fichier source SFRIA.DOC, les figures 13 et 14 sont fournies en images attachées (fichiers FIG13.GIF et FIG14 .GIF).