Une méthode d'indexation fondée sur l'analyse sémantique de documents spécialisés : le prototype RIME et son application à un corpus médical

Catherine Berrut

THESE

présentée par

Catherine BERRUT

pour obtenir le titre de **docteur**
de l'**Université Joseph Fourier - Grenoble I**
(arrêté ministériel du 5 juillet 1984)

spécialité *Informatique*

Une méthode d'indexation fondée sur l'analyse sémantique de documents spécialisés.

Le prototype RIME et son application à un corpus médical.

date de soutenance : le 13 décembre 1988

composition du jury :

président :	M. Michel ADIBA
rapporteurs :	M. Patrick BOSC
	M. Patrice POGNAN
examinateurs :	M. Yves CHIARAMELLA
	M. Philippe CINQUIN

Thèse préparée au sein du Laboratoire de Génie Informatique - IMAG -
à l'Université Joseph Fourier - Grenoble I -

Je tiens à remercier,

Monsieur Michel Adiba, Professeur à l'Université Joseph Fourier, pour avoir accepté de présider ce jury, pour l'intérêt qu'il a porté à ce travail, pour la gentillesse et le soutien moral qu'il m'a toujours témoignés pendant ces années ;

Monsieur Patrick Bosc, Professeur à l'Ecole Nationale Supérieure des Sciences Appliquées et de Technologies (ENSSAT) de Lannion, qui a bien voulu rapporter ce travail, et qui, grâce à questions, ses remarques, m'a aidée à faire progresser ce document ;

Monsieur Patrice Pognan, Professeur des Universités, affecté à l'Institut National des Langues et Civilisations Orientales (INALCO), responsable du Centre d'Etudes et de Recherches sur le Traitement Automatique des Langues (CERTAL), qui a accepté de rapporter ce travail, et qui par ses remarques a contribué à la qualité de ce document ;

Monsieur Philippe Cinquin, assistant à l'Université Joseph Fourier, dont l'aide, la gentillesse et la disponibilité permanentes ont été fondamentales dans ce travail ;

Monsieur Yves Chiaramella, Professeur à l'Université Joseph Fourier, Directeur du Laboratoire de Génie Informatique et responsable de l'équipe Systèmes Intelligents de Recherche d'Informations, qui m'a accueillie dans son équipe, et qui a dirigé mes travaux de recherches. Il a, par ses critiques constructives, fait avancer ce travail, et par ses lectures améliorer ce document. Il m'a donné au sein de son équipe une liberté d'action qui m'a permis d'apprécier le travail de recherche dans toutes ses dimensions. Il a su pendant toutes ces années par ses encouragements m'aider à surmonter tous les problèmes rencontrés. Et surtout il a toujours su avec calme et compréhension supporter ma nervosité dans les moments les plus difficiles ;

L'Université Joseph Fourier, l'UFR Informatique et Mathématiques Appliquées et son directeur Monsieur Pierre-Claude Scholl, pour m'avoir accueillie dans son équipe d'enseignants. C'est avec beaucoup de plaisir et énormément d'intérêt que j'ai pendant ces années découvert et exercé le métier d'enseignante ;

Monsieur Jacques Demongeot, Professeur à l'Université Joseph Fourier et responsable du service de Biostatistiques et d'Informatique Médicale du Centre Hospitalier de Grenoble, pour l'intérêt qu'il a toujours porté à ce travail, et la valorisation qu'il lui a toujours donnée dans le monde de la recherche en informatique médicale ;

Monsieur Pierre Gressier, étudiant à l'ENSIMAG, qui a développé certaines parties du prototype de RIME ;

Monsieur Patrick Palmer et Madame Catherine Péquegnat, membres de l'équipe Systèmes Intelligents de Recherche d'Informations, pour avoir par leurs compétences largement contribué à ce travail ;

Madame Marie-France Bruandet et Monsieur Bruno Defude pour leurs encouragements, leur amicale complicité, et les améliorations qu'ils ont apportées à ce manuscrit ;

Les membres de l'équipe Systèmes Intelligents de Recherche d'Informations pour avoir su me donner l'énergie nécessaire pour terminer ce travail ;

Tous les collègues (et néammoins amis) du Laboratoire Marie-Christine, Christine, Philippe, Michel, Jean-Pierre, tous les amis Myriam, Arlette, Françoise, Chantal, Marie-Laure, Alain, Christian, tous les Jean-François qui m'ont toujours aidée, encouragée et soutenue dans ce travail ;

Jean-Lucien, sans qui je n'aurais sans doute jamais tenté une telle expérience ;

Et surtout, mon père et toute ma famille, qui malgré toutes les épreuves difficiles que nous avons subies pendant toutes mes années d'études, m'ont toujours aidée et entourée de la plus grande affection.

Sommaire

Sommaire des figures

Chapitre 1
Introduction

1.1. présentation des systèmes de recherche d'informations

1.1.1. généralités

La définition et l'implantation d'une fonction de correspondance entre une requête d'un utilisateur et un corpus de documents constitue un sujet de recherche connu depuis plusieurs années sous le nom de *recherche d'informations* ou *recherche documentaire* en français, *information retrieval* ou *text retrieval* en anglais. Les fonctionnalités essentielles des systèmes de recherche d'informations sont la gestion et l'accès à des bases documentaires, de manière à permettre à tout utilisateur d'obtenir des réponses à une question posée [RIJS79] [SALT80].

Les systèmes présentent donc un ensemble de fonctions de manipulation de documents permettant tout d'abord de les représenter, de les stocker et de les organiser : ils offrent ensuite des outils permettant de déduire du corpus traité l'ensemble des documents répondant à une requête posée par un utilisateur. Une requête peut porter sur les attributs externes des documents, comme leur titre, leur auteur, ou bien sur le contenu même des documents. Généralement la réponse à une requête se compose quant à elle d'un ensemble de références vers les documents sélectionnés [GOLL72], ou bien du contenu même des

documents, voire de certaines parties plus particulièrement pertinentes de leur contenu [SALT71].

1.1.1.1. correspondance requête-document

La fonction de correspondance requête/documents implique l'utilisation possible des attributs externes et des attributs de contenu, séparément ou simultanément. En ce qui concerne les attributs externes, la recherche des documents pertinents est déterministe, et la correspondance requête/document exacte. En fait, il s'agit de requêtes tout à fait analogues à celles traitées dans les systèmes de gestion de bases de données classiques (SGBD). En ce qui concerne les attributs de contenu par contre, la correspondance ne peut être exacte dans le cas général. Il faut en effet, à ce niveau, établir une comparaison sémantique (et non une égalité) entre des concepts figurant dans un document et d'autres figurant dans une requête. Une telle comparaison aboutit rarement à des équivalences strictes, du fait des nuances de langage et d'expression des connaissances. Il en résulte que les systèmes de recherche d'informations ne peuvent être, à l'image des SGBD, fondés sur une stratégie de correspondance exacte requête/base de données ; une telle fonction retournerait le plus souvent des réponses vides (par exemple, lorsqu'on recherche les documents parlant du *premier ministre* dans un corpus où n'apparaît que *chef du gouvernement*) ou de très mauvaise qualité (par exemple, lorsqu'on retourne en réponse à une requête portant sur la *marine* tous les documents traitant de la couleur *bleu marine*).

1.1.1.2. évaluation d'un système de recherche d'informations

Une autre conséquence du type de comparaison utilisé dans ces systèmes est que leurs performances qualitatives s'évaluent principalement par rapport à l'aptitude que présente effectivement la fonction de correspondance à réaliser cette comparaison sémantique requête/document. Une réponse (un document) est évaluée à travers sa pertinence, qui est censée mesurer la *distance sémantique* entre le document et la requête. Cette notion de pertinence est

éminemment subjective, et ne peut être qu'estimée à travers des fonctions de calcul fondée sur le modèle de correspondance utilisé. Un premier critère pour juger de la qualité d'un système de recherche d'informations, consiste donc en sa capacité de fournir des réponses ayant une *bonne* mesure de pertinence. Les critères classiques, qui permettent de mesurer le résultat global de la requête (ensemble de réponses obtenues) et de les comparer, font donc intervenir cette notion de *pertinence* (adéquation d'un document à la requête) au travers de deux paramètres qui sont le *rappel* et la *précision* :

- les mesures de *rappel* donnent la proportion d'informations pertinentes retrouvés par rapport au nombre total de réponses pertinentes ;
- la *précision* mesure la proportion d'informations pertinentes retrouvées par rapport au nombre total de réponses données.

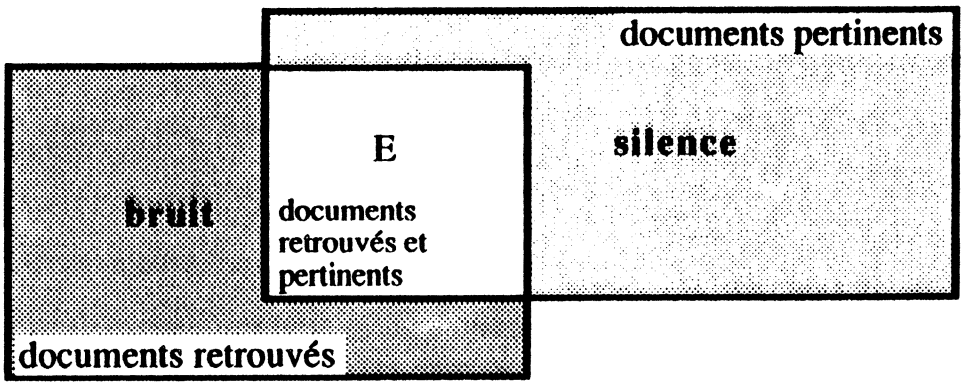
On peut également mesurer :

- le *silence* le complémentaire du rappel, c'est-à-dire le nombre de documents pertinents non retrouvés par rapport à tous les documents pertinents ;
- le *bruit* le complémentaire de la précision, c'est-à-dire la proportion de documents non pertinents donnés en réponse par rapport au total de documents donnés en réponse.

La figure 1 représente toutes ces notions.

Il est donc évident que, lors d'une recherche, on va chercher à diminuer silence et bruit. Ce double objectif est contradictoire, dans la mesure où diminuer le silence consiste à élargir la requête initiale (par rapport à ses concepts initiaux), de façon à atteindre tous les documents pertinents. Ce qui ne peut se faire sans risque d'introduire du bruit lors d'un élargissement excessif. Inversement, diminuer le bruit consiste à restreindre la requête initiale (toujours par rapport à ses concepts initiaux), ce qui peut augmenter le silence. Ainsi ce problème est difficile à résoudre , et doit être mener avec vigilance lors de la phase d'interrogation.

figure 1
les mesures des systèmes de recherche d'informations



précision = E / documents retrouvés

rappel = E / documents pertinents

1.1.2. indexation

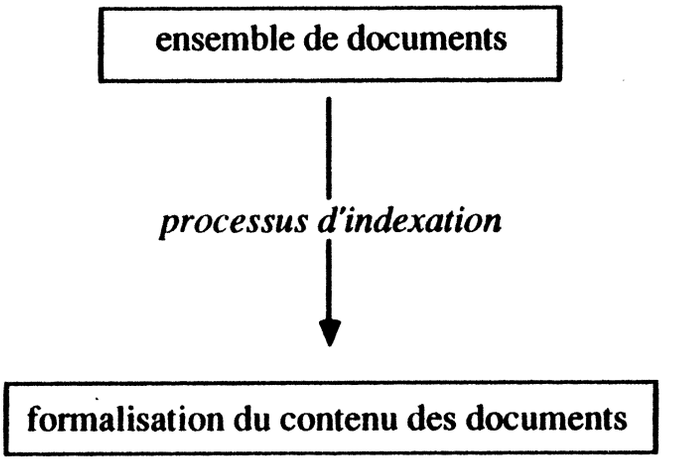
Pour fournir la réponse à une requête, un système de recherche d'informations pourrait procéder en deux étapes :

- prendre connaissance du contenu de chacun des documents stockés ;
- examiner si le contenu de chaque document correspond à la requête.

Il est évident que, non seulement pour atteindre ce double objectif, mais également pour obtenir de bonnes évaluations de la fonction de correspondance, il est impossible de traiter les requêtes de manière exhaustive sur tous les documents. Il paraît par conséquent nécessaire d'introduire dans les systèmes de recherche d'informations une étape préalable, appelée *indexation*, permettant l'analyse des documents de manière à les représenter sous une forme permettant cette évaluation performante [SPAR72].
Un tel processus sous-entend une définition a priori du contenu sémantique des documents : l'indexation consiste en fait à choisir un modèle conceptuel des documents, appelé *langage d'indexation*, contenant les éléments jugés a priori les plus porteurs de l'information véhiculée par les documents [KERK84]. Selon ce langage, l'indexation normalise et sélectionne les concepts retenus

dans chaque document : la forme indexée est une perception plus ou moins arbitraire des documents, dont le contenu sémantique apparaît plus restreint que celui des documents.
La figure 2 nous montre le processus d'indexation d'un corpus.

figure 2
l'indexation d'un corpus



1.1.3. interrogation

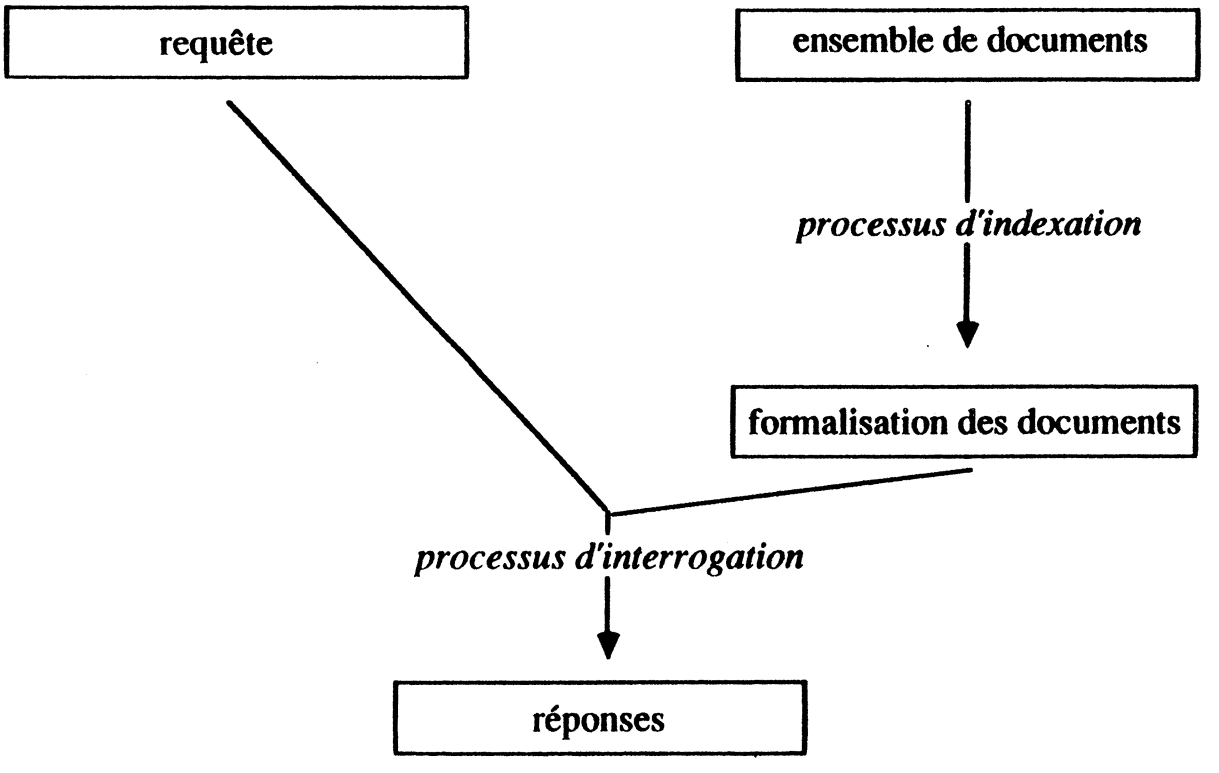
Le traitement des requêtes d'interrogation correspond à la finalité essentielle de tout système de recherche d'informations, et comprend pour cela toutes les possibilités que le système offre de manière à satisfaire l'utilisateur face à ses besoins d'informations. Globalement, le processus d'interprétation d'une requête établit une correspondance, via une *fonction de correspondance*, entre la forme indexée des documents et le contenu sémantique d'une requête. Cette correspondance réussit pour un document lorsque l'expression de la requête est incluse dans la forme indexée du document, ou plus généralement si elle peut en être déduite [DEFU86] [NIE88].

Classiquement, les systèmes de recherche d'informations utilisent une interface entre indexation et interrogation, appelée thésaurus, qui contient une représentation normalisée des concepts du corpus traité. Chaque concept peut ainsi être désigné, à l'indexation tout comme à l'interrogation, par l'ensemble

des termes du vocabulaire qui le décrivent. Un thésaurus peut être vu comme un ensemble de termes et un ensemble de relations sémantiques entre ces termes [RIJS79] [KERK84]. Ces relations correspondent généralement à une interprétation sémantique telle que la synonymie (pure ou partielle), la hiérarchie (généricité, spécificité), la causalité, ou bien dans certaines applications particulières à une interprétation contextuelle au corpus traité [BRUA89].

La figure 3 constitue une présentation schématique d'un système de recherche d'information

figure 3
schéma général d'un système de recherche d'informations



1.2. la problématique de l'indexation

1.2.1. présentation

L'indexation constitue la fonction d'un système de recherche d'informations, traduisant de manière homogène, i.e. selon un langage d'indexation prédéfini, le contenu sémantique des documents.

En fait, si l'on veut détailler un peu plus précisément le processus d'indexation, il faut au préalable présenter les documents indexés comme formés de deux parties bien distinctes :

- tout d'abord une partie objective, contenant les attributs externes des documents ; par exemple lorsque le document est un livre, il s'agit du nom de l'auteur, de l'éditeur, du numéro ISBN, etc ;

- ensuite une partie décrivant, selon un langage d'indexation fixé de manière relativement arbitraire, le contenu du document. C'est l'obtention de ces informations que nous allons étudier plus particulièrement ici.

1.2.2. les méthodes d'indexation

La représentation d'un document selon un langage d'indexation est un travail délicat qui doit être abordé dans le but de satisfaire trois objectifs principaux [KERK84] :

- reconnaître tous les concepts des documents : c'est la phase d'*analyse* ;

- choisir le vocabulaire qui traduise le mieux ces concepts : c'est la phase de *sélection* ;

- établir une relation d'indexation entre cet ensemble de concepts et les documents : c'est la phase d'*enregistrement* et de *pondération.*

Différentes méthodes existent pour atteindre ces trois objectifs, chacune dépendant généralement d'un environnement particulier d'indexation. On peut cependant montrer de façon très générale quelques notions communes à toutes, et qui, dans la manière dont elles sont appréhendées, permettent de procéder à des distinctions fondamentales entre les différentes méthodes d'indexation [SALT & MCGI83].

1.2.2.1. indexation manuelle et indexation automatique

La première notion concerne la manière dont est effectuée l'indexation. Il faut tout d'abord faire la distinction entre les systèmes à indexation *automatique* et les systèmes à indexation *manuelle*. Historiquement les premières opérations d'indexation de documents ont été faites manuellement par des "experts"

chargés de dégager manuellement de chaque document le contenu sémantique qui sera considéré comme la forme indexée du document traité. Cette tâche est de nos jours facilitée dans les gros systèmes de recherche d'informations par des aides automatiques à l'indexation [THUR86]. Par ailleurs les systèmes actuels tendent à définir des fonctions d'indexation complètement automatiques basées sur différents modèles de correspondance (vectoriel, probabiliste, ...) que nous reprendrons par la suite.

1.2.2.2. langages contrôlés et langages non contrôlés

La seconde notion concerne les classes, donc les limites, du langage d'indexation. Il existe à ce niveau deux grandes classes de langages d'indexation : les *langages d'indexation contrôlés* et les *langages d'indexation non contrôlés*. Les premiers sont des langages figés a priori pour un corpus donné de documents, les seconds sont des langages "libres" qui peuvent admettre l'introduction de nouveaux concepts pour un document donné.
Choisir entre ces deux types de langages n'est pas une tâche évidente. En effet, les langages d'indexation non contrôlés incluent a priori les mêmes problèmes qu'un langage naturel en introduisant éventuellement des possibilités d'erreurs ou d'ambiguïtés, mais permettent par contre leur propre enrichissement lorsque de nouveaux concepts sont effectivement rencontrés dans un document. Les langages d'indexation contrôlés éliminent toute possibilité de mots, de concepts nouveaux en ne permettant que les mots jugés a priori représentatifs de concepts, et éliminent par ce biais les introductions d'erreurs typographiques, ou de synonymes. Le problème essentiel pour arriver à des résultats réellement efficaces dans ce contexte contrôlé est en fait de passer par un intermédiaire (par exemple un expert du langage d'indexation) pour traduire toute requête de manière pertinente par rapport à ce langage.

1.2.2.3. langages simples et langages complexes

La troisième notion fondamentale concerne la nature des entités ou des termes qui définissent le langage d'indexation et la façon dont ils sont extraits des

documents de manière à en caractériser le contenu. Là encore on distingue deux grandes classes de langages, la première correspond aux langages manipulant des termes simples, et la seconde aux langages manipulant des termes complexes :

- dans les systèmes à termes simples, les entités, connues généralement sous l'appellation de termes d'indexation, de mots clés ou de descripteurs, sont représentées par des mots isolés. Le contenu de chaque document est par conséquent représenté par un ensemble de mots isolés, qui peuvent éventuellement être combinés ou *coordonnés* ultérieurement lors d'une phase d'interrogation. Un tel processus se nomme *postcoordination*. Par exemple, un document traitant de la production de charbon est, dans un tel processus, indexé par les termes d'indexation *production* et *charbon*, et sera retrouvé à l'interrogation par la conjonction *production ET charbon* ;

- lorsque des termes complexes sont utilisés comme termes d'indexation (sous la forme par exemple de syntagmes), avec des indicateurs de relations à l'intérieur des syntagmes, un tel processus est dit *précoordination*. Pour reprendre l'exemple donné ci-dessus, le document est, dans un cas de précoordination, indexé par le terme complexe *production de charbon.*

Dans les systèmes de recherche d'informations où l'on dispose d'experts à l'indexation, l'utilisation de langages d'indexation contrôlés utilisant des termes d'indexation complexes précoordonnés est très fréquente. Par contre les systèmes d'indexation automatique utilisent plus fréquemment des termes simples, car l'extraction automatique de termes simples pertinents est une technique relativement bien dominée actuellement. Dans de tels systèmes les termes sont généralement combinés par postcoordination lors de la phase d'interrogation.

1.2.3. conclusion

Il est assez normal qu'un système de recherche d'informations cherche à être le plus précis possible tant lors de l'indexation que de l'interrogation : les systèmes respectant ce type de critère sont des systèmes à indexation dite

profonde. Ceci implique le respect des deux caractéristiques que sont l'exhaustivité de l'indexation et la spécificité/généricité des termes d'indexation :

- plus un système est exhaustif, plus il assure que toutes les notions, tous les concepts d'un document sont retenus lors de son indexation ;

- la spécificité/généricité permet de mesurer le degré de généralité des termes d'indexation retenus pour chaque document. Plus un terme d'indexation est spécifique, plus son sens est précis et restreint. Inversement, plus il est générique, plus son sens est large et peut être décrit par des termes plus spécifiques.

Ces propriétés ne sont pas systématiquement les plus recherchées dans les systèmes de recherche d'informations : certains systèmes à indexation dite *de surface* ou *superficielle* permettent l'extraction de termes d'indexation très larges de manière à réaliser un gain de temps et de coût.

1.3. objets et approches de l'étude

Notre travail est relatif à la mise en œuvre de la fonction d'indexation du système de recherche d'informations RIME. Notre approche consiste ici à définir une méthode d'indexation automatique qui, au travers de la représentation "formatée" des documents (représentation décrite en détail au chapitre 3), permette une compréhension profonde. Une telle stratégie ne s'avère viable que si le corpus traité est relatif à un **domaine sémantique bien délimité** : l'application choisie pour illustrer cette méthode et valider notre approche, est **l'indexation d'un corpus médical constitué de comptes rendus radiologiques**. Cette application constitue une suite à l'étude effectuée par [MUNO87], relative à la gestion de ces comptes rendus et des documents iconographiques associés dans le contexte d'un SGBD multimédia. En effet, notre travail consiste à permettre, en plus de l'accès classique, l'accès aux images radiologiques par leur contenu sémantique, via les documents textuels (les comptes rendus) qui les décrivent.

Ce travail permet ainsi de traiter des requêtes très précises telles que, par exemple, *Donnez-moi toutes les radios sur lesquelles apparaissent des opacités*

pulmonaires pouvant être considérées comme cancéreuses. La définition de la fonction d'interrogation de RIME fait par ailleurs l'objet du travail de J. NIE.

Le corpus de RIME est constitué de comptes rendus médicaux qui sont des documents textuels très courts, généralement une page, voire exceptionnellement deux, rédigés en langue naturelle, et dont nous montrons en annexe 1 un exemple. A titre de remarque, notons que nous utilisons dans tout de document les termes *langage naturel* et *langue naturelle*, généralement utilisés en informatique comme traduction du terme anglais *natural language*. En fait, la traduction exacte de *natural language* est *langue*, comparativement à *language* dont la traduction est *langage*.

L'étude des propriétés des comptes rendus médicaux nous a amenés à définir une fonction d'indexation qui en dégage une compréhension profonde. En effet ces documents sont très courts et surtout très denses, et constituent en quelque sorte un résumé décrivant les images radiologiques. Toutes les informations qu'ils véhiculent sont fondamentales et ne peuvent être partiellement ignorées lors de l'indexation, sans prendre le risque de perdre des informations médicales importantes. Pour ces raisons, il est donc nécessaire de définir une stratégie d'indexation exhaustive basée sur une extraction de termes très spécifiques et significatifs pour le domaine médical, que seuls des traitements morphologique, syntaxique et sémantique de la langue naturelle nous permettent d'approcher.

Pour situer notre approche par rapport aux propriétés générales des fonctions d'indexation, nous proposons une **indexation automatique** précoordonnée basée sur un langage d'indexation contrôlé et complexe. Nous insistons sur le fait que, par rapport à d'autres systèmes de recherche d'informations, il n'y a dans notre approche aucune sélection de concepts lors de l'indexation, c'est-à-dire que les concepts présents dans le document initial sont représentés dans le document indexé. L'indexation proposée ici consiste à obtenir une représentation canonique de tous les concepts des documents, quelle qu'en soit

leur formulation en langue naturelle. Ceci est réalisé dans notre application au travers de différents outils de traitement de la langue naturelle.

1.3.1. le modèle sémantique de RIME

Le langage d'indexation de RIME, que nous appelons *modèle sémantique*, est un langage complexe et contrôlé, permettant de relier des termes par différents types de liens sémantiques. Ce modèle sémantique, que nous décrivons en détail au chapitre 3, est un modèle de représentation fin et riche, qui se rapproche des représentations en Dépendances Conceptuelles de Schank [SCHA72]. Il contient les termes utilisés dans notre corpus médical, des relations sémantiques contextuelles entre ces termes, et intéressantes dans le domaine traité.
Par exemple, nous avons des relations sémantiques de type *porte sur* (représenté par *p_sur*), ou bien *a pour valeur locative* (représenté par *a_pr_val_loc*), ... qui permettent de relier des catégories sémantiques précises de termes. En effet, il est indispensable de savoir qu'une lésion observée "porte sur" un organe particulier (le poumon par exemple), et d'en préciser davantage la localisation (le poumon droit par exemple).

Face aux différentes techniques de correspondance traditionnelles (booléenne, probabiliste ou vectorielle), seules les techniques linguistiques nous permettent, à partir de textes en langue naturelle, une **génération automatique** de leur traduction dans le modèle sémantique de RIME. En effet, les méthodes dites *classiques* de correspondance ne permettent pas de dépendances complexes entre termes : les modèles probabiliste et vectoriel ignorent complètement cet aspect du problème, et le modèle booléen ne permet que deux types de relations entre termes : le *et* et le *ou*.

Pour atteindre notre objectif d'indexation exhaustive, nous n'avons pu disposer de système de construction automatique de ce modèle, car bien que de tels outils soient en très grande évolution ces dernières années, ils ne permettent pas, pour le moment, de générer automatiquement le contenu

précis d'un corpus donné. Aussi, le modèle sémantique a été construit manuellement avec l'aide de médecins, et offre ainsi la représentation fidèle, dont nous avons besoin, du contenu sémantique du domaine traité. Cependant, nous offrons des outils permettant de vérifier la validité du modèle, d'effectuer sa mise à jour, et éventuellement faire des propositions pour représenter un nouveau concept n'existant pas dans le modèle sémantique (cf parties 4.4.3 et .5.4.2).

1.3.2. les outils de l'indexation de RIME

La fonction d'indexation de RIME consiste en un travail de traduction de textes en langue naturelle dans le langage cible représenté par le modèle sémantique de RIME [BERR & CINQ88a], qui nous donne la représentation interne du contenu des documents.

Il s'agit pour cela de mettre en œuvre des outils de traduction semblables, dans les principes seulement, à ceux développés dans les systèmes de traduction automatique ou de compréhension de langue naturelle. Cependant les outils de la fonction d'indexation de RIME sont moins complexes que les outils nécessaires dans de tels systèmes dans la mesure où :

- d'une part, la langue naturelle, utilisée dans les documents que nous traitons est un sous-ensemble du français. En ce sens, elle offre moins de possibilités d'ambiguïtés tant syntaxiques que sémantiques ou pragmatiques ;
- d'autre part, la langue cible (le langage d'indexation) est une langue artificielle, et par là même offre moins de complexité lors de sa génération qu'une langue naturelle.

Les outils nécessaires à cette indexation sont donc les outils classiques des traitements de la langue naturelle :

- un outil d'analyse morphologique, permettant d'une part l'extraction des mots des textes traités, et d'autre part la déduction des connaissances

morphologiques, syntaxiques et sémantiques nécessaires pour le traitement de chacun des mots ;

- un outil d'analyse syntaxique dont le rôle consiste à aider l'analyseur sémantique, au travers de deux tâches essentielles : simplifier les possibilités initiales issues du travail morphologique, et proposer une première description des textes traités ;

- finalement un outil d'analyse sémantique capable de générer du langage cible de RIME dans le but de traduire les comptes rendus médicaux.

1.3.3. conclusion

RIME est un système de recherche d'informations à indexation automatique, et à modèle sémantique complexe et contrôlé, dont le but est d'obtenir **automatiquement** une représentation relativement **fine** du contenu des documents couvrant un domaine sémantique limité.
Par ailleurs, l'indexation de RIME a nécessité une analyse détaillée des phénomènes linguistiques qui interviennent lors de la génération du modèle sémantique : cette analyse a permis de fixer les processus linguistiques nécessaires à ce travail, ainsi que le rôle joué par chacun d'eux.
Notre hypothèse de travail a été de considérer les trois processus de base d'un traitement linguistique, à savoir la morphologie, la syntaxe et la sémantique, comme des processus totalement indépendants. Dans cette optique, nous avons défini précisément quels étaient leurs rôles respectifs, la répartition des tâches linguistiques entre eux, et le niveau de finesse auquel chacun d'entre eux doit arriver pour satisfaire notre objectif final.
La notion d'autonomie de ces trois processus nous a amenés à définir un processus de coopération permettant l'enchaînement et les échanges entre ces processus. Le but de RIME est de montrer la validité de notre hypothèse de travail, et également de mettre en évidence les améliorations apportées par une approche sémantique de l'indexation dans un contexte d'univers fermé [BERR & CINQ88b].

Nous présentons dans le chapitre 2 différentes approches de traitement linguistique en recherche d'informations.

Les trois données importantes de RIME sont décrites dans le chapitre 3 :

- le modèle sémantique servant de base à toute l'indexation du système ;
- le langage source traité, c'est-à-dire la langue naturelle utilisée dans les comptes rendus médicaux ;
- ainsi que les phénomènes linguistiques apparaissant dans la langue traitée, et dont le traitement s'avère nécessaire pour permettre une traduction de ce langage source dans le modèle sémantique de RIME.

Nous montrons, dans les chapitre 4 et 5, les moyens mis en œuvre pour permettre la génération automatique du modèle sémantique à partir de comptes rendus médicaux :

- un lexique, qui contient les mots des comptes rendus médicaux, et auquel nous avons associé une interface de mise à jour permettant l'insertion des mots nouveaux accompagnés de leurs informations morphologiques, syntaxiques et sémantiques ;
- trois processus linguistiques indépendants (morphologie, syntaxe et sémantique) capables de traiter les phénomènes linguistiques relevés dans le chapitre 3 ;
- un processus de coopération garantissant l'indépendance entre les processus linguistiques, et capable de gérer leurs différents appels, et de transmettre entre eux les informations nécessaires à leur fonctionnement ;

Le chapitre 6 montre la réalisation et l'expérimentation qui ont été faites. Les programmes développés pour cette réalisation ont un caractère expérimental dont l'intérêt immédiat est simplement de montrer la validation de la stratégie d'indexation proposée.

Chapitre 2
Les outils linguistiques en recherche d'informations

Ainsi que nous l'avons montré dans l'introduction, tout système de recherche d'informations se caractérise par une fonction d'indexation et une fonction d'interrogation, qui à partir d'un corpus de documents en permettent la recherche. La correspondance requête-document est évaluée par rapport à une fonction de correspondance, qui est elle-même fondée sur un modèle abstrait appelé modèle de correspondance. Notre travail s'apparentant à un modèle de correspondance *linguistique*, nous présentons en détail, dans ce chapitre, les principes de ce modèle linguistique, ainsi que des exemples de systèmes fondés sur une approche analogue.

2.1. intérêts des outils linguistiques en recherche d'informations

L'idée d'introduire des outils linguistiques dans les systèmes de recherche d'informations n'est pas nouvelle dans les modèles théoriques. Cependant, et malgré leur grande évolution, les techniques linguistiques sont encore très rarement utilisées dans les systèmes opérationnels.
Bien que longtemps décriés par les tenants des modèles classiques, les outils linguistiques peuvent de façon évidente apporter certaines améliorations dans

les systèmes de recherche d'informations [SPAR79], [SMIT80], [SALT85], [DOSZ86] :

- tout d'abord au niveau des interfaces de dialogue avec l'utilisateur, notamment au moment de l'interrogation du système lors d'une session de recherche ;

- au niveau de la définition des termes d'indexation qui peuvent être envisagés comme des unités linguistiques complexes, telles que par exemple des syntagmes nominaux complexes, au lieu des traditionnels termes d'indexation simples ;

- au niveau des réponses données à l'utilisateur, qui au lieu d'être de simples références à un livre par exemple, pourraient être des réponses directes aux questions posées au travers d'un dialogue relevant d'un système de question-réponse ;

- éventuellement on peut également imaginer des fonctions de transformation de textes, tels que des générateurs de résumé ou bien encore des procédures de traduction partielle ;

- etc

Certains de ces objectifs peuvent paraître irréalistes à l'heure actuelle, et nous n'envisageons pas de présenter dans cette partie tous les systèmes mettant en exergue l'un ou l'autre de ces aspects. Mais étant donné que nous nous intéressons particulièrement dans ce travail à valider le deuxième aspect de cette liste, nous présentons certains systèmes permettant également l'extraction de termes d'indexation complexes au travers de procédures linguistiques.

Une langue naturelle offre dans sa globalité suffisamment de complexités, d'ambiguïtés pour interdire toute présomption de traitement automatique et complet. Aussi tout traitement automatique de la langue naturelle ne peut être envisagé que pour atteindre des objectifs bien délimités, de manière à pouvoir développer les outils nécessaires pour les atteindre. Classiquement, tout traitement peut se décomposer en quatre niveaux : la morphologie, la syntaxe, la sémantique et la pragmatique. Il faudrait en fait, pour être complet, y ajouter la phonologie et la lexicologie. Chaque traitement présente, en fonction des objectifs qui lui sont assignés, un degré d'intervention, de

profondeur propre pour chacun de ces quatre niveaux ; il se peut par ailleurs qu'un ou plusieurs de ces niveaux soient inexistants pour un traitement donné. En fait, quel que soit le domaine traité, cette discussion porte généralement sur la profondeur respective demandée aux niveaux syntaxique et sémantique. Il est admis que ces deux processus ne peuvent isolément résoudre tous les problèmes qui leur sont soumis ; ainsi la syntaxe a besoin de la sémantique, et réciproquement. Mais il est également clair que, selon les objectifs assignés à un traitement, la syntaxe peut être priviligiée par rapport à la sémantique, ou l'inverse.

Cette discussion a donné naissance à deux grands courants de modèles linguistiques en recherche d'informations :

- le premier prône l'utilisation de la syntaxe essentiellement dans le but d'extraire des structures syntaxiques relativement simples, et représentatives a priori du contenu des documents. Les méthodes utilisées dans ces systèmes sont relativement performantes, et permettent une indexation assez fine de corpus dont le contenu peut difficilement être cerné (voir partie 2.2) ;

- le second courant met en exergue une sémantique forte permettant une représentation relativement fine des documents. Il est clair que de tels systèmes offrent une possibilité de représentation, et par conséquent de recherche, très fine des documents, mais que cette précision est coûteuse de par la nécessité d'obtenir une représentation sémantique de l'univers du domaine couvert par le corpus, ce qui ne peut généralement se faire sans le concours de spécialistes de ce domaine (voir partie 2.3).

2.2. les modèles à dominante syntaxique

Le premier système que nous présentons ici est le prototype IOTA développé à Grenoble [CHIA & al86]. La base de l'indexation de IOTA repose sur l'hypothèse que l'information véhiculée dans les textes se retrouve essentiellement dans les syntagmes nominaux des textes. Aussi le travail de l'indexation dans IOTA se divise-t-il en deux phases :

- tout d'abord une phase d'extraction des syntagmes nominaux basée sur un analyseur de surface de la langue naturelle entièrement automatique et

capable d'enrichissement automatique lors de la rencontre de mots nouveaux [BERR & PALM86], [PALM88] ;

- ensuite une phase de génération et de pondération des termes d'indexation à partir des syntagmes nominaux extraits par consultation d'un thésaurus prédéfini, tout en tenant compte de la structure logique des documents du corpus traité ainsi que de certains éléments textuels particuliers tels que les titres des chapitres, paragraphes, etc [KERK84].

La génération des termes d'indexation est fondée sur un processus de transformation des groupes nominaux reconnus dans le texte, à partir d'une comparaison avec les éléments d'un thésaurus. Ce processus de transformation est fondée sur des critères essentiellement linguistiques.

Le thésaurus de IOTA, utilisé tant à l'indexation qu'à l'interrogation, est construit par un processus entièrement automatisé, fondé sur des critères syntaxiques et statistiques. Ce travail consiste à tout d'abord évaluer les liaisons contextuelles entre certaines classes de termes dans un corpus représentatif du domaine, et à enregistrer ces mesures dans une matrice terme-terme. L'exploitation de cette matrice permet d'en extraire des sous-graphes maximaux complets, appelés *cliques*, qui sont considérés comme représentants de classes de concepts : une clique représente l'ensemble des concepts correspondant aux groupes nominaux qui peuvent être construits à partir de ses constituants. L'étude de ces cliques permet par ailleurs de mettre en évidence des relations sémantiques intéressantes entre concepts [BRUA85] [BRUA89].

L'interrogation dans IOTA est bâtie autour un système expert permettant une interrogation en langue quasi-naturelle du système, la modélisation - la "typologie" de l'usager, l'évaluation des références résultat et la reformulation automatique des requêtes lors d'une évaluation jugée insuffisante des références données à un utilisateur [DEFU86].

Le système REALIST, développé au centre de recherche SIEMENS à Münich [THUR86], propose deux outils, l'un syntaxique, l'autre statistique, d'aide à l'indexation. Ces outils permettent la génération lors de la phase d'indexation d'un ensemble de relations entre les termes d'indexation. L'analyse syntaxique

des textes est implantée en PROLOG, et permet l'identification de syntagmes syntaxiques particuliers tels que par exemple des syntagmes nominaux. Lorsqu'un texte a été traité, le système connaît les termes d'indexation candidats syntaxiquement et statistiquement et les propose à la personne chargée de valider l'indexation. Ce type d'aide à une indexation manuelle travaillant essentiellement sur des outils syntaxiques est souvent proposée comme première étape dans les gros systèmes à indexation manuelle [DEJA & GARB86].

Le système FASIT - Fully Automatic Syntactic Indexing of Text - développé par Dillon, Gray et McDonald [DILL & GRAY83] propose une fonction d'indexation complètement automatique basée sur un processus d'analyse syntaxique. Chaque texte est décomposé mot par mot en assignant tout d'abord à chacun une catégorie syntaxique, de manière à extraire des termes ou des syntagmes supposés forts selon des critères syntaxiques consignés dans des schémas syntaxiques prédéfinis. Ce travail a été testé sur 75 catégories syntaxiques et 1087 règles de désambiguïsation syntaxique, et propose des performances intéressantes [DILL & MACD83] sur un corpus relativement réduit, et il faudrait mesurer ses réelles performances sur une collection de documents plus large. Le principal défaut de FASIT est en fait de ne prévoir aucune procédure de récupération en cas de textes syntaxiquement incorrects.

A. Smeaton et C.J. Van Rijsbergen [SMEA87] [SMEA & RIJS88] proposent une expérimentation très intéressante, portant sur des calculs d'améliorations apportées par un traitement linguistique. Ce traitement linguistique consiste ici en l'utilisation d'un analyseur syntaxique qui permet, lors de la phase d'interrogation, une analyse fine des syntagmes nominaux. L'expérimentation a porté sur 48 questions-tests de la collection CACM mise en œuvre par E. Fox à Cornell University. Sur ces 48 questions, 17 n'ont pu être traitées par l'analyseur syntaxique, et il faut par conséquent considérer pour ces questions une amélioration nulle par ce traitement. Cependant le résultat de l'expérience montre que sur les 31 questions restantes, on observe une amélioration en précision allant jusqu'à 13% des réponses fournies.

Tous ces systèmes offrent des possibilités particulièrement intéressantes d'indexation et d'interrogation sur des domaines très larges. Cependant ces systèmes ne sont pas suffisamment riches pour l'approche que nous avons dans RIME, où nous voulons aller très loin en compréhension des documents. RIME entre de ce fait dans les systèmes à modèle sémantique que nous présentons ci-après.

2.3. les modèles à dominante sémantique

Les modèles sémantiques utilisés dans les systèmes de recherche d'informations dérivent en fait directement des modèles utilisés dans les systèmes classiques de compréhension de la langue naturelle. Par conséquent, nous distinguons ici trois parties :

- une première partie où nous présentons les systèmes classiques de compréhension de la langue naturelle, sur lesquels nous insisterons particulièrement dans la mesure où leurs principes sont repris dans les systèmes de recherche d'informations basés sur des modèles sémantiques ;
- une deuxième partie dans laquelle nous montrons certains exemples de systèmes de recherche d'informations utilisant des outils de nature sémantique ;
- finalement, nous montrons des exemples d'applications médicales.

2.3.1. les systèmes classiques de compréhension de la langue naturelle

2.3.1.1. présentation de différents systèmes

MARGIE a été un des premiers systèmes de traitement de la langue naturelle [SCHA72]. Son but consiste en la traduction de phrases en langue naturelle dans un modèle conceptuel très profond, indépendant de toute langue naturelle, appelé Dépendance Conceptuelle.
MARGIE travaille sur trois modules :

- un analyseur conceptuel qui transforme les phrases données en entrée en leur représentation en Dépendance Conceptuelle ;

- un générateur capable de traduire des représentations en Dépendance Conceptuelle en langue naturelle ;

- un programme capable de chercher des références, et capable d'inférences sur les textes donnés en entrée, et qui permet de travailler sur deux modes :

- le mode paraphrase qui permet à partir d'une représentation d'une phrase en Dépendance Conceptuelle de regénérer toutes les possibilités de sa traduction en langue naturelle. Par exemple :

entrée : *John killed Mary by choking Mary*

sortie 1 : *John strangled Mary*

sortie 2 : *John choked Mary and she died because she could not breathe*

sortie 3 : *Mary died because she was unable to inhale some air and she was unable to inhale some air because John grabbed her neck* ;

- le mode inférence capable de générer les inférences potentielles à partir d'une phrase. Par exemple :

entrée : *John gave Mary an aspirin*

sortie 1 : *John believes that Mary wants an aspirin*

sortie 2 : *Mary is sick*

sortie 3 : *Mary wants to feel better*

sortie 4 : *Mary will ingest the aspirin.*

MARGIE est cependant un système très irréaliste dans la mesure où il ne fonctionne que sur des phrases isolées et indépendantes. Ce qui ne permettait la levée des ambiguïtés que le contexte des phrases aurait permis. De plus le mode inférence explose très rapidement puisqu'il est incapable de mesurer la pertinence de certaines inférences par rapport à d'autres. Cependant ce système est historiquement le premier à avoir permis des manipulations de la langue naturelle aussi performantes que les manipulations en modes paraphrasage et inférence qu'il propose. Et le principe de la représentation en Dépendance Conceptuelle est une idée qui a été largement reprise dans beaucoup d'autres systèmes de manipulations de la langue naturelle.

SAM est un système de compréhension d'histoires, dont le but est de construire correctement les inférences, dont l'importance avait été sous-estimée dans

MARGIE [CULL78]. Les connaissances prennent la forme de chaînes causales stéréotypées d'événements appelées *scripts*. Dans les exemples donnés pour l'expérimentation de SAM, les scripts représentent des scènes de la vie quotidienne telles qu'aller au restaurant, prendre le bus, visiter un musée, ...
SAM identifie le script dont il est question dans un contexte donné, et y détermine la place de chaque événement traité. Les événements non attendus ne sont pas traités. Par exemple à partir des phrases :

John went to a restaurant
He ordered a hamburger
He paid the check and left

SAM reconnaît tout d'abord que le script approprié pour cette histoire est le script *restaurant*. SAM détermine ensuite que la première phrase est du type *patron goes to restaurant*, et est un événement du script, de même que *patron orders his meal*, *patron leaves the restaurant*. Si l'on demande ensuite à SAM *what did John eat at the restaurant?*, SAM répond *John ate a hamburger*, car la connaissance du script *restaurant* permet le minimum d'inférences pour déduire que généralement une personne mange ce qu'elle commande.
SAM permet de montrer qu'un problème de compréhension ne peut se résoudre que si la connaissance de l'univers des textes traités est connue du système.
Le problème principal de SAM consiste en ce que les liens entre événements doivent être spécifiés en extension. SAM n'est pas capable de réagir à des situations nouvelles, sortant du cadre d'un script. Il est cependant applicable à toutes les descriptions de séquences d'événements stockés dans des scripts qui selon [SCHA & ABEL77] constituent une part importante des situations réelles. D'autre part, SAM demande de rencontrer les événements dans le même ordre que les prototypes des scripts, ce qui n'est pas toujours le cas dans un texte.

FRUMP (Fast Reading and Understanding Memory Program) s'efforce de son côté, d'appliquer des scripts d'une manière intégrée [DEJO79]. Son but est de produire des résumés succincts de dépêches de l'agence de presse UPI, dans des domaines variés. Une fois un script reconnu, il se sert des prédictions

tirées de la définition de ce script pour guider l'analyse du texte. Il lui est ainsi possible de lire un texte "en diagonale", en n'en retenant que les informations essentielles, d'où son nom de Fast Reading and Understanding Memory Program.

La reconnaissance de l'occurrence d'un événement n'est plus le fruit d'une comparaison directe de la représentation linéaire d'une phrase et de l'évolution du script, mais le résultat d'un dialogue entre un module Substantiateur, et un module Prédicteur : le Prédicteur prédit l'apparition de conceptualisations correspondant aux prototypes du script, et demande au Substantiateur si ces conceptualisations sont effectivement instanciables. Pour reconnaître une conceptualisation, le Substantiateur fait en général appel à un analyseur. Il cherche d'abord un mot bâtissant une partie quelconque de la conceptualisation demandée, et retourne cette partie de conceptualisation au Prédicteur. Le Prédicteur lui commande alors de trouver un par un les concepts remplissant les rôles vides de la conceptualisation initiale. Le Substantiateur applique des informations sémantiques ainsi que des informations syntaxiques pour mener à bien sa tâche.

Grâce aux prédictions sans cesse reformulées par le Prédicteur en fonction du contexte, l'analyse du texte dispose du maximum d'informations disponibles à chaque moment. FRUMP utilise ces informations non seulement pour aider à la compréhension, mais aussi pour focaliser la lecture sur les points qui l'intéressent. Il en résulte qu'il ne lit que ce qu'il cherche, certaines parties des dépêches étant simplement sautées. Ce traitement essentiellement descendant ne permet pas de prendre en compte des faits non prédits par le mécanisme de FRUMP, et qui, même sans être immédiatement rattachables à la représentation du texte, pourraient être utiles à une compréhension plus complète. De par cette limitation due à la nature même des scripts, FRUMP présente sur ce plan les mêmes limitations que SAM.

2.3.1.2. mise en œuvre des traitements linguistiques

Tout traitement de la langue naturelle est généralement divisé en quatre étapes fondamentales :

- la **morphologie**, c'est-à-dire la segmentation d'un texte en mots, à partir de l'étude de la forme des mots (lexèmes) ;

- la **syntaxe**, c'est-à-dire la détermination de la structure des phrases selon les catégories syntaxiques des mots qui les composent ;

- la **sémantique**, c'est-à-dire la construction du sens des assertions à partir du sens des mots les composant ;

- la **pragmatique**, c'est-à-dire la connaissance du monde réel, nécessaire à la compréhension globale de tout texte.

Selon les besoins requis pour sa mise en œuvre, tout traitement de la langue naturelle traite plus ou moins partiellement chacune de ces étapes, privilégiant éventuellement l'une par rapport aux autres. Et il s'agit pour tout système de justifier de tous ces choix.

Classiquement, la discussion porte essentiellement sur les rapports entre la syntaxe et la sémantique qui peuvent sur certains aspects apparaître concurrentes, et sur d'autres se révéler complémentaires [SCHA & BIRN80], [SCHA81]. Chaque système justifie de par ses objectifs finaux les problèmes qu'il place dans chacune de ces deux fonctions, et montre au travers de transferts d'informations entre les deux une éventuelle prééminence de l'une par rapport à l'autre.

Nous pouvons relever dans les systèmes existants quatre méthodes très générales d'appréhender ce problème :

a - la syntaxe et la sémantique sont complètement séparées, l'analyse syntaxique est un processus complétement indépendant et prioritaire par rapport au processus sémantique. Dans ces systèmes, la syntaxe contrôle d'un bout à l'autre toute l'analyse [CHOM65] ;

b - l'analyse sémantique et l'analyse syntaxique coopèrent un peu. Il y a ici encore une priorité à l'analyseur syntaxique qui fournit en sortie l'entrée de l'analyseur sémantique. De plus, l'analyseur syntaxique peut faire appel à un composant sémantique pour prendre une décision d'ordre syntaxique, mais cette entraide est entièrement contrôlée par la syntaxe [FODO74] [WOOD70],

encore que [MARC79] permette un peu plus de flexibilité dans la communication entre syntaxe et sémantique ;

c - l'analyse sémantique et l'analyse syntaxique coopèrent un peu plus. L'interaction n'est plus contrôlée exclusivement par un mécanisme syntaxique. Le composant syntaxique travaille, donne la main au composant sémantique qui, quand il a terminé, rappelle la syntaxe pour plus d'informations [WINO72] [WINO77] ;

d - l'analyse sémantique et l'analyse syntaxique sont utilisées parallèlement. La décision d'utiliser du savoir syntaxique ou du savoir sémantique est prise par une structure de contrôle indépendante [SCHA72] [RIES75] [MARS78] [SCHA80].

Tout ceci nous montre les différents choix qu'il faut faire dans tout traitement de la langue naturelle :

- d'une part quant à la profondeur demandée tant à la syntaxe qu'à la sémantique ;

- d'autre part quant à la gestion des différents processus linguistiques.

Nous reviendrons dans la conclusion de ce chapitre, ainsi que dans les chapitres suivants sur nos choix dans RIME quant à ces différents problèmes.

2.3.2. les applications en recherche d'informations

Il existe en recherche d'informations un certain nombre d'exemples de systèmes utilisant, tant à l'indexation qu'à l'interrogation, des outils de nature sémantique soit comme amélioration des performances d'un système déjà existant, soit comme base d'un nouveau prototype. Nous ne présentons ici que deux exemples de tels systèmes :

- ADRENAL de W.B. Croft, D.D. Lewis, et N. Bhandaru, développé à l'University of Massachusetts ;

- HAVANE de P. Bosc, M. Courant et S. Robin, développé à l'IRISA.

L'idée du système ADRENAL [CROF & LEWI87] est de construire une enveloppe autour du système de recherche d'informations I^3R : cette

enveloppe est chargée d'une part d'une analyse fine des requêtes de manière à les transmettre au système, d'autre part d'analyser plus précisément le contenu des documents donnés en réponse par le système interne. Le but d'ADRENAL est de traduire requêtes et documents sélectionnés dans un modèle de représentation du corpus traité, appelé REST, et reprenant les principes de la Dépendance Conceptuelle de Schank. Dans une première expérimentation sur la collection CACM, les auteurs ont montré tout l'intérêt de leur proposition, en montrant des résultats apportant une amélioration remarquable du système. La base de cette expérimentation repose cependant sur une traduction manuelle des requêtes-tests dans REST. Les auteurs ne donnent actuellement qu'une proposition très floue de traduction automatique de la langue naturelle dans REST, au travers d'une part de dictionnaires de mots et de syntagmes, et d'autre part d'heuristiques générales de génération automatique de Dépendances Conceptuelles pour les mots inconnus.

HAVANE [BOSC, COUR & ROBI85] est un système expert en mise en relation de petites annonces. Les petites annonces se caractérisent par une absence de structure syntaxique globale, et une grande richesse sémantique et surtout pragmatique. Havane est construit autour de spécifications relatives, d'une part au langage des petites annonces, d'autre part à la mise en relation d'annonces sur des critères flous. Nous ne rappelons ici que la partie de Havane permettant le traitement du langage des petites annonces. Ce traitement utilise deux outils principaux bâtis autour du formalisme des grammaires hors-contexte attribuées :

- un descripteur de structure, capable de décrire les annonces acceptables comme des arborescences sémantiques. Le descripteur est donc représenté par une grammaire hors-contexte, telle que l'on puisse construire un axiome *annonce* accompagné d'une liste d'attributs contenant tous les éléments sémantiques jugés significatifs pour la comparaison d'annonces ;

- des grammaires d'analyse linguistique, dont le but est d'établir le lien entre les éléments textuels pertinents pour la comparaison, et les attributs des arborescences sémantiques. Elles sont également décrites autour du formalisme des grammaires hors-contexte attribuées, dont, pour cette partie,

les terminaux sont des chaînes de caractères directement issues du langage des petites annonces.
HAVANE constitue une étape intéressante dans les systèmes utilisant des outils linguistiques comme base d'un nouveau prototype. De plus, comparativement aux langages non spécialisés, le langage restreint des petites annonces permet une expérimentation linguistique plus aisée.

2.3.3. les applications médicales

La médecine offre des possibilités très intéressantes pour les modèles linguistiques.
Nous citons entre autres le Linguistic String Processor to medical reports [SAGE78] dont le but est de générer des formes canoniques de comptes rendus médicaux : chaque composant de phrase de comptes rendus traités remplit un champ spécifique dans un format standard de l'information. Par exemple, la phrase *Patient 1st had sickle cell anemia diagnosed at age 2 years when he complained of leg pain, he was worked up and diagnosis was made* est traduite sur la figure 4.

Le système comprend une normalisation de la sémantique véhiculée dans les documents lors du remplissage de chacune des colonnes des tables, mais ne propose par ailleurs aucun affinement syntaxique ou sémantique tel qu'un remplacement des mots issus des textes par des représentants de catégorie sémantique, ou bien une normalisation syntaxique des mots mis dans les colonnes. Le format ne propose non plus aucun lien sémantique même implicite entre les différentes colonnes d'un tableau.

HELENE [ZWEIG &al87] est un système de compréhension automatique de comptes rendus en langue naturelle, construit à l'aide de KONTROL-META, un laboratoire d'expérimentation de modèles de compréhension, fondé sur une architecture de tableau noir. Ce modèle organise en modules autonomes les différents processus linguistiques mis en œuvre pour ce traitement : morphologie, syntaxe, sémantique, ... Il permet ainsi d'appeler ces différents

processus lors de la construction des scripts. Bien que les auteurs ne précisent pas actuellement le contenu exact des différents processus utilisés, la structure de données choisie, et les ordonnancements des différents processus par le tableau noir, ce travail mérite une grande attention par son choix d'utilisation de tableau noir comme outil de coopération des processus linguistiques.

figure 4
un exemple de représentation dans le système LSP

	CONJ	PATIENT	TREATMENT		PATIENT STATE				TIME	
			INST	V-MED	V-PAD	B-PART	SIGN	DIAG	P2	REF PT
1		patient		1st had diagnosed				sickle cell anemia	at	age 2 years
2	when	he			complained of	leg	pain			(" ")
3		he		was worked up						
4	and			diagnosis was made						

MEDIAL, pour MEdical DIALogues, est un système d'analyse en ligne de comptes rendus [BORS, WHER & SCHE84]. Ce système permet aux utilisateurs de saisir leurs textes en français, puis le système en extrait les informations les plus pertinentes pour ensuite les stocker dans une base de données. L'utilisateur peut également utiliser MEDIAL comme un système de questions-réponses en langue naturelle, les réponses sont dans ce cas puisées dans les données stockées dans la base. La traduction est essentiellement fondée sur une analyse syntaxique très performante des textes [WHER83], et utilise également un processus sémantique chargé, à l'aide d'une base de connaissances, d'uniformiser le vocabulaire source.

2.4. conclusion

Tous ces systèmes présentent à la fois des aspects tant syntaxiques que sémantiques intéressants, mais aussi des imperfections souvent liées à des procédures linguistiques à la fois insuffisantes et lourdes :

- insuffisantes dans la mesure où ces procédures ne gèrent que certains phénomènes linguistiques, sans que les phénomènes gérés fassent l'objet d'une étude très fine de la part des constructeurs du système, ce qui est le cas dans les systèmes LSP et HELENE ;

- lourdes dans la mesure où la discussion pour le partage des tâches entre les différents processus linguistiques potentiels, notamment la syntaxe et la sémantique, est généralement inexistante : la plupart du travail est généralement confié à un seul processus, alourdi en ce sens par des tâches qu'un autre processus pourrait mener à bien à moindre coût , ce que l'on retrouve dans le système MEDIAL.

Le système RIME propose d'étudier des textes en langue naturelle, de les représenter dans un modèle sémantique relativement souple (cf chapitre 3), et ce au travers de différentes procédures morphologique, syntaxique et sémantique *autonomes* :

- la profondeur et le moment d'intervention de chacun de ces processus a été fixée à partir d'une étude des phénomènes linguistiques (cf chapitre 3), ce qui est une condition nécessaire à une bonne mise en œuvre du traitement linguistique (cf chapitre 5) [BERR & CINQ88b] ;

- un processus de coopération réalise l'ordonnancement et l'indépendance des processus linguistiques tel que nous l'avons présenté dans la partie 2.3.1.2.d (cf partie 5.5).

Chapitre 3
Les données de RIME

Le but de notre travail est de traiter des documents structurés que nous présentons dans la partie 3.2.1, et qui correspondent dans RIME à des comptes rendus médicaux décrivant des images radiologiques. Une partie essentielle de notre travail consiste à traiter les parties textuelles de ces comptes rendus en les traduisant selon un format interne de représentation des connaissances. Le but de ce chapitre est de définir :

- ce format de représentation que nous appelons *modèle sémantique* ;
- les phénomènes linguistiques qu'il faut traiter pour permettre la traduction de textes en langue naturelle dans ce modèle sémantique.

D'une manière très générale, le modèle sémantique de RIME permet la représentation des comptes rendus par l'intermédiaire de faits médicaux, et de liens sémantiques entre faits médicaux : ce sont ces liens que nous appelons par la suite opérateurs sémantiques. L'idée de base consiste à représenter le sens de chaque phrase par un graphe sémantique de type arborescent. Cette approche, comme nous le verrons plus loin, s'inspire à la fois de la notion de dépendance conceptuelle et de frame [SCHA72]. Elle se distingue de celles-ci par l'existence d'une grammaire définissant les arborescences autorisées. Par exemple, la phrase *augmentation du volume du foie* contient les faits médicaux *augmentation du volume* et *foie*, et se traduit dans notre modèle sémantique par *volume* ***a pour valeur*** *augmenté et* ***porte sur*** *foie*, où les mots indiqués

en gras représentent ce que nous appelons les opérateurs sémantiques entre faits médicaux. Nous expliquerons que la représentation réelle de cette phrase dans le modèle sémantique est en fait une expression préfixée de l'arbre : *[p_sur [a_pr_val, volume, augmenté], foie]*. Pour pouvoir comprendre ce modèle sémantique et pour pouvoir s'exprimer selon ses possibilités, il faut connaître la grammaire sous-jacente que nous présentons dans la première partie de ce chapitre.

Dans la seconde partie, nous présentons le document *compte rendu médical* en en montrant ses différentes parties :

- les *attributs externes* ;
- les *attributs internes* correspondant aux parties textuelles auxquelles nous nous intéressons plus particulièrement dans ce travail.

Finalement, nous mettons en exergue les phénomènes linguistiques nécessaires à la génération du modèle sémantique de RIME.

3.1. le modèle sémantique de RIME

3.1.1. principes

Pour définir la grammaire du modèle sémantique, nous avons choisi de reprendre l'idée de dépendance conceptuelle introduite dans [SCHA72]. En effet, comme nous le montrons ci-dessous, les comptes rendus médicaux ont un contenu sémantique facilement représentable selon des principes de dépendance conceptuelle. Les connaissances véhiculées dans ces documents seront donc représentées au travers de schémas mettant en évidence des relations sémantiques entre différents concepts intervenant dans les textes traités.

L'organisation interne des comptes rendus médicaux présente de manière évidente différents niveaux : un tel document contient généralement des informations qui décrivent la technique utilisée lors de l'examen subi par le patient, les constatations faites suite à cet examen, et éventuellement un diagnostic (voir en annexe 1 un exemple de compte rendu médical). Chacune de ces parties peut à son tour être redéfinie en termes de sous-notions telles que des signes, des lésions, ... Le niveau le plus bas d'une représentation de ce type

est constitué de termes médicaux ou techniques que nous appelons vocabulaire de base. Cette hiérarchie représentative du contenu d'un compte rendu médical peut s'exprimer au travers d'une grammaire dans laquelle toutes ces notions - ou concepts - sont reliées entre elles à l'aide de relations sémantiques prédéfinies représentées par des opérateurs sémantiques pour fournir de nouveaux concepts d'ordre plus élevé.

Nous reprenons ces idées de concepts et de relations sémantiques de la façon suivante :

a. Tout d'abord, les concepts et les opérateurs sémantiques se représentent par des structures d'arbres binaires complets dans lesquelles :

- les nœuds correspondent aux opérateurs sémantiques. Ces opérateurs sont binaires, et explicitent le lien sémantique entre les concepts représentés par les sous-arbres gauche et droit de l'arborescence. Les opérateurs sémantiques sont des terminaux de la grammaire de notre modèle. Par exemple, *a_pr_val* établit un lien de valeur entre *volume* et *augmenté* dans ***[a_pr_val**, volume, augmenté]*. Nous remarquons au travers de cet exemple que nous représentons les arborescences binaires complètes en indiquant entre crochets leur parcours préfixé, c'est-à-dire *[père, fils gauche, fils droit]* ;
- les feuilles de l'arborescence correspondent à des faits médicaux ou à des termes techniques relatifs par exemple à l'examen effectué ; nous retrouvons à ce niveau ce que nous appelons le vocabulaire de base de notre modèle, c'est-à-dire, au sens de notre grammaire, des terminaux, et qui exprime la couverture sémantique de notre application. Par exemple, *poumon, cancer, opacité.*

b. Chaque phrase d'un compte rendu médical doit être traduite dans une arborescence de ce type, et l'ensemble des arborescences représente le sens du compte rendu traduit : nous appelons **compte rendu conceptuel** l'ensemble des arborescences représentant la traduction d'un compte rendu selon notre modèle. Par exemple, *condensation pulmonaire en projection du lobe supérieur droit* se traduit par ***[projection**, [p_sur, [a_pr_val, densité,*

augmenté], poumon], [a_pr_val, [partie_de, lobe, poumon], [et, supérieur, droit]].

c. Les arbres construits doivent respecter le modèle formel que nous définissons au travers d'une grammaire, et le langage défini par cette grammaire s'appelle le **langage conceptuel**.

3.1.2. la grammaire

L'organisation interne des comptes rendus médicaux peut se décrire au travers d'une grammaire dans laquelle les méta-symboles correspondent à des concepts intermédiaires tels que les signes, les lésions ou les constatations, et les symboles terminaux à des opérateurs sémantiques, à des concepts atomiques tels que les constituants de l'organisme ou encore les différentes fonctions de l'organisme. La grammaire permet ainsi de définir les concepts intermédiaires, et de spécifier les différentes structures possibles pour les arborescences construites. Les méta-symboles de la grammaire sont représentés en majuscules, les terminaux en minuscules. Nous ne présentons par la suite qu'un sous-ensemble de la grammaire, dont nous donnons en annexe 2 la définition complète.

Nous relevons trois niveaux fondamentaux dans l'organisation des comptes rendus médicaux [CHIA, BERR & CINQ87] :
a. Le premier niveau exprime qu'un compte rendu médical est constitué d'une ou plusieurs phrases, et donne une définition formelle des concepts de plus haut niveau, c'est-à-dire les constatations, le diagnostic, permettant ainsi de relier les différents composants d'un compte rendu. Par exemple :

CR ::= CONSTAT
CR ::= DIAGNOSTIC
CR ::= [**permet_de_déduire**, CONSTAT, DIAGNOSTIC]

Ces règles définissent un CR (Compte Rendu), qui est le symbole terminal de notre grammaire, comme étant constitué d'un seul constat, d'un seul diagnostic, ou bien des deux. Dans ce cas le constat permet la déduction du

diagnostic, et ce lien sémantique entre constat et diagnostic se traduit par l'opérateur sémantique *permet_de_déduire*.

b. Le second niveau définit les notions de constat, de diagnostic, ainsi que des sous-notions qui leur sont associées :

CONSTAT ::= [**dû-à**, CONSTAT, CONSTAT]

Ce qui exprime que des constats peuvent être interdépendants.

CONSTAT ::= [**montre_par**, SIGNE, EXAMEN]

CONSTAT ::= SIGNE

Ces règles définissent un constat à partir d'un signe seul, ou bien par un signe révélé par un examen ; un signe est une entité observable, telle *une cavité*.

DIAGNOSTIC ::= [**et**, DIAGNOSTIC, DIAGNOSTIC]

Cette règle exprime un diagnostic comme une combinaison de plusieurs diagnostics.

DIAGNOSTIC ::= LESION

Cette règle exprime un diagnostic comme une simple lésion, par exemple *un emphysème*.

SIGNE ::= [**a_pr_val**, SIGNE, QUAL]

Cette règle définit un signe comme un signe modifié par un qualificatif, tel que *aérique, bombé*.

SIGNE ::= [**p_sur**, SIGNE, LOC]

Cette règle définit un signe portant sur une localisation particulière, par exemple *une opacité du poumon*.

LESION ::= [**p_sur**, LESION, LOC]

Cette règle définit une lésion portant sur une partie de l'organisme, par exemple *une tumeur du poumon*.

LESION ::= [**en_rel_topo_avec**, LESION, LOC]

Cette règle situe une lésion par rapport à une partie de l'organisme, par exemple *une tumeur derrière le gril costal.*

LOC ::= [**a_pr_val**, LOC, POS]

LOC ::= CONST_ORG

Cette règle définit une localisation comme un constituant de l'organisme, ou bien comme une notion plus complexe combinant une localisation et une position particulière, par exemple *la partie supérieure du poumon droit.*

c. Le troisième niveau contient les règles pré-terminales et les règles terminales de la grammaire, et correspond aux concepts de plus bas niveau dans notre modèle. L'ensemble des symboles terminaux de la grammaire, hormis les opérateurs sémantiques, est consigné dans un lexique. Une information sémantique est attribuée à chaque entrée du lexique, et dans cette information se trouve notamment une catégorie sémantique correspondant à un méta-symbole de la grammaire ; par exemple, *augmentation* possède la catégorie sémantique *signe*. Nous reviendrons sur le lexique ultérieurement. Ce type de méthode est à rapprocher des grammaires sémantiques de Fillmore [FILL68]. Nous reviendrons sur le processus de traduction dans le chapitre 5. Par exemple :

SIGNE ::= {t ∈ V_T / catégorie_sémantique(t) = 'signe'}

Ce qui signifie que tout terme de catégorie sémantique *signe* est un SIGNE dans la grammaire, par exemple, *volume* est un signe.

LESION ::= {t ∈ V_T / catégorie_sémantique(t) = 'lésion'}

Par exemple, *tumeur, cancer.*

CONST_ORG ::= {t ∈ V_T / catégorie_sémantique(t) = 'const_org'}

Par exemple, *poumon, cœur.*

POS ::= {t ∈ V_T / catégorie_sémantique(t) = 'position'}

Par exemple, *côté.*

Cette grammaire définit donc le langage conceptuel de RIME, dont l'utilisation permet une interrogation fine, notamment en traitant les méta-

symboles de la grammaire comme des concepts intermédiaires, comme des accès à l'information, au lieu de rester simplement au niveau des faits de base.

La grammaire est composée actuellement

- d'une soixantaine de règles ;

- d'une dizaine d'opérateurs sémantiques terminaux (a_pr_val, a_pr_val_loc, dû_à, montre_par, par_rap, partie_de, permet_de_déduire, p_sur), et d'un opérateur "général" EN_REL_TOPO_AVEC, qui représente différents opérateurs sémantiques distincts, mais dont le rôle dans la grammaire est strictement identique (gauche, droit, haut, bas, avant, arrière, intra, extra, contact, près, loin, cerné, fistule, séquestre, envahissement, vers, à_partir_de, jusqu'à, contraste_entre, contraste_avec, en_projection_de) ;

- et le lexique, contenant entre autres tous les terminaux de la grammaire, distingue une dizaine de catégories sémantiques.

Pour qualifier certaines constructions de cette grammaire, nous utilisons les termes suivants : une *arborescence* pour désigner tout arbre binaire complet correspondant aux règles de la grammaire, nous utilisons également les notions classiques de *partie gauche* d'une règle, de *branche* et de *feuille* d'une arborescence.

La hiérarchie imposée par la définition même de la grammaire se répercute par des relations d'inclusion entre parties gauches de règle ; inclusion que nous utilisons dans le processus de traduction décrit au chapitre 5, et que nous définissons de la façon suivante : *une partie gauche A inclut une partie gauche B* si et seulement si l'une des assertions suivantes est vraie :

(i) A et B sont identiques. Par exemple *LOC inclut LOC* ;

(ii) il existe une règle telle que A ::= B.

Par exemple, *DIAGNOSTIC inclut LESION* au vu de la règle *DIAGNOSTIC ::= LESION* ;

(iii) il existe une partie gauche C, telle que *A inclut C*, et une règle telle que *C ::= B*.

Par exemple, *CR inclut LESION* au vu des règles *CR ::= DIAGNOSTIC* et *DIAGNOSTIC ::= LESION*.

3.1.3. conclusion

L'indexation de RIME est un processus de traduction de textes en langue naturelle dans le modèle sémantique que nous venons de décrire. Nous décrivons dans la suite de ce chapitre une autre des données de RIME : les documents à indexer que sont les comptes rendus médicaux. Cette étude se décompose en deux parties :

- tout d'abord, nous présentons les comptes rendus en tant que documents dont on peut donner la structure ;
- nous étudions ensuite plus particulièrement le contenu des parties textuelles de ces documents.

3.2. les comptes rendus médicaux

3.2.1. le document *Compte Rendu Médical*

Les comptes rendus médicaux sont des documents courts (moins d'une page) rédigés par des spécialistes - voir en annexe 1 un exemple de compte rendu médical - ; ils y décrivent entre autres l'examen médical subi par le malade, les constatations faites suite à cet examen. Ces documents combinent des attributs simples tels que le nom du malade ou la date de l'examen, et des attributs plus complexes tels que les constatations faites suite à cet examen ou encore le diagnostic médical déduit de ces constatations.

Dans le cadre de RIME, nous avons décidé de travailler sur le modèle de compte rendu médical décrit page suivante. Ce modèle fait apparaître des attributs soulignés : ***nom***, ***date*** et ***constatations*** qui sont les seuls attributs obligatoirement remplis du document. Nous montrons après le modèle général un compte rendu médical exprimé selon ce modèle.

Certains des attributs d'un modèle de document sont dits externes; ils décrivent des informations non liées au contenu sémantique du compte rendu. Dans notre modèle de documents, les attributs externes sont ***nom, adresse, date, médecin traitant, type de l'examen et médecin hôpital***. Ces attributs

EXEMPLE DE COMPTE RENDU MEDICAL APPLIQUE A CE MODELE

nom : Monsieur Hyde
adresse : pavillon D
date de l'examen : Le 12.09.85
médecin traitant :
EXAMEN
type de l'examen : TOMODENSITOMETRIE
antécédents médicaux : homme de 47 ans - Adressé pour bilan d'épanchement pleural droit avec température et signes de compression médiastinale - FIBROSCOPIE : compression extrinsèque sur la bronche lobaire moyenne mais aspect endoscopique normal - ETUDE CYTOLOGIQUE DU LIQUIDE PLEURAL pas de cellule maligne
motifs de l'examen :
incidents éventuels :
commentaires : Les coupes ont été pratiquées depuis le sommet thoracique jusqu'à la région cœliaque.
constatations : Les constatations sont les suivantes :
- processus expansif et invasif de la loge médiastinale antérieure, depuis l'étage supra-aortique jusqu'à l'étage cardiaque. Plusieurs critères sémiologiques peuvent être précisés,
  - densité tissulaire et structure hétérogène faisant craindre la présence de larges zones de nécrose,
  - déformation du versant antérieur de la veine cave supérieure,
  - déplacement modéré mais indiscutable du médiastin à gauche et en arrière.
- important épanchement pleural de la grande cavité droite avec par place épaississement nodulaire du feuillet pleural faisant craindre une extension pleurale et/ou sous-pleurale de la tumeur médiastinale antérieure.
- sous l'épanchement, collapsus incomplet et incarcération du poumon droit (lobe moyen et lobe inférieur).

conclusion :
processus expansif et invasif du médiastin antérieur, de grandes dimensions, de densité tissulaire, associé à des images suggérant une greffe pleurale et/ou sous-pleurale à distance. Cet aspect TDM suggère volontiers le diagnostic de thymome lymphoépithélial invasif avec greffe pleurale.
médecin hôpital : Docteur Jekill.

correspondent à ceux des SGBD classiques; lors de l'interrogation, leur utilisation met en œuvre une fonction de correspondance exacte (identité).
Les autres attributs d'un document sont dits internes dans le sens où ils expriment un contenu sémantique au travers d'une langue naturelle ou artificielle. Dans notre modèle de documents, les attributs internes sont ***antécédents médicaux, motifs de l'examen, incidents éventuels, commentaires, constatations et conclusion.***

Ce sont ces attributs internes que nous étudions plus particulièrement ici, puisque ce sont eux que nous désirons traduire dans le modèle sémantique de RIME. L'étude de la langue naturelle utilisée dans ces attributs va nous permettre par la suite de dresser la liste de phénomènes linguistiques que doivent traiter les processus linguistiques de RIME pour permettre la génération du modèle sémantique.

3.2.2. conclusion

De par leur nature et leur élaboration, ces documents sont d'une part très techniques, d'autre part écrits dans un style concis et direct : la langue dans laquelle sont exprimés les comptes rendus relève d'une langue de spécialité, c'est-à-dire d'une langue utilisant toutes les ressources morphologiques, syntaxiques, sémantiques voir pragmatiques d'une langue mère, et y ajoutant ses propres spécificités notamment dans ce corpus médical au niveau du vocabulaire et de la syntaxe employés. En effet, le vocabulaire employé dans les comptes rendus médicaux est constitué de termes techniques permettant la description de la technologie relative aux examens médicaux, et de termes médicaux permettant d'exprimer un constat médical et un diagnostic. La syntaxe utilisée ici est un sous-ensemble des possibilités de la langue naturelle mère, puisque les phrases écrites sont généralement des groupes nominaux éventuellement complexes.
Toutes ces caractéristiques vont nous permettre dans la dernière partie de ce chapitre de mettre en évidence les phénomènes linguistiques qu'il est

nécessaire de traiter pour réaliser la traduction des comptes rendus. Cette étude se présente en deux parties :

- tout d'abord, nous donnons, dans la partie 3.3.2., une liste générale des phénomènes linguistiques existant en français ;
- ensuite nous distinguons plus particulièrement, dans la partie 3.3.3., les phénomènes linguistiques concernant notre processus de traduction.

Finalement, nous montrons comment les différents processus linguistiques de RIME se partagent ce travail.

3.3. les besoins linguistiques

L'idée que nous développons ici est de donner une liste aussi complète que possible des besoins ou **tâches** intervenant dans un traitement de la langue naturelle, et de montrer ensuite lesquelles doivent être prises en compte dans RIME, compte tenu des besoins linguistiques particuliers de l'application. Il faut par ailleurs déterminer quelle fonction du système est à même de signaler chacune de ces tâches, et laquelle peut les réaliser. Nous spécifions ainsi chacune des fonctions de RIME par les tâches qu'elle se doit de détecter, et par celles qu'elle a à résoudre. Nous verrons dans le chapitre 5 les outils à mettre en œuvre dans chacune des fonctions pour que ces spécifications soient réalisées. Nous nous devons également de définir des outils de décision au cas où il y aurait un choix entre plusieurs solutions pour une même tâche : par exemple, il faut prévoir des outils permettant la résolution des problèmes sémantiques tels que la polysémie.

3.3.1. vocabulaire

Nous prenons ici les définitions de P.Palmer dans [PALM88].

3.3.1.1. lettres

Une lettre est un caractère appartenant à l'alphabet classique du français en minuscule ou en majuscule, ou la réunion d'un signe orthographique et d'un

caractère pouvant former un voyelle accentuée ou le "c cédille". Si l'on désigne par LET l'ensemble des lettres, on a :

LET = {a,b,c, ... ,z,A,B, ... ,Z,é,è,à,ù,â,ê,î,ô,û,ë,ï,ü,ç }

Nous ne prenons pas en compte les signes orthographiques "œ" et "æ" que nous considérons comme les successions des caractères "o" et "e" d'une part, et "a" et "e" d'autre part.
Notons par ailleurs que cette définition n'est valable que dans le cas du français.

3.3.1.2. chiffre

L'ensemble des chiffres noté CHIF est donné par l'énumération suivante :

CHIF = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 }

3.3.1.3. mot simple

Un mot simple M est une suite contiguë finie de caractères pris dans l'ensemble noté ALPH, constitué par l'union de l'ensemble des lettres et de l'ensemble des chiffres. On dit *mot simple* par opposition à *mot composé.*

ALPH = LET ∪ CHIF

3.3.1.4. séparateur

On appelle séparateurs tous les autres caractères utilisables pour l'écriture du français. Cet ensemble noté SEP comprend :
- les signes de ponctuation;
- les caractères représentant des opérateurs : + - = < > etc;
- les signes orthographiques que sont l'apostrophe ' et le trait d'union - ;
- le caractère "espace";
- ainsi qu'un certain nombre de symboles : & % etc.

3.3.1.5. séparateur non-impératif

On distingue parmi l'ensemble des séparateurs un sous-ensemble qui est constitué de ce que nous nommons des séparateurs non-impératifs. Cette distinction sera utilisée pour définir les mots composés. Cet ensemble noté SNI est composé du caractère blanc, de l'apostrophe et du trait d'union. Par opposition, tout autre séparateur est appelé *séparateur impératif.*

SNI = { " ", " ' ", "-" }

3.3.1.6. mot composé et mot

Un mot composé noté MC est une suite contiguë et finie de mots simples séparés les uns des autres par un séparateur non impératif. Exemple : *aujourd'hui, pomme de terre, ci-dessus.*
On remarque que tout mot composé MC n'appartient pas forcément à une langue naturelle, même si les mots simples le composant appartiennent à cette langue : *l'élève, viendra-t-il.*
Les nombres constituent un ensemble de mots particuliers, dans la mesure où ils sont constitués d'une suite contiguë de chiffres, auxquels peuvent se mêler dans certaines conditions des points ou des virgules : *1.986.456,123*

On appelle *mot* tout mot simple ou composé appartenant à une langue naturelle.

3.3.1.7. attributs

Nous appelons attributs toute connaissance de type morphologique, syntaxique ou sémantique pouvant être attribuée à un mot. Nous distinguerons plus particulièrement par la suite les attributs syntaxiques et les attributs sémantiques d 'un mot.
Nous appelons *attributs virtuels* les attributs hors contexte d'un mot, et *attributs actuels* les attributs dans un contexte donné d'un mot. Les attributs actuels forment un sous-ensemble déduit des attributs virtuels. Par exemple,

les attributs syntaxiques virtuels de *porte* sont *substantif, verbe conjugué* et dans le contexte *il porte les bagages* l'attribut syntaxique actuel de *porte* est *verbe conjugué.*

3.3.2. la liste des tâches

Nous allons dans cette partie décrire les tâches dont la réalisation peut être exigée par un traitement de la langue naturelle. Nous appelons tâche tout phénomène linguistique de la langue naturelle, et nous verrons dans 3.3.3. les tâches retenues dans RIME.

De manière très générale, tout traitement de la langue naturelle peut comprendre trois grandes classes de tâches : tout d'abord les tâches qui portent au niveau des mots d'un texte, puis les tâches de regroupement de mots pour former ce que nous appelons des structures, et finalement les tâches créant des liens entre structures :
- tout d'abord les premières tâches que nous traitons sont celles portant sur la reconnaissance des mots, permettant d'une part l'attribution de leurs attributs virtuels et d'autre part la déduction de leurs attributs actuels à partir de leurs attributs virtuels. Nous appelons ces tâches les *tâches infra-structurelles*;
- après avoir considéré les mots comme des entités individuelles, nous regardons les possibilités d'agrégation des mots pour construire et nommer des structures, ce sont les tâches que nous appelons *tâches intra-structurelles*;
- finalement, nous relevons les cas de liens implicites ou explicites entre structures, ces liens apparaissent au travers de transformations plus ou moins substantielles à l'intérieur des structures à cause de la proximité topologique ou sémantique d'autres structures. Il faut donc au travers de ce que nous appelons les *tâches inter-structurelles* mettre en exergue ces transformations, et créer les liens nécessaires entre structures.

Il s'agit maintenant de décrire plus en détail toutes ces tâches. Nous verrons par ailleurs que ces tâches s'accompagnent lors de leur mise en œuvre d'un

certain nombre de problèmes généralement liés à des ambiguïtés; la figure 5 nous montre toutes les tâches et leurs problèmes associés.

3.3.2.1. les tâches infra-structurelles

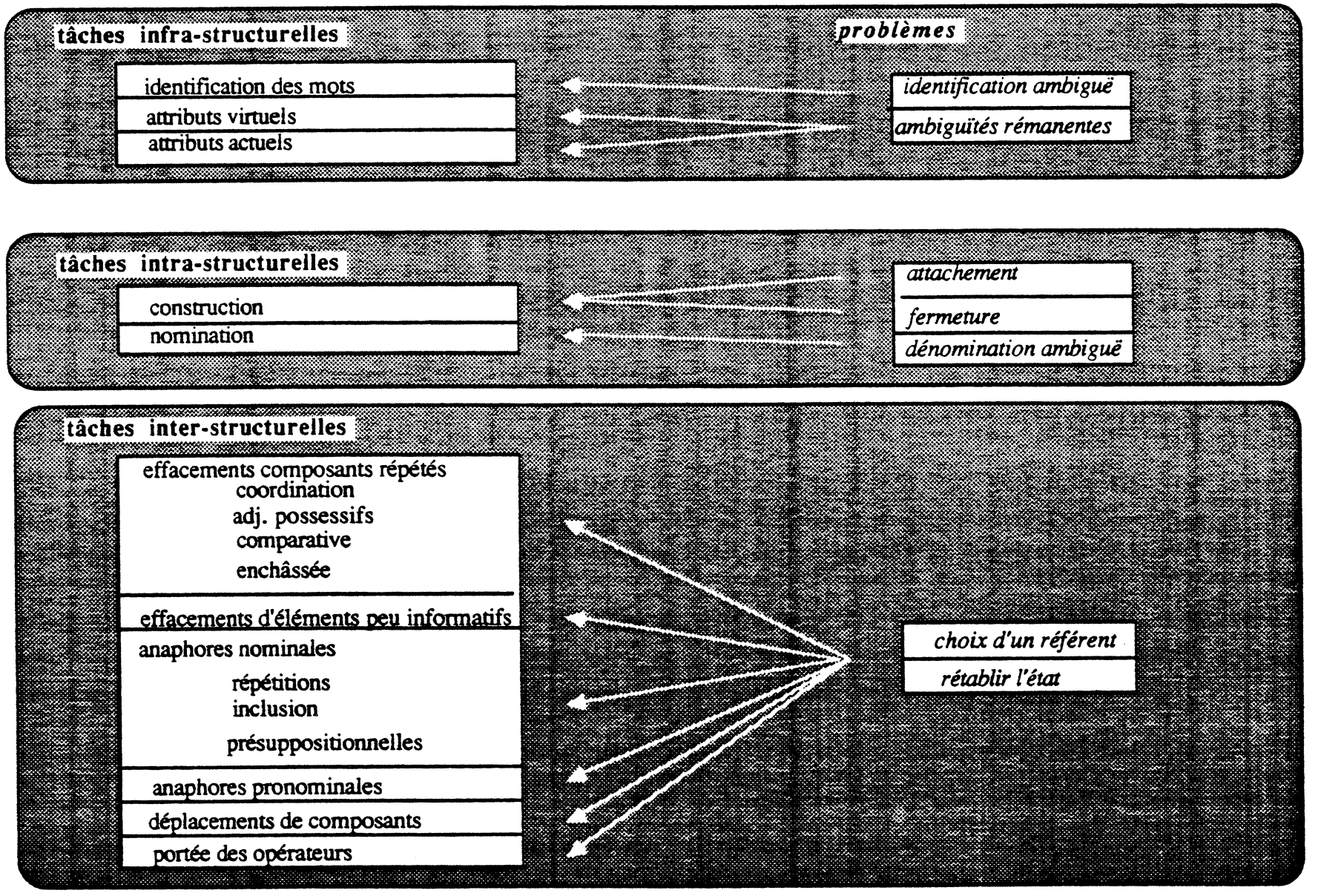
Nous appelons tâches infra-structurelles les tâches qui considèrent le mot en tant qu'entité individuelle. Elles sont au nombre de trois :

1 - **l'identification des mots :** ce qui sous-entend d'une part isoler chaque mot simple, et d'autre part déduire de cet ensemble de mots simples l'ensemble des mots selon le modèle représenté.
Par exemple, la phrase *extension ganglionnaire médiastinale au niveau du groupe de la bifurcation de la chaîne para-trachéale droite* est composée de 15 mots simples mais de 13 mots dans notre modèle, car les trois mots simples *au niveau du* correspondent à un mot composé;

2 - **la déduction des attributs virtuels**, c'est-à-dire la déduction des attributs morphologiques, syntaxiques et sémantiques potentiels de chaque mot. Tout mot possédant plusieurs attributs virtuels morphologiques, syntaxiques ou sémantiques est dit *ambigü*, ainsi un mot peut être une ambiguïté morphologique, syntaxique ou encore sémantique.
Par exemple, *ganglionnaire* est une ambiguïté morphologique car il peut être *féminin* ou *masculin*, son attribut syntaxique virtuel est *adjectif qualificatif*, et son attribut sémantique dans notre modèle est *([ganglion], const_org)*, c'est-à-dire le constituant de l'organisme ganglion;

3 - **la déduction des attributs actuels** de chaque mot, sous-ensemble déduit de ses attributs virtuels. Généralement il faut arriver à lever les ambiguïtés et ne déduire qu'un et un seul attribut actuel.
Dans l'exemple ci-dessus, *extension* qui forme le contexte droit immédiat de *ganglionnaire*, et qui a comme attribut morphologique *féminin* nous permet de déduire l'attribut morphologique actuel de ganglionnaire : *féminin* et ainsi de lever l'ambiguïté morphologique.

figure 5 : les tâches



. Ces tâches posent deux types différents de problèmes :

1 - tout d'abord l'identification des mots peut ne pas être complètement déterministe à cause des ambiguïtés posées par les mots composés.
Par exemple, dans la phrase *il est si bien* ***attendu que*** *nous n'avons pas à nous soucier de son arrivée,* et dans la phrase *nous ne viendrons pas* ***attendu que*** *nous n'avons pas été conviés,* les deux mots simples *attendu* et *que* correspondent dans un cas à deux mots, et dans le second à un seul;

2 - les ambiguïtés ne se lèvent pas toujours simplement, et certains mots restent, à ce niveau infra-structurel, ambigus. On retrouve alors les trois types d'ambiguïtés possibles :

- les *ambiguïtés morphologiques* comme par exemple *gaz* et *dangereux* qui peuvent être au singulier ou au pluriel dans la phrase *attention : gaz dangereux;*
- les *ambiguïtés syntaxiques* telles que *porte* qui peut être un substantif ou un verbe conjugué;
- les *polysémies* comme par exemple *bar* dans la phrase *le bar est ouvert* qui peut représenter soit un lieu public soit un poisson.

3.3.2.2. les tâches intra-structurelles

Après avoir considéré les mots comme des entités individuelles auxquelles sont rattachés des attributs de type morphologique, syntaxique et sémantique, nous considérons maintenant l'agrégation de mots pour constituer des structures généralement syntaxiques ou sémantiques.
Par exemple, dans la phrase *le petit garçon joue dans la chambre de sa mère,* nous associons les mots *le petit garçon,* syntaxiquement en tant que groupe nominal sujet du groupe verbal *joue,* et sémantiquement comme l'acteur de l'action *jouer.*
Nous avons pour la suite besoin de définir la notion de **composant** et d'**opérateur**. Un **composant** est une sous-partie autonome d'une structure, ci-dessus *la chambre* et *sa mère* sont deux composants de la structure *la*

chambre de sa mère. Un **opérateur** est un mot faisant office de lien explicite entre des structures, ou entre des composants d'une structure. Ci-dessus, *de* est un opérateur entre les composants *la chambre* et *sa mère*, *dans* est un opérateur entre la structure *joue* et la structure *la chambre de sa mère*.

On distingue deux tâches intra-structurelles :
- la **construction** des structures, c'est-à-dire dans l'exemple ci-dessus construire le groupe *le petit garçon* ;
- la **nomination** des structures, c'est-à-dire dans l'exemple ci-dessus déduire que syntaxiquement *le petit garçon* est un groupe nominal sujet du groupe verbal *joue*.

On relève trois problèmes spécifiques à ces tâches, les deux premiers relèvent de la construction des structures, le dernier de la nomination :

1 - **l'attachement** : ce problème se pose quand il faut déduire des dépendances dans une structure à au moins trois composants, dans laquelle le rattachement du troisième composant au premier ou au deuxième composant peut éventuellement poser problème. Considérons en effet une structure à trois composants XYZ où Z doit être relié à X ou Y, il existe alors au moins trois configurations possibles de liens entre X, Y et Z : (X (Y Z)) ou (X Y (Z)) ou encore (X (Y) (Z)); les parenthèses représentent les rattachements potentiels entre les composants, par exemple (X (Y Z)) représente Z relié à Y, et le composant (Y Z) est relié à X.
Par exemple, *confirmation d'une hypertrophie de densité tissulaire* peut être a priori analysée comme *(confirmation (d'une hypertrophie de densité tissulaire))*, ou bien comme *(confirmation d'une hypertrophie (de densité tissulaire))*, ou encore comme *(confirmation (d'une hypertrophie) (de densité tissulaire))*. Il faut donc déduire de cet ensemble de possibilités théoriques que *(confirmation (d'une hypertrophie de densité tissulaire))* est la bonne solution ;

2 - **la fermeture** : ce problème se pose quand il faut déduire des dépendances dans des structures à au moins trois composants dont le premier et le troisième peuvent éventuellement être liés. Considérons en effet une structure à trois composants XYZ où Y et Z sont liés, et où X peut être lié à Z. On peut alors avoir deux interprétations : ou bien X est un composant libre et la structure se définit à deux composants X d'une part et YZ d'autre part, ou bien X est lié à Z et on obtient une structure à deux composants XZ d'une part et YZ d'autre part.
Par exemple, *le développement et la maintenance de logiciels* peut être interprété comme *(le développement et la maintenance) de logiciels* ou bien comme *(le développement) et (la maintenance de logiciels).*

3 - on relève enfin un problème spécifique à la **nomination** des structures : celle-ci peut quelquefois être **ambiguë**, ce qui généralement est la conséquence d'une ambiguïté rémanente du niveau infra-structurel.
Par exemple, on peut relever un tel problème dans la phrase *devant cette somme, Cédric hésite* où *devant cette somme* peut être différemment nommé selon l'interprétation de *devant* comme préposition ou bien comme participe présent du verbe devoir.

3.3.2.3. les tâches inter-structurelles

Le but des tâches inter-structurelles est de mettre en évidence des liens implicites ou explicites entre des structures. Ces liens se retrouvent généralement au travers de transformations de structures, transformations généralement dues à la proximité tant topologique que sémantique d'autres structures. Il faut donc, pour les cas les plus intéressants, d'une part détecter des traces de ces transformations, et d'autre part restituer plus ou moins partiellement ces transformations au travers de liens entre structures, liens indiquant les dépendances entre elles.

On relève quatre grandes tâches inter-structurelles :

1 - on peut relever des cas d'**effacement d'une partie d'une structure** [FUCH83], effacement dû soit à la présence d'éléments présents dans une autre structure, et que donc on ne répète pas, soit à des éléments peu informatifs :

- effacement de composants répétés :

- coordination entre structures : ce problème se pose quand une structure de type XY précède une structure XZ, il y a effacement d'un élément répété quand la succession des deux structures se retrouve dans le texte en langue naturelle sous la forme *XY coordination Z*, où *coordination* représente toute coordination autorisée telle que la virgule, ou les conjonctions de coordination.

Par exemple, dans la phrase *opacité pulmonaire en projection du lobe supérieur droit et d'aspect alvéolaire avec signe de nécrose*, une partie de la structure *opacité pulmonaire d'aspect alvéolaire avec signe de nécrose* a été effacée;

- l'utilisation d'adjectifs possessifs référençant un composant d'une autre structure.

Par exemple, ***La lobaire supérieure droite,*** *notamment au niveau de la coupe 13, présente une amputation de* ***sa*** *lumière, pouvant correspondre au bourgeon décrit à l'endoscopie.*

- les structures comparatives : il est fréquent dans des phrases comparatives d'effacer l'élément sur lequel porte la comparaison.

Par exemple, *la tumeur est plus grosse que sur la radio du 30.10.87* qui peut être rétablie en *la tumeur est* ***de taille*** *plus grosse que* ***la taille de la tumeur*** *de la radio du 30.10.87;*

- les structures enchâssées : *Stéphane demande à Médor d'aboyer* qui peut être rétablie en *Stéphane demande à Médor que Médor aboie*, et non en *Stéphane demande à Médor que Stéphane aboie.*

- effacement des composants peu informatifs

- effacement des compléments implicites de certains composants : *tomodensitométrie thoracique découverte à l'occasion d'une fièvre* où la

personne agent de la découverte - en l'occurrence un radiologue - de *découverte* est effacé;

- effacement d'opérateurs : *le produit a réagi 10 minutes* au lieu de *le produit a réagi pendant 10 minutes;*

- effacement de classifieurs : *aime a quatre lettres* au lieu de *le mot aime a quatre lettres.*

2 - un composant d'une structure peut être anaphorique lorsqu'il est nécessaire, pour lui donner une interprétation, de se reporter à un autre composant d'une autre structure [MILN82] [RODE85]. Nous appelons *interprétant* le composant auquel on est renvoyé par l'anaphorique. Cet interprétant se trouve soit au travers d'un substantif, ce que nous appelons anaphore nominale, soit au travers d'un pronom, ce que nous appelons anaphore pronominale.

- on détecte plusieurs types d'**anaphores nominales**

- au travers de répétitions de composants : ***opacités*** *alvéolaires en projection de la lingula, qui est le siège d'une rétraction modérée;* ***ces opacités...***

- au travers d'une inclusion des attributs sémantiques d'un premier composant d'une structure dans un second composant d'une autre structure, de plus le second composant s'avère avoir des attributs sémantiques dépendants : *aspect TDM en faveur d'****un cancer*** *de siège périphérique. Extension de* ***la lésion*** *au niveau médiastinal et pédiculaire.*

- anaphores présuppositionnelles : les attributs sémantiques de deux composants sont identiques, et peuvent, bien qu'indépendants, référer à la même entité : ***processus expansif ganglionnaire****, partiellement nécrosé, intéressant la chaîne médiastinale droite et surtout la chaîne paratrachéale droite.* ***L'adénopathie*** *refoule la veine cave supérieure...*

- **anaphores pronominales :** *mise en évidence au niveau de la loge latéro-trachéale droite d'****opacités****.* ***Elles*** *mesurent environ 10 mm de diamètre.*

3 - déplacement de composants ou de structures

Nous utilisons le terme *déplacement* dans un sens très général, nous entendons par *déplacement* toute permutation de composants ou structures ordonnés par rapport à un ordre fixé a priori.
Par exemple, *Dans cet examen, on remarque une extension ganglionnaire* présente un déplacement de *Dans cet examen.*

4 - portée des opérateurs

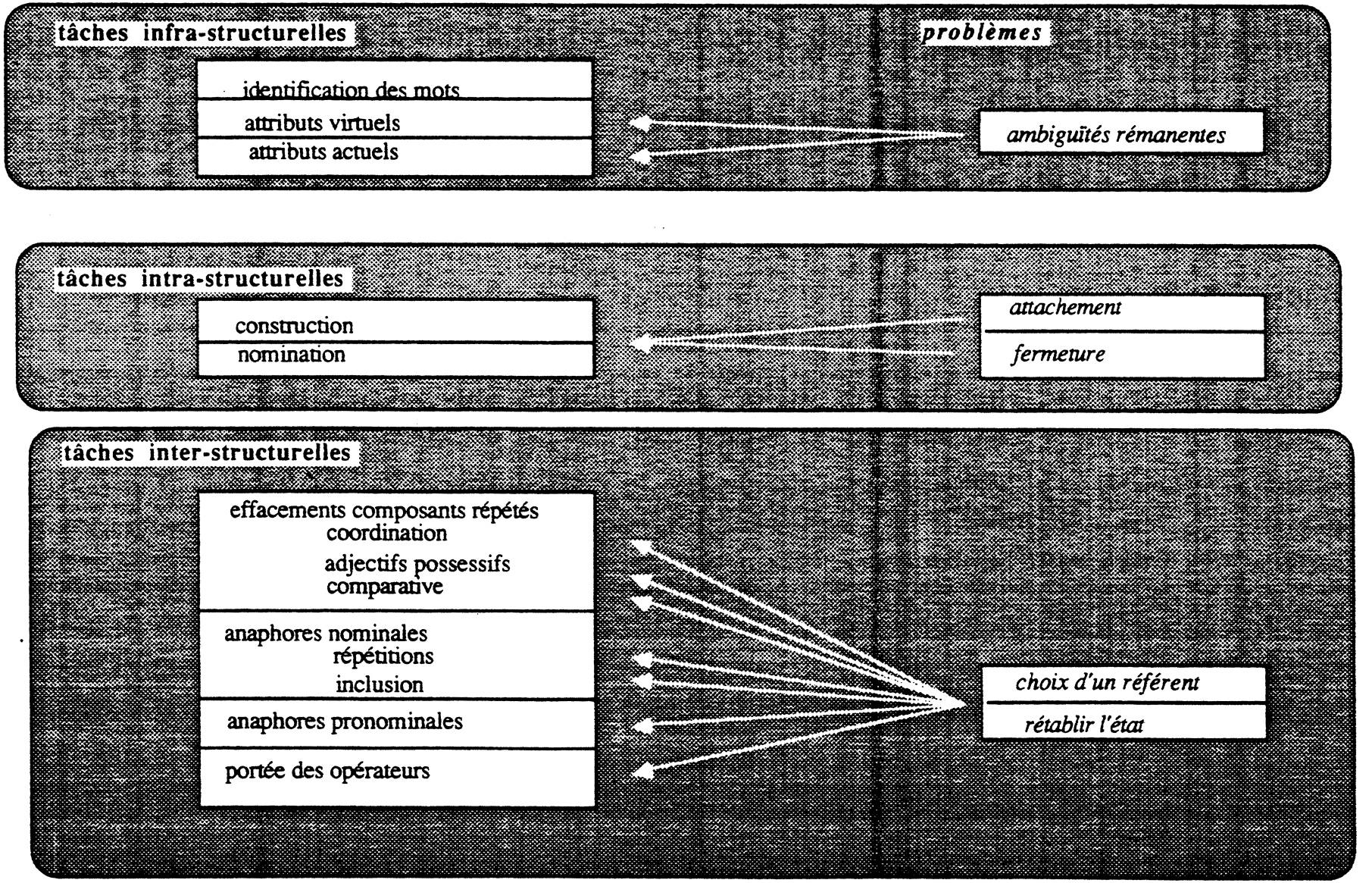
Il s'agit là de regarder la portée de certains opérateurs notamment la négation ou la probabilité.
Par exemple, *Sonia ne mange pas une glace au chocolat* présente un problème au niveau de la négation puisqu'on ne sait pas a priori si Sonia ne mange pas de glace du tout, ou bien si elle mange une glace à un autre parfum.

En plus de l'identification de traces de ces tâches inter-structurelles, et de recherche de candidat(s) éventuel(s) dans une autre structure, nous devons par ailleurs résoudre **deux problèmes** :
1 - parmi les candidats possibles à l'établissement d'un lien entre eux et une structure présentant une trace de tâche inter-structurelle, il faut décider d'une solution;
2 - enfin, il faut rétablir le mieux possible l'état "avant-transformation" des structures, ce qui veut dire dans ce cas créer des liens entre les différentes structures.

3.3.3. les tâches nécessaires dans RIME

Après avoir donné une liste des phénomènes linguistiques que l'on peut trouver dans tout traitement de la langue naturelle, nous voulons indiquer celles qui doivent être intégrées dans RIME (voir la figure 6 résumant les tâches et les problèmes traités dans RIME). Nous verrons dans le chapitre 5 comment elles sont traitées au travers des fonctionnalités de notre système.

figure 6 : les tâches dans RIME



Avant toute chose, nous devons montrer les spécificités du traitement de la langue naturelle de RIME. Ce sont ces spécificités qui distinguent un traitement de la langue naturelle par rapport à tout autre traitement, et qui ainsi autorisent à privilégier des tâches par rapport à d'autres, voire de décider de ne pas traiter certaines d'entre elles. Dans RIME, tout notre traitement est basé sur deux hypothèses fondamentales :

- **les textes traités sont écrits à partir d'un vocabulaire très technique et peu ambigu**, et de ce fait nous verrons par exemple que nous ne mettrons pas en œuvre des outils de résolution d'ambiguïtés très lourds;
- si l'on juge de l'objectif de traduction sémantique de notre application, les quatre niveaux d'analyse (morphologique, syntaxique, sémantique et pragmatique) sont bien entendu corrélés, mais surtout hiérarchisés dans leur degré d'importance, dans le sens où **l'aspect sémantique du problème est primordial** ici. Nous reviendrons par la suite sur cette hypothèse. Par ailleurs, nous ne traitons pas dans notre application le niveau pragmatique à part entière, mais par contre nous incluons les connaissances de ce niveau dans le processus sémantique.

3.3.3.1. les tâches infra-structurelles dans RIME

1 - tout comme les deux autres tâches infra-structurelles, **l'identification des mots** est une fonction nécessaire à tout traitement de la langue naturelle. Dans notre approche, tous les mots sont stockés dans un lexique offrant une description arborescente des mots [PALM81], lexique que nous décrivons par la suite. Nous verrons ci-après quelles propositions sont possibles dans notre approche face à des mots non consignés dans le lexique.

Nous ne traitons pas dans RIME le problème lié à d'éventuelles identifications ambiguës de mots : en effet si nous reprenons l'hypothèse de base de tout notre processus de traduction qui nous dit que les textes traités sont écrits sur du vocabulaire très technique et de ce fait peu ambigu, les identifications ambiguës de mots sont des cas trop rares pour exiger l'utilisation d'outils déterministes lourds de résolution.

2 - **la déduction des attributs virtuels** de chaque mot doit être envisagée sous deux angles : tout d'abord d'un point de vue statique, c'est-à-dire la déduction des attributs virtuels des mots du lexique donc identifiés, ensuite d'un point de vue dynamique c'est-à-dire la déduction des attributs virtuels des mots non consignés dans le lexique.
D'un point de vue statique, toutes les formes et leurs attributs sont stockés dans un lexique vérifiant les fonctionnalités définies dans le chapitre 4. Ainsi pour tout mot reconnu comme une entrée dans le lexique, tous ses attributs sont connus.
D'un point de vue dynamique, nous avons vu que l'initialisation syntaxique du lexique permettait un certain nombre de déductions syntaxiques automatiques [BERR & PALM86]. Sémantiquement aucun outil automatique n'est par contre envisageable. Ceci étant dit il est possible comme nous le montrons dans le chapitre 4 de fournir à l'expert chargé de la mise à jour du lexique un outil d'aide à la décision.

Dans la phrase *condensation pulmonaire du lobe supérieur droit*, nous déduisons les attributs suivants :

condensation

attributs morphologiques *féminin singulier*

attributs syntaxiques *substantif commun*

attributs sémantiques *([a_pr_val, densité, augmenté], signe)*

pulmonaire

attributs morphologiques *masculin/féminin singulier*

attributs syntaxiques *adjectif qualificatif*

attributs sémantiques *([poumon], const_org)*

du

attributs morphologiques *masculin singulier*

attributs syntaxiques *préposition*

attributs sémantiques *([],creux)*

lobe

attributs morphologiques *masculin singulier*

attributs syntaxiques *substantif*

attributs sémantiques *([lobe], détail)*

droit

attributs morphologiques *masculin singulier*
attributs syntaxiques *adjectif qualificatif*
attributs sémantiques *([droit], val_qual_abs)*

3 - **la déduction des attributs actuels** de chaque mot à partir de ses attributs virtuels peut apparaître a priori comme une triple tâche puisque les attributs des mots appartiennent aux trois niveaux morphologique, syntaxique et sémantique. Cependant si l'on reprend l'hypothèse de base de notre processus de traduction qui nous dit que l'aspect sémantique du problème est primordial ici, nous ne développons, à ce niveau infra-structurel, des outils de niveaux morphologique et syntaxique que dans un aspect d'aide ou de simplification sémantique.
Dans un premier temps, nous procédons à un rapide "dépoussiérage" morpho-syntaxique, c'est-à-dire que nous élaguons les impossibilités morpho-syntaxiques évidentes, en ne regardant au niveau de chaque mot que son contexte immédiat gauche et droit. Pour cela, nous utilisons un filtre syntaxique rapide et performant sous la forme d'une matrice de précédence [PALM 88] que nous présentons plus en détail dans la partie 5.3 traitant l'analyse syntaxique. A la suite de ce filtrage morpho-syntaxique, certains mots présentent encore des ambiguïtés morpho-syntaxiques, et nous ne pouvons pas envisager leur résolution systématique. Cependant, nous verrons en analysant le traitement des tâches intra ou inter-structurelles, que, par exemple, la détection de certaines transformations structurelles est de nature syntaxique, et que pour ce type de traitement, certaines occurrences syntaxiques particulières nous intéressent. Aussi pour les mots syntaxiquement ambigus, qui sont des représentants potentiels de telles occurrences, nous devons prévoir des outils permettant la levée de ces ambiguïtés. Pour cela nous utilisons des schémas d'ambiguïtés que nous décrivons également dans la partie traitant l'analyse syntaxique.

Par ailleurs, les polysémies pures sont très rares dans le vocabulaire très technique de notre application. Nous n'envisageons pas d'outils de levée des ambiguïtés polysémiques, et nous gardons a priori toutes les solutions sémantiques potentielles.

Dans la phrase *condensation pulmonaire du lobe supérieur droit*, nous simplifions via le filtre syntaxique les attributs de *pulmonaire*

attributs morphologiques *féminin singulier*

attributs syntaxiques *adjectif qualificatif*

attributs sémantiques *([poumon], const_org)*

3.3.3.2. les tâches intra-structurelles dans RIME

Que l'on considère la tâche de construction de structures ou bien la tâche de nomination des structures construites, il faut fixer pour les tâches intra-structurelles les niveaux de construction ou de nomination syntaxique et sémantique qui nous intéressent dans notre application. Pour cela, rappelons une de nos deux hypothèses de travail : si l'on considère l'objectif de traduction sémantique de notre application, les différents niveaux d'analyse sont hiérarchisés dans leur degré d'importance, dans le sens où l'aspect sémantique du problème est primordial ici. En ce qui concerne les tâches intra-structurelles, il faut fixer le niveau d'intervention de structures syntaxiques dans le processus de traduction.

Pour cela nous pouvons revenir sur la description de la sémantique véhiculée dans les comptes rendus médicaux : nous trouvons dans les comptes rendus médicaux des faits médicaux, et des articulateurs entre ces faits médicaux. Les faits médicaux sont des groupes de mots à attribut sémantique complet dans le sens où la sémantique qu'ils véhiculent est complètement connue. Les articulateurs sont des mots ou des groupes de mots à attribut sémantique incomplet dans le sens où la sémantique qu'ils véhiculent n'est pas complètement connue, et que seul leur contexte nous permet de la connaître.

Par exemple, dans *extension ganglionnaire médiastinale au niveau du groupe de la chaîne para-trachéale droite*, nous relevons les faits médicaux *extension*

ganglionnaire médiastinale, et *groupe de la chaîne para-trachéale droite*; par ailleurs nous relevons l'articulateur *au niveau du*.

Si on examine la représentation syntaxique des faits médicaux et des articulateurs, les faits médicaux sont représentés par des syntagmes nominaux complexes, et les articulateurs par des prépositions qui ne sont pas de type *de* ou bien des syntagmes verbaux qui peuvent éventuellement être nominalisés (par exemple, *déviation* est une nominalisation du verbe *dévier*). Cette brève présentation nous permet de montrer quelle aide syntaxique peut être donnée au niveau intra-structurel : en effet, connaissant syntaxiquement une partie des besoins du processus sémantique, nous pouvons faire des propositions de construction et de nomination syntaxiques au processus sémantique, en fournissant une décomposition des phrases en syntagmes nominaux, syntagmes verbaux, ...

1 - la construction et la nomination de structures

Sémantiquement, nous voulons construire et nommer des structures correspondant au modèle sémantique décrit précédemment : nous voulons extraire les lésions, les signes, les diagnostics, ... Schématiquement, nous voulons extraire et construire toute structure correspondant à une règle de la grammaire du modèle sémantique, et nommer toute structure extraite de la même façon que la partie gauche de la règle de la grammaire permettant sa construction. Cette contruction est fondée sur l'utilisation d'un système de réécriture que nous décrivons ultérieurement.

De manière très générale, ce processus travaille sur des décompositions des phrases en groupes de mots, et il se doit de modeler ces groupes de mots de façon à répondre aux critères du modèle sémantique. Pour aider ce processus sémantique, le processus syntaxique fait une première proposition de décomposition des phrases en structures intéressantes syntaxiquement et sémantiquement : les syntagmes nominaux, les syntagmes verbaux et les syntagmes prépositionnels. Ce travail syntaxique s'effectue au travers d'une analyse de surface des textes, analyse visant à une reconnaissance de certaines constructions syntaxiques spécifiques. Nous reviendrons là-dessus dans la suite

du chapitre. Travaillant sur cette première décomposition, le processus sémantique la validera plus ou moins partiellement, proposera à son tour des découpages partiels, jusqu'à trouver une interprétation cohérente.
Par exemple, dans la phrase *extension ganglionnaire au niveau du groupe de la chaîne para-trachéale droite*, le découpage syntaxique donne
extension ganglionnaire comme *syntagme nominal;*
au niveau du comme *syntagme prépositionnel;*
groupe de la chaîne para-trachéale droite comme *syntagme nominal,* nous verrons ultérieurement que les syntagmes nominaux sont à nouveau décomposés en syntagmes nominaux simples séparés par des prépositions de type *de*, ce qui donne ici un syntagme formé de *groupe* et de *chaîne para-trachéale droite.*
Le processus sémantique regroupe les syntagmes de la façon suivante :
extension ganglionnaire comme *un fait médical* qu'il nomme *signe;*
au niveau du comme *un articulateur;*
groupe de la chaîne para-trachéale droite comme *un fait médical* qu'il nomme *localisation*.

2 - les problèmes liés aux tâches intra-structurelles

L'attachement se présente sous la forme de triplets XYZ où Z peut se rattacher à X ou à Y. La détection se fait automatiquement dans le processus sémantique dans toute structure proposée possédant au moins trois composants. En effet quand les deux premiers d'entre eux sont reliés, se pose le problème du rattachement du troisième à l'un des deux. Après avoir vérifié les différentes possibilités, le problème demeure si l'attachement de Z reste ambigu entre X et Y, sinon le problème est résolu. Dans le cas d'une ambiguïté de rattachement, nous faisons le choix - qui peut s'avérer être une erreur par la suite du traitement - de rattacher au dernier constituant, soit à Y dans notre exemple. Ce choix est complètement arbitraire, et nous verrons comment en cas d'invalidation d'un tel choix le processus sémantique revient sur ses décisions.

La fermeture se traite de la même façon que l'attachement.

Par ailleurs, nous ne traitons pas le cas de nomination ambiguë de structures que, sans doute grâce au faible nombre de polysémies dans nos textes, nous n'avons pas encore détecté dans les comptes rendus médicaux.

3.3.3.3. les tâches inter-structurelles dans RIME

Les tâches traitées jusqu'à maintenant permettent la création des structures autonomes, mais ne permettent pas la mise en évidence de dépendances entre les différentes structures du document. Le but des tâches inter-structurelles est de connecter ces structures jusqu'alors indépendantes.
Parmi toutes les tâches décrites, nous ne considérons pas certaines d'entre elles : l'effacement des composants peu informatifs, les anaphores présuppositionnelles, car ce sont des phénomènes linguistiques peu intéressants dans notre application. Toutes les informations manquantes devront être retrouvées si besoin est lors de la phase d'interrogation. Nous ne traitons pas non plus les effacements d'éléments dans des structures enchâssées, car ce sont des effets de style que nous ne rencontrons pas dans les comptes rendus médicaux. Finalement, nous ne traitons pas non plus les déplacements de structures qui sont des effets de style très rares dans les documents que nous traitons.
Il reste ainsi à traiter dès l'indexation : l'effacement d'éléments dans les cas de coordination, d'utilisation d'adjectifs possessifs, et de comparatives, les anaphores nominales dues aux répétitions ou à l'inclusion, les anaphores pronominales, et la portée de certains opérateurs. Nous allons voir ci-dessous que nous ne les traitons pour la plupart d'entre elles que partiellement.
Nous verrons également que le processus syntaxique aide généralement à la détection de ces tâches, mais que seul le processus sémantique est à même de décider de la nécessité d'un travail inter-structurel : nous disons par la suite que le processus syntaxique ou le processus sémantique *signalent* une tâche. Ce signalement, lorsqu'il est syntaxique, doit être *confirmé* par le processus sémantique pour que le travail complet quant à la tâche repérée soit mis en

œuvre; nous entendons par là tout d'abord la recherche de composants référents potentiels dans d'autres structures, le choix de l'un d'entre eux, et finalement la remise dans l'état "avant transformation".
Avant de présenter chacune des tâches, nous voudrions donner une idée générale de ce type de traitement dans notre application.
Tout d'abord, la détection de toute tâche est *signalée* syntaxiquement ou sémantiquement, mais seul le processus sémantique peut *confirmer* une occurrence réelle.
Ensuite le processus sémantique doit chercher les composants référents potentiels dans d'autres structures et valider l'un d'entre eux. Pour cela, le processus sémantique ne cherche pas toutes les solutions possibles, mais n'envisage que la plus probable qui s'avère généralement être la plus proche dans le contexte gauche de la tâche détectée. Dans certains cas, nous verrons que le processus sémantique attend une *validation* de son choix par le processus syntaxique. Tout comme pour l'attachement et la fermeture, ce choix est complètement arbitraire et dans le cas d'invalidation par la suite du traitement, le processus devra revenir sur cette décision. Dans notre contexte médical, il s'avère en fait très rare d'avoir à revenir sur ces décisions.

1 - effacements de composants répétés

La détection des problèmes d'effacement d'éléments répétés par **coordination** entre structures est signalée par le processus syntaxique dans certains cas particuliers tels qu'une conjonction de coordination suivie d'une préposition, ou encore une virgule suivie d'une préposition. Nous ne traitons pas d'autres cas de coordination actuellement, car ce sont les seuls cas recensés dans nos comptes rendus médicaux. Ce sont cependant des cas très fréquents. Nous procédons alors comme décrit ci-dessus, sans aucune validation syntaxique.

Nous traitons de la même façon les structures **comparatives** que le processus syntaxique signale lors d'apparition de séquences *moins ... que, plus ... que.*

Les références par **adjectifs possessifs** sont signalées syntaxiquement; le processus sémantique doit ensuite trouver le premier référent satisfaisant l'anaphore.

2 - les anaphores nominales

Les anaphores nominales **par répétition de constituants** ne sont identifiables dans nos documents que si elles s'expriment au travers d'adjectifs démonstratifs ou de certains adjectifs indéfinis. Dans les autres cas il est beaucoup trop difficile d'arriver à une résolution d'anaphore. La détection de ces anaphores est donc syntaxique, et la recherche de l'interprétant est ici beaucoup plus simple, puisqu'il s'agit de trouver dans le contexte gauche de la tâche détectée sa propre occurrence.

L'anaphore **par inclusion** n'est détectable que sémantiquement, lors d'une occurrence d'un mot dit insuffisant sémantiquement. Il s'agit dès lors de trouver dans le contexte gauche de l'anaphore un mot permettant de compléter les attributs sémantiques de l'anaphore.

3 - les anaphores pronominales

Leur détection est purement syntaxique; le processus sémantique doit, après avoir trouvé le premier interprétant satisfaisant l'anaphore, demander une validation au processus syntaxique.

4 - portée des opérateurs

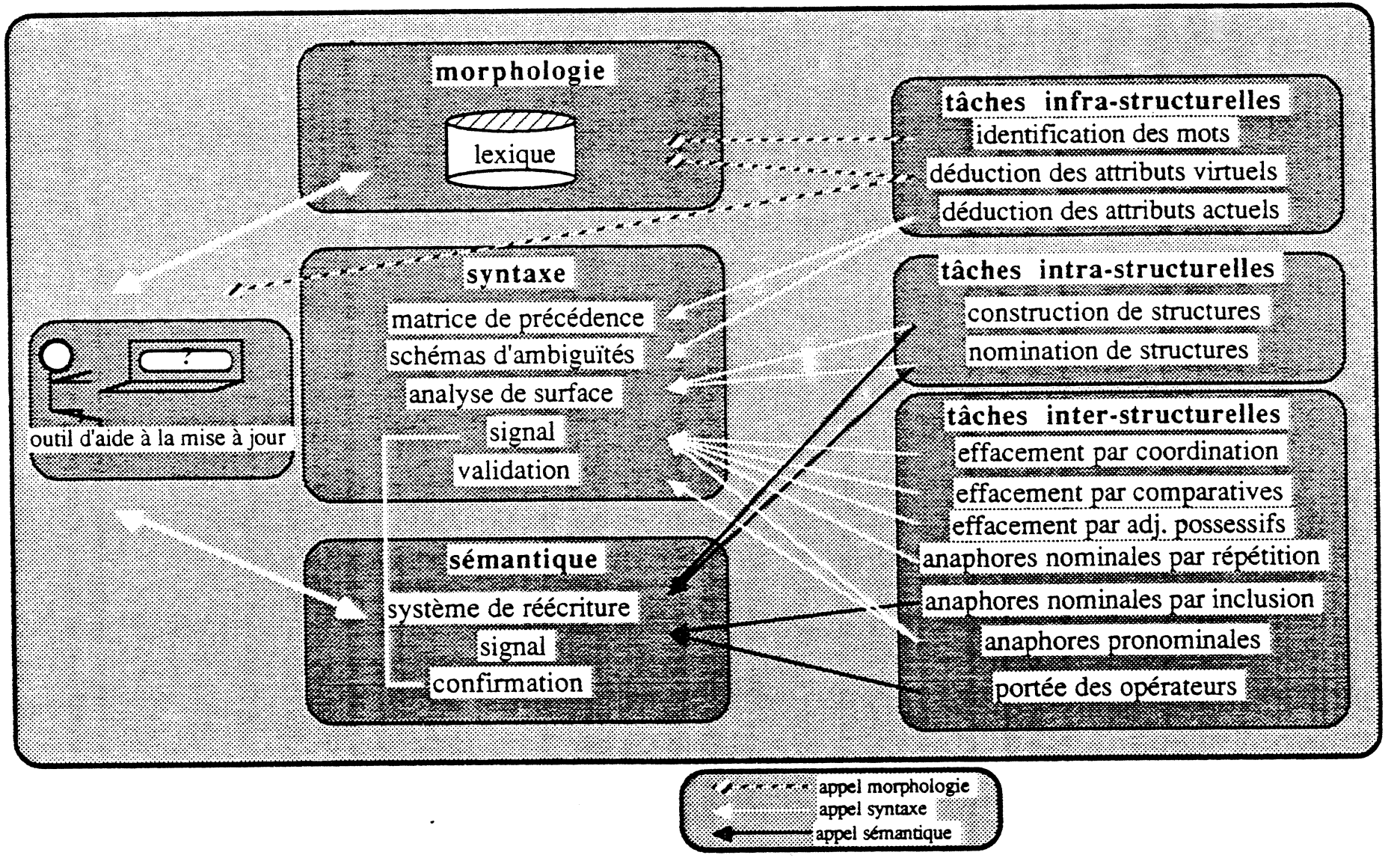
Ce problème est traité de manière simplifiée ici, en ne faisant porter les opérateurs - négation, par exemple - sur des syntagmes dont la tête est un fait médical. Notre approche consiste à faire systématiquement répercuter l'opérateur sur la tête de syntagme au travers de certaines règles de la grammaire. Nous reviendrons plus en détail sur cet aspect dans le chapitre suivant.

3.3.4. conclusion

La figure 7 montre la répartition des différentes tâches nous intéressant dans RIME, au travers des trois processus que sont la morphologie, la syntaxe et la sémantique. Rappelons que nous n'avons pas de niveau pragmatique explicite, mais que par contre nous incluons des connaissances de niveau pragmatique dans le processus sémantique, notamment pour la résolution des tâches inter-structurelles.
Ces trois processus interagissent pour la résolution de certaines tâches. Comme nous l'avons déjà indiqué, dans tout traitement de la langue naturelle, il faut d'une part fixer la profondeur demandée à chaque processus - ce que nous venons de faire au travers des tâches -, et d'autre part fixer la gestion de l'utilisation des différentes sources de connaissances. Pour gérer cette coopération entre processus, nous avons décidé de définir **un processus de coopération indépendant**, permettant ainsi un contrôle autonome des processus constructifs. Ce processus possède donc la connaissance du pilotage de tout le processus de traduction de RIME : il sait quel processus constructeur appeler, enregistre les résultats de chaque processus appelé, enchaîne les processus constructeurs nécessaires à la réalisation de toute tâche, ...

Nous allons dans les chapitres suivants décrire tout le mécanisme de traduction de RIME, en présentant le lexique dans le chapitre 4, ainsi que, dans le chapitre 5, les quatre processus constructeurs que sont :
- le processus morphologique (partie 5.2.);
- le processus syntaxique (partie 5.3.);
- le processus sémantique (partie 5.4.);
- le processus de coopération (partie 5.5.).

figure 7 : la répartition des tâches dans RIME



Chapitre 4
Le lexique de RIME

Le but final de notre travail est de traduire de la langue naturelle issue de comptes rendus médicaux dans le modèle sémantique défini dans le chapitre 3. Considérant les choix faits quant à ce modèle sémantique, nous voulons présenter dans ce chapitre le lexique de RIME. Il s'agit de décrire le vocabulaire autour duquel s'articule notre processus de traduction.

En effet, après avoir présenté la grammaire, qui correspond au résultat du processus, il nous faut maintenant présenter les données du processus qui sont constituées par les mots des comptes rendus médicaux : pour traiter ces textes, il est nécessaire de connaître pour chaque mot extrait l'information indispensable au processus de traduction dans le modèle sémantique. Cette information qui est à la fois syntaxique et sémantique est entièrement stockée dans le lexique que nous présentons ici.

Nous montrons quelles informations au niveau des mots sont nécessaires pour le traitement de traduction : dans un premier temps nous présentons l'information sémantique liée à chaque mot dans RIME, et dans un second temps nous abordons le problème de l'information syntaxique associée à chaque mot. Finalement, nous montrons comment le lexique de RIME peut être mis à jour.

4.1. aspect sémantique du vocabulaire

Nous devons entre autre connaître pour chaque mot reconnu dans un compte rendu médical sa traduction selon le modèle sémantique. Cette traduction sémantique est en fait un couple que nous appelons *expression sémantique* et qui est formé d'une part d'un *trait sémantique* et d'autre part d'une *catégorie sémantique* :

- le trait sémantique d'un mot définit son interprétation sémantique dans le modèle. Pour un mot du vocabulaire de base, son trait sémantique est le mot lui-même (d'où la dénomination *vocabulaire de base*). Pour un mot du vocabulaire complexe, sa signification dans le modèle est exprimée par une expression du modèle : une arborescence. Cette expression peut être résolue (les feuilles sont des mots du vocabulaire de base) ou non résolue (les feuilles sont des méta-symboles), par exemple :

poumon, qui est un mot du vocabulaire de base, a donc pour trait sémantique *[poumon]* ;

hypertrophie, qui est du vocabulaire complexe, a pour trait sémantique une expression résolue ***[a_pr_val**, volume, augmenté]* ;

dévier, qui est également du vocabulaire complexe, a pour trait sémantique une expression non résolue ***[dû_à**, **[p_sur**, **[a_pr_val**, position, déviation]*, *localisation]*, *lésion]*

- la catégorie sémantique correspond à un symbole terminal de la grammaire. La catégorie sémantique permet de rattacher un mot à une classe de concept du modèle, et par là, de l'interpréter correctement dans son contexte. Par exemple, le mot *poumon* a la catégorie sémantique *const_org* .

Nous allons maintenant revenir sur ces deux aspects des expressions sémantiques. Dans un premier temps, nous expliquons les différentes catégories sémantiques de notre modèle en en donnant une liste ainsi que des exemples. Dans un second temps, nous décrivons les traits sémantiques du vocabulaire en fixant tout d'abord le vocabulaire de base, et en montrant

ensuite comment le vocabulaire complexe s'exprime au travers de la grammaire, des catégories sémantiques et du vocabulaire de base.

4.1.1. les catégories sémantiques

Avec l'aide d'experts du domaine médical traité, nous avons dressé la liste des catégories sémantiques :

- Tous les représentants de constituants de l'organisme ont pour catégorie sémantique la catégorie *const_org*. On y distingue deux sous-catégories : *détail* et *org*. Les termes comme *lobe*, que l'on retrouve dans différentes parties de l'organisme et pour lesquels il faut déduire la partie de l'organisme désignée, ont pour catégorie sémantique la sous-catégorie *détail* ; par exemple *lobe du poumon*. Par ailleurs, on trouve dans la sous-catégorie *org* les termes désignant un constituant de l'organisme, par exemple *poumon* ;

- La catégorie *creux* regroupe les mots sémantiquement vides de notre modèle. Ces mots sont lors de la traduction ignorés sémantiquement, mais ils peuvent par contre apporter des renseignements syntaxiques intéressants. On trouve dans cette catégorie des termes tels que *type* ou *zone* ;

- La catégorie *degré* contient les termes comme *pas du tout, un peu, beaucoup* ;

- La catégorie *examen* regroupe les termes relatifs aux examens et à leurs techniques : *fibroscopie, tomodensitométrie* ;

- La catégorie *fonction* rassemble les termes médicaux représentant les fonctions médicales telles que la *vascularisation* ou la *ventilation* ;

- La catégorie *lésion* comporte les lésions au sens médical : *nécrose, goître, hernie* ;

- La catégorie *position* regroupe les termes tels que la *localisation* ;

- La catégorie *signe* se divise en deux sous-catégories : *sgn* et *car_phy*. Par exemple, *obstruction, liseré* ou *masse* représentent des *sgn*, et *hauteur* ou *volume* des car_phy ;

- Les valeurs quantitatives notées *val_quan* telles que *0, 1, 2* ou encore *moitié, quart, ...* se divisent dans deux sous-catégories : *val_quant_absolue* dans laquelle on retrouve les chiffres *0,1,2, ...* ainsi que *tout, peu, plusieurs*, et d'autre part *val_quant_relative* pour *tiers, moitié, quart, ...* et tous les adjectifs numéraux ordinaux tels que *premier, dixième, ...* ;

- La catégorie *val_qual_abs* regroupe des termes tels que *convexe, congénital, carcinomateux.*

Nous pouvons en conclusion de cette partie définir en extension l'ensemble *catégories_sémantiques* de notre modèle :

catégories_sémantiques = {const_org, creux, degré, examen, fonction, lésion, position, signe, val_quan, val_qual_abs}

Nous appelons *CAT_SEM(m)* l'ensemble des catégories sémantiques du mot *m*.

Comme nous l'avions déjà remarqué dans la partie 3.1.2, il existe une inclusion entre les parties gauches des règles de la grammaire. Cette inclusion se répercute sur les catégories sémantiques (propriétés que nous utilisons dans le chapitre 5) de la façon suivante : *une partie gauche AA inclut une catégorie sémantique a* si et seulement si il existe une partie gauche *A* telle que *AA inclut A*, et une règle telle que *A ::= a*. Par exemple, *QUAL* inclut *val_qual*, car d'une part *QUAL* inclut *VAL_QUAL*, et d'autre part il existe une règle de la grammaire telle que *VAL_QUAL ::= val_qual.*

4.1.2. les traits sémantiques du vocabulaire

Après avoir présenté les différentes catégories sémantiques autorisées, nous devons maintenant montrer les différents traits sémantiques admis dans notre modèle. Rappelons l'exemple de *poumon* qui a comme expression sémantique *(poumon, const_org)*, et dont le trait sémantique est *poumon*. Nous allons dans cette partie montrer les différentes façons d'exprimer l'élément de gauche d'une expression sémantique, que nous appelons trait sémantique.

Fondamentalement, le vocabulaire des comptes rendus médicaux se divise en deux grandes familles sémantiques :

- les faits médicaux tels que *foie, cancer ;*
- les articulateurs entre les faits médicaux tels que *en projection de, déborder* ou *dévier.*

Par exemple, dans la phrase *le foie déborde du gril costal*, les faits médicaux *foie* et *gril costal* sont reliés entre eux par *déborde.*

Si nous considérons individuellement ces mots, nous remarquons que les faits médicaux sont sémantiquement autonomes, alors que les articulateurs sont sémantiquement dépendants des faits médicaux, c'est-à-dire qu'ils ne peuvent s'employer sans fait médical dans leur contexte immédiat : *foie* ou *cancer* se suffisent sémantiquement à eux-mêmes, par contre lors de l'utilisation de *en projection de* il est nécessaire de signaler *ce qui* est en projection de *quoi.* Ainsi par la suite, les traits sémantiques des faits médicaux sont dits *complets*, alors que ceux des articulateurs sont eux dits *incomplets.*

En fait, les faits médicaux se classent en deux grandes catégories :

- le vocabulaire de base qui regroupe les faits médicaux de niveau sémantique le plus bas dans notre modèle. Par exemple, *poumon, cancer* ;
- le vocabulaire à trait sémantique complexe et complet, qui regroupe les faits médicaux de niveau sémantique plus élevé et dont les traits sémantiques s'expriment au travers du vocabulaire de base d'une part, et des règles de la grammaire du modèle sémantique d'autre part. Par exemple, *hypertrophie* qui s'exprime en utilisant d'une part les faits de base *volume* et *augmenté*, le trait sémantique de *hypertrophie* est ***[a_pr_val**, volume, augmenté].*

Les articulateurs sont représentés par des traits sémantiques complexes et incomplets. Par exemple, *dévier* a pour trait sémantique ***[dû-à***, ***[p-sur***, ***[a_pr_val***, *position, déviation]*, *localisation]*, *lésion]* où l'on ne sait pas a priori quel *localisation* ou quelle *lésion* entourent le mot *dévier*. On retrouve ici une notion très voisine des frames, où les catégories sémantiques terminales jouent ici le même rôle que les slots, qui peuvent être instanciés par d'autres frames du vocabulaire de base ou complexe.

On distingue donc trois sous-ensembles dans le lexique des faits médicaux :

- le vocabulaire de base ;
- le vocabulaire complexe et complet ;
- le vocabulaire complexe et incomplet.

4.1.2.1. le vocabulaire de base

Les mots du vocabulaire de base sont les mots représentant les faits médicaux sémantiquement élémentaires de notre modèle. Tous ces mots du vocabulaire de base ont un trait sémantique dit simple, puisqu'il s'agit d'eux-mêmes :
poumon --> ([poumon], const_org)
cancer --> ([cancer], lésion)
concave --> ([concave], val_qual_abs)
échographie --> ([échographie], examen)
De manière plus générale, tout mot *m* du vocabulaire de base a pour expression sémantique *(m, catégorie_sémantique)*.

Le problème le plus délicat pour le vocabulaire de base est en fait d'en dresser la liste : en effet ceci fixe par là-même le niveau de compréhension de tout notre système puisque d'une part c'est-à-partir du vocabulaire de base que s'exprime tout le reste du vocabulaire, et que d'autre part tout notre processus de traduction a pour but de générer des arborescences dont les feuilles sont des mots du vocabulaire de base. Cette liste du vocabulaire de base a été dressée, pour chacune des catégories sémantiques de notre modèle, par les médecins, experts du domaine.

Par la suite, nous appelons *vb* l'ensemble des mots du vocabulaire de base, c'est-à-dire dont le trait sémantique est eux-mêmes.

4.1.2.2. le vocabulaire complexe et complet

Nous retrouvons dans cette catégorie du vocabulaire les faits médicaux d'un niveau de compréhension plus complexe que le vocabulaire de base. Cette complexité s'exprime pour leur trait sémantique par une substitution par un mot du vocabulaire de base, ou par une expression conforme à la grammaire du modèle sémantique. Par exemple :
pulmonaire --> *([poumon], const_org)* où l'on voit que le trait sémantique de *pulmonaire* s'exprime par *poumon ;*
hypertrophie --> *([a_pr_val, volume, augmenté], signe)*
Ces traits sémantiques sont dits *complexes* dans le sens où généralement ils sont constitués d'une arborescence respectant les règles de la grammaire, ils sont également dits *complets* car toutes les feuilles de l'arborescence sont connues et correspondent en outre à des mots du vocabulaire de base.
Par la suite, nous appelons *vcc* l'ensemble des traits sémantiques complexes et complets.

4.1.2.3. le vocabulaire complexe et incomplet

Les articulateurs entre faits médicaux sont représentés par des traits sémantiques complexes et incomplets : *complexes* dans le sens où leur trait sémantique s'exprime au travers d'une expression de notre modèle, *incomplets* dans la mesure où certaines feuilles de leur trait sémantique peuvent ne pas être complètement instanciées et doivent l'être par des faits médicaux repérés dans le contexte de l'articulateur. Cette incomplétude se repère par l'attribution de catégories sémantiques, et non de mots du vocabulaire de base, à certaines feuilles de l'articulateur représenté.

Par la suite, nous appelons *vci* l'ensemble des traits sémantiques complexes et incomplets.

Nous appelons par ailleurs

traits_sémantiques = vb U vcc U vci ;

TRAIT_SEM(m, cat_sém) l'ensemble des traits sémantiques du mot *m*, de catégorie sémantique *cat_sém* ; les éléments de cet ensemble sont des éléments de *traits_sémantiques*.

4.1.3. conclusion

En conclusion, pour tout mot d'un compte rendu médical, nous devons connaître son ou ses expression(s) sémantique(s), c'est-à-dire des couples formés d'un trait sémantique et d'une catégorie sémantique. De manière plus précise, nous désirons ici rappeler toutes les propriétés que nous venons devoir en les formulant dans le langage de modélisation Z_0 :

CAT_SEM

mot ---------->> catégories_sémantiques

TRAIT_SEM

mot X catégories_sémantiques ----------|->> traits_sémantiques

EXP_SEM

mot ---------->> expressions_sémantiques

où *mot* est l'ensemble des mots figurant dans les comptes rendus ;

traits_sémantiques est l'ensemble des traits sémantiques :

traits_sémantiques = vb U vcc U vci ;

catégories_sémantiques est l'ensemble des catégories sémantiques ;

expressions_sémantiques = catégories_sémantiques X traits_sémantiques ;

CAT_SEM est une fonction multivaluée totale (tout mot a au moins une catégorie sémantique) ;

TRAIT_SEM est une fonction multivaluée partielle, car définie sur un sous-ensemble de *mot X catégories_sémantiques* : cet ensemble est réduit aux doublets (m, c) où c ∈ CAT_SEM(m) ;

EXP_SEM est une fonction multivaluée totale dans *expressions_sémantiques.*

Dans ce système redondant de fonctions, on a les contraintes d'intégrité suivantes :

(1) $\forall$ (m, c) $\in$ *mot* $\times$ *catégories_sémantiques*, c $\in$ CAT_SEM(m)

(2) $\forall$ m $\in$ *mot*, EXP_SEM(m) = {(c, v)}

où c $\in$ CAT_SEM(m), et v $\in$ TRAIT_SEM((m,c))

La décomposition des traits sémantiques en trois sous-ensembles correspond à une classification des concepts en faits élémentaires ou complexes. L'assignation d'un mot à l'une de ces catégories ne peut être effectuée que par un spécialiste du domaine ; elle constitue une opération capitale, car elle détermine la finesse du modèle sémantique, qui sera d'autant plus grande qu'il y a plus de traits sémantiques complexes, par rapport aux traits sémantiques simples (identique aux mots).

4.2. aspect syntaxique du vocabulaire

Après avoir montré l'information sémantique à connaître pour chaque mot du vocabulaire médical, nous présentons maintenant les connaissances syntaxiques retenues au niveau de chacun de ces mots. Nous avons, pour cette partie, repris partiellement les principes du modèle linguistique de P.PALMER, décrit dans [PALM88], à l'aide de [GREV80].

Ce modèle présente une classification des mots de la langue en fonction de leur rôle syntaxique ou des différentes positions que peuvent prendre ces mots dans l'ordonancement d'une phrase de la langue française. Ces catégories se nomment des *catégories lexicales* ou encore des *catégories grammaticales.* De plus, pour chaque catégorie grammaticale, un ensemble de *variables grammaticales* permet de préciser un ensemble d'*attributs grammaticaux* ou *valeurs grammaticales* qui constituent les renseignements linguistiques associés aux éléments de ces catégories.

Par exemple, la catégorie grammaticale SUBC contenant les substantifs communs est associée à deux variables grammaticales : le *genre* et le *nombre*

qui peuvent respectivement prendre leurs valeurs dans *{masculin, féminin}* et *{singulier, pluriel}*.

Le modèle syntaxique proposé par P.PALMER dans [PALM88] est particulièrement intéressant pour notre application notamment de par ses qualités de mise en œuvre et de mise à jour aisée (voir également la partie 4.4.3). Nous l'avons de ce fait entièrement intégré dans notre système.

4.2.1. les catégories grammaticales

Nous avons considéré d'une part les catégories grammaticales dites *fermées* dont la liste exhaustive des éléments peut être établie a priori, et dont le rôle syntaxique permet de faciliter l'identification de structures particulières de la langue naturelle (syntagmes nominaux, syntagmes verbaux, ...), et d'autre part les catégories dites *ouvertes* pour lesquelles il est pratiquement impossible de dresser des listes exhaustives. Les catégories fermées regroupent les déterminants (articles, adjectifs déterminatifs), les pronoms (personnels, possessifs, ...), les prépositions et les conjonctions. Les catégories ouvertes comportent les substantifs, les adjectifs qualificatifs, les adverbes et les verbes. Pour les éléments des catégories ouvertes, nous avons déterminé 9 catégories grammaticales :

SUBC substantif commun
SUBP substantif propre
ADJQ adjectif qualificatif
ADVB adverbe
VBIF verbe à l'infinitif
VBPA verbe au participe passé
VBPR verbe au participe présent
VBCJ verbe conjugué
ABRV abréviation et sigle

La liste complète des catégories grammaticales est donnée en annexe 3. Par la suite, nous appelons *catégories_grammaticales* l'ensemble des catégories grammaticales de notre modèle. Nous définissons par ailleurs

CAT_GRAM(m) comme l'ensemble des catégories grammaticales du mot *m* : par exemple, *CAT_GRAM(arbre) = {SUBC}*.

4.2.2. les variables grammaticales

Les variables grammaticales que nous utilisons sont les variables traditionnelles pour le français. Elles sont au nombre de cinq :
- la variable grammaticale *genre* notée GNR, dont l'ensemble des valeurs possibles est *{masculin, féminin, neutre}* ;
- la variable grammaticale *nombre* notée NBR, dont l'ensemble des valeurs possibles est *{singulier, pluriel}* ;
- la variable grammaticale *mode* notée MOD, dont l'ensemble des valeurs possibles est *{indicatif, subjonctif, conditionnel, impératif}*. Les modes infinitif et participe sont représentés par des catégories grammaticales particulières ;
- la variable grammaticale *temps* notée TMP, dont l'ensemble des valeurs possibles est *{présent, imparfait, passé simple, futur}*. Les temps composés ne sont pas analysés en tant que tels mais comme la succession d'un auxiliaire conjugué et d'un participe passé ;
- la variable grammaticale *personne* notée PRS dont l'ensemble des valeurs possibles est *{1ère, 2ème, 3ème}*.

Par la suite, nous appelons *var_grammaticales,* l'ensemble des variables grammaticales :
variables_grammaticales = { GNR, NBR, MOD, TMP, PRS }
Nous appelons *INST(i)*, la liste des valeurs possibles de *i*, *i* ∈ *variables_grammaticales* :
INST(GNR) = {masculin, féminin}
INST(NBR) = {singulier, pluriel}
INST(MOD) = {présent, imparfait, passé simple, futur}
INST(PRS) = {1ère, 2ème, 3ème}
Il faut donc par la suite mémoriser pour chaque *cat_gram* ∈ *catégories_grammaticales* un élément de *variables_grammaticales*$^+$, élément

que nous appelons *VG_CAT(cat_gram)*, correspondant aux variables grammaticales qui lui sont associées. Par exemple, *VG_CAT(SUBC) = {(GNR, NBR)}*. Nous donnons en annexe 3 la liste des catégories grammaticales associées à leurs variables grammaticales.
De la même façon, il faut pour chaque mot *m*, d'une catégorie grammaticale *cat_gram* donnée, connaître les valeurs de ses variables grammaticales, c'est-à-dire l'instantiation pour ce mot *m* de chacun des éléments de *VG_CAT(cat_gram)* : nous appelons cet ensemble *VAL_GRAM(m, cat_gram)*. Par exemple, pour le mot *arbre*, *VAL_GRAM(arbre, SUBC) = {(masculin, singulier)}* avec *VG_CAT(SUBC) = {(GNR, NBR)}*, *masculin* $\in$ *INST(GNR) et singulier* $\in$ *INST(NBR)*.

4.2.3. conclusion

C'est l'ensemble des associations catégorie grammaticale - variables grammaticales qui forme la trame du modèle syntaxique que nous décrivons plus tard. Et c'est à partir de ces associations que l'on pourra définir par exemple un certain nombre de restrictions grammaticales, dont les principales sont les accords en genre, nombre et personne, ou bien que nous repérons dans les textes des entités syntaxiques susceptibles d'aider le mécanisme de traduction : les syntagmes nominaux ou verbaux, ...
Pour cela, à chaque mot d'un compte rendu médical, nous voulons dans une phase syntaxique connaître sa (ses) catégorie(s) grammaticale(s) ainsi que ses (leurs) variables grammaticales. Par exemple, le mot *arbre* est associé à la catégorie grammaticale *SUBC* et aux instantiations *(masculin, singulier)* des variables grammaticales de *VG_CAT(SUBC)*, soit *(GNR, NBR)*.
De manière plus formelle,
$\forall$ *m* $\in$ *mot*,
il faut connaître l'ensemble *EXP_SYNT(m)* tel que
EXP_SYNT(m) = { (cat_gram, inst_var_gram) / cat_gram $\in$ *CAT_GRAM(m) et inst_var_gram* $\in$ *VAL_GRAM(m, cat_gram) }*
Par exemple, *EXP_SYNT(arbre) = { (SUBC, (masculin, singulier)) }*

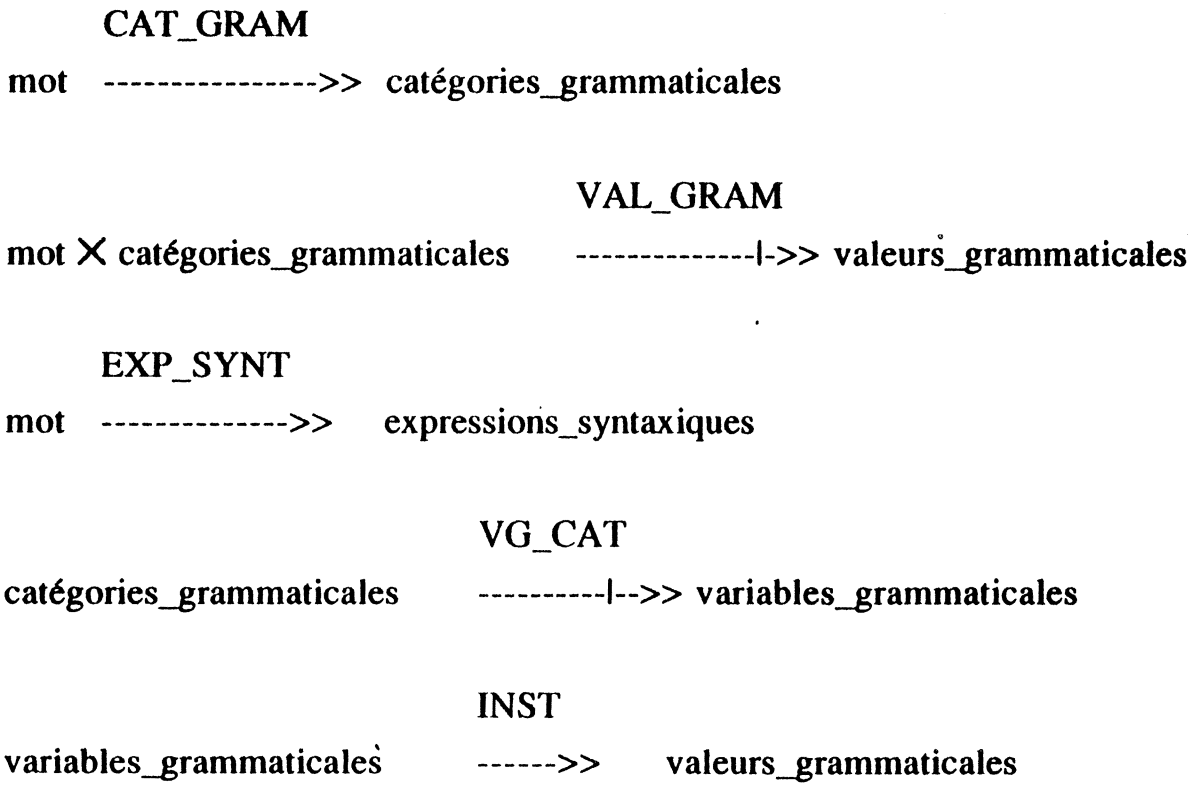
Il est cependant important de préciser qu'une catégorie grammaticale peut avoir plusieurs mêmes ensembles de variables grammaticales (avec des valeurs différentes) associés pour un mot, et qui correspondent à des interprétations grammaticales différentes. Par exemple, *EXP_SYNT(page) = {(SUBC (masculin, singulier)), (SUBC (féminin, singulier)) }* ou encore *EXP_SYNT(rédigions) = { (VBCJ (indicatif, imparfait, 1ère, pluriel)) , (VBCJ (subjonctif, présent, 1ère, pluriel)) }*

Nous définissons par ailleurs l'ensemble *expressions_syntaxiques* tel que

expressions_syntaxiques =

catégories_grammaticales $\times$ *variables_grammaticales*

Le schéma Z_0 ci-dessous représente les possibilités syntaxiques que nous venons de présenter :



où *mot* est l'ensemble des mots figurant dans les comptes rendus ;

catégories_grammaticales est l'ensemble des catégories grammaticales ;

variables_grammaticales est l'ensemble des variables grammaticales associées aux catégories grammaticales ;

valeurs_grammaticales représente l'ensemble des valeurs des variables grammaticales : *valeurs_grammaticales* = {masculin, féminin, singulier, pluriel, indicatif, subjonctif, conditionnel, impératif, présent, imparfait, passé simple, futur, 1ère, 2ème, 3ème} ;

expressions_syntaxiques est l'ensemble des expressions syntaxiques défini précédemment ;

CAT_GRAM est une fonction multivaluée totale (tout mot a au moins une catégorie grammaticale) ;

VG_CAT est une fonction multivaluée partielle : toute catégorie possède 0 ou plusieurs variables grammaticales. Par exemple, la catégorie adverbe ne possède aucune variable grammaticale ;

INST est une fonction multivaluée totale : toute variable possède au moins 1 valeur ;

VAL_GRAM est une fonction multivaluée partielle, car définie sur un sous-ensemble de *mot* $\times$ *catégories_grammaticales* : cet ensemble est réduit aux doublets (m, c), où c $\in$ CAT_GRAM(m) ;

EXP_SYNT est une fonction multivaluée totale dans *catégories_grammaticales* $\times$ *valeurs_grammaticales.*

Dans ce système redondant de fonctions, on a les contraintes d'intégrité suivantes :

(1) $\forall$ (m, c) $\in$ *mot* $\times$ *catégories_grammaticales*, c $\in$ CAT_GRAM(m)

(2) $\forall$ (m, c) $\in$ *mot* $\times$ *catégories_grammaticales*, VAL_GRAM((m,c)) = V,

où $\forall$ v $\in$ V, v $\in$ INST(VG_CAT(c))

(3) $\forall$ m $\in$ *mot*, EXP_SYNT(m) = {(c, V)}

où c $\in$ CAT_GRAM(m), et V $\subset$ INST(VG_CAT(c))

4.3. vue générale du lexique

4.3.1. présentation fonctionnelle

Les informations associées aux mots, et nécessaires pour que la traduction se déroule sans problème relèvent des deux niveaux que nous venons de décrire.

Le lexique doit donc permettre de connaître pour chaque mot extrait d'un compte rendu les informations que nous venons de mettre en exergue.
Pour cela, nous présentons le lexique fonctionnellement sur la figure 8.
Nous avons ajouté la relation *EXP_SYN_SEM* qui est telle que :
$\forall$ *m*, *m* $\in$ *mot*, EXP_SYN_SEM(m) = {(cs, es), où cs $\in$ EXP_SYNT(m), et es $\in$ EXP_SEM(m)}
EXP_SYN_SEM(m) donne la liste des couples formés d'informations syntaxiques et sémantiques liées à m. Si l'on veut donner un exemple informel, on dirait que *EXP_SYN_SEM(car) = {(préposition invariable, lien de cause à effet), ((substantif commun, masculin, singulier), moyen-de-transport)}*.
Par la suite nous appelons *expression syntaxique de m* l'ensemble *EXP_SYNT(m)*, et *expression sémantique de m* l'ensemble *EXP_SEM(m)*.
Utilisant ce formalisme, on peut relever certaines propriétés intéressantes du vocabulaire, par exemple :
ambiguïtés syntaxiques = $\{m \in mot, card(EXP\_SYNT(m)) > 1\}$
ambiguïtés sémantiques = $\{m \in mot, card(EXP\_SEM(m)) > 1\}$
polysémies pures = $\{m \in mot, (card(EXP\_SEM(m)) > 1) \& (card(EXP\_SYNT(m)) = 1)\}$

4.3.2. initialisation du lexique

Le problème de l'initialisation du lexique de RIME a été abordé en deux temps, sous un aspect tout d'abord syntaxique, ensuite sémantique.

4.3.2.1. initialisation syntaxique

Avant toute chose, il est nécessaire de préciser que pour tout mot *m* saisi, toutes ses déclinaisons (en genre et/ou en nombre pour les substantifs, les adjectifs et certains pronoms, ou bien en temps, mode, nombre et personne pour les verbes et auxiliaires) sont déduites automatiquement : là encore, nous utilisons les possibilités du modèle linguistique de P.PALMER dans lequel sont décrits tous les modèles de déclinaison d'une part en genre et en nombre, d'autre part en temps, mode, nombre et personne. Par exemple, lors de la

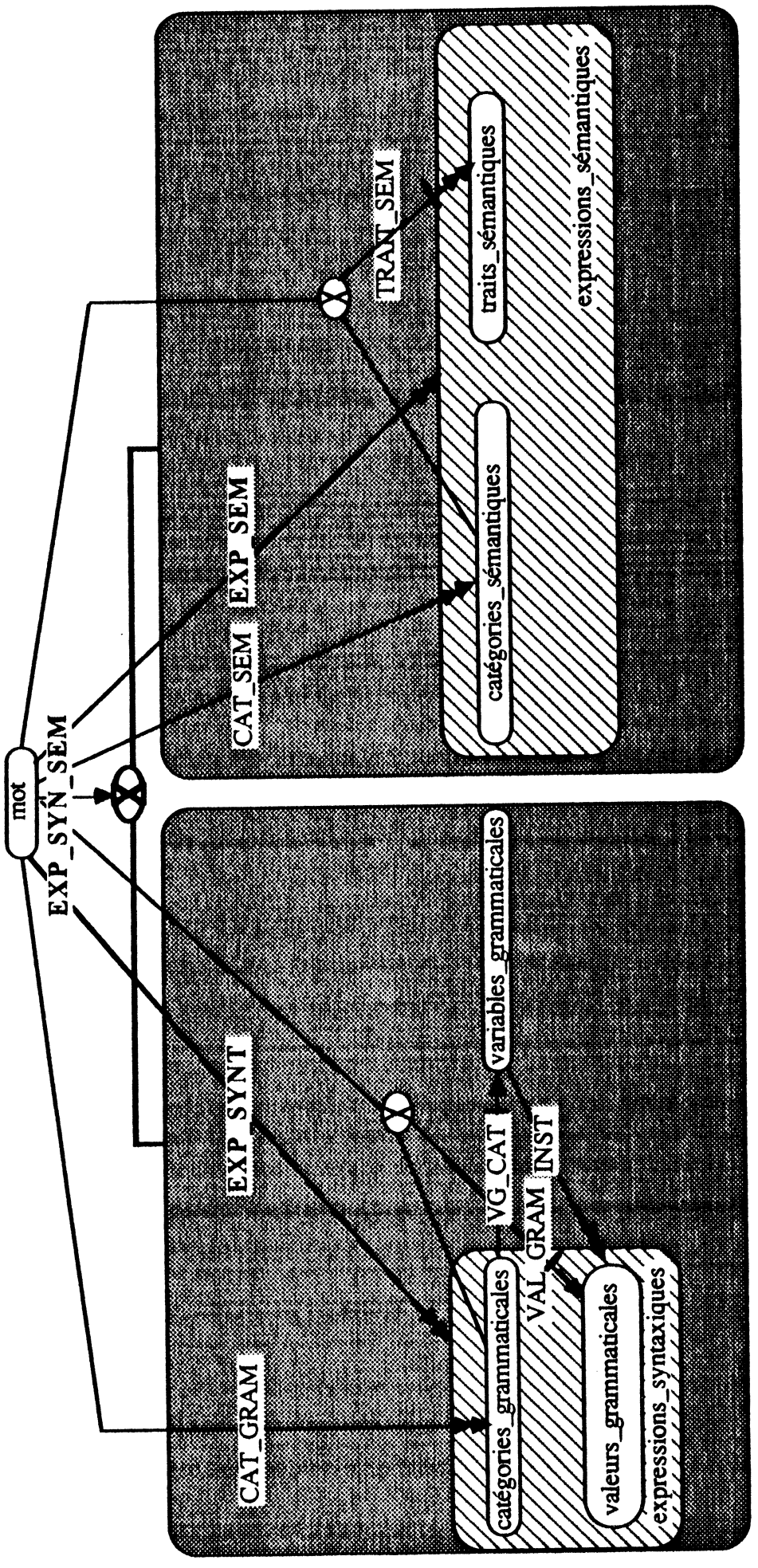


figure 8 : le lexique de RIME

saisie des informations syntaxiques et sémantiques du mot *arbre*, nous déduisons également celles du mot *arbres*.

Dans un premier temps, nous avons abordé l'aspect syntaxique du problème syntaxiquement en stockant tous les mots des catégories fermées. Rappelons que cette liste de mots est exhaustive : pour chacun d'eux nous avons, avec l'aide d'un expert, stocké manuellement leurs attributs syntaxiques et sémantiques. En conséquence de quoi, tous les mots restants à saisir dans le lexique sont des mots des catégories ouvertes de notre modèle linguistique.

4.3.2.2. initialisation sémantique

Dans un second temps, nous avons abordé l'aspect sémantique du problème en demandant aux experts une liste des mots correspondant au vocabulaire de base : ainsi tout mot *m* du vocabulaire de base d'une catégorie sémantique *cat_sém* donnée est stocké dans le dictionnaire en complétant *EXP_SYN_SEM(m)* par *((cat_gram, var_gram), (m, cat_sém))* où *cat_gram* $\in$ *{ SUBC, SUBP, ADJQ, ADVB, VBIF, VBPA, VBPR, VBCJ, ABRV}* et *var_gram* est une instantiation de *VG_CAT(cat_gram)*.

Finalement, d'une part nous avons demandé aux experts une liste des mots du vocabulaire complexe complet ou incomplet, d'autre part nous avons analysé un certain nombre de comptes rendus médicaux de manière à en extraire les mots inconnus du lexique actuel.

Tous les mots sont ensuite saisis dans le lexique avec l'aide d'un expert et d'un outil d'aide à la mise à jour du lexique que nous décrivons ci-après. Cet outil est basé sur une étude morphologique du vocabulaire médical, et permet dans certains cas de faire des propositions de trait et de catégorie sémantiques. Par ailleurs, toutes les mises à jour après cette initialisation du lexique sont faites avec l'aide de cet outil.

4.4. proposition de mise à jour du lexique

Le lexique mis en œuvre pour traduire un compte rendu selon le modèle sémantique regroupe, pour chaque mot du langage médical spécialisé, des informations de nature syntaxique et sémantique, ces dernières étant directement liées au modèle sémantique. Ce modèle est par hypothèse correct : toute insertion ou modification doit donc être vérifiée et validée par un expert. Il est clair cependant que la grammaire et son vocabulaire ne sont pas des données faciles à manipuler, pour ne pas dire ésotériques, et que ces manipulations doivent être dans la mesure du possible simplifiées :

- pour un domaine donné, les règles de la grammaire évoluent très peu. Elles peuvent éventuellement être modifiées ou complétées, et nous montrons au chapitre 5 les moyens mis en œuvre pour l'aide et surtout la vérification de cette évolution ;

- le vocabulaire évolue lui plus régulièrement, et pour cela nous présentons ici un outil d'aide à la mise à jour que nous définissons ci-après.

Comme nous l'avons vu, la langue des comptes rendus médicaux relève d'une langue de spécialité.
D'un point de vue vocabulaire, ceci implique l'utilisation d'un vocabulaire de la langue mère et du vocabulaire de la langue de spécialité. Nous entendons par vocabulaire de la langue mère d'une part les mots outils de cette langue - c'est-à-dire le contenu des catégories fermées que nous avons présentées précédemment - , et d'autre part les mots des catégories ouvertes ne correspondant ni à un terme relevant d'une technologie médicale ni à un terme médical. Nous étudions plus particulièrement le vocabulaire propre à la langue de spécialité utilisé dans les comptes rendus médicaux (que nous appelons désormais *vocabulaire médical*), pour tenter de voir comment certaines propriétés de la construction morphologique de ce vocabulaire peuvent permettre de déduire des propriétés syntaxiques ou sémantiques de ces mots.
Nous verrons ensuite dans quelle mesure ces connaissances déduites peuvent aider, notamment lors de mises à jour du lexique.

4.4.1. étude morphologique du vocabulaire médical

4.4.1.1. présentation

Le but ici est d'analyser les propriétés morphologiques du vocabulaire médical, ce qui nous permettra, en admettant que les problèmes liés à une telle analyse soient résolus, de nous interroger sur l'information associée au vocabulaire traité : la morphologie permet en effet de dégager de chaque mot des morphèmes, c'est-à-dire des formes minimum douées de sens, dont il faut tirer une certaine productivité en terme d'informations syntaxiques ou sémantiques.

Une telle étude nous permettra ultérieurement de mettre en œuvre des outils permettant des décisions quant au contenu sémantique des nouveaux mots du vocabulaire, allégeant ainsi le travail de l'expert travaillant à la mise à jour du lexique de notre système.
Il faut cependant souligner que tous les outils proposés suite à cette analyse ne peuvent être que des aides à la décision, et non des outils de décision, et ce pour plusieurs raisons :

- la première est l'impossibilité de trouver des règles systématiques de décomposition morphologique non contredites par nombre de contre-exemples ;

- la seconde de ces raisons est l'interdiction que nous nous sommes fixée quant à l'introduction d'erreurs dans RIME. Une telle restriction nécessite la validation de toute insertion dans RIME par un expert du domaine quand l'insertion porte sur une connaissance médicale, ce qui est le cas quand on essaie de déduire des connaissances au niveau du vocabulaire médical.

Il faut également souligner que dans un système comme le nôtre, l'aide à la décision est fondamentale, et qu'il faut dans la mesure du possible soulager le travail de décision au travers de propositions parmi lesquelles l'expert décidera de la solution finale. Nous verrons, en conclusion de cette partie concernant le vocabulaire médical d'autres possibilités d'aide à la décision que l'analyse morphologique, et comment le tout peut former un outil relativement puissant.

4.4.1.2. les différentes compositions du vocabulaire médical

Le vocabulaire médical que nous étudions est essentiellement constitué de mots composés, c'est-à-dire selon [CATA82] des *mots formés d'éléments morphologiques ou lexicaux à l'origine distincts, tendant vers l'unité sémantique et grammaticale, unité qu'ils finissent en général par atteindre au point que les composants ne sont plus sentis.* Nous ne traitons pas ici les mots non-composés du vocabulaire médical, car nous n'avons pour eux aucun moyen automatique de déduction morphologique. Nous verrons dans la conclusion de cette étude du vocabulaire médical quels autres moyens peuvent être utilisés pour aider l'expert dans sa décision.

On distingue classiquement trois sortes de composition :

- **la composition par particules** qui va de la préfixation à la composition sur prépositions, adverbes, etc (exemple : *juxta+position*, *exo+coriation*). Ce type de composition n'est pas typique du vocabulaire médical, et nombre de mots du français courant sont construits sur ce mode. Les préfixes (prépositions, adverbes, etc) sont d'origine grecque, latine, française ou assimilée, et leur liste est connue. L'analyse morphologique dans ce cas de composition peut être envisagée comme une consultation de la liste des préfixes. Cependant il faut être prudent quant aux déductions hâtives ; par exemple, tous les mots commençant par *anti* ne sont pas des mots composés par la particule *anti : antichar* en est un, mais pas *antimoine* ;

- **la composition par thèmes** nominaux ou verbaux (exemple : *cox+alg+ie, cox+arthr+ose, arthr+alg+ie*). Ce type de composition est particulièrement utilisé dans le vocabulaire médical, d'autant plus que c'est un processus vivant et productif de construction de mots nouveaux. On ne trouve dans ces mots composés par thèmes que des *radicaux nus, dépouillés de toute flexion, et suivis seulement d'une terminaison qui donne au composé son unité et son individualité.* Les matériaux de cette construction proviennent le plus souvent du grec ou du latin, leur liste est connue mais par contre elle peut évoluer. On peut remarquer que les différents constituants n'ont pas de place a

priori dans le mot (par exemple *arthr* ci-dessus). L'analyse morphologique pour ce genre de composition ne peut généralement se faire que par connaissance des morphèmes pouvant composer un mot et par connaissance des frontières entre morphèmes. Ces frontières ne sont généralement pas marquées ; cependant on peut remarquer certains cas fréquents :

**o* dans *électr+o+cardiogramme* ;

* une séquence de voyelles formant digramme dans *anacrob+io+se* ou *ox+ya+cétylénique* ;

* *i* dans *iso+i+onique* ou *rhombo+ï+de* ;

* - dans *cardio+vasculaire* ou *génito+urinaire.*

Là encore les contre-exemples sur ce type de composition ne sont pas rares, et il faut donc être également très prudent quant aux conclusions de construction morphologique sur ce type de composition ;

- **la composition par éléments** grammaticaux ou lexicaux autonomes où l'on trouve des locutions invariables d'une part, adjectifs et noms d'autre part (exemple : *garde-à-vous, cul-de-jatte*). La mise en évidence d'une telle composition offre moins de possibilités que les deux précédentes, car d'une part les éléments autonomes potentiels ne sont pas connus a priori, et d'autre part il est difficile dans ce type de composition de déduire des informations syntaxiques ou sémantiques sur un mot à partir de ses morphèmes constructeurs. Pour ces différentes raisons, nous n'étudierons pas ce type de composition pour le vocabulaire médical, et nous considérons les mots composés par éléments comme les mots non-composés du vocabulaire médical. Nous reviendrons sur ces mots dans la conclusion de cette analyse morphologique du vocabulaire médical.

4.4.2. les informations déductibles de ces compositions

Il s'agit maintenant de voir quelles informations tirer du vocabulaire médical lorsque sa construction relève d'une composition par particule ou d'une composition par thème. Comme nous l'avons déjà fait remarquer ci-dessus, il faut être très prudent quant à des décisions hâtives puisqu'il existe nombre de contre-exemples pour chacune des compositions.

4.4.2.1. morphologie et syntaxe

Certaines études [PAC & PRA69], [ERLI82] ont regardé la possibilité qu'offre une analyse morphologique au point de vue syntaxe, notamment en dressant des listes de morphèmes dérivationnels (morphèmes qui permettent de passer d'une catégorie grammaticale à une autre). On peut par exemple étudier le passage de la catégorie grammaticale adjectif qualificatif à la catégorie substantif, en dressant des listes de terminaisons possibles (on sait pour la langue anglaise qu'un adjectif se terminant par *ie* a des correspondants substantifs se terminant par *o, a, e,* etc). Ces règles ne sont pas des règles systématiques de déduction de catégories grammaticales , mais elles délivrent un ensemble de possibilités pour un mot d'une catégorie grammaticale donnée de mots d'une autre catégorie ayant même racine mais un suffixe différent. Ces études donnent des résultats remarquables ; mais il est cependant nécessaire de dire que de tels processus ne sont pas très intéressants dans RIME dans la mesure où nous utilisons un lexique initialisé de telle façon que les catégories grammaticales potentielles d'un mot inconnu sont suffisamment restreintes. Ajouter là-dessus un mécanisme similaire à ceux que nous venons de voir ne ferait qu'alourdir le processus général de RIME sans vraiment améliorer ses capacités.

4.4.2.2. morphologie et sémantique

Contrairement à la relation morphologie/syntaxe qui dans RIME ne nous apporte pas des résultats intéressants, nous allons voir que l'on peut tirer des résultats intéressants en étudiant la relation morphologie/sémantique, notamment en étudiant les compositions par thème et par particule.

- **les mots composés par particules** sont formés d'un préfixe - voir en annexe 4 la liste des préfixes admis dans la composition par particule - et d'une racine. Les préfixes dans la composition par particule ont un sens précis, et apparaissent comme des entités sémantiquement dépendantes de la racine, et

dont le rôle consiste essentiellement en une modification partielle du sens de la racine.

Par exemple, *juxtaaortique* est un mot composé du préfixe *juxta* et de la racine *aortique* où la racine *aortique* représente une entité sémantiquement autonome, et le préfixe *juxta* un modificateur de la racine.

Quelle que soit la racine associée, un préfixe modifiera toujours de la même façon le sens des racines qui lui sont associées dans une composition par particule : la sémantique drainée par un préfixe se traduit, en terme du modèle sémantique, en une instanciation de l'opérateur sémantique *EN_REL_TOPO_AVEC*, soit un trait sémantique complexe et incomplet (à gauche et à droite) ; le fils droit sera dans ce cas complété par le trait sémantique de la racine du mot. Il faut cependant noter ici que certains préfixes sont sémantiquement ambigus, c'est-à-dire qu'ils offrent plusieurs traductions selon notre modèle sémantique, par exemple *in* peut représenter la notion *à l'intérieur de* tout comme la notion *sans*.

Nous pouvons sérialiser le problème de la façon suivante :

1) décomposer morphologiquement le mot ;

2) connaître la traduction sémantique du préfixe et de la racine. Pour le préfixe, nous supposons que la liste des préfixes et de leur traduction sémantique est connue. Par contre dans le cas où la racine n'est pas déjà connue du système, il faut faire appel à un expert ;

3) unir les deux traductions sémantiques - celle du préfixe et celle de la racine, ce qui ici se traduit en validation, selon le modèle sémantique, du rattachement de la racine en tant que fils droit de l'arborescence : ce qui implique une vérification de l'inclusion de la catégorie sémantique de la racine dans la catégorie sémantique du fils droit du trait sémantique incomplet.

Rappelons une fois de plus que les contre-exemples ne sont pas rares, que toute proposition de ce type de composition ne doit constituer qu'une aide à la décision, que seul un expert est à même de décider quant à la validité de toute proposition. Notamment l'expert décidera de la solution à choisir parmi les possibilités offertes par les préfixes ou les racines ambigus.

Exemple :

- *juxta* dont la traduction sémantique est l'instanciation *contact* de l'opérateur sémantique *EN_REL_TOPO_AVEC* et a pour traduction sémantique *([contact, lésion, localisation], lésion)* ;

- et *aortique* dont la traduction sémantique est *([aorte], const_org)* ;

donnent dans une composition par particule *juxtaaortique* dont la traduction sémantique *([contact, lésion, aorte] , lésion)*.

- **les mots composés par thème** sont formés de morphèmes autonomes. Ces morphèmes représentent des entités sémantiquement indépendantes qu'il faut relier pour former l'entité sémantique représentée par le mot composé.

Cette réunion sémantique de morphèmes pose des problèmes que nous n'avons pas rencontrés dans la composition par particule :

- Ici aucune indication ne nous est donnée quant aux relations sémantiques qui permettent de relier les différents morphèmes. Par exemple, dans le mot composé *staphylocoque* qui est composé des deux morphèmes *staphyl* et *coque*, *staphyl* représente une forme et *coque* un germe ; nous devons dans ce cas les réunir de la façon suivante : *[a_pr_val, germe, staphyl]*. Par contre dans le mot composé *gonocoque* qui est composé des morphèmes *gon* et *coque* , *gon* représente un lieu et nous devons les réunir de la façon suivante : *[p_sur, germe, gon]* ;

- Il peut également exister dans ce type de composition des phénomènes elliptiques, c'est-à-dire des effacements partiels de connaissances. Par exemple, le mot *hépatomégalie* qui se décompose en *hépat (foie)* et *mégalie (augmentation)* présente une ellipse de *volume* pour indiquer qu'une *hépatomégalie* est une *augmentation du volume du foie*, ellipse que nous devons résoudre pour en arriver à la traduction sémantique *([p_sur [a_pr_val, volume, augmenté], foie], signe)*

En plus de ces problèmes particuliers à la composition par thème, nous retrouvons ici le problème classique d'ambiguïtés sémantiques que nous avions rencontré dans la composition par particule.

Nous pouvons là également sérialiser le problème de la façon suivante :

1) décomposer morphologiquement le mot composé. Bien que la liste des morphèmes acceptés dans une composition par thème soit connue - encore qu'elle puisse évoluer -, il n'est pas évident, vus les différents cas de frontières entre morphèmes, de les extraire d'un mot, et de ne pas arriver à différentes décompositions possibles ;
2) connaître le sens associé à chacun des morphèmes. Travaillant sur la liste des morphèmes connus, nous supposons que la traduction sémantique de chacun est consignée ;
3) relier les morphèmes entre eux. On ne connait pas, dans ce type de composition, les liens sémantiques entre les différents morphèmes, ce qu'il faut traiter de la même façon que l'union de deux mots consécutifs issus d'un compte rendu médical - cf partie 5.4. -. Les cas elliptiques ne peuvent être traités que via l'expert qui est à même de donner au système l'information manquante, notamment en indiquant le groupe complet synonyme du mot composé : par exemple, *augmentation du volume du foie* pour le mot composé *hépatomégalie*. De la même façon que dans la composition par particule, l'expert choisit parmi les différentes propositions dues aux morphèmes ambigus laquelle est correcte.

4.4.3. une aide à la mise à jour du lexique

Nous avons vu dans le chapitre 3 le noyau bicéphale de RIME : une grammaire décrivant le modèle sémantique, et un lexique contenant le vocabulaire des comptes rendus médicaux.
La grammaire peut être vue de manière relativement statique ; elle peut bien entendu évoluer de manière ponctuelle : une règle à ajouter ou à modifier. La grammaire peut également être complétement réécrite si le domaine d'application change. De telles modifications sont rares, mais elles nécessitent beaucoup de précautions car elles engendrent des conséquences qui influenceront toute la fonction d'indexation de RIME.
Face à une certaine statique de la grammaire, le vocabulaire lui évolue beaucoup plus librement. Et pour pallier l'ésotérisme du modèle sémantique exposé au chapitre 3, nous proposons ici une aide à la mise à jour du

vocabulaire des comptes rendus médicaux, outil qui devrait permettre de réaliser cette tâche à toute personne connaissant le milieu décrit, sans pour autant remettre en cause la description faite au travers de la grammaire. Rappelons que la langue des comptes rendus médicaux correspond à une langue de spécialité, et que nous étudions ici le vocabulaire qui relève uniquement de cette langue de spécialité. Comme nous venons de le voir une partie de ce vocabulaire répond à des critères précis de composition morphologique, et nous allons maintenant les utiliser pour décrire un outil d'aide à la saisie du vocabulaire de la langue de spécialité : voir la figure 9 où nous présentons l'outil de mise à jour du vocabulaire médical.

Ainsi que l'indique la figure 9, tout mot nouveau est automatiquement décomposé morphologiquement. Après validation par l'expert de cette décomposition, plusieurs cas peuvent se produire :

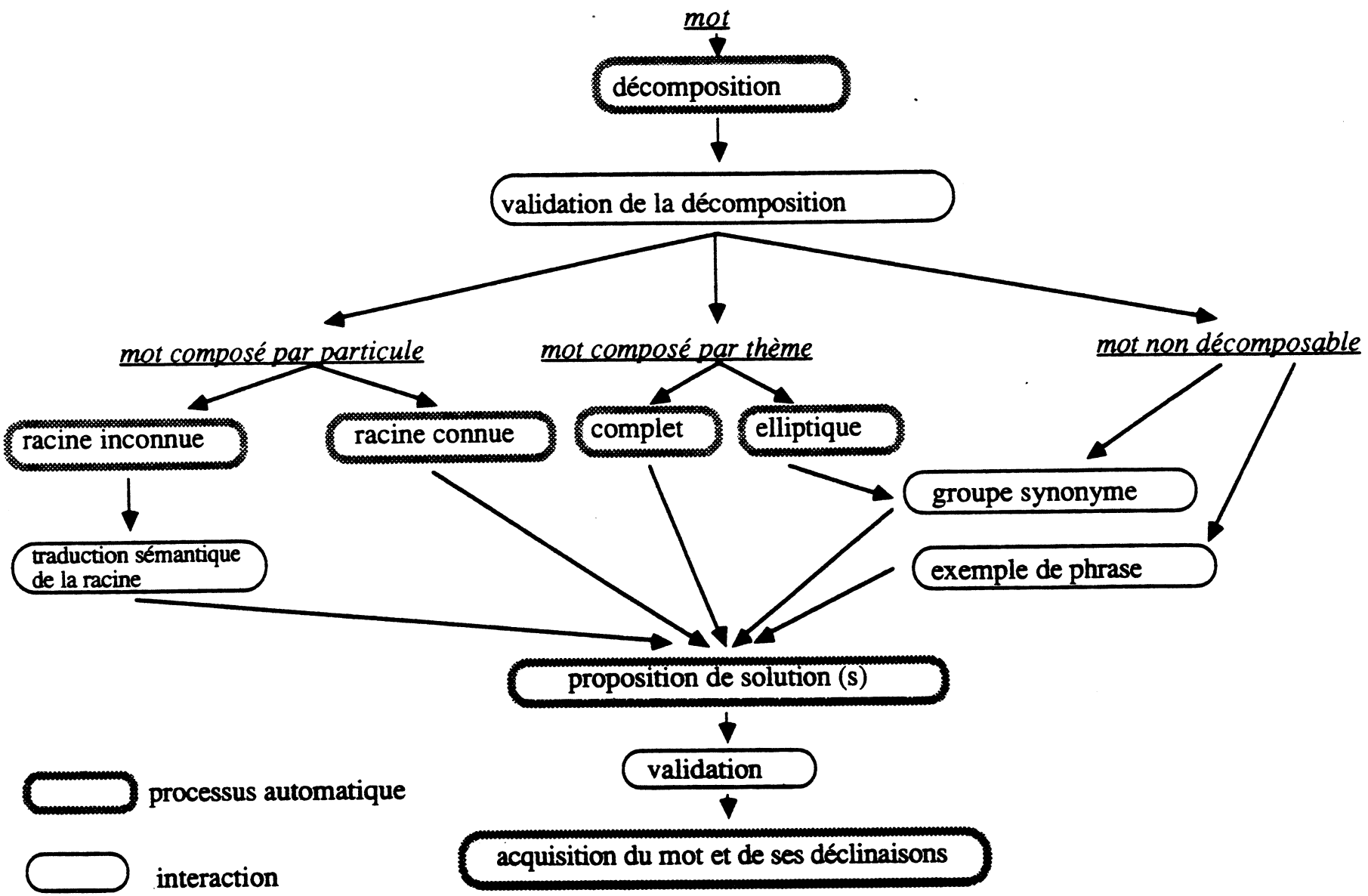
- le mot est composé par particule ou par thème : les stratégies décrites dans la partie 4.4.2.2. sont appliquées ;

- le mot n'est pas décomposable : on demande à l'expert un nom (ou une groupe nominal) synonyme, ou une phrase contenant le mot nouveau, de manière à le traduire dans le modèle sémantique de RIME, au travers du processus sémantique décrit dans la partie 5.4.

Dans tous les cas, les propositions de traduction sémantique doivent être validées par l'expert. Syntaxiquement, nous connaissons un ensemble restreint de catégories syntaxiques pour un mot inconnu - les catégories ouvertes - et l'expert aura à choisir parmi cet ensemble. Quand l'expert aura décidé et validé une (ou plusieurs) solution, elle sera consignée ainsi que toutes ses déclinaisons en nombre pour les substantifs, en genre et en nombre pour les adjectifs, en temps, mode, personne et nombre pour les verbes.

Il est clair par ailleurs que nous ne couvrons pas dans cet outil toutes les possibilités de composition des mots médicaux : ainsi nous n'incluons pas les cas très rares de composition mixte, c'est-à-dire par thème et par particule : comme par exemple pour *interhépatodiaphragmatique*. Cependant face à la complexité initiale du modèle sémantique de RIME, ceci offre une proposition intéressante d'aide à la mise à jour du modèle.

figure 9 : une aide à la mise à jour du vocabulaire médical



Chapitre 5
Les outils linguistiques

Le but de ce chapitre est définir les quatre processus constructeurs de RIME, qui sont :

- la morphologie, que nous présentons dans la partie 5.2 ;
- la syntaxe, que nous présentons dans la partie 5.3 ;
- la sémantique, que nous présentons dans la partie 5.4 ;
- la coopération, que nous présentons dans la partie 5.5.

Nous donnons tout d'abord un premier aperçu du processus de coopération avant de le compléter dans la partie 5.5, ainsi qu'un exemple de traduction d'une phrase dans le modèle sémantique de RIME au travers des différents processus constructeurs.

5.1. introduction

5.1.1. le processus de coopération

Le processus de coopération de RIME a pour tâche essentielle de piloter les processus constructeurs que sont la morphologie, la syntaxe et la sémantique, ce que nous rappelons au travers de la figure 10.

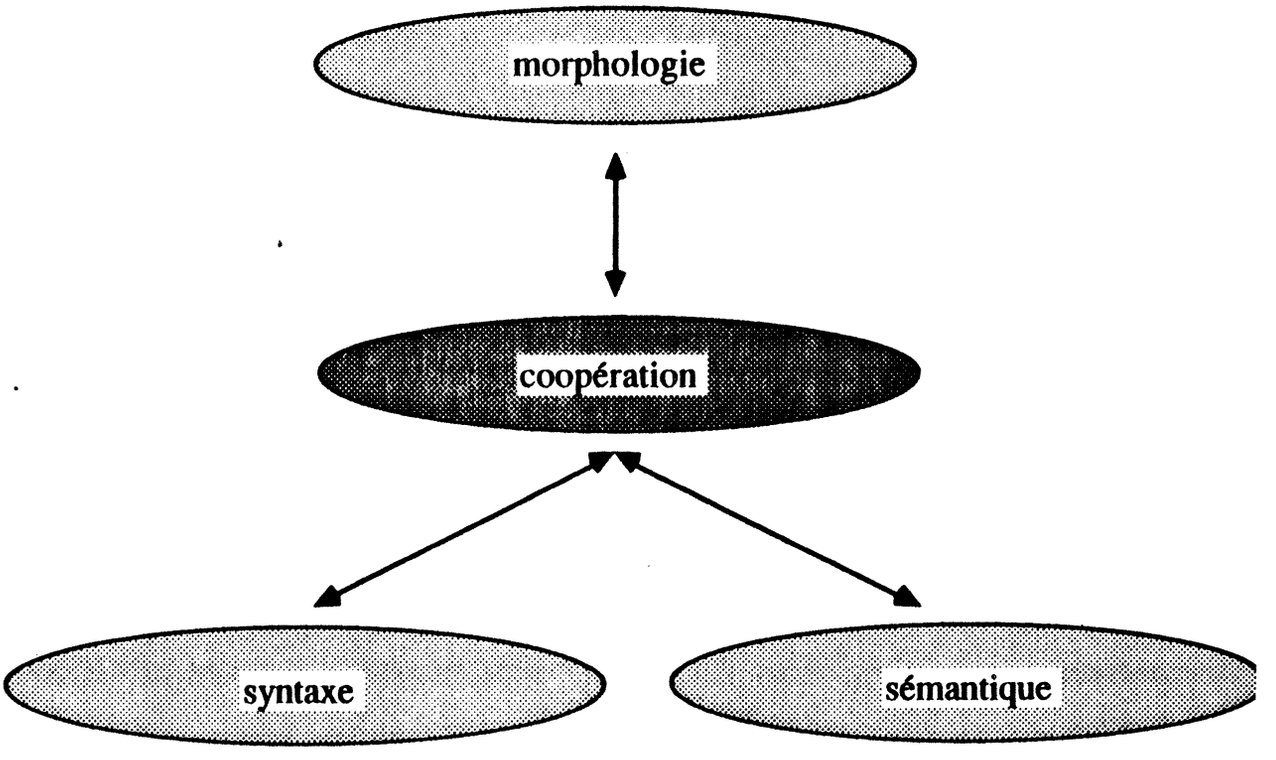
Le travail du processus de coopération se divise essentiellement en deux parties :

- d'une part l'enchaînement des trois processus constructeurs de manière à permettre la traduction de textes en langue naturelle ;

- d'autre part la transmission de résultats entre les processus constructeurs.

Nous reviendrons plus en détail sur cet aspect dans la partie 5.5.

figure 10 : les différents processus de RIME



5.1.2. un exemple complet de traduction

Avant de montrer en détail chacun des processus constructeurs de RIME, nous voulons ici donner un simple exemple de traduction de phrase dans le modèle sémantique de RIME, en montrant le rôle joué par chacun des processus.

Prenons comme phrase à traduire :

phrase initiale : *le cancer dévie l'uretère*

5.1.2.1. coopération-morphologie

Le premier travail du processus de coopération est de confier cette phrase initiale au processus morphologique :

appel morphologie : *le cancer dévie l'uretère*

La morphologie traite cette phrase par consultaion du lexique décrit au chapitre 4 (cf partie 5.2).

figure 11 : résultat de la morphologie

le	*cancer*	*dévie*	*l'*	*uretère*
(ARTD, (masc, sing)) (PRPV, (masc, sing)) (le, creux)	(SUBC, (masc, sing)) (cancer, lésion)	(VBCJ, (indicatif, présent, 3ème, sing)) (dû_à p_sur lésion a_pr_val loc position déviation , diagnostic)	(ARTD, (masc/fém, sing)) (PRPV, (masc/fém, sing)) (le, creux)	(SUBC, (fém, sing)) (uretère, const_org)

5.1.2.2. coopération-syntaxe

Récupérant le résultat de la morphologie, le processus de coopération le transmet au processus syntaxique :

figure 12 : appel de la syntaxe

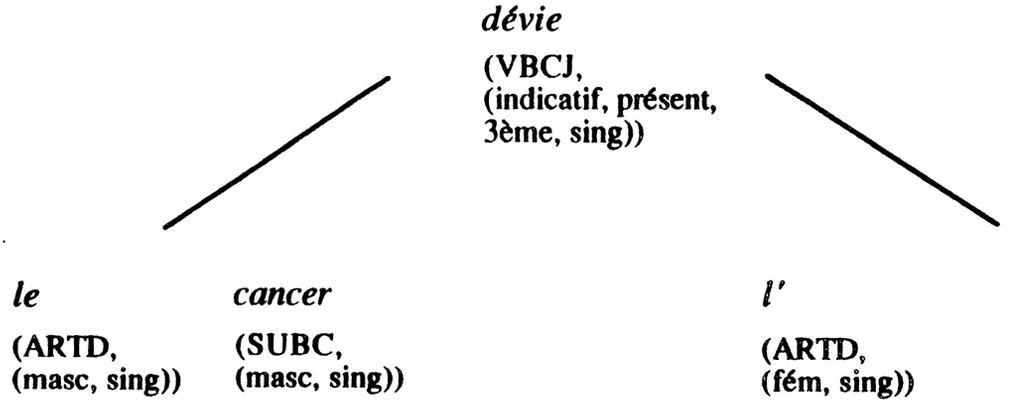
le	*cancer*	*dévie*	*l'*	*uretère*
(ARTD, (masc, sing)) (PRPV, (masc, sing))	(SUBC, (masc, sing))	(VBCJ, (indicatif, présent, 3ème, sing))	(ARTD, (masc/fém, sing)) (PRPV, (masc/fém, sing))	(SUBC, (fém, sing))

Le processus syntaxique traite cette donnée en deux temps :

- tout d'abord, il filtre les possibilités morphologiques de chacun des mots, en utilisant une matrice de précédence et des schémas d'ambiguïtés (cf partie 5.3.1) ;

- ensuite, il construit des structures syntaxiques (syntagmes nominaux, prépositionnels, verbaux) (cf partie 5.3.2, 5.3.3).

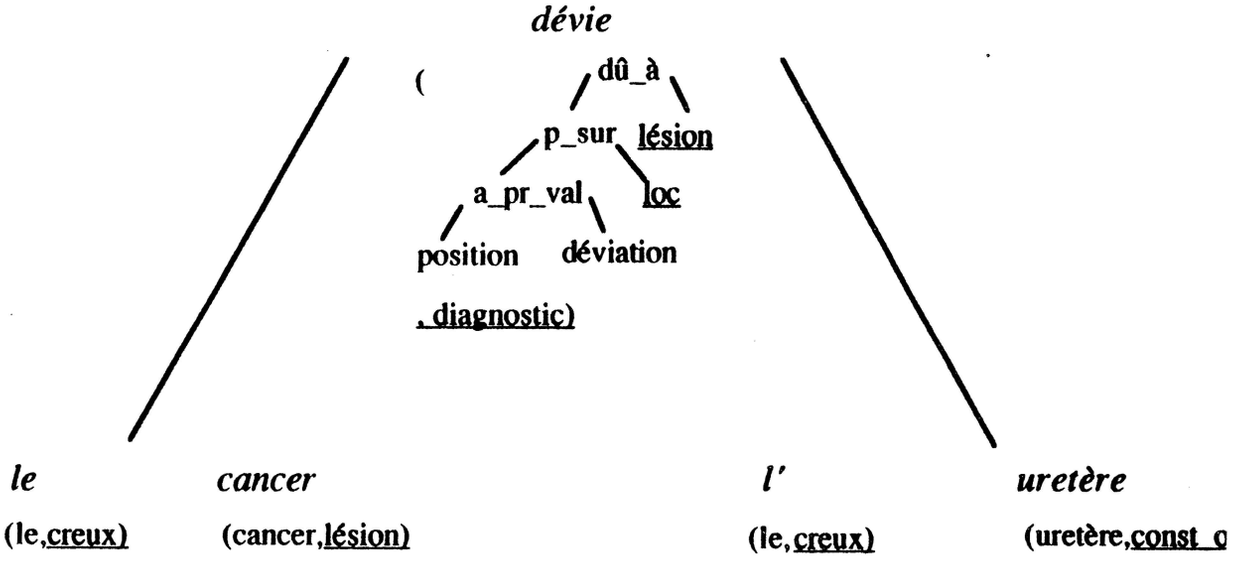
figure 13 : résultat de la syntaxe



5.1.2.3. coopération-sémantique

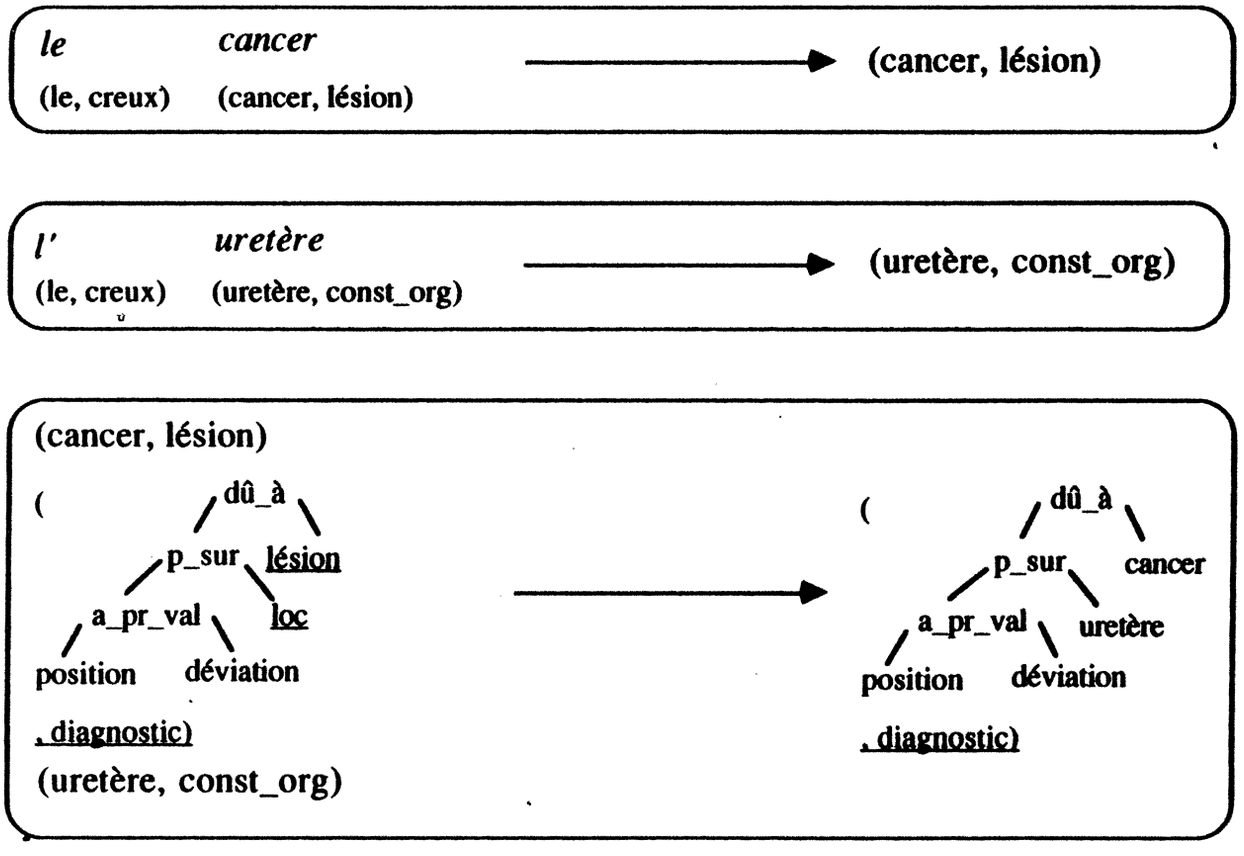
Le processus de coopération récupère ce résultat pour le transmettre au processus sémantique de la façon suivante :

figure 14 : appel de la sémantique



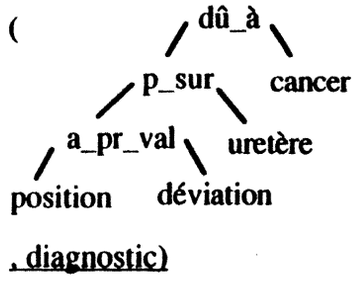
Le processus sémantique traite cette entrée au travers de ses deux composantes que sont l'enveloppe (cf partie 5.4.1) et le noyau sémantiques (cf partie 5.4.2) :

figure 15 :
appel de l'enveloppe sémantique et réponse du noyau sémantique



Finalement le processus sémantique rend au processus de coopération le résultat final :

figure 16 : résultat final



qui est considéré comme le résultat final du traitement.

Nous présentons maintenant plus en détail chacun des processus constructeurs, en montrant pour chacun d'eux comment sont mises en œuvre les tâches qui leur ont été confiées.

5.2. l'analyse morphologique

Si l'on reprend la figure 7 montrant la répartition des tâches dans RIME, les tâches relevant de la morphologie sont l'identification des mots et la déduction automatique via le lexique, ou manuelle via l'aide d'un expert de leurs attributs virtuels.
Pour montrer comment nous concrétisons ces tâches morphologiques, nous donnons quelques définitions que nous reprenons de P.PALMER dans [PALM88].

5.2.1. définitions

5.2.1.1. forme

On appelle *forme*, notée F_i, où i désigne le rang de la forme dans le texte, un mot ou un séparateur impératif.

5.2.1.2. solution d'une forme

Si nous reprenons les définitions données au chapitre 4 (figure 8), nous appelons *solution* d'une forme f_i, l'ensemble $EXP\_SYN\_SEM(f_i)$ qui représente les interprétations hors-contexte de f_i dans notre modèle. Nous appelons *solutions morphologiques* la restriction aux éléments $ess_{i,n}$ de l'ensemble solution $EXP\_SYN\_SEM(f_i)$. Cet ensemble $EXP\_SYN\_SEM(f_i)$ constitue en fait l'ensemble solution intrinsèque dans notre modèle de f_i.
Si $EXP\_SYN\_SEM(f_i)$ fournit p solutions morphologiques pour la forme f_i et que l'on note $ess_{i,k}$ la kième, alors :
$EXP\_SYN\_SEM(f_i) = \{ess_{i,1}, \ldots, ess_{i,k}, \ldots, ess_{i,p}\}$
Par exemple, pour f_i = *"poumon"*, on a

$EXP\_SYN\_SEM(f_i) = \{ess_{i,1}\} = \{((SUBC, (masculin, singulier), (poumon, const\_org))\}$

5.2.2. fonction de l'analyse morphologique

Nous pouvons après ces définitions préciser la fonction de l'analyse morphologique : segmenter en formes f_i un compte rendu médical écrit en langue naturelle, et les interpréter par rapport à notre modèle, en leur attribuant l'ensemble solution noté $EXP\_SYN\_SEM(f_i)$ de solutions morphologiques.

Les deux problèmes essentiels qui se posent lors de l'analyse morphologique sont :

- d'une part, le cas où il n'y a pas de solution morphologique unique pour une forme ;

- d'autre part, le cas où l'analyseur manque d'informations pour traiter une forme.

Pour cela nous rappelons les définitions suivantes :

- une forme f est dite *ambiguë*, si et seulement si $card(EXP\_SYN\_SEM(f)) > 1$. Cette ambiguïté est rémanente soit de l'information syntaxique soit de l'information sémantique au niveau de la forme f : une forme f est *syntaxiquement ambiguë*, respectivement *sémantiquement ambiguë*, si et seulement si $card(EXP\_SYNT(f)) > 1$, et $card(EXP\_SEM(f)) > 1$;

- une forme f est dite *inconnue* de notre analyse morphologique si et seulement si la fonction EXP_SYN_SEM n'est pas définie pour cette forme. Une forme peut être inconnue pour deux raisons : d'une part la forme n'appartient pas à la langue naturelle traitée (faute d'orthographe, faute de frappe, ...), d'autre part la forme appartient à la langue traitée, et sa non-reconnaissance provient d'une insuffisance de l'analyse morphologique (données linguistiques incomplètes, ...). Quelle qu'en soit la raison, nous n'envisageons pas de traiter automatiquement le problème posé par les formes inconnues : pour le résoudre, nous nous retranchons derrière l'expert et l'outil d'aide à la saisie des mots dans le lexique.

Nous effectuons une analyse morphologique classique de gauche à droite consistant en une segmentation d'une forme en racine + désinence. Les racines se trouvent dans un dictionnaire d'analyse, les désinences dans de multiples tables spécifiques [PALM & BERR85] [PALM88], le tout formant le lexique décrit précédemment. Pour une même forme, toutes les segmentations possibles connues sont mémorisées.

Nous utilisons un dictionnaire d'analyse où toutes les racines sont factorisées sous forme d'arbre lexicographique et rangées selon un critère de fréquence pour optimiser les accès [PALM81]. Chaque fin de racine est matérialisée par le positionnement d'un indicateur. Au cours de l'analyse, la rencontre d'un tel indicateur valide un ensemble de tables de désinences. La reconnaissance est alors poursuivie parallèlement dans ces tables et dans le dictionnaire sans retour arrière.

Cette organisation est compatible avec un enrichissement du vocabulaire, puisque toute insertion d'une nouvelle racine se fait facilement soit en "feuille" dans l'arbre, soit par le positionnement d'un indicateur de fin de racine, si elle est préfixe d'une racine déjà insérée.

5.2.3. le résultat de l'analyse morphologique

Par exemple, l'analyse morphologique de la phrase *les tumeurs dévient l'uretère* comportant une séquence de 5 formes f_1, f_2, f_3, f_4 et f_5 donne pour chaque forme f_i son ensemble $EXP\_SYN\_SEM(f_i)$ de solutions morphologiques $ess_{i,n}$. On a :

$card(EXP\_SYN\_SEM(f_1)) = 2$
$card(EXP\_SYN\_SEM(f_2)) = 1$
$card(EXP\_SYN\_SEM(f_3)) = 1$
$card(EXP\_SYN\_SEM(f_4)) = 2$
$card(EXP\_SYN\_SEM(f_5)) = 1$

le résultat de la morphologie peut se représenter de la façon suivante :

les	*tumeurs*	*dévient*	*l'*	*uretère*
f1	*f2*	*f3*	*f4*	*f5*
$ess_{1,1}$	$ess_{2,1}$	$ess_{3,1}$	$ess_{4,1}$	$ess_{5,1}$
$ess_{1,2}$		$ess_{4,2}$		

Pour l'exemple *les tumeurs dévient l'uretère*, nous obtenons :

f1 = les card(*EXP_SYN_SEM(les)*) = 2

EXP_SYN_SEM(les) = {((ARTD, (féminin, pluriel)), (les, creux)), ((PRPV, (masculin, pluriel)), (le, creux))}

f2 = tumeurs card(*EXP_SYN_SEM(tumeurs)*) = 1

EXP_SYN_SEM(tumeurs) = {((SUBC, (féminin, pluriel)), ([tumeur], lésion))}

f3 = dévient card(*EXP_SYN_SEM(dévient)*) = 1

EXP_SYN_SEM(dévient) = {((VBCJ, (indicatif, présent, 3ème, pluriel)), ([dû_à,[p_sur, [a_pr_val, position, déviation], localisation], lésion], constat))}

f4 = l' card(*EXP_SYN_SEM(l')*) = 2

EXP_SYN_SEM(l') = {((ARTD, (féminin, singulier)), (l', creux)), ((ARTD, (masculin, singulier)), (le, creux))}

f5 = uretère card(*EXP_SYN_SEM(uretère)*) = 1

EXP_SYN_SEM(uretère) = {((SUBC, (féminin, singulier)), ([uretère], const_org))}

5.3. l'analyse syntaxique

Comme nous l'avons vu dans le chapitre 3 (figure 7), la syntaxe dans RIME est chargée de certaines tâches, de signaler et de valider certaines autres :

- tout d'abord, la syntaxe est chargée dans une certaine mesure de la déduction des attributs actuels de chaque forme extraite d'un compte rendu médical : elle tente, pour les mots syntaxiquement ambigus, de minimiser le nombre de leurs solutions morphologiques. Il est clair qu'une analyse syntaxique, si élaborée soit-elle, ne peut résoudre toutes ces ambiguïtés, et nous montrons par ailleurs les limites de l'analyse syntaxique développée dans

RIME, qui correspondent aux objectifs analysés au chapitre 3 et que nous rappelons brièvement ici ;

- la syntaxe est également chargée de certaines tâches intra-structurelles, notamment de la construction et de la nomination de structures syntaxiques particulières. Par exemple, les syntagmes nominaux et verbaux représentent pour l'analyse sémantique une première proposition intéressante de découpage des phrases ; aussi la syntaxe essaie-t-elle d'extraire de telles structures des comptes rendus médicaux ;

- la syntaxe doit également signaler un certain nombre de tâches inter-structurelles : l'effacement par coordination, l'effacement par comparatives, l'effacement par adjectifs possessifs, les anaphores pronominales et les anaphores nominales par répétitions ;

- finalement, la syntaxe est chargée de la validation de la résolution des anaphores pronominales.

Concrètement, la syntaxe traite des séquences telles que :

le	*cancer*	*dévie*	*l'*	*uretère*
(ARTD, (masc, sing)) (PRPV, (masc, sing))	(SUBC, (masc, sing))	(VBCJ, (indicatif, présent, 3ème, sing))	(ARTD, (masc/fém, sing)) (PRPV, (masc/fém, sing))	(SUBC, (fém, sing))

elle simplifie tout d'abord cette séquence pour donner un *parcours syntaxique* (partie 5.3.1) tel que :

le	*cancer*	*dévie*	*l'*	*uretère*
(ARTD, (masc, sing))	(SUBC, (masc, sing))	(VBCJ, (indicatif, présent, 3ème, sing))	(ARTD, (masc/fém, sing))	(SUBC, (fém, sing))

et rend en résultat final une arborescence syntaxique (partie 5.3.2) telle que :

le	*cancer*	*l'*	*uretère*
(ARTD, (masc, sing))	(SUBC, (masc, sing))	(ARTD, (fém, sing))	(SUBC, (fém, sing))

Nous montrons ci-après comment tout ceci est mis en œuvre dans la partie syntaxique de RIME.

5.3.1. la déduction syntaxique des attributs virtuels

5.3.1.1. définitions

Notre stratégie de résolution des ambiguïtés syntaxiques est fondée sur la définition d'un graphe syntaxique associé à la chaîne de formes à analyser; nous en donnons tout d'abord une définition.

a. chaîne syntaxique associée à une séquence de formes

Notons esy_i l'ensemble correspondant à $EXP\_SYNT(f_i)$, qui représente l'ensemble des solutions syntaxiques d'une forme f_i. Nous appelons alors *chaîne syntaxique* associée à la séquence de formes $f_i, \ldots, f_n$ la suite ordonnée $esy_1, \ldots, esy_n$ des traductions syntaxiques de ces formes.

b. transition syntaxique entre deux formes consécutives

Considérons une sous-suite constituée de deux éléments consécutifs esy_i et esy_{i+1} d'une chaîne syntaxique. Nous appelons *transition syntaxique* t_i de la forme f_i à la forme f_{i+1}, le graphe bi-parti ainsi défini :

- ensembles de nœuds : les deux ensembles esy_i et esy_{i+1};

- ensemble des arcs : tous les arcs définis par le produit $a_i = esy_i \times esy_{i+1}$.

Nous notons $a_{i,k,l} = (esy_{i,k}, esy_{i+1,l})$ un élément de a_i.

On a donc $t_i = (esy_i, esy_{i+1}, a_i)$.

Nous appelons *transitions consécutives* les transitions t_i et t_{i+1}, soit

$t_i = (esy_{i,}, esy_{i+1}, a_i)$

$t_{i+1} = (esy_{i+1,}, esy_{i+2}, a_{i+1})$

c. graphe syntaxique associé à une séquence

Etant donnée une séquence de formes $f_1, ..., f_n$, nous appelons *graphe syntaxique gs* associé à cette séquence le graphe n-parti obtenu par union des n-1 transitions syntaxiques consécutives (où l'union est considérée au sens de l'union des graphes).

Exemple : reprenons l'exemple donné précédemment

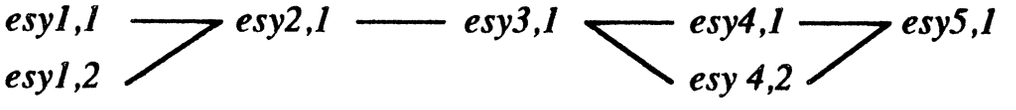
séquence de formes :

les	*tumeurs*	*dévient*	*l'*	*uretère*
f1	*f2*	*f3*	*f4*	*f5*

chaîne syntaxique associée :

esy1,1	*esy2,1*	*esy3,1*	*esy4,1*	*esy5,1*
esy1,2			*esy 4,2*	

graphe syntaxique associé :



Le graphe comprend 4 transitions t_1, t_2, t_3, t_4 avec

$t_1 = (esy_1, esy_2, a_1)$ où $a_1 = \{(esy_{1,1}, esy_{2,1}), (esy_{1,2}, esy_{2,1})\}$

$t_2 = (esy_2, esy_3, a_2)$ où $a_2 = \{(esy_{2,1}, esy_{3,1})\}$

$t_3 = (esy_3, esy_4, a_3)$ où $a_3 = \{(esy_{3,1}, esy_{4,1}), (esy_{3,1}, esy_{4,2})\}$

$t_4 = (esy_4, esy_5, a_4)$ où $a_4 = \{(esy_{4,1}, esy_{5,1}), (esy_{4,2}, esy_{5,1})\}$

Nous notons, par extension, $gs_{i,j}$ le sous-graphe m-parti (m = j-i+1) de gs correspondant à la sous-chaîne syntaxique $esy_i, \ldots, esy_j$.

d. chemin syntaxique

Nous appelons *chemin syntaxique* $c_{i,j}$ associé à la séquence de formes $f_i, \ldots, f_j$, tout chemin sans cycle dans le sous-graphe syntaxique $gs_{i,j}$ correspondant.
Si on restreint la définition d'un tel chemin à la séquence des nœuds constituants, on peut dire que :

$$c_{i,j} \in esy_i \times esy_{i+1} \times \ldots \times esy_j$$

Lorsqu'un sous-graphe syntaxique $gs_{i,j}$ comporte plusieurs chemins syntaxiques, on les notera $c^k_{i,j}$.
Exemple : dans l'exemple précédent, le sous-graphe syntaxique $gs_{1,3}$ correspond à deux chemins syntaxiques :

$$c^1_{1,3} = (esy_{1,1}, esy_{2,1}, esy_{3,1})$$
$$c^2_{1,3} = (esy_{1,2}, esy_{2,1}, esy_{3,1})$$

Un chemin syntaxique correspond à une interprétation syntaxique de la séquence de formes associée; lorsqu'il existe plusieurs chemins associés, cela traduit l'existence d'une ambiguïté au moins.

e. parcours syntaxique

Nous appelons *parcours syntaxique* $p_{i,j}$ associé à une séquence de formes $f_i, \ldots, f_j$, l'ensemble des chemins syntaxiques associés à cette séquence.
Exemple : si nous reprenons l'exemple donné ci-dessus, on a

$$p_{1,3} = \{c^1_{1,3}, c^2_{1,3}\}.$$

Un parcours syntaxique peut éventuellement comprendre plusieurs éléments correspondant à des formes ambiguës. Par assimilation entre une séquence et son parcours associé, nous parlerons désormais de parcours linéaire, ou de parcours ambigu.

Si nous considérons une séquence de formes $f_1, \ldots, f_n$, le graphe syntaxique $gs_{1,n}$ correspondant, ainsi que le parcours syntaxique $p_{1,n}$, chacun des chemins de $p_{1,n}$ correspond à une solution d'analyse de la séquence. Toutes ces solutions, définies jusqu'ici selon des critères purement combinatoires, ne sont évidemment pas forcément correctes au sens des règles de syntaxe de la langue naturelle. Après avoir ainsi défini un espace de solutions possibles pour une séquence de formes, il nous faut à présent définir un processus de sélection des seules solutions correctes. Résoudre les ambiguïtés consiste donc à rendre linéaire un parcours par élimination des chemins incorrects (non conformes à la syntaxe, ou à la sémantique).

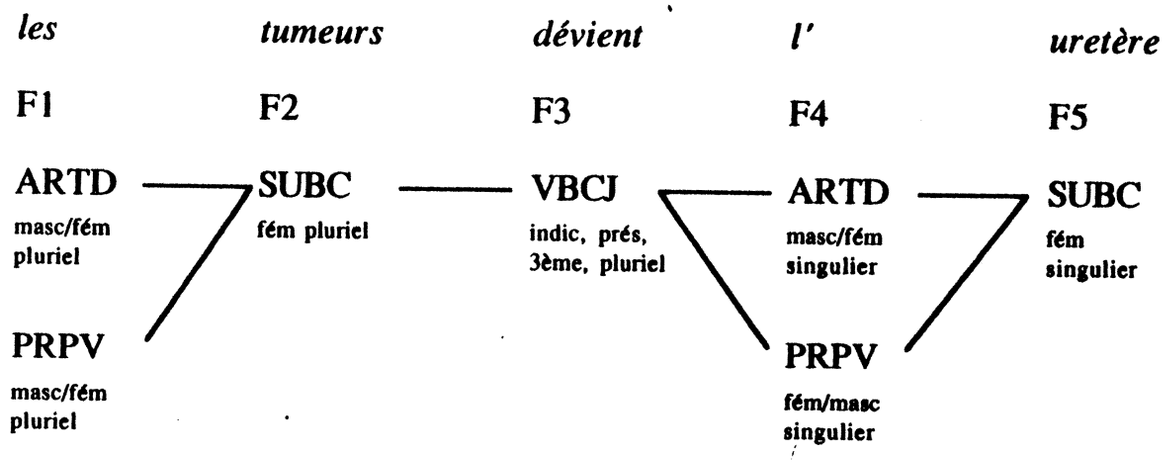
5.3.1.2. le filtrage syntaxique par la matrice de précédence

Les contraintes syntaxiques propres à une langue naturelle induisent généralement des possibilités restreintes quant aux successions licites de catégories grammaticales dans une phrase. On appelle filtrage syntaxique tout mécanisme syntaxique permettant d'éliminer un certain nombre de successions illicites, c'est-à-dire permettant de simplifier un certain nombre de transitions sur le parcours potentiel établi à partir du résultat de l'analyse morphologique. L'effet du filtrage est d'obtenir un parcours plus proche de la linéarité pour chaque phrase analysée.
Dans un premier temps, pour chaque forme f_i du texte analysé, l'analyseur syntaxique ne connaît que l'ensemble $EXP\_SYNT(f_i)$ représentant les possibilités syntaxiques de f_i. Nous montrons sur la figure 17 un exemple de la "vue" de l'analyseur syntaxique.

La première intervention de la syntaxe consiste donc en un filtrage syntaxique qui permet d'élaguer les impossibilités syntaxiques évidentes, en regardant pour chaque forme ses liens avec la forme qui la précède, et la forme qui la suit. On procède à la simplification de certaines transitions en éliminant des arcs qui correspondent soit à des successions syntaxiques interdites, soit à des accords grammaticaux en nombre, (ou en genre et nombre) non vérifiés. Pour

cela, nous utilisons un filtre syntaxique rapide et performant fondé sur une matrice de précédence binaire [PALM88].

figure 17 : une vue de l'analyseur syntaxique



a. la matrice de précédence

Cet outil syntaxique s'inspire directement de la matrice de précédence binaire booléenne de C.FLUHR dans [FLUH77], c'est-à-dire une matrice basée sur les règles positionnelles du français, stockant les liaisons entre deux catégories grammaticales consécutives et donnant en résultat vrai ou faux selon que deux catégories peuvent se suivre ou non : par exemple, un nom peut être suivi d'un adjectif qualificatif, par contre un article défini ne peut être suivi d'un verbe conjugué. La matrice de précédence correspond à la définition d'une fonction *précède* à deux variables *x* et *y* prenant leurs valeurs dans l'ensemble des catégories du modèle :

$\forall$ x, y, *précède*(x,y) $\in$ {vrai, faux}

la valeur *vrai* est fournie si et seulement si un mot de catégorie *x* peut précéder un mot de catégorie *y* ;

la valeur *faux* est fournie sinon.

A la différence de C.FLUHR, nous travaillons sur une matrice de précédence pouvant délivrer cinq types de valeurs :

$\forall$ x, y, *précède*(x,y) $\in$ {0,1,2,3,Δ}

- Δ (ou blanc) quand la succession entre les deux catégories n'a pas encore été envisagée (cas d'intermination) ;

- 0 si la succession est interdite : par exemple, la séquence substantif commun - article défini ;

- 1 si la succession est autorisée sans aucune vérification : par exemple, la séquence substantif commun - préposition ;

- 2 : la succession est autorisée si et seulement si il y accord en nombre entre les deux formes correspondantes, par exemple la séquence pronom personnel - verbe conjugué ;

- 3 : la succession est autorisée si et seulement si il y a accord en genre et en nombre entre les deux formes correspondantes, par exemple la séquence article défini - substantif commun.

Ainsi pour chaque couple de formes consécutives dans le texte, la matrice est consultée et les impossibilités dues à des liaisons interdites, ou à des accords non-vérifiés permettent de simplifier la combinatoire du graphe syntaxique en éliminant des arcs correspondant à des successions interdites.

Cette matrice a été construite par apprentissage sur des textes codés manuellement et par une analyse plus poussée des liaisons entre les catégories jouant un rôle syntaxique intéressant [PALM88]. La construction de cette matrice a été élaborée dans une phase préliminaire et n'a été intégrée à l'analyse syntaxique que lorsqu'elle a été jugée opérationnelle, c'est-à-dire que son taux de remplissage a été jugé suffisamment élevé. De cette manière, nous considérons les cases encore vides, ce que nous appelions les cas d'intermination, comme des successions interdites.

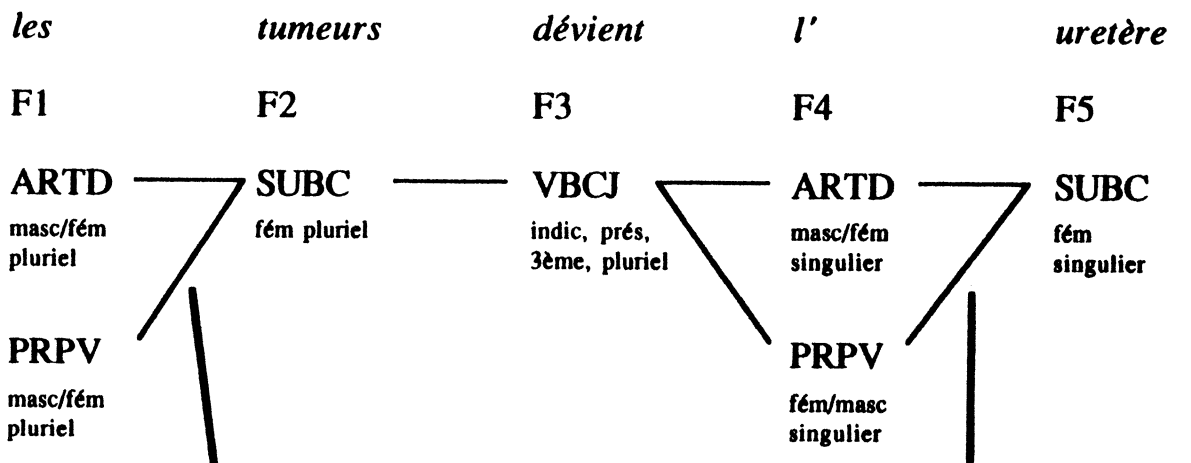
b. l'utilisation de la matrice de précédence

Nous montrons au travers de la figure 18 une simplification syntaxique par utilisation de la matrice de précédence.

De manière plus formelle, nous savons que l'entrée de l'analyseur syntaxique correspond à un parcours syntaxique p_{ij}.

figure 18
un exemple de simplification par la matrice de précédence

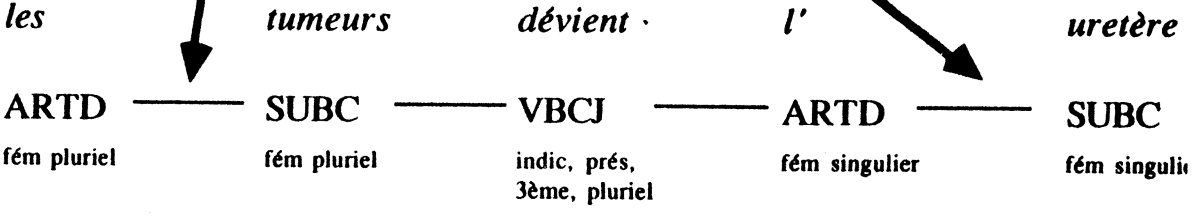
entrée du filtrage syntaxique :



extrait de la matrice de précédence :

	SUBC	VBCJ	ARTD	PRPV
ARTD	3	0	0	0
PRPV	0	1	0	1
SUBC		1	1	1
VBCJ	1	0	1	1

résultat du filtrage syntaxique :



A la suite du premier filtrage syntaxique d'une séquence, le parcours initial est simplifié de manière à obtenir le parcours $ps_{i,j}$ tel que :
$ps_{i,j} \subset p_{i,j}$ (où l'inclusion est considérée au sens de l'inclusion des graphes)
et $\forall\, c_{i,j} \in ps_{i,j}$, *et* $\forall\, p = i, \ldots, j-1$,
et $\forall\, a_{p,k,l} \in c_{i,j}$,
et $précède(a_{p,k,l}) = 1$,
ou $précède(a_{p,k,l}) \in \{2,3\}$ *et les conditions grammaticales sont vérifiées entre* f_p *et* f_{p+1}.
Pour simplifier les notations ultérieures, nous noterons $esyf_{i,k}$ les éléments de l'ensemble simplifié $EXP\_SYNTF(f_i) \subset EXP\_SYNT(fi)$ obtenu après un filtrage syntaxique de la forme f_i :

Il est clair que tous les parcours ne peuvent être complètement linéarisés après ce filtrage, et que certains restent malgré tout partiellement ambigus. Les linéariser complètement correspond à résoudre toutes les ambiguïtés ; cela n'est pas nécessaire dans notre contexte, où la reconnaissance non ambiguë est limitée à des groupements syntaxiques particuliers : syntagmes nominaux ou verbaux. Nous nous limitons par conséquent à résoudre la linéarisation des sous-chemins correspondant à ces unités. Cette linéarisation partielle se fait via un second filtrage syntaxique que l'on effectue en utilisant ce que nous appelons des schémas d'ambiguïtés.

5.3.1.3. le filtrage syntaxique par les schémas d'ambiguïtés

Un schéma d'ambiguïté peut succinctement se décrire comme une règle de transformation admettant en partie gauche un *parcours ambigu*, auquel est associé en partie droite un chemin appelé *chemin solution*. Pour les utiliser lors d'un filtrage syntaxique, nous dressons avant tout une *liste de schémas d'ambiguïtés*, c'est-à-dire une liste de telles règles de transformation. Ainsi lorsqu'un schéma ambigu de la liste est reconnu dans un texte analysé, on procède à son remplacement par le chemin solution qui lui est associé : ce remplacement permettant la linéarisation partielle du parcours analysé.

Nous donnons ci-après une définition plus précise des schémas d'ambiguïtés et de leur mode d'application, ainsi que les éléments principaux de la stratégie de mise en œuvre. Nous montrons par la suite l'utilisation de ces schémas d'ambiguïtés dans RIME.

a. définition

Les schémas d'ambiguïtés sont des règles de transformations qui se présentent classiquement sous la forme :

< partie gauche > --> < partie droite >

et dont l'interprétation est "si la partie gauche d'un schéma d'ambiguïté est trouvée dans un parcours, alors la remplacer dans le parcours par la partie droite du schéma" [BERR85].

La partie gauche d'un schéma d'ambiguïté est appelée *parcours ambigu*. Elle est composée par définition de plusieurs chemins, dont il faut détecter les occurrences dans les textes analysés. Nous définissons plus précisément un parcours ambigu comme un parcours dont tous les chemins commencent par un même nœud (catégorie grammaticale) appelé *point d'ancrage gauche*, et se terminent tous par un même nœud appelé *point d'ancrage droit* : la non-linéarité exprime une configuration syntaxiquement ambiguë. Nous formalisons un schéma ambigu *SA* en le décrivant comme un ensemble de n chemins c_k:

$SA = \{c_1, c_2, \ldots, c_n\}$

tel que tous les chemins soient issus du même point :

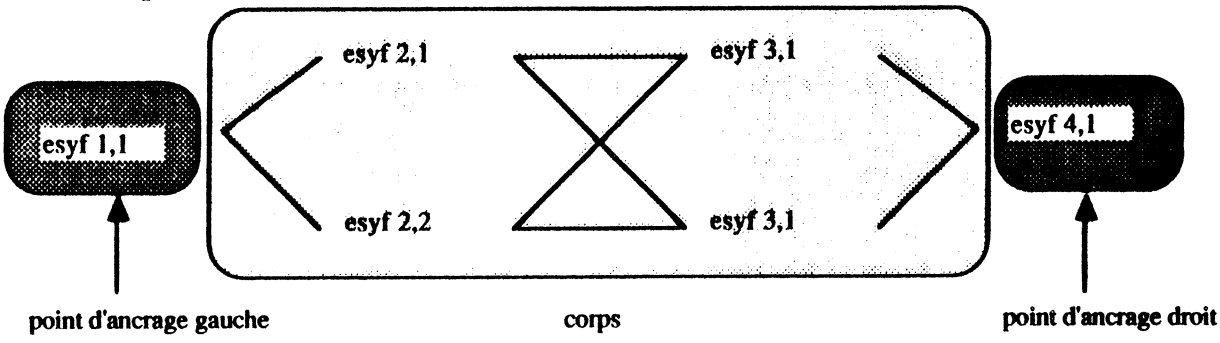
$card(esy_1) = 1$

tel que tous les chemins arrivent au même point , soit :

$card(esy_k) = 1$

et tel que $\exists k, k=1, \ldots, n-1$ *tel que* $card(esy_k))>1$

Par exemple,



avec SA = { c1, c2, c3, c4}

où c1 = (esyf 1,1, esyf 2,1, esyf 3,1, esyf 4,1)

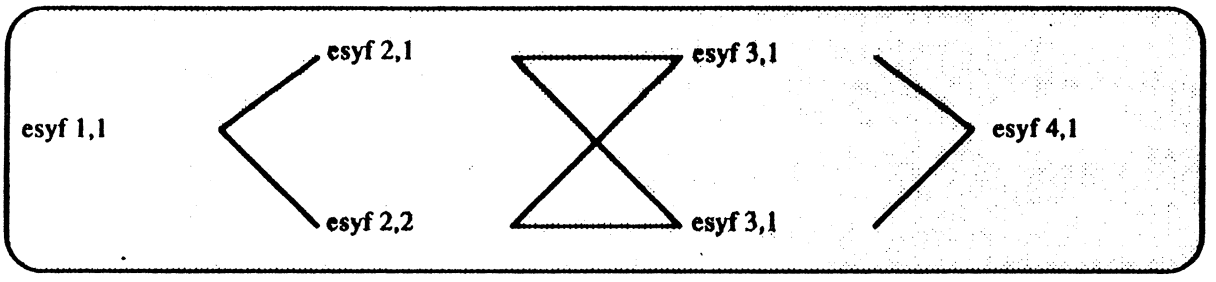
c2 = (esyf 1,1, esyf 2,1, esyf 3,2, esyf 4,1)

c3 = (esyf 1,1, esyf 2,2, esyf 3,1, esyf 4,1)

c4 = (esyf 1,1, esyf 2,2, esyf 3,2, esyf 4,1)

La partie droite du schéma d'ambiguïté est appelée *chemin solution*, et correspond à l'élément du schéma ambigu apparaissant comme la résolution de ce schéma. Le chemin solution *CHS* d'un schéma ambigu *SA* est donc un élément de *SA* : $CHS \in SA$. Le schéma ci-après nous indique le chemin solution du schéma d'ambiguïté précédent. La linéarisation des réseaux sous-entend qu'on dispose de plusieurs schémas d'ambiguïtés, correspondant à autant de configurations possibles. Lors du filtrage, cet ensemble est consulté et quand l'un d'eux est reconnu, il faut procéder à son remplacement par son chemin solution. Pour cela, il faut décrire la stratégie d'application des schémas : nous ne le faisons ici que succintement, et pour plus de détails nous référençons pour cette partie [BERR85] et [PALM88].

schéma ambigu :



chemin solution :

CHS = { esyf 1,1, esyf 2,1, esyf 3,1, esyf 4,1}

L'application des schémas est supervisée par un automate d'états finis qui active, désactive ou valide certains schémas selon l'évolution du parcours à filtrer. Ainsi, pour chaque transition, cet automate doit :
- permettre l'exploration de la liste des schémas d'ambiguïtés de manière à obtenir la liste des schémas dits *activés*, c'est-à-dire ceux dont la première transition correspond à la transition courante du parcours étudié ;
- de *désactiver* les schémas jusqu'alors activés et qui ne correspondent plus à l'état courant du parcours étudié ;
- de *valider* le schéma activé, s'il existe, qui se termine à la transition courante du parcours et qu'il faut donc appliquer en le remplaçant par son chemin solution.

b. application dans RIME

Les schémas d'ambiguïtés sont appliqués sur les résultats du premier filtrage par la matrice de précédence. Ce deuxième filtrage a pour but de simplifier les parcours demeurés ambigus à l'issue du premier filtrage. Rappelons que dans

RIME ce second filtrage ne doit s'effectuer que sur des zones très précises (cf ci-après), pour simplifier le travail de la syntaxe dans les autres tâches qui lui sont assignées.

Rappelons en effet que la syntaxe est chargée :

- du signal des effacements par coordination, comparatives et adjectifs possessifs ;
- du signal d'anaphores, et de la validation de certaines d'entre elles ;
- de la construction et nomination de structures syntaxiques.

Pour que ces tâches soient facilement menées à bien, il faut que les zones susceptibles de présenter de tels phénomènes linguistiques ne soient pas "trop" ambiguës, de manière à ce que le verdict syntaxique soit aussi fiable que possible. Pour cela nous devons étudier les possibilités d'ambiguïtés liées à chacune des tâches syntaxiques, et mettre en évidence les ambiguïtés qu'il est particulièrement intéressant à lever.

Le signal des effacements par coordination se traduit dans RIME par la succession d'une conjonction de coordination et d'une préposition, ou bien d'une virgule et d'une préposition. Il existe quelques cas d'ambiguïtés dans les conjonctions de coordination (*car* est une conjonction de coordination et un substantif commun), ainsi que dans les prépositions (*concernant* est en même temps une préposition et un verbe au participe présent) qu'il faut dans la mesure du possible essayer de lever en étudiant le contexte de ces occurrences dans nos textes.

De même, les autres signaux peuvent être "gênés" par certaines occurrences précises, qu'il s'agit d'étudier dans notre contexte médical de manière à pouvoir lever les ambiguïtés correspondantes. Nous avons ainsi relevé les ambiguïtés suivantes :

- *leurs* est un pronom et un adjectif possessif. Cette ambiguïté doit être levée pour faire la distinction entre une anaphore pronominale et un effacement par adjectif possessif. De la même façon, *ce* est un adjectif et un pronom démonstratif. Il faut résoudre cette ambiguïté pour faire la distinction entre une anaphore pronominale, et une anaphore nominale par répétition ;

- *leur* est un pronom possessif, un adjectif possessif ou un pronom préverbal. Il est nécessaire de lever l'ambiguïté entre la double possibilité pronom et la possibilité adjectif possessif, pour faire la distinction entre un signal d'effacement ou d'anaphore. Il ne nous est pas par ailleurs nécessaire dans RIME de faire la distinction entre les deux types de pronoms ;

- les articles définis et pronoms préverbaux *l', le les* engendrent également des ambiguïtés qu'il faut résoudre pour permettre la détection des anaphores pronominales, ou bien d'une partie de syntagme nominal.

Pour tous ces cas d'ambiguïtés, nous avons fait une étude systématique des occurrences de ces formes dans nos textes, et nous avons établi à partir de là l'ensemble des schémas d'ambiguïtés permettant une linéarisation partielle des parcours non-linéaires.

La syntaxe est par ailleurs également chargée de construire et de nommer des structures syntaxiques simples : syntagmes nominaux, verbaux ou prépositionnels. Pour simplifier cette tâche, nous essayons de lever les ambiguïtés au niveau des formes à catégories grammaticales pouvant appartenir à ces différents types de syntagmes : par exemple, *le* est un *article défini* que l'on retrouve dans un syntagme nominal, ou bien un *pronom préverbal* que l'on retrouve dans un syntagme verbal.

Pour cela, nous avons essentiellement repris le travail de P.Palmer sur la résolution des ambiguïtés dans les syntagmes nominaux, réalisé dans le cadre du projet IOTA. P.Palmer met en effet en œuvre des schémas d'ambiguïtés permettant la mise en évidence de syntagmes nominaux, jugés par ailleurs les plus porteurs sémantiquement dans le cadre d'une indexation de textes ouverts.

5.3.2. la construction et nomination de structures syntaxiques

Le but final de la traduction de RIME est de relier des concepts appelés *faits médicaux* extraits des comptes rendus, par des *articulateurs* également extraits de ces mêmes comptes rendus. La construction et la nomination de structures

syntaxiques permet de faire une première hypothèse de décomposition des textes analysés, de manière à en extraire une première proposition mettant en évidence les faits médicaux et leurs articulateurs. En effet, nous avons observé que d'une part les faits médicaux se retrouvent dans les syntagmes nominaux, et que d'autre part les articulateurs se retrouvent généralement dans les syntagmes verbaux et dans les syntagmes prépositionnels. Ainsi, la construction et nomination de structures syntaxiques doit nous permettre de mettre en évidence :

- les syntagmes nominaux ;
- les syntagmes prépositionnels, que nous avons décomposés en deux classes : les syntagmes prépositionnels dits "de type *de*", c'est-à-dire toutes les prépositions dérivées de la préposition *de* (*du, de, d', des*), et tous les autres syntagmes prépositionnels dits "non de type *de*". Cette distinction nous permet de faire la différence entre les prépositions de type *de* dont le trait sémantique est vide, et les autres prépositions qui ont un trait sémantique complexe et incomplet ;
- les syntagmes verbaux.

Cette analyse se fait par une simple reconnaissance de catégories grammaticales :

- un syntagme nominal est constitué de substantifs communs ou propres, d'adjectifs qualificatifs ou déterminatifs, d'articles, de verbes et auxiliaires au participe passé, de conjonctions de coordination, de pronoms personnels sujet, complément, possessifs, démonstratifs, relatifs, indéfinis, d'adverbes, de pronoms adverbiaux ;
- un syntagme verbal est constitué au moins d'un verbe ou d'un auxiliaire à l'infinitif, au participe présent ou conjugué. Il peut être accompagné de conjonctions, d'adverbes, de pronoms adverbiaux, et d'adverbes de négation.

Le résultat de cette analyse syntaxique peut se représenter par une arborescence à quatre niveaux, chacun de ces niveaux décrivant les quatre types de syntagmes reconnus, c'est-à-dire les syntagmes nominaux, les deux types de syntagmes prépositionnels, et les syntagmes verbaux :

- au niveau 4, le plus bas de l'arborescence, se trouvent les syntagmes nominaux simples. Ils sont considérés comme simples dans la mesure où ils ne comportent aucune préposition. Nous notons *nom* chacun des éléments des syntagmes de ce type : ainsi ce niveau peut s'exprimer de la façon suivante :

Niveau4_du_procsynt ::= [nom EXP_SYNTF(nom)]$^{+}$;

- au niveau 3 se trouvent les prépositions dites de type *de*, c'est-à-dire les prépositions *de, du, des, d'*, que nous appelons par la suite *prépde*. Elles relient les syntagmes du niveau 4 de l'arborescence de la façon suivante :

Niveau3_du_procsynt ::= Niveau4_du_procsynt [prépde EXP_SYNTF(prépde) Niveau4_du_procsynt]*

Nous appelons *représentation linéaire* cette représentation des syntagmes nominaux entre eux ;

- au niveau 2, se trouvent les syntagmes prépositionnels formés d'une part des locutions prépositionnelles, et d'autre part des prépositions qui ne sont pas de type *de* (que nous notons *prépnonde*) et qui relient là aussi de façon linéaire les structures du niveau 3 :

Niveau2_du_procsynt ::= Niveau3_du_procsynt
[prépnonde EXP_SYNTF(prépnonde) Niveau3_du_procsynt]* ;

- au niveau 1, le plus haut, se trouvent les syntagmes verbaux dont les éléments sont notés *verb*, et qui relient entre eux les structures du niveau 2 :

Niveau1_du_procsynt ::= Niveau2_du_procsynt [[verb EXP_SYNTF(verb)]$^{+}$ Niveau2_du_procsynt]* ;

Nous montrons sur la figure 19 un exemple de décomposition syntaxique de phrase.

5.3.3. le signal de tâches inter-structurelles

Les différents signaux attribués au processus syntaxique sont activés dans RIME dans des configurations syntaxiques très précises :

figure 19
un exemple de décomposition syntaxique de phrase

phrase :

opacité à disposition péri-bronchique de 25 mm de diamètre, en projection de la bronche lobaire supérieure droite et du tronc bronchique intermédiaire correspondant très probablement à la masse tumorale

nom :

opacité
disposition péri-bronchique
25 mm
diamètre
bronche lobaire supérieure droite
tronc bronchique intermédiaire
masse tumorale

prépde :

de
de
(et) du

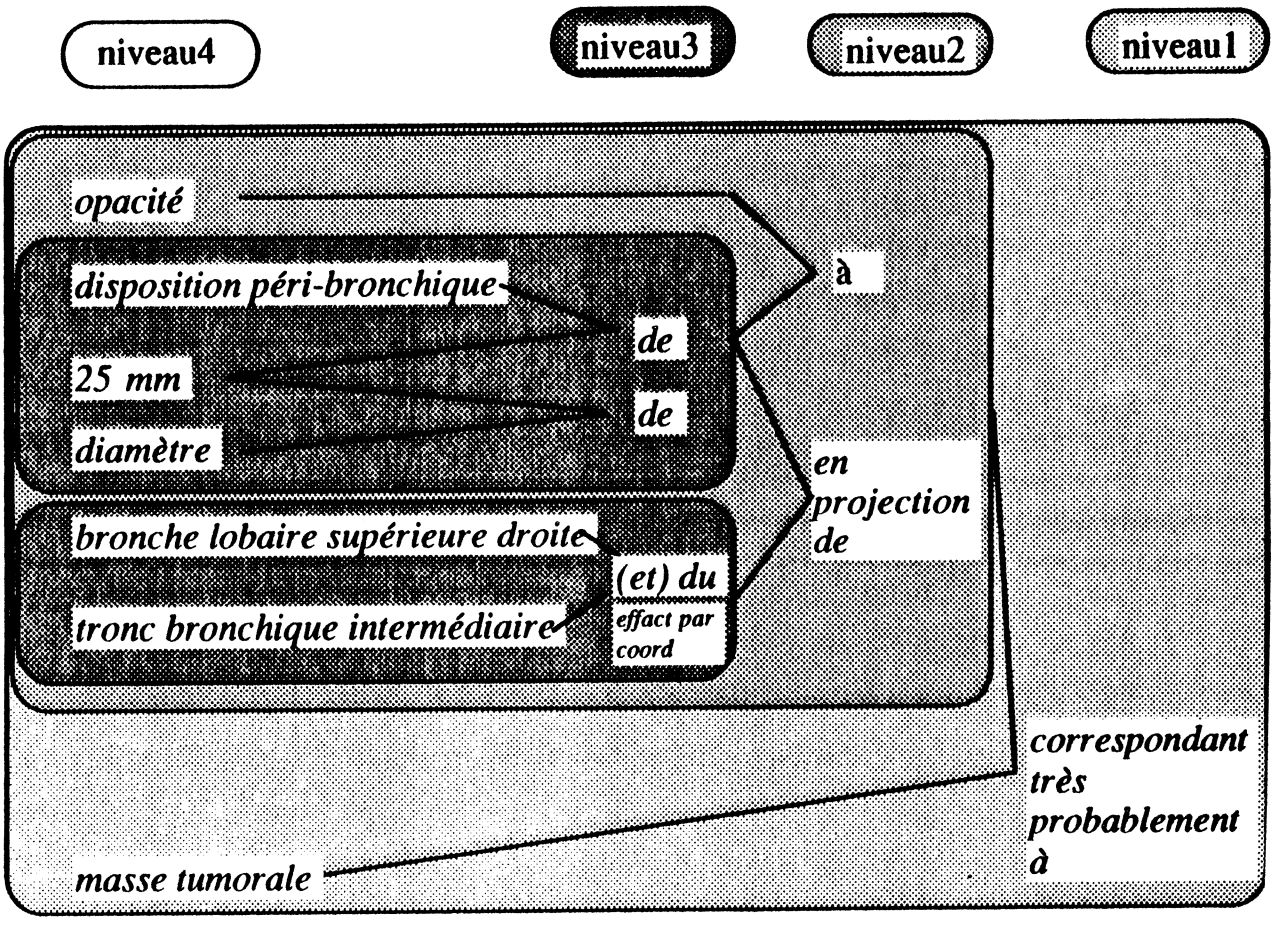
prépnonde :

à
en projection de

verb :

correspondant très probablement à

arborescence construite :



- les effacements par **coordination** sont signalés par l'occurrence d'une conjonction de coordination suivie d'un syntagme prépositionnel, ou bien d'une virgule suivie d'un syntagme prépositionnel ;

- les effacements par **comparatives** sont signalés par apparition de séquences *moins ... que* ou bien *plus ... que* ;

- les effacements par **adjectifs possessifs** sont signalés par occurrence d'adjectifs possessifs. De la même façon les **anaphores pronominales** sont signalées par occurrence de pronoms, et les **anaphores nominales par répétition** par l'occurrence d'adjectifs démonstratifs.

Ces signaux sont directement intégrés dans l'arborescence syntaxique au niveau même de chaque syntagme présentant l'une de ces particularités linguistiques. Par exemple, la coordination se retrouve dans les liaisons entre syntagmes nominaux par des syntagmes prépositionnels, c'est-à-dire aux niveaux 2 ou 3, les effacements par pronoms se trouvent aux niveaux 1 ou 4, etc. Dans l'exemple de la figure 19, il existe un signal d'effacement par coordination au niveau 2.

Ces différents signaux s'intègrent dans la grammaire que nous décrivions précédemment de la façon suivante :

Niveau4_du_procsynt ::=
[nom EXP_SYNTF(nom)]$^{+}$
/ [nom EXP_SYNTF(nom)]$^{+}$*signal_eff_comp*
/ [nom EXP_SYNTF(nom)]$^{+}$ *signal_ana_pro*
/ [nom EXP_SYNTF(nom)]$^{+}$*signal_eff_adj_po ;*

Niveau3_du_procsynt ::=
Niveau4_du_procsynt [Nœud3 Niveau4_du_procsynt]*
Nœud3 ::= prépde EXP_SYNTF(prépde)
/ prépde EXP_SYNTF(prépde) signal_eff_coord

Niveau2_du_procsynt ::=
Niveau3_du_procsynt [Nœud2 Niveau3_du_procsynt]*

Nœud2 ::= prépnonde EXP_SYNTF(prépnonde)
/ prépnonde EXP_SYNTF(prépnonde) signal_eff_coord ;

Niveau1_du_procsynt ::=
Niveau2_du_procsynt [Nœud1 Niveau2_du_procsynt] ;*
Nœud1 ::= [verb EXP_SYNTF(verb)]+
/ [verb EXP_SYNTF(verb)]+ signal_eff_comp
/ [verb EXP_SYNTF(verb)]+ signal_ana_pro
/ [verb EXP_SYNTF(verb)]+ signal_ana_nom_rép

5.3.4. la validation de tâches inter-structurelles

Par ailleurs, le processus syntaxique doit assurer la validation des anaphores pronominales.
Cette validation consiste à vérifier si dans toute paire telle que
((mot1, EXP_SYNTF(mot1)), (mot2, EXP_SYNTF(mot2)))
avec *EXP_SYNTF(mot)* $\subset$ *EXP_SYNT(mot)*
et *EXP_SYNT(mot) = {(cat_gram, inst_var_gram) / cat_gram* $\in$ *CAT_GRAM(mot) et inst_var_gram* $\in$ *VAL_GRAM(mot, cat_gram) }*,
mot1 est un interprétant syntaxiquement correct pour *mot2*.
La syntaxe valide cette résolution d'anaphore si et seulement si
inst_var_gram1 = inst_var_gram2.

5.4. l'analyse sémantique

Lors de l'attribution des tâches aux différents processus de RIME, le processus sémantique s'est vu attribuer :

- des tâches de niveau intra-structurel, en l'occurrence la construction et la nomination de structures respectant le modèle sémantique de RIME ;

- des tâches de niveau inter-structurel que sont les anaphores nominales par inclusion, et de la portée des opérateurs, ainsi que la confirmation et la résolution des signaux inter-structurels rémanents du processus syntaxique.

Pour décrire le processus sémantique de RIME, nous en présentons les deux parties constituées par l'*enveloppe sémantique*, et par le *noyau sémantique* :

- l'*enveloppe sémantique*, dans laquelle interviennent les tâches de niveau inter-structurel, propose des solutions aux tâches inter-structurelles détectées, et fait valider ses solutions par le *noyau sémantique*. Ces tâches sont matérialisées comme des méta-règles, permettant le guidage de la construction faite par le noyau sémantique ;

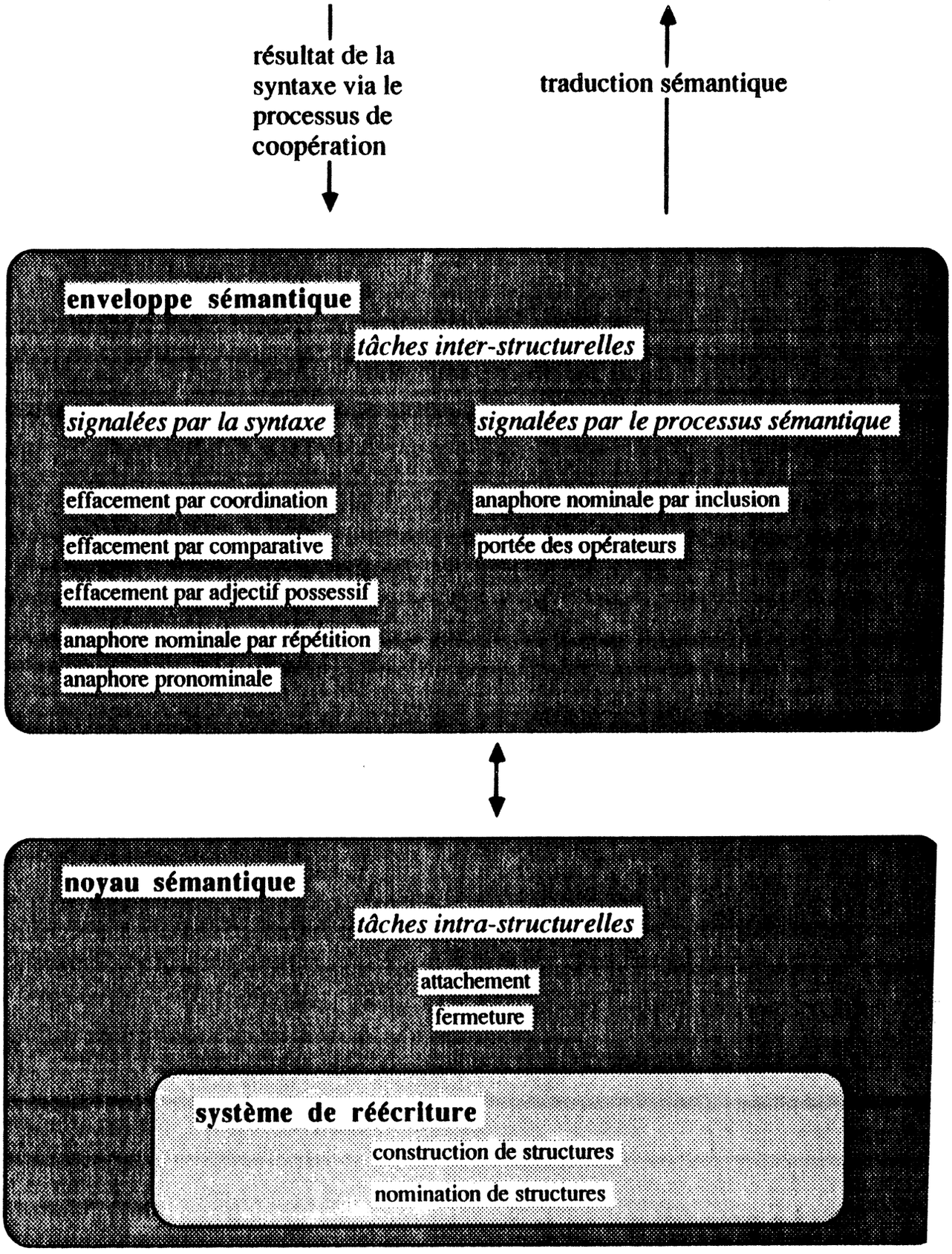
- le *noyau sémantique* admet en entrée des données ne comportant que des problèmes de niveau intra-structurel; le but est de résoudre ces tâches intra-structurelles, et de générer en sortie des arborescences respectant le modèle sémantique de RIME. Comme nous le montrons par la suite, cette génération du modèle sémantique de RIME se fait par unification de structures au travers d'un système de réécriture.

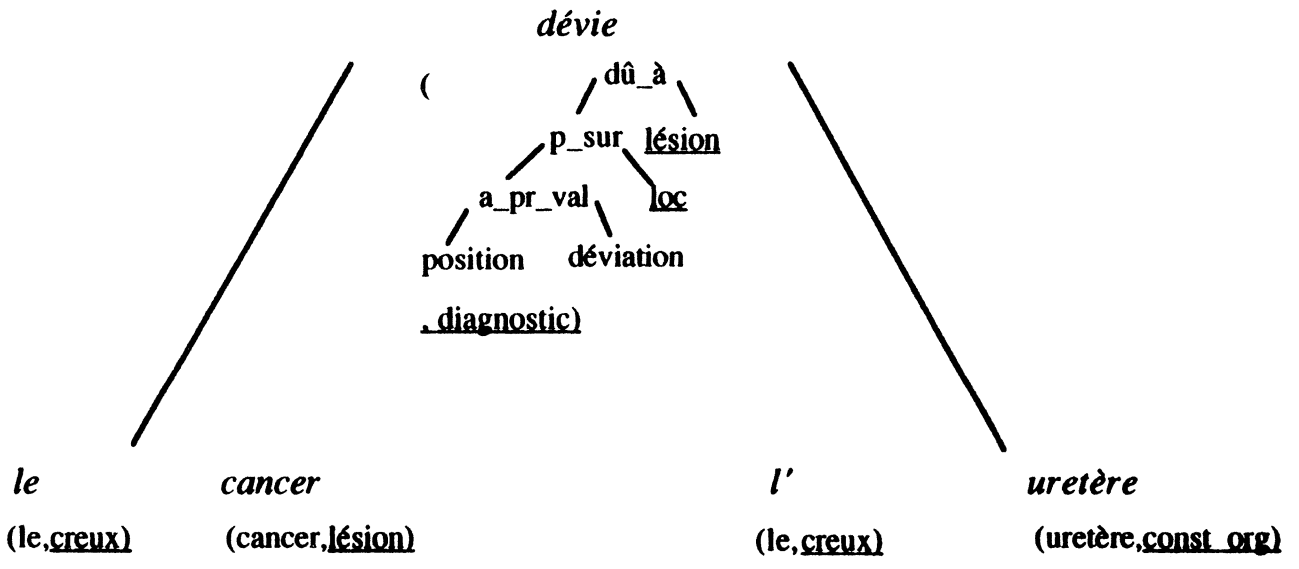
Nous présentons sur la figure 20 une vue globale du processus sémantique de RIME, vue dans laquelle apparaissent les deux composantes principales de ce processus, ainsi que les tâches qui leur sont attribuées.

Globalement, l'enveloppe sémantique traite le résultat de l'analyse syntaxique, qui se trouve sous la forme d'une arborescence comportant éventuellement des signaux syntaxiques de tâches inter-structurelles. L'enveloppe sémantique se doit de confirmer les signaux syntaxiques, de signaler à son tour les tâches inter-structurelles attribuées au processus sémantique, et de trouver une solution à tous ces signaux. Quand l'enveloppe sémantique considère avoir rempli ce rôle, elle appelle le noyau sémantique en lui donnant en entrée une proposition d'arborescence ne comportant plus que des tâches de niveau intra-structurel.

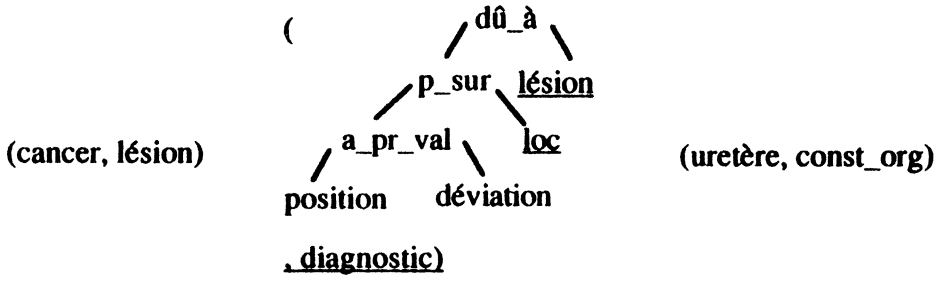
Exemple : si l'on reprend l'exemple donné au début du chapitre (partie 5.1), une donnée du processus sémantique de RIME peut être :

figure 20 : le processus sémantique de RIME

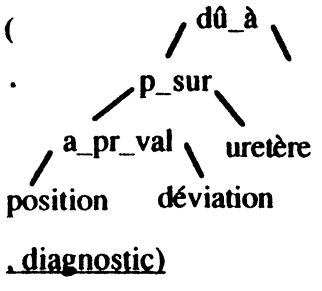




L'enveloppe sémantique traite cette arborescence, et active le noyau sémantique en lui donnant en entrée :



le noyau sémantique traite cette donnée, et donne en résultat :



qui sera considéré comme le résultat final du processus sémantique.

Nous allons, dans les parties 5.4.1 et 5.4.2, donner plus en détail le contenu de chacune des deux parties du processus sémantique.

5.4.1. l'enveloppe sémantique

L'enveloppe sémantique constitue la couche du processus sémantique dans laquelle interviennent les tâches inter-structurelles traitées dans RIME.
Après avoir récupéré une entrée dérivée de l'arborescence résultat du processus syntaxique, l'enveloppe sémantique doit d'une part confirmer les signaux inter-structurels de la syntaxe, et d'autre part mettre en évidence les tâches inter-structurelles qui ont été attribuées au processus sémantique. Par ailleurs, après avoir fourni ce travail inter-structurel, l'enveloppe sémantique appelle le noyau sémantique pour lui faire traiter des structures arborescentes ne comportant que des problèmes intra-structurels qui sont de sa compétence. L'enveloppe sémantique répercute par la suite les tâches inter-structurelles sur le résultat du noyau sémantique.
Les entrées de l'enveloppe sémantique sont des arborescences à au plus quatre niveaux, dérivées par le processus de coopération à partir de l'arborescence résultat de l'analyse syntaxique. Ces entrées sont donc représentées par des arborescences contenant tous les signaux inter-structurels issus du travail syntaxique, et dont les feuilles et les nœuds sont constitués d'expressions sémantiques, et éventuellement de signaux inter-structurels. La transformation opérée par le processus de coopération sur le résultat de la syntaxe peut se résumer comme une substitution des possibilités syntaxiques par leur expression sémantique. Nous reprendrons cette transformation du processus de coopération dans la partie 5.5.
Ces arborescences données en entrée de l'enveloppe sémantique correspondent à la grammaire suivante :

Niveau4_de_envsém ::=
[nom EXP_SEM_DED(nom)]$^+$
/ [nom EXP_SEM_DED(nom)]$^+$ *signal_eff_comp*
/ [nom EXP_SEM_DED(nom)]$^+$ *signal_ana_pro*
/ [nom EXP_SEM_DED(nom)]$^+$ *signal_eff_adj_po*
/ [nom EXP_SEM_DED(nom)]$^+$ *signal_ana_nom_rép ;*
avec *EXP_SEM_DED(mot)* $\subset$ *EXP_SEM(mot) ;*

Niveau3_de_envsém ::=
Niveau4_de_envsém [Nœud3 Niveau4_de_envsém] ;*
Nœud3 ::= prépde EXP_SEM_DED(prépde) | prépde
EXP_SEM_DED(prépde) signal_eff_coord ;

Niveau2_de_envsém ::=
Niveau3_de_envsém [Nœud2 Niveau3_de_envsém] ;*
Nœud2 ::= prépnonde EXP_SEM_DED(prépnonde)
| prépnonde EXP_SEM_DED(prépnonde) signal_eff_coord ;

Niveau1_de_envsém ::=
Niveau2_de_envsém [Nœud1 Niveau2_de_envsém] ;*
Nœud1 ::= [verb EXP_SEM_DED(verb)]+
| [verb EXP_SEM_DED(verb)]+ signal_eff_comp
| [verb EXP_SEM_DED(verb)]+ signal_ana_pro ;

L'enveloppe sémantique traite de telles arborescences, pour ensuite activer le noyau sémantique, qui, après éventuellement plusieurs échecs, donne un résultat final. L'enveloppe sémantique récupère ce résultat du noyau sémantique, considéré comme la traduction de l'arborescence donnée en entrée du processus sémantique.

Pour parvenir à ce résultat, l'enveloppe sémantique analyse l'arborescence qui lui est donnée en entrée niveau par niveau de manière à construire pour chacun une arborescence compatible pour le noyau sémantique. Cette construction est obtenue par un repérage des tâches inter-structurelles dans l'arborescence initiale. Ces tâches peuvent avoir été signalées par le processus syntaxique, ou bien avoir été entièrement laissées au processus sémantique :

- chaque signal inter-structurel émanant de la syntaxe est examiné pour être sémantiquement soit infirmé, soit confirmé. Dans ce second cas, l'enveloppe sémantique doit être à même de proposer des résolutions à ces tâches inter-structurelles ;

- de la même façon, chaque occurrence de tâches inter-structurelles confiées au processus sémantique doit être résolue.

Ces résolutions permettent par la suite d'interroger le noyau sémantique sur des propositions ne contenant que des tâches intra-structurelles.

5.4.1.1. confirmation et résolution des signaux inter-structurels du processus syntaxique

Les arborescences données en entrée de l'enveloppe sémantique contiennent à des niveaux précis cinq types de signaux inter-structurels issus du travail syntaxique :

- des signaux d'effacements par coordination aux niveaux 2 et 3 des arborescences ;
- des signaux d'effacements par comparatives aux niveaux 1 et 4 des arborescences ;
- des signaux d'effacements par adjectifs possessifs au niveau 4 des arborescences ;
- des signaux d'anaphores nominales par répétition au niveau 4 des arborescences ;
- des signaux d'anaphores pronominales aux niveaux 1 et 4 des arborescences.

Chacun de ces signaux doit être

- confirmé, dans la mesure où ce ne sont pour la plupart que des hypothèses syntaxiques de tâches inter-structurelles. L'enveloppe sémantique, au travers de critères spécifiques à chacune des tâches, se doit ainsi de valider chacun des signaux ;
- résolu, lorsque la tâche en question a été confirmée par l'enveloppe sémantique.

Pour cela, nous allons reprendre un à un chacun des signaux syntaxiques, et montrer dans quelle mesure l'enveloppe sémantique les valide et les résoud.

a. effacement par coordination

L'effacement par coordination peut apparaître aux niveaux 2 et 3 des arborescences.

Au niveau 3 des arborescences où l'on trouve des syntagmes nominaux reliés par des prépositions de type *prépde*, la sous-arborescence contenant l'effacement est donc de type *A prépde Signal_eff_coord B*. L'effacement se trouve confirmé par l'enveloppe sémantique dans deux cas :

- si et seulement si *A* est de type *A' prépde A"*. Et dans ce cas, il faut corriger l'effacement en travaillant sur *(A' prépde A")* et *(A' prépde B)* ;

- ou bien si *A prépde Signal_eff_coord B* est sous-arborescence gauche d'un niveau 2, c'est-à-dire que l'on ait *C prépnonde (A prépde Signal_eff_coord B)*. Dans ce cas, il faut corriger l'effacement en travaillant sur *C prépnonde A* et *C prép B* ou *C prépde B* si le précédent crée une impossibilité au niveau du noyau sémantique ;

Au niveau 2 des arborescences, où l'on trouve des syntagmes de niveau 2 reliés par des prépositions de type *prépnonde*, la sous-arborescence contenant l'effacement est de type *A prépnonde Signal_eff_coord B*. L'effacement se trouve confirmé par l'enveloppe sémantique

- si et seulement si *A* est de niveau 2 ou 3, c'est-à-dire de type *A' prép A"*, où *prép* est de type *prépde* ou *prépnonde*. Et dans ce cas, il faut corriger l'effacement en travaillant sur *(A' prép A")* et *(A' prépnonde B)* ;

- ou bien si *A prépnonde Signal_eff_coord B* est sous-arborescence gauche d'un niveau 1, c'est-à-dire que l'on ait *C verb (A prépnonde Signal_eff_coord B)*. Dans ce cas, il faut corriger l'effacement en travaillant sur *C verb A* et *C verb prépnonde B* ou *C prépnonde B* si le précédent crée une impossibilité au niveau du noyau sémantique.

b. effacement par comparative

Les effacements par comparatives se retrouvent dans des arborescences telles que *A plusoumoins B que C*. L'effacement se trouve systématiquement confirmé, et de telles arborescences sont dans la version actuelle de RIME

traitées de façon simplifiée : on traite indépendamment *AB* et *AC*. Or dans la version actuelle du modèle sémantique de RIME, il n'existe aucun lien sémantique permettant de représenter la comparaison entre deux structures, *AB* et *AC* sont donc pour le moment reliés par l'opérateur ***et***.

c. effacement par adjectif possessif

L'effacement par adjectif possessif ne peut être que systématiquement confirmé par le processus sémantique, dont le rôle à ce niveau consiste à trouver un interprétant à l'élément effacé par l'adjectif possessif.
Pour cela, le processus sémantique travaille de la façon suivante :

- soient *A* le syntagme comportant l'adjectif possessif, *CAT_SEM(A)* sa catégorie sémantique ;
- soit *R* une règle de la grammaire telle que *R ::= [op, A', C']* ou *R ::= [op, C', A']* avec *CAT_SEM(A)* soit inclus dans *A'*.

Le travail du processus sémantique consiste à trouver dans le contexte gauche de *A* le premier syntagme *C* tel que *CAT_SEM(C)* soit inclus dans *C'*. Dès lors, la partie *A Signal_eff_adj_pos* est substituée par *A prépde C* dans l'arborescence construite par l'enveloppe sémantique.

Si une telle décision crée une impossibilité au niveau du noyau sémantique, l'enveloppe sémantique revient sur sa décision dans plusieurs cas :

- tout d'abord, il peut exister plusieurs règles de la grammaire *R* répondant aux critères donnés ci-dessus. Il faut donc réessayer sur une nouvelle règle ;
- ensuite, il peut exister un autre syntagme *C*, situé plus à gauche de *A* et répondant aux critères énoncés ci-dessus.

d. anaphore nominale par répétition

L'anaphore nominale par répétition est confirmée par l'enveloppe sémantique de la façon suivante : soit *A* le syntagme nominal contenant l'anaphore repérée par le processus syntaxique, l'anaphore est confirmée si et seulement

si il existe dans le contexte gauche de *A* un syntagme *A'* de trait sémantique incluant celui de *A*.

e. anaphore pronominale

Tout comme pour l'effacement par adjectif possessif, l'enveloppe sémantique confirme systématiquement les anaphores pronominales. Son rôle à ce niveau consiste à tout d'abord trouver dans le contexte de l'anaphore détectée un interprétant potentiel, et à le faire valider par le processus syntaxique.
La procédure utilisée pour trouver un interprétant potentiel est la même que celle utilisée pour les effacements par adjectifs possessifs. De plus ici, l'enveloppe sémantique demande au processus syntaxique une validation quant à son choix : pour cela le processus sémantique transmet au processus syntaxique, via le processus de coopération, la paire

((interprétant_potentiel, EXP_SEM_DED(interprétant_potentiel)), (pronom, EXP_SEM_DED(pronom))).

5.4.1.2. tâches inter-structurelles du processus sémantique

Parallèlement à cette phase de confirmation et de résolution des tâches inter-structurelles signalées par le processus syntaxique, l'enveloppe sémantique doit également mettre en évidence et résoudre les tâches inter-structurelles qui lui ont été confiées lors de la distribution des tâches dans RIME. Il s'agit là de traiter et de trouver des solutions aux anaphores nominales par inclusion et au problème de la portée des opérateurs.

a. anaphore nominale par inclusion

L'anaphore nominale par inclusion apparaît dans nos documents lors de l'utilisation de certains mots, tels que *lésion*, *examen*, ou *diagnostic*, à sens très général. Ces mots représentent une forme d'instantiation de termes utilisés précédemment dans le texte : par exemple, lors d'utilisation de séquence *le cancer ... la lésion.*

Il faut dans ce cas retrouver le premier terme de catégorie sémantique inclus dans celle de l'anaphore, et situé dans le contexte gauche de celle-ci.

b. portée des opérateurs

La portée des opérateurs de négation ou d'éventualité est traitée dans l'indexation de RIME de manière simplifiée. La phase d'interrogation aborde par ailleurs ce problème de façon beaucoup plus fine, de manière à permettre une compréhension précise des requêtes des utilisateurs [NIE88].
Ces opérateurs portent sur des faits médicaux, c'est-à-dire des syntagmes dont la tête est un fait médical. Notre approche consiste à faire systématiquement répercuter l'opérateur sur la tête de syntagme au travers des règles de la grammaire contenant l'opérateur sémantique *DEGRE*, et à faire appel au noyau sémantique avec ce fait médical pondéré par ce biais.

5.4.1.3. appel du noyau sémantique

Après ce traitement inter-structurel, les arborescences initiales sont simplifiées, puisque les signaux qu'elles contenaient n'apparaissent plus. Les arborescences traitées à ce stade de l'analyse vérifient donc la grammaire suivante :

Niveau4_de_envsém ::=
[nom EXP_SEM_DED(nom)]$^{+}$
avec *EXP_SEM_DED(mot)* $\subset$ *EXP_SEM(mot)* ;

Niveau3_de_envsém ::=
Niveau4_de_envsém [prépde EXP_SEM_DED(prépde)
Niveau4_de_envsém]* ;

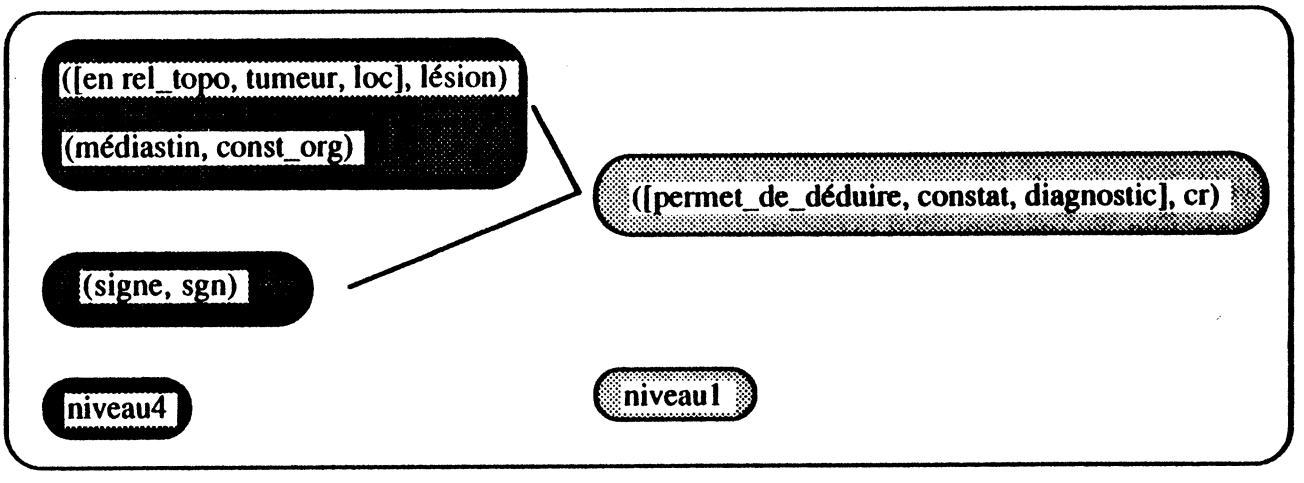
Niveau2_de_envsém ::=
Niveau3_de_envsém [prépnonde
EXP_SEM_DED(prépnonde)Niveau3_de_envsém]* ;

Niveau1_de_envsém ::=

Niveau2_de_envsém [[verb EXP_SEM_DED(verb)]+

Niveau2_de_envsém] ;*

Par exemple,

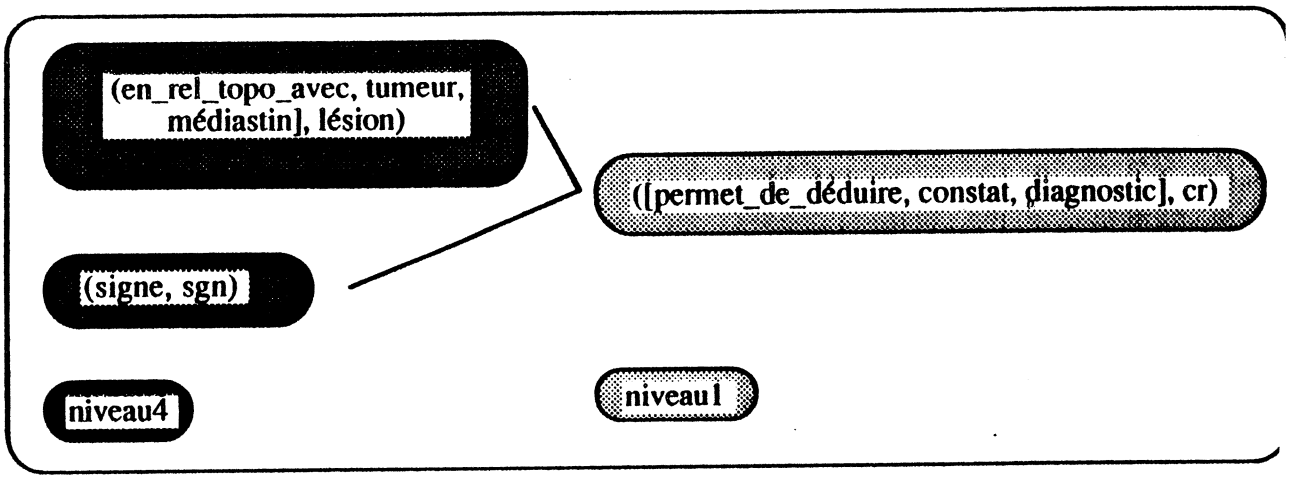


L'enveloppe sémantique va activer le noyau sémantique sur une lecture *niveau par niveau* de ces arborescences.

Dans un premier temps, l'enveloppe sémantique donne au noyau sémantique tous les groupes de plus d'un élément de niveau 4 de l'arborescence. Les groupes de niveau 4 composés d'un seul élément sont en effet corrects vis à vis du modèle sémantique, puisqu'ils sont directement issus du lexique. Les éléments de chaque groupe transmis sont séparés par *()*, ce qui représente un lien sémantique vide entre les éléments. C'est-à-dire dans notre exemple, *([en_rel_topo, tumeur, loc], lésion) () (médiastin, const_org).*

Le noyau sémantique donne la traduction de chacun d'eux dans le modèle sémantique pour chacun d'entre eux. Ce qui dans notre exemple se traduit par *([en_rel_topo, tumeur, médiastin], lésion).*

Chaque résultat du noyau sémantique est reporté dans l'arborescence donnée initialement. Dans notre exemple, après le traitement des niveaux 4, nous obtenons :



Les niveaux 4 étant individuellement traduits dans le modèle sémantique, l'enveloppe sémantique va traiter les groupes de niveau 3, en proposant au noyau sémantique des *structures sémantiques* qui sont une représentation linéaire des sous-arborescences traitées. Les arborescences traitées sont :

- soit des feuilles *f* isolées, que l'on ne transmet pas au noyau sémantique, puisqu'elles ont été traitées au niveau 4. Dans l'exemple ci-dessus, nous avons deux feuilles isolées : *([en_rel_topo, tumeur, médiastin], lésion)* et *(signe, sgn)*;

- soit des arborescences composées de deux feuilles $f1$ et $f2$, et d'une racine n_2 de niveau 3, qui sont transmises au noyau sémantique sous la forme d'une structure sémantique $struct = f_1\ n_2\ f_2$.;

- soit des arborescences composées de m feuilles (m>2) et de m-1 racines de niveau 3, qui sont transmises au noyau sémantique sous la forme d'une structure sémantique $struct = f_1\ n_2 f_2\ n_2 \ldots n_m f_m$.

Le noyau sémantique traduit chacune des structures qui lui sont transmises, et chacun des résultats qu'il fournit est reporté dans l'arborescence initiale.

L'enveloppe procède ainsi de suite pour les niveaux 2 puis 1, jusqu'à obtenir une traduction finale.

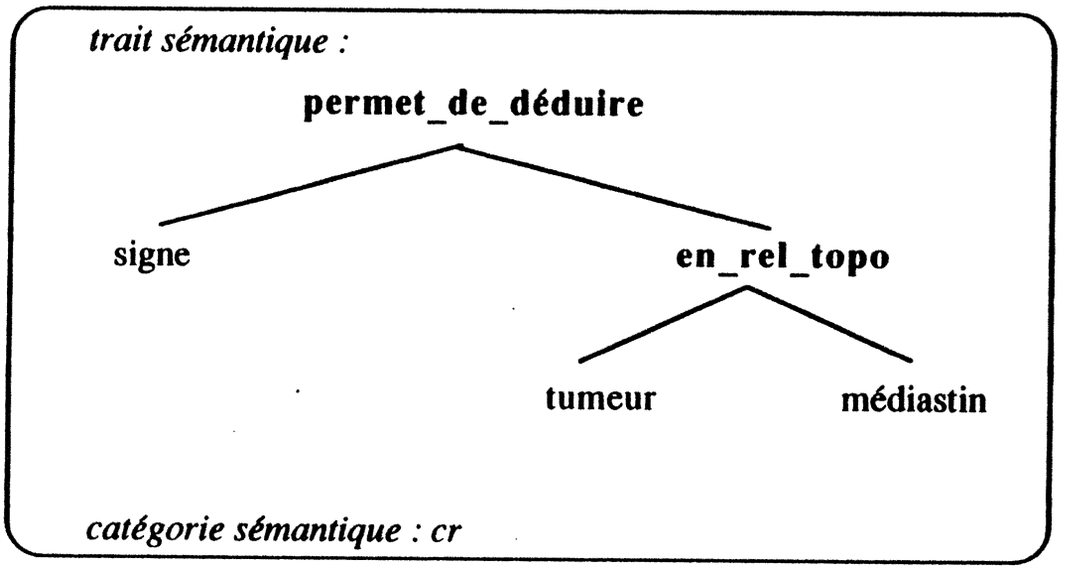
Ainsi dans notre exemple, l'enveloppe sémantique active le noyau sémantique avec une structure $struct = n_1\ f_2\ n_2$, telle que

$n1$ = *([en_rel_topo, tumeur, médiastin], lésion)*,

f_2 = *([permet_de_déduire, constat, diagnostic], cr)*,

et n_2 = *(signe, sgn)*.

Le noyau sémantique donne en résultat



5.4.2. le noyau sémantique

Le but de la traduction de RIME est de construire et de nommer des structures correspondant à son modèle sémantique : nous voulons identifier des lésions, des signes, ou bien des diagnostics et éventuellement les lier entre eux.
Le noyau sémantique de RIME est chargé de cette traduction à partir des propositions de l'enveloppe sémantique, propositions ne contenant que des problèmes de niveau intra-structurel : sur une décomposition donnée par l'enveloppe sémantique, le noyau sémantique construit et nomme des structures selon le modèle sémantique de RIME, et doit résoudre les problèmes intra-structurels que sont la fermeture et l'attachement. Pour cela, nous utilisons deux outils :

- d'une part un jeu de règles permettant l'appel d'un système de réécriture, et de le relancer en cas d'échecs dûs à des problèmes d'attachement ou de fermeture (partie 5.4.2.1) ;

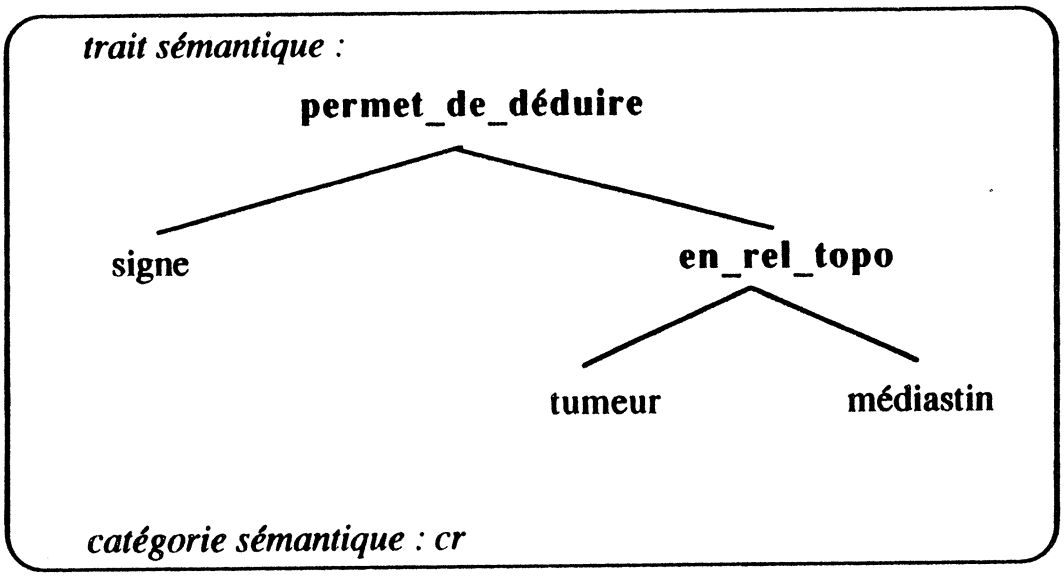
- d'autre part la grammaire du modèle sémantique de RIME représentée au travers d'un système de réécriture (partie 5.4.2.2).

Globalement, le noyau sémantique est activé par l'enveloppe sémantique, qui lui transmet des structures sémantiques telle que *struct* = $n_1 f_2 n_2$, avec :

*n1 = ([**en_rel_topo**, tumeur, médiastin], lésion),*
f_2 *= ([**permet_de_déduire**, constat, diagnostic], cr),*
et n_2 *= (signe, sgn).*

Cette première proposition est traitée par le système de réécriture qui tente la traduction dans le modèle sémantique de cette structure.

Ainsi dans notre exemple, le système de réécriture génère le résultat :



Dans certains cas, le système de réécriture aboutit à un échec, et le travail du jeu de règles consiste à résoudre les problèmes intra-structurels de la structure initale, de manière à pouvoir relancer le système de réécriture sur une nouvelle proposition.

5.4.2.1. appel du système de réécriture, et résolution intra-structurelle en cas d'échec

a. présentation

Ce processus constitue la couche intra-structurelle du noyau sémantique, et son rôle essentiel est d'appeler le système de réécriture, de récupérer son résultat

et éventuellement de le réappeler en cas d'impossibilités de résoudre des problèmes intra-structurels.
L'entrée de cette couche est une structure sémantique, fournie par le processus syntaxique, et qui ne contient que des problèmes de niveaux intra-structurels : ainsi les problèmes à résoudre à ce niveau sont des problèmes de fermeture et d'attachement.
Le résultat de ce processus est une arborescence conforme à la grammaire de RIME, et correspondant à la traduction de l'arborescence donnée en entrée.
Pour passer de l'un à l'autre, cette couche du noyau sémantique travaille en deux temps :

- un temps d'activation du système de réécriture ;
- lorsque le système de réécriture donne une réponse négative, il faut prévoir son activation sur une nouvelle proposition, déduite d'une résolution des problèmes intra-structurels que contenait la première proposition.

Chacun des éléments de la structure sémantique initiale est soumis au système de réécriture. Soit $f_1\ n_2 \ldots n_m f_m$ l'un de ces éléments, les f_i représentent les feuilles de l'élément, et n_i les nœuds : le système de réécriture est interrogé en lui donnant une telle entrée.

Par ailleurs, on observe dans certaines structures, des feuilles correspondant à des traits sémantiques complexes et incomplets (ce sont généralement des nominalisations de verbes). Aussi faut-il les positionner correctement, c'est-à-dire en nœuds, dans la proposition faite au système de réécriture.
Ainsi, dans l'exemple que nous donnons, le noyau sémantique est interrogé dans un premier temps sur

*([**en_rel_topo**, tumeur, localisation], sgn) () ([médiastin], const_org)*

où *([**en_rel_topo**, tumeur, localisation], sgn)* , situé en feuille, a un trait sémantique complexe et incomplet.
Cette structure est donc transformée en

*() ([**en_rel_topo**, tumeur, localisation], sgn) ([médiastin], const_org)*

avant d'être soumise au système de réécriture.

b. résolution intra-structurelle

Lorsque le système de réécriture rend un résultat négatif, il faut le réinterroger sur une nouvelle proposition. La négation est dûe à des problèmes intra-structurels, c'est-à-dire soit un problème d'attachement, soit un problème de fermeture. Face à ces problèmes, le noyau essaie dans un premier temps de garder entièrement les syntagmes en faisant de nouvelles propositions de regroupement. Cependant si le système de réécriture rend à nouveau un résultat négatif, le noyau n'a pas d'autre solution que de casser le syntagme de manière à ce que ses différentes parties hétérogènes soient dissociées pour par la suite être à nouveau associées à d'autres syntagmes.

Le problème est de savoir lorsque l'on traite une structure du type $f_1\ n_2\ f_2\ n_3\ f_3$, si l'on doit le comprendre comme $(f_1\ n_2\ f_2)\ n_3\ f_3$ ou bien comme $f_1\ n_2\ (f_2\ n_3\ f_3)$ Lors de la première activation du système de réécriture, la version $(f_1\ n_2\ f_2)\ n_3\ f_3$ est testée, et peut éventuellement être rejetée. Il faut alors demander le test de $f_1\ n_2\ (f_2\ n_3\ f_3)$. Si le résultat est également négatif, le syntagme initial doit être scindé en différents syntagmes. Cette cassure doit cependant essayer de rendre des structures aussi grandes que possible, et pour cela le résultat donné par le noyau sémantique est le premier des trois résultats qui s'avère vrai entre $(f_1\ n_2\ f_2)$ et f_3 ou bien f_1 et $(f_2\ n_3\ f_3)$, ou sinon f_1 et f_2 et f_3.

5.4.2.2. le système de réécriture

La grammaire du modèle sémantique de RIME est représentée dans un système de réécriture présentant les propriétés de confluence et de terminaison intéressantes quant à notre processus de traduction.
Cet outil est chargé de traiter les informations qui lui sont données en entrée, de manière à fournir soit une réponse positive sous la forme d'une traduction selon le modèle sémantique de RIME, soit une réponse négative dans le cas où l'entrée fournie ne remplit pas les conditions nécessaires à sa traduction.

a. principes

Nous avons décidé de confier la réalisation d'une partie de la traduction à un système de réécriture du fait des propriétés de *confluence* et de *terminaison* qui permettent d'en contrôler le fonctionnement, et du fait des possibilités de mise à jour aisée du système de règles, très précieuses dans le contexte de la mise au point d'un modèle linguistique [YAN86], [RAMO87].
Nous ne voulons pas ici reprendre la théorie des systèmes de réécriture, et pour cela nous référençons [KNUT & BEND70], [JOUA & LESC86], [HUET81]. Nous nous limitons simplement à présenter les quelques principes des systèmes de réécriture qui nous paraissent intéressants, sans pour autant trop entrer dans le formalisme et le vocabulaire détaillés de ces systèmes.

On appelle *système de réécriture* tout ensemble R de paires de termes <A,D> avec une condition : $V(A) \subset V(D)$ (V : ensemble de variables). On appelle règles de réécriture une expression de la forme : A --> D.
Les systèmes de réécriture sont des outils de décision intéressants notamment lorsqu'ils possèdent deux propriétés particulières : la terminaison et la confluence :

- la propriété de *terminaison* exprime que tous les calculs faits par un système de réécriture possédant cette propriété sont finis. Bien que cette propriété soit généralement indécidable, il existe des algorithmes qui permettent de l'établir à l'aide de l'orientation des règles de réécriture en suivant un ordre de simplification ;

- par ailleurs, un système de réécriture est *confluent* s'il a la propriété de Church-Rosser qui exprime que, quel que soit l'ordre dans lequel les règles de réécriture sont appliquées, si l'on obtient une forme normale, alors elle est unique. Nous appelons forme normale, tout résultat du système de réécriture sur lequel il est impossible de réappliquer des règles de réécriture (et donc de le modifier).

Nous avons donc décidé d'utiliser un système de réécriture pour implanter le processus de génération sémantique, de manière à ce que nous soyons sûrs

d'obtenir d'une part des résultats en assurant la propriété de terminaison, et d'autre part des résultats uniques par la propriété de confluence, même si les chemins d'accès à une même solution peuvent être différents [RAMO87].

b. utilisation

b.1. introduction

L'entrée du système de réécriture est une structure sémantique comportant deux types d'éléments :

- ce que nous appelons les *feuilles* notées f_i sont des expressions sémantiques ;

- ce que nous appelons les *nœuds*, notés n_j, qui peuvent être des expressions sémantiques à trait sémantique complexe et incomplet, qui sont à instancier par les feuilles de la structure sémantique, dans ce cas ces nœuds expriment les liens explicites entre les feuilles de la structure qu'ils relient. Les nœuds peuvent être également vides, dans le cas où les liens entre les feuilles sont implicites, et il faut pour résoudre ces cas prévoir des liens par défaut entre les feuilles qu'ils relient. Cette différence est une rémanence du niveau syntaxique où nous faisons la distinction entre des niveaux où interviennent des liens représentés par des mots creux entre syntagmes - par exemple les prépositions de type *de* -, et des niveaux où interviennent des liens représentés par des prépositions qui ne sont pas de type *de* ou bien des syntagmes verbaux. Par conséquent, une structure soumise au système de réécriture ne peut contenir que des nœuds explicites ou que des nœuds implicites, mais en aucun cas elle ne peut contenir ces deux types de nœuds simultanément.

Soient *struct* une structure fournie au système de réécriture, f_i ses feuilles, et n_i ses nœuds, où i ∈ [1 .. m], on a

$$struct = f_1\ n_2\ f_2 \ldots\ n_m f_m.$$

Pour une telle structure, le système de réécriture tente de construire et de donner le résultat déduit de la construction $((\ldots(f_1\ n_2\ f_2)\ \ldots)\ n_m f_m)$.

Par exemple, soit *struct* une structure à liens explicites, on a

struct1 = ([cancer], lésion) ([dû_à, [p_sur, [a_pr_val, position, déviation], localisation], diagnostic], diagnostic) ([uretère], const_org)
où *f1 = (cancer, lésion),*
n2 = ([dû_à, [p_sur, [a_pr_val, position, déviation], localisation], diagnostic], diagnostic)
et *f2 = ([uretère], const_org),*
où *localisation* et *diagnostic* représentent les valeurs à instancier du lien explicite n_2.

Soit *struct2* une structure à liens implicites, on a par exemple

struct2 = ([cancer], lésion) () ([poumon], const_org)

où *()* représente un lien implicite ; structure pour laquelle il faut rechercher dans la grammaire du modèle sémantique une façon de lier une *lésion* et un *const_org*, de manière à lui attribuer d'une part un trait sémantique et d'autre part une catégorie sémantique.

Le but du système de réécriture est d'associer les éléments de la structure donnée en entrée de manière à obtenir une expression sémantique, c'est-à-dire une catégorie sémantique et un trait sémantique qui peut ne pas être totalement instancié. Il se peut que cet objectif ne soit pas atteint, et dans ce cas le système de réécriture renvoie une réponse négative.
Pour cela, le système de réécriture cherche s'il est possible d'associer de gauche à droite les éléments de la structure, c'est-à-dire qu'à partir de la donnée

$$struct = f_1\ n_2 f_2 \ldots n_m f_m$$

il construit l'expression sémantique $v_1 = f_1$, puis essaie de construire v_2 à partir de $f_1\ n_2 f_2$, puis v_3 à partir de $v_2\ n_3 f_3$ et ainsi de suite chaque $v_i = v_{i-1}\ n_i f_i$, jusqu'à $v_m = v_{m-1} n_m f_m$. Pour comprendre ce processus, il est nécessaire d'expliquer la notion de construction dans notre système de réécriture en regardant la construction à partir d'une structure à nœuds vides d'une part, à nœuds explicites d'autre part.

b.2. cas des structures sémantiques à nœuds vides

Si la structure est à **nœuds vides**, c'est-à-dire chaque n_i est vide (pour i>1, n_i = ()), construire de gauche à droite la structure consiste à associer chaque v_{i-1} et f_i, i>1, en cherchant une règle de la grammaire telle que

$$A_i ::= [op_i, cs_{i1}, cs_{i2}]$$

avec $cat\_sém(v_{i-1})$ inclus dans cs_{i1}, et $cat\_sém(f_i)$ inclus dans cs_{i2} (voir la définition de l'inclusion donnée dans les parties 3.1.2 et 4.1.1).
Ainsi, on obtient l'expression sémantique v_i avec

- $trait\_sém(v_i) = [op_i, trait\_sém(v_{i-1}), trait\_sém(f_i)]$;
- et $cat\_sém(v_i) = A_i$.

Par exemple, pour *struct2* telle que

struct2 = ([cancer], lésion) () ([poumon], const_org)

on cherche une règle de la grammaire telle que $A ::= [op_i, cs1, cs2]$ avec *lésion* inclus dans *cs1*, et *const_org* inclus dans *cs2*, et répondant aux critères de choix exprimés ci-dessus : *LESION ::= [p_sur, LESION, LOC]*. Ainsi le résultat fourni pour *struct2* est *([p_sur, cancer, poumon], lésion)*.

Il se peut qu'il y ait ambiguïté entre différentes règles de la grammaire. Dans ce cas, notre premier critère de choix est le niveau des règles candidates : nous choisissons la règle de niveau le plus bas dans la grammaire. Il existe cependant quelques règles de même niveau et qui relient les mêmes catégories sémantiques, par exemple :

LESION ::= [EN_REL_TOPO_AVEC, LESION, LOC]
LESION ::= [p_sur, LESION, LOC]

Il existe dans ces cas un choix "par défaut" à faire parmi les règles candidates. Pour résoudre ces cas, nous avons dressé la liste des règles qui peuvent être concurrentes, et nous avons, pour chaque cas, choisi une règle parmi les différentes candidates.
Il se peut que cette construction ne soit pas possible, c'est-à-dire qu'il n'existe pas de règle satisfaisant les conditions décrites ci-dessus; il faut alors rendre une réponse négative, de manière à ce que le système de réécriture puisse être à nouveau activé sur une nouvelle proposition.

b.3. cas des structures sémantiques à nœuds explicites

Si la structure est à **nœuds explicites**, c'est-à-dire si tous les n_i sont non vides, la construction d'une expression sémantique d'une structure consiste à obtenir chaque v_i (i>1) par transformation de v_{i-1} à partir des expressions sémantiques n_i et f_i. Pour cela, il s'agit de montrer comment obtenir tout d'abord la catégorie sémantique, puis le trait sémantique de v_i à partir des informations contenues dans v_{i-1}, n_i, f_i (i>1).

- La catégorie sémantique de v_i est celle de n_i;

- Le trait sémantique de v_i est obtenu par instantiation des feuilles incomplètes du trait sémantique de n_i. En effet n_i a un trait sémantique complexe et incomplet, que nous voulons partiellement remplir en utilisant son contexte immédiat représenté par v_{i-1} et f_i : comme tout trait sémantique complexe et incomplet, certaines feuilles de la structure sont incomplètes, car instanciées par des catégories sémantiques. Le rôle de v_{i-1} et de f_i est de compléter certaines de ces feuilles de la façon suivante : la structure représentant le trait sémantique de n_i est parcourue selon un parcours préfixé, et les deux premières feuilles *F1* et *F2*, respectivement, de n_i vérifiant les propriétés :

cat_sém(v_{i-1}) inclus dans F1

cat_sém(f_i) inclus dans F2

sont remplacées par v_{i-1} et f_i respectivement.

Pour l'exemple *struct1 = ([cancer], lésion) ([dû_à, [p_sur, [a_pr_val, position, déviation], localisation], diagnostic], diagnostic) (poumon, const_org)*, on construit l'expression sémantique *([dû_à, [p_sur, [a_pr_val, position, déviation], uretère], cancer], diagnostic)*.

Si une telle construction s'avère impossible, il faut là également retourner une réponse négative et réactiver le système de réécriture.

5.5. le processus de coopération

Le processus de coopération de RIME a pour tâche essentielle de piloter les processus constructeurs que sont la morphologie, la syntaxe et la sémantique. Ce travail consiste essentiellement en l'établissement d'une communication entre ces trois processus constructeurs. Cette communication se traduit dans RIME par plusieurs niveaux de connaissances définis a priori dans le processus de coopération même :

- l'ordre partiel d'appel des processus constructeurs. En effet le processus de coopération doit connaître a priori par quel processus commencer la traduction, et comment ensuite enchaîner les processus de manière à mener à terme cette traduction ;
- le type des données à fournir à chacun des processus ;
- le type des résultats à récupérer pour chacun des processus ;
- la transformation des résultats de certains processus en données d'autres processus, de manière à permettre un enchaînement cohérent.

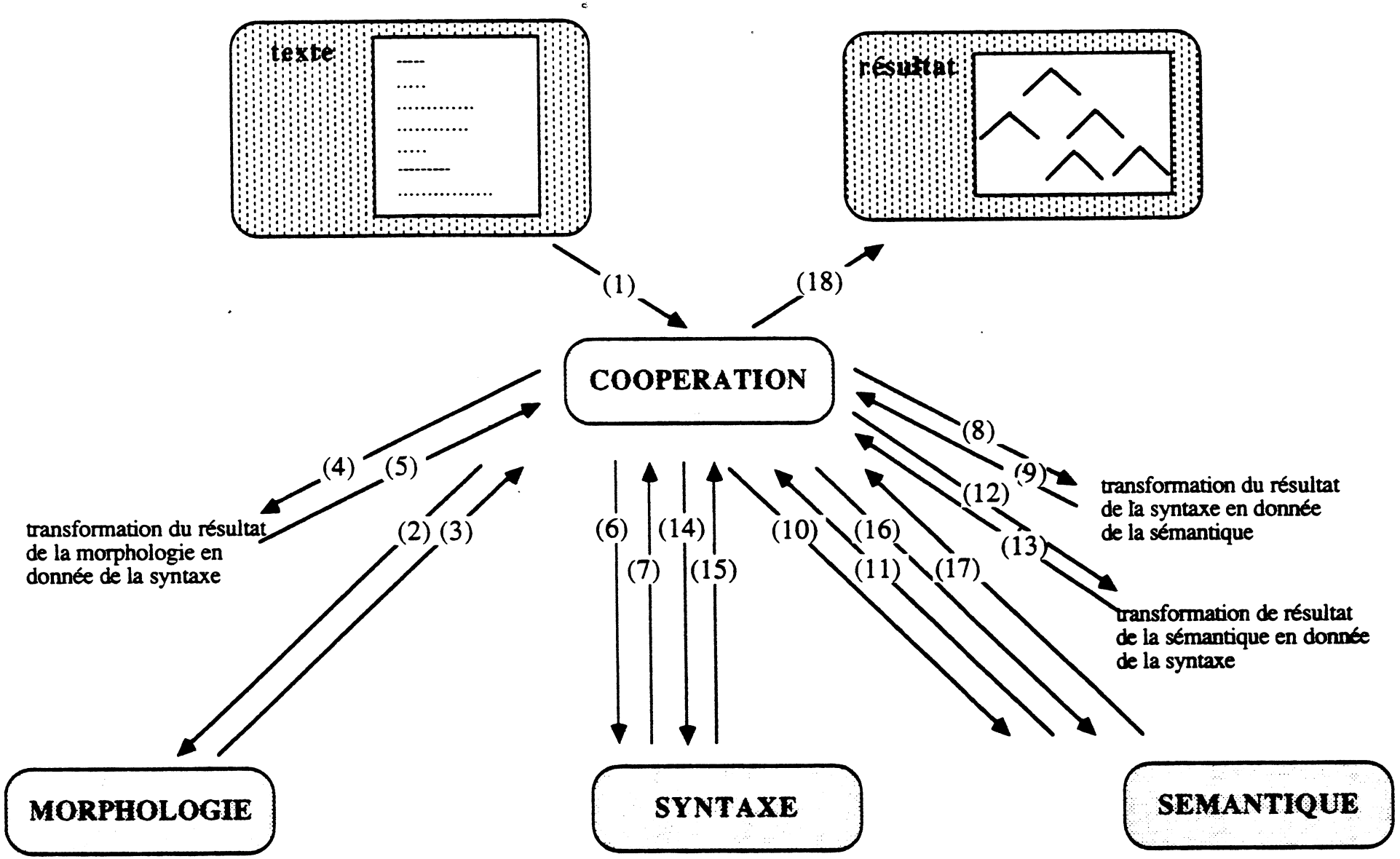
Ce sont ces quatre points que nous allons aborder pour présenter le processus de coopération de RIME.

5.5.1. les appels des processus constructeurs

L'ordre d'intervention des processus constructeurs de RIME est résumé par la figure 21.
Cette figure nous montre l'ordonnancement, par le processus de coopération, des trois processus constructeurs :

(1) un texte (un compte rendu médical) est donné en entrée du système ;
(2) le processus de coopération transmet ce texte à la morphologie, et la morphologie rend son résultat (3) ;
(4-5) le processus de coopération transforme le résultat de la morphologie en donnée acceptable pour la syntaxe ;
(6) le processus de coopération active la syntaxe avec cette donnée ;
(7) la syntaxe rend son résultat ;
(8-9) le processus de coopération transforme le résultat de la syntaxe en donnée acceptable pour la sémantique ;

figure 21 : l'ordonnancement des processus dans RIME



(10) le processus de coopération active la sémantique avec cette donnée ;

(11) la sémantique peut dans certains cas demander des validations à la syntaxe, via le processus de coopération, qui (12-13) transforme la requête du processus sémantique en donnée interprétable par la syntaxe ;

(14-15) la syntaxe interprète cette requête, et donne son résultat ;

(16) le processus de coopération transmet ce résultat à la sémantique ;

(17) après, éventuellement, plusieurs interrogations de la syntaxe, la sémantique donne son résultat final, que le processus de coopération considère comme le résultat final de l'indexation (18).

Nous rappelons ci-après

- les données de chacun des processus constructeurs (partie 5.5.2.) ;
- les résultats fournis par chacun d'eux (partie 5.5.3.) ;

Finalement, nous montrons dans la partie 5.5.4., comment le processus de coopération transforme les résultats des processus constructeurs. Par ailleurs, nous montrons dans le chapitre 6 des exemples complets de traduction de phrases.

5.5.2. les données des processus constructeurs

5.5.2.1. processus morphologique

L'entrée du processus morphologique est tout simplement constitué d'un texte correspondant au compte rendu médical à traiter :

entrée_morphologie ::= texte.

5.5.2.2. processus syntaxique

L'entrée du processus syntaxique est un ensemble de mots accompagnés de leurs attributs syntaxiques virtuels :

entrée_syntaxe::= [mot EXP_SYNT(mot)] $^{+}$;

5.5.2.3. processus sémantique

Rappelons que l'entrée du processus sémantique est composée d'arborescences à au plus quatre niveaux régies par la syntaxe suivante :

Niveau4_du_procsém ::=
[nom EXP_SEM_DED(nom)]+
/ [nom EXP_SEM_DED(nom)]+ signal_eff_comp
/ [nom EXP_SEM_DED(nom)]+ signal_ana_pro
/ [nom EXP_SEM_DED(nom)]+ signal_eff_adj_po
/ [nom EXP_SEM_DED(nom)]+ signal_ana_nom_rép ;
avec *EXP_SEM_DED(mot)* $\subset$ *EXP_SEM(mot) ;*

Niveau3_du_procsém ::=
Niveau4_du_procsém [Nœud3 Niveau4_du_procsém] ;*
Nœud3 ::= prépde EXP_SEM_DED(prépde) / prépde EXP_SEM_DED(prépde) signal_eff_coord ;

Niveau2_du_procsém ::=
Niveau3_du_procsém [Nœud2 Niveau3_du_procsém] ;*
Nœud2 ::= prépnonde EXP_SEM_DED(prépnonde)
/ prépnonde EXP_SEM_DED(prépnonde) signal_eff_coord ;

Niveau1_du_procsém ::=
Niveau2_du_procsém [Nœud1 Niveau2_du_procsém] ;*
Nœud1 ::= [verb EXP_SEM_DED(verb)]+
/ [verb EXP_SEM_DED(verb)]+ signal_eff_comp
/ [verb EXP_SEM_DED(verb)]+ signal_ana_pro ;

entrée_sémantique ::= Niveau1_du_procsém +.

5.5.3. les résultats des processus constructeurs

5.5.3.1. processus morphologique

La sortie du processus morphologique est une liste de mots accompagnés de leurs attributs virtuels :

sortie_morphologie ::= [mot EXP_SYN_SEM(mot)] $^{+}$;

5.5.3.2. processus syntaxique

Rappelons que la sortie du processus syntaxique est un ensemble d'arborescences à au plus quatre niveaux régies par la syntaxe suivante :

Niveau4_du_procsynt ::=
[nom EXP_SYNTF(nom)]$^{+}$
/ [nom EXP_SYNTF(nom)]$^{+}$*signal_eff_comp*
/ [nom EXP_SYNTF(nom)]$^{+}$ *signal_ana_pro*
/ [nom EXP_SYNTF(nom)]$^{+}$*signal_eff_adj_po ;*

Niveau3_du_procsynt ::=
Niveau4_du_procsynt [Nœud3 Niveau4_du_procsynt]*
Nœud3 ::= prépde EXP_SYNTF(prépde)
/ prépde EXP_SYNTF(prépde) signal_eff_coord

Niveau2_du_procsynt ::=
Niveau3_du_procsynt [Nœud2 Niveau3_du_procsynt]*
Nœud2 ::= prépnonde EXP_SYNTF(prépnonde)
/ prépnonde EXP_SYNTF(prépnonde) signal_eff_coord ;

Niveau1_du_procsynt ::=
Niveau2_du_procsynt [Nœud1 Niveau2_du_procsynt]* ;
Nœud1 ::= [verb EXP_SYNTF(verb)]$^{+}$
/ [verb EXP_SYNTF(verb)]$^{+}$ *signal_eff_comp*
/ [verb EXP_SYNTF(verb)]$^{+}$ *signal_ana_pro*
/ [verb EXP_SYNTF(verb)]$^{+}$ *signal_ana_nom_rép*

sortie_syntaxe ::= Niveau1_du_procsynt $^{+}$.

5.5.3.3. processus sémantique

Le résultat du processus sémantique est un ensemble d'arborescences vérifiant le modèle sémantique décrit au chapitre 3.

5.5.4. la transformation des résultats

5.5.4.1. la transformation morphologie-syntaxe

Le passage entre la morphologie et la syntaxe se fait par une simple projection de chaque *EXP_SYN_SEM(mot)* sur chaque *EXP_SYNT(mot)*, qui en extrait les informations syntaxiques.

5.5.4.2. la transformation syntaxe-sémantique

Le passage syntaxe-sémantique est obtenu par transformation de chaque
mot EXP_SYNTF(mot) où *EXP_SYNTF(mot)* $\subset$ *EXP_SYNT(mot)*
en
EXP_SEM_DED(mot) où *EXP_SEM_DED(mot)* $\subset$ *EXP_SEM(mot).*

Ceci se fait par interrogation du lexique de la façon suivante :
EXP_SEM_DED(mot) = {exps $\in$ *expressions_sémantiques /* $\exists$ *cats* $\in$ *EXP_SYNTF(mot) tel que (cats, exps)* $\in$ *EXP_SYN_SEM(m)}*

5.4.4.3. la transformation sémantique-syntaxe

Cette transformation consiste à modifier toute paire telle que

((mot1, EXP_SEM_DED(mot1)), (mot2, EXP_SEM_DED(mot2)))

en

((mot1, EXP_SYNTF(mot1)), (mot2, EXP_SYNTF(mot2))).

Ceci se fait également par consultation du lexique de la façon suivante :

EXP_SEM_DED(mot) = {exps ∈ expressions_sémantiques / ∃ cats ∈ EXP_SYNTF(mot) tel que (cats, exps) ∈ EXP_SYN_SEM(m)}

5.6. conclusion

Un des aspects de RIME, que nous voudrions résumer ici, est le paraphrasage, tel qu'il a été abordé et traité dans l'indexation. En effet, les documents traités peuvent tout aussi bien contenir la phrase

la tumeur dévie l'uretère

que la phrase

l'uretère est déviée par la tumeur ;

ou bien la phrase

inflammation des bronches

que

inflammation des bronches

sans que le contenu sémantique de ces documents soit différent.

Il est évident que nous ne pouvons pas, dès l'indexation, traiter tous les cas de paraphrasages. Il nous semble cependant utile de citer les paraphrasages que nous traitons ici, de manière que l'on sache parfaitement quels cas restent à traiter lors du processus d'interrogation.

Nous avons montré, au travers des chapitres 3 et 4, les phénomènes linguistiques que nous désirions traiter, et comment nous les traitions. Cette étude nous a donc amené à résoudre un certain nombre de paraphrasages, tels que ceux apparaissant au travers d'anaphores nominales ou pronominales, ou bien de divers types d'effacements. Il est clair que cette étude du paraphrasage n'est pas exhaustive, et que les études pour l'indexation de RIME doivent pousser plus loin cette analyse. Il semble en effet intéressant de pousser autant que possible le traitement du paraphrasage à l'indexation, de façon à simplifier le processus d'interrogation. Il est clair par ailleurs que le système de réécriture est un outil bien adapté à la résolution de ce problème.

Chapitre 6
Réalisations et expérimentations

Les réalisations pour l'application de RIME ont porté sur le développement des quatre processus principaux : la coopération, la morphologie, la syntaxe et la sémantique. Ces quatre processus doivent, à partir d'un compte rendu médical, générer sa traduction selon le modèle sémantique défini au chapitre 3.
La figure 22 montre ces quatre processus, et rappelle de manière très succinte le travail attendu pour chacun d'entre eux.

6.1. le prototype réalisé
6.1.1. schéma général

La figure 23 montre le prototype actuel, et nous revenons ci-après et plus en détails sur chacune des parties réalisées.
Ce prototype n'inclut pas encore toutes les possibilités théoriques de RIME. Par exemple, certains modules existent par ailleurs dans d'autres langages de programmation - Pascal et Lisp notamment - (ce qui est le cas entre autre du filtrage syntaxique), et pour des raisons de temps, nous ne les avons pas encore intégrés dans le prototype actuel.
Cependant ce prototype permet la traduction automatique d'un compte rendu médical : il convient suffisamment pour permettre la validation qualitative de

figure 22 : le schéma général du prototype de RIME

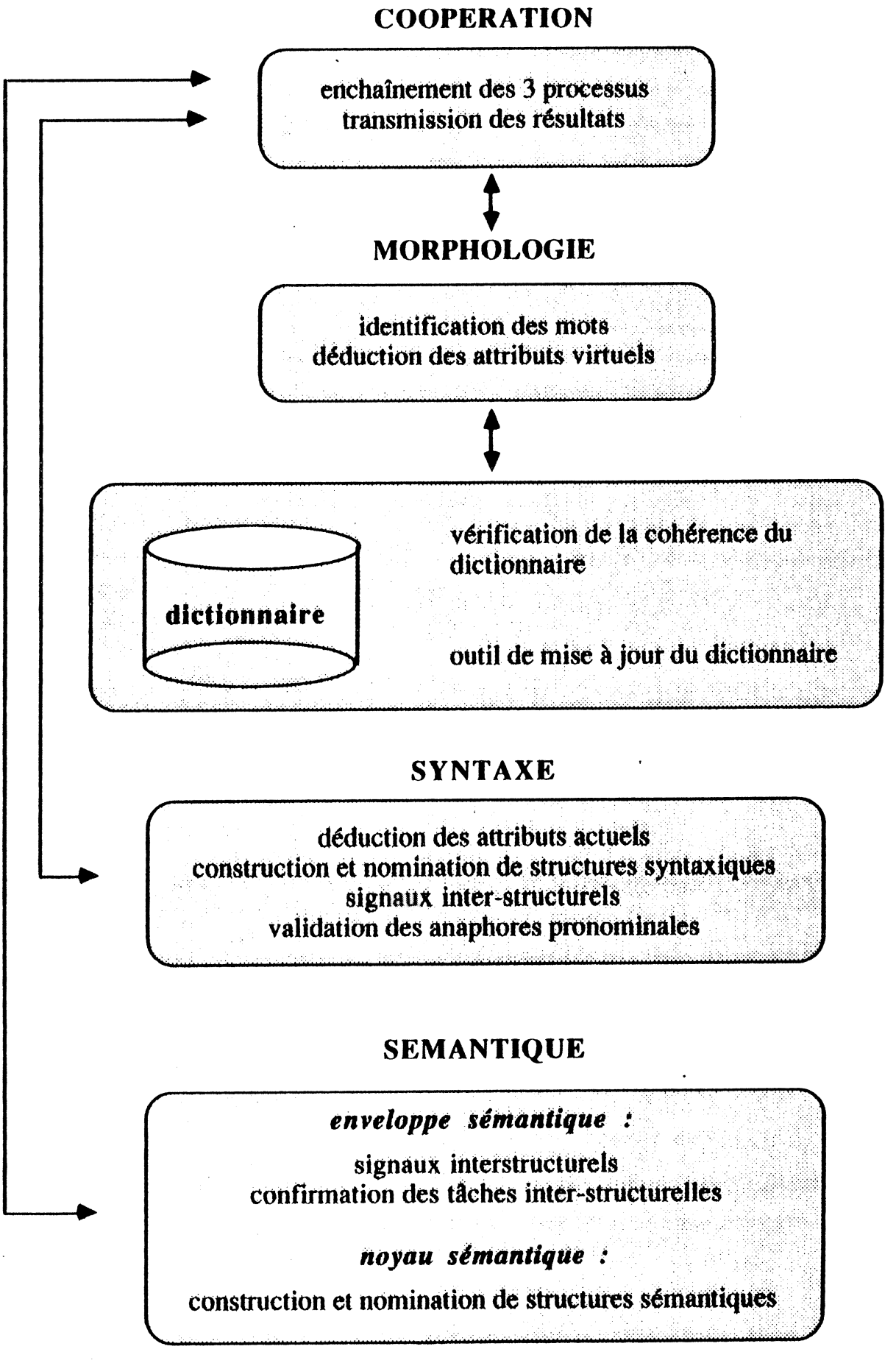
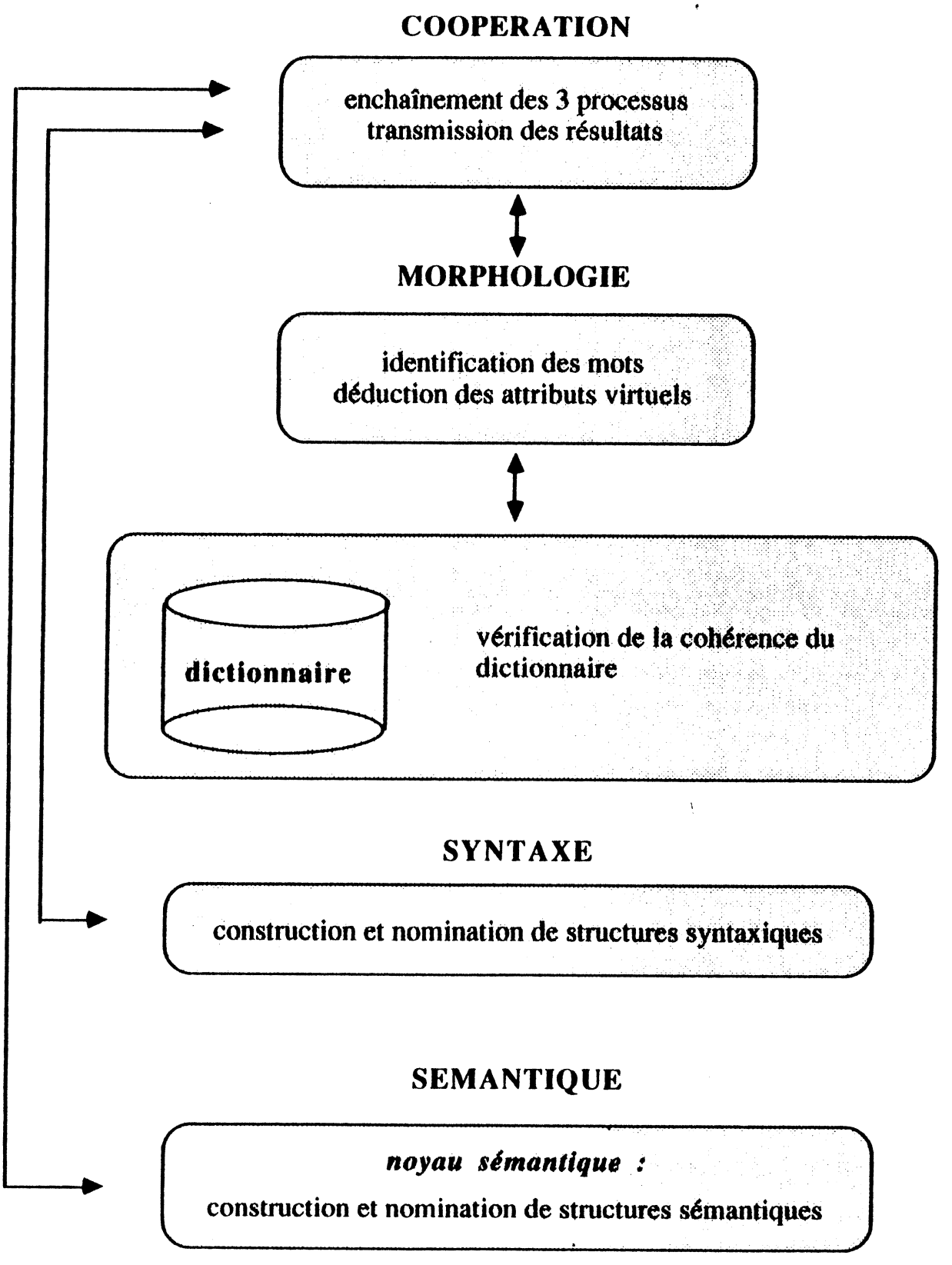


figure 23 : le schéma du prototype actuellement réalisé



la stratégie d'indexation proposée. Un prototype complet de RIME est prévu pour juillet 1989. Il a été développé en C.Prolog 1.5 (interpréteur), sur le Vax 785 du Laboratoire de Génie Informatique de l'IMAG, sous UNIX 4.3BSD. La version actuelle contient environ 3500 lignes de Prolog.

6.1.2. le lexique

Le lexique de RIME (module *dico*) a été construit manuellement dans un premier temps sur le vocabulaire d'une trentaine de comptes rendus. Par ailleurs, deux modules permettent d'une part l'interrogation du dictionnaire (le module *primit_dico*) et d'autre part la vérification de la cohérence du lexique (le module *verif_dico*).

6.1.2.1. le module *dico*

Dans sa version actuelle, le module *dico* contient chacun des mots des comptes rendus traités sous forme de clauses prolog :

dico(mot, (cat_gram, var_gram)), (trait_sém, cat_sém)).

où, en reprenant la modélisation du lexique du chapitre 4, (voir figure 8),

cat_gram ∈ *catégories_grammaticales,*

var_gram ∈ *valeurs_grammaticales,*

trait_sém ∈ *traits_sémantiques,*

et *cat_sém* ∈ *catégories_sémantiques.*

Par exemple, on a

dico(lobe, (subc, (masc,sing)), (lobe, <u>détail</u>)).

6.1.2.2. le module *primit_dico*

Ce module contient toutes les primitives d'accès au lexique, telles qu'elles sont décrites dans le chapitre 4 (cf figure 8) : *EXP_SYN_SEM(mot)* nous donne toutes les informations syntaxiques et sémantiques de *mot*, *CAT_GRAM(mot)* donne la ou les catégories grammaticales de *mot*.

Par exemple, *CAT_GRAM(lobe) = {subc}*.

6.1.2.3. le module *verif_dico*

Ce module effectue toutes les vérifications syntaxiques et sémantiques de cohérence du dictionnaire.

a. vérification des informations syntaxiques *cat_gram* et *var_gram*

Les informations syntaxiques de chaque mot sont considérées comme correctes si *cat_gram* et *var_gram* respectent les trois conditions suivantes :

cat_gram ∈ *catégories_grammaticales*,

var_gram ⊂ *valeurs_grammaticales*,

et *var_gram* ∈ *INST(VG_CAT(cat_gram))*.

Par exemple, si l'on reprend le mot *lobe*,

subc ∈ *catégories_grammaticales*,

(masc,sing) ⊂ *valeurs_grammaticales*,

VG_CAT(subc) = {*genre*, *nombre*},

INST(genre) = {*masc*, *fém*}, et *INST(nombre)* = {*sing*, *plu*}.

b. vérification des informations sémantiques *cat_sém* et *trait_sém*

Les informations sémantiques de chaque mot sont considérées comme correctes si *cat_sém* et *trait_sém* respectent les conditions suivantes :

cat_sém ∈ *catégories_sémantiques*,

si l'on reprend notre exemple, *détail* ∈ *catégories_sémantiques*.

trait_sém ∈ *traits_sémantiques*, ce qui nécessite un traitement plus complexe, car trois cas peuvent se produire :

1 - *trait_sém* = *mot*, ce qui est le cas pour le vocabulaire de base, comme dans notre exemple, où *trait_sém* = *lobe*.

2 - *trait_sém* = *mot'*, et *mot'* fait partie du vocabulaire de base, il faut donc vérifier également que *cat_sém* = *CAT_SEM(mot')*.

Par exemple, pour *pulmonaire*, on a
dico(pulmonaire, (adjq, (fem/masc, sing), (poumon, const_org)),
dico(poumon, (subc, (masc, sing), (poumon, const_org)),
avec la même catégorie sémantique pour les deux mots.

3 - *trait_sém* est complexe (complet ou incomplet), il faut donc vérifier que ce trait sémantique est compatible avec le modèle sémantique de RIME. Ceci se fait par appel du module sémantique *reduct* , que nous présentons dans la partie 6.1.5. Ce module vérifie *trait_sém*, et rend, en cas de succès, en résultat un couple *(trait_sém, cat_sém_bis)*. Il faut donc également vérifier que *cat_sém = cat_sém_bis*.
Par exemple, on a
dico(collapsus, (subc, (masc, sing), ([a_pr_val, volume, [a_pr_val, diminué, beaucoup]], signe))
L'appel de *reduct* avec *[a_pr_val, volume, [a_pr_val, diminué, beaucoup]]* réussit et donne en résultat *([a_pr_val, volume, [a_pr_val, diminué, beaucoup]], signe)*.
Finalement, on teste l'égalité entre *cat_sém_bis* (= *signe*) et *cat_sém* (= *signe*).

Par ailleurs, ce module permet la création automatique de clauses *ambiguité* qui contiennent la liste des mots dont le début est identique :
par exemple, on a *ambiguité(carcinome, [carcinome, embryonnaire])*.
Cet ensemble de clauses est consulté lors de la morphologie, de façon à découper correctement les phrases en mots (voir partie 6.1.3).

6.1.2.4. améliorations prévues

Nous n'avons pas pour le moment utilisé le dictionnaire de P.Palmer décrit dans [PALM81]. En effet, ce dictionnaire est d'une part écrit en Pascal, ce qui pose des problème de communications avec Prolog. D'autre part, il ne permet pas pour le moment d'inclure des informations sémantiques. Il nécessite donc des transformations avant de pouvoir être intégré dans RIME. Cependant ce

travail doit être rapidement fait de façon à bénéficier des avantages certains de ce dictionnaire : par exemple, nous devons actuellement saisir manuellement toutes les déclinaisons d'un mot, ce qui est une tâche fastidieuse que le dictionnaire de P.Palmer évite.

6.1.3. la morphologie, les modules *morphologie* et *parser*

La morphologie traite les textes qui lui sont donnés en entrée pour les découper en mots. Ce travail se fait par une double consultation :

- par consultation tout d'abord des clauses *ambiguité*, créées par le module *vérif_dico* (cf partie 6.1.2.3.). Cette consultation permet de découper correctement les textes en mots, sans problème d'identification des mots à partir des mots simples composant les phrases (cf partie 3.3.2.1.).

Par exemple, la phrase *opacité pulmonaire en projection du lobe pulmonaire supérieur droit avec signe de nécrose* est découpée en 11 mots :

'opacité',
pulmonaire,
'en projection du', par consultation de la clause *ambiguité (en, [en, projection, du]),*
lobe,
pulmonaire,
'supérieur',
droit,
avec,
signe,
de,
'nécrose'.

- par accès au lexique (modules *dico* et *primit_dico* décrits dans les parties 6.1.2.1. et 6.1.2.2.) pour chacun des mots extraits, de manière à extraire les informations sémantiques et syntaxiques de chacun. Ainsi pour l'exemple donné ci-dessus, on obtient :

[
['opacité',[subc,[fem,sing]],[[a_pr_val,'densité','augmenté'],signe]],
[pulmonaire,[adjq,[masc/fem,sing]],[[poumon],const_org]],
['en projection du',[prep,'[]'],[[en_projection_de, signe, loc], signe]],

[lobe,[subc,[masc,sing]],[[lobe],detail]],
[pulmonaire,[adjq,[masc/fem,sing]],[[poumon],const_org]],
['supérieur',[adjq,[masc,sing]],[['supérieur'],position]],
[droit,[adjq,[masc,sing]],[[droit],position]],
[avec,[prep,'[]'],[[avec],creux]],
[signe,[subc,[masc,sing]],[[signe],creux]],
[de,[prep,'[]'],[[de],creux]],
['nécrose',[subc,[fem,sing]],[['nécrose'],lesion]]
]

6.1.4. la syntaxe, les modules *syntaxe* et *grammaire*

Dans la version actuelle du prototype de RIME, nous n'avons programmé que des tâches intra-structurelles. Ainsi, au niveau syntaxique, nous disposons pour le moment de la construction et nomination de structures syntaxiques : syntagmes nominaux, syntagmes prépositionnels, syntagmes verbaux.
La grammaire que nous avons décrite dans la partie 5.3.2 est utilisée dans le module *grammaire*, et permet la génération de syntagmes intéressants pour notre traitement. La mise en œuvre de cette grammaire, à partir des textes fournis à la syntaxe est géré par le module *syntaxe*.
Ainsi en reprenant l'exemple donné précédemment, nous appelons la syntaxe avec :

[
['opacité',[subc,[fem,sing]]],
[pulmonaire,[adjq,[masc/fem,sing]]],
['en projection du',[prep,'[]']],
[lobe,[subc,[masc,sing]]],
[pulmonaire,[adjq,[masc/fem,sing]]],
['supérieur',[adjq,[masc,sing]]],
[droit,[adjq,[masc,sing]]],
[avec,[prep,'[]']],
[signe,[subc,[masc,sing]]],
[de,[prep,'[]']],
['nécrose',[subc,[fem,sing]]]
]

la syntaxe rend la résultat suivant :

[
[gn, ['opacité', [subc, [fem, sing]]], [pulmonaire, [adjq, [masc/fem, sing]]]],
[prep_non_de, ['en projection du', [prep,'[]']]],
[gn, [lobe, [subc, [masc, sing]]], [pulmonaire, [adjq, [masc/fem, sing]]],
['supérieur', [adjq, [masc, sing]]], [droit, [adjq, [masc, sing]]]],
[prep_non_de, [avec, [prep,'[]']]],
[gn, [signe, [subc, [masc, sing]]]],
[prep_de, [de, [prep,' []']]],
[gn, ['nécrose', [subc, [fem, sing]]]]
]

6.1.5. la sémantique

La version actuelle du prototype de RIME n'incluant pas de tâches inter-structurelles, le processus sémantique contient pour le moment :

- une partie seulement de l'enveloppe sémantique permettant l'appel du noyau sémantique (module *sémantique*) ;
- le noyau sémantique complet, tel qu'il a été décrit dans la partie 5.4.2 , c'est-à-dire deux parties : le système de réécriture et la résolution des tâches intra-structurelles (modules *gram* et *reduct*).

6.1.5.1. l'enveloppe sémantique, le module *sémantique*

Dans la version actuelle de RIME, l'enveloppe sémantique se réduit à l'appel du noyau sémantique, tel que nous l'avons décrit dans la partie 5.4.1.3. C'est-à-dire qu'à ce niveau, on analyse le résultat de la syntaxe, pour le transformer en structures sémantiques interprétables par le noyau sémantique.
Ainsi dans notre exemple, l'enveloppe sémantique reçoit le résultat de la syntaxe sous la forme suivante :

[
*[gn, ['opacité', [**a_pr_val**, 'densité',' augmenté'], signe]], [pulmonaire, [[poumon], const_org]]],*
*[prep_non_de, ['en projection du', [**en_projection_de**, signe, loc], signe]]],*
[gn, [lobe, [[lobe], detail]],[pulmonaire, [[poumon], const_org]], ['supérieur', ['supérieur'], position]], [droit, [[droit], position]]],

[prep_non_de,[avec,[[avec],creux]]],
[gn, [signe, [[signe], creux]]],
[prep_de, [de, [[de], creux]]],
[gn, ['nécrose', [['nécrose'], lesion]]]
].

L'enveloppe sémantique active le noyau sémantique en lui donnant tout d'abord les structures sémantiques correspondant aux syntagmes nominaux de plus d'un élément, c'est-à-dire pour notre exemple :

(1) *[[**a_pr_val**, densité, augmenté], signe]*
[]
[[poumon],const_org]

puis

(2) *[lobe, detail]*
[]
[[poumon], const_org]
[]
[[supérieur], position]
[]
[[droit], position]

Ensuite l'enveloppe sémantique active le noyau sémantique avec une structure sémantique comprenant une préposition de type *de* :

(3) *[[signe],creux]*
[[de],creux]
[['nécrose'],lesion]

et enfin avec :

(4) *résultat(1)*
*[[**en_projection_dè**, signe, loc], signe]*
résultat(2)
[[avec],creux]]
résultat(3)

dont le résultat donné par le noyau sémantique est la traduction du texte initial dans le modèle sémantique de RIME.

6.1.5.2. le noyau sémantique, les modules *gram* et *reduct*

Le noyau sémantique interprète une structure sémantique, telle qu'elle est définie dans la partie 5.4.2.1.a.
Le rôle du noyau sémantique consiste à tout d'abord générer une traduction sémantique de la structure qui lui est proposée. Ce travail se divise en deux parties très générales :

- un système de réécriture (module *gram*) capable de générer une traduction d'une structure proposée ;
- une partie permettant la résolution des tâches intra-structurelles (module *reduct*) en cas d'échec du système de réécriture.

Dans un premier temps, le noyau sémantique (via le module *reduct)* traite la structure en activant le système de réécriture (contenu dans le module *gram)* en lui fournissant tout d'abord la structure complète. En cas d'échec du système de réécriture, le module *reduct* essaie de résoudre le problème intra-structurel, de la façon que nous avons décrite dans la partie 5.4.2.1.b.
Le module *gram* (contenant le système de réécriture) ne contient actuellement que le modèle sémantique. Nous n'avons pas pour le moment inclus de programmes de vérification des propriétés de confluence et de terminaison (alogorithme de Church-Rosser) décrites dans la partie 5.4.2.2.a. Mais nous avons testé par ailleurs notre système avec le laboratoire de réécriture REVE, disponible sur le Vax du laboratoire. Ce laboratoire nous a permis de prouver que le système de réécriture que nous utilisons vérifie les propriétés de terminaison et de confluence recherchées [RAMO87].

Si l'on reprend l'exemple ci-dessus, le noyau sémantique est appelé tout d'abord avec :

(1) *[[**a_pr_val**, densité, augmenté], signe]*
[]
[[poumon], const_org]

qu'il traduit par

*[[**p_sur**,*
*[**a_pr_val**, densité, augmenté],*

poumon],
signe]

puis il est activé avec :

(2) *[lobe, detail]*
[]
[[poumon], const_org]
[]
[[supérieur], position]
[]
[[droit], position]

qu'il retourne avec

[[a_pr_val_loc,
[partie_de, lobe, poumon],
[et, supérieur, droit]],
loc]

ensuite avec

(3) *[[signe],creux]*
[[de],creux]
[['nécrose'],lesion]

qu'il retourne avec

[['nécrose', lesion]

Finalement il travaille sur la structure

(4) *[[p_sur, [a_pr_val, densité, augmenté], poumon], signe],*
[[en_projection_de, signe, loc], signe]
[[a_pr_val_loc, [partie_de, lobe, poumon], [et, supérieur, droit]], loc]
[[avec],creux]]
[['nécrose', lesion]

qu'il traduit par

[[permet_de_déduire,
['en_projection_de',
[p_sur,
[a_pr_val, densité, augmenté],
poumon],

[a_pr_val_loc,
[partie_de, lobe, poumon],
[et, supérieur, droit]],
'nécrose']
, cr]

6.1.6. la coopération

La coopération (module *cooperation*) permet l'enchaînement des trois processus constructeurs, ainsi que la transformation des résultats des processus. La coopération est écrite exactement comme elle a été décrite dans la partie 5.5.

6.1.7. conclusion

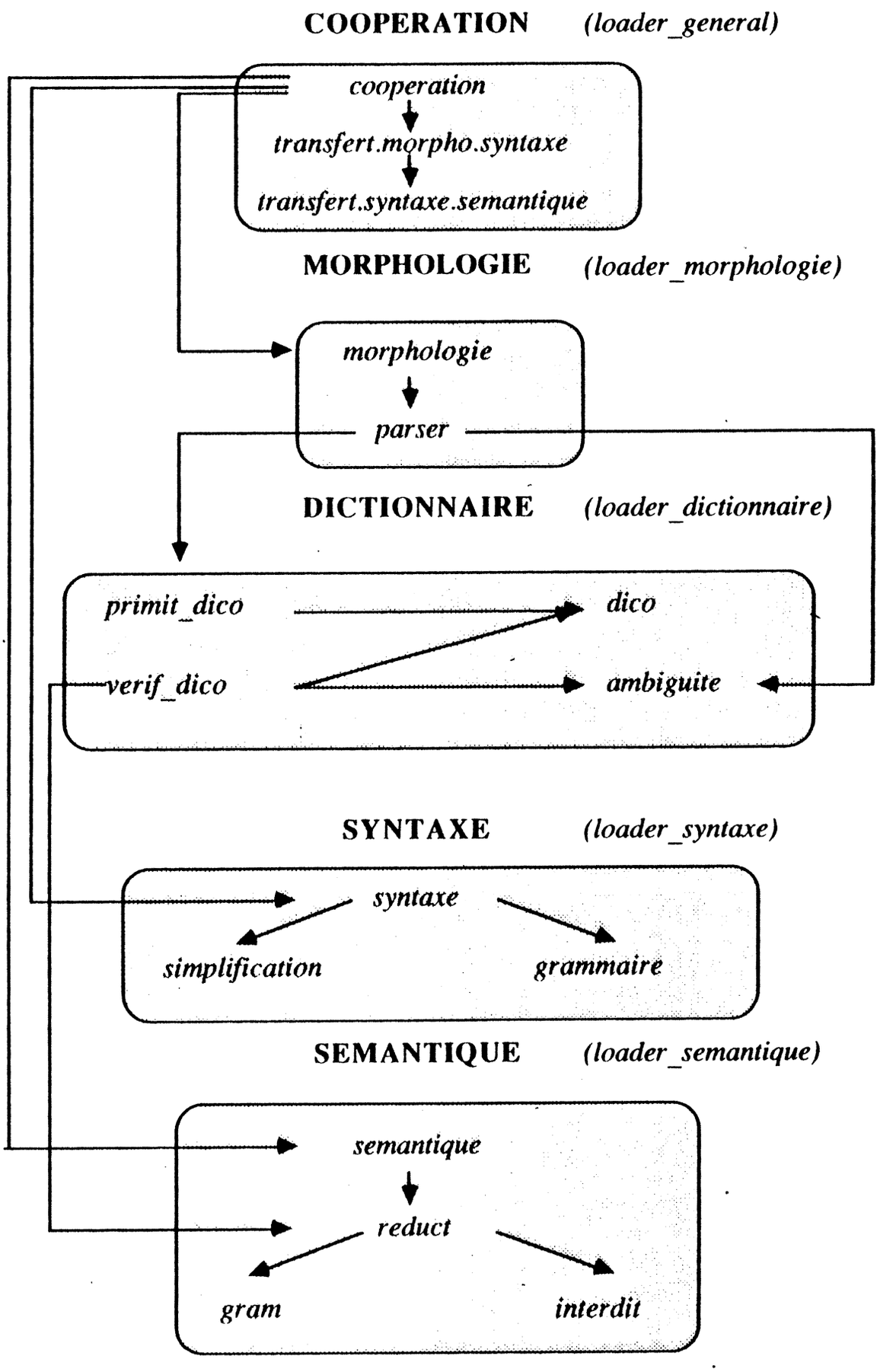
La figure 24 résume tous les modules que nous venons de présenter, et qui constituent le prototype actuel de RIME.

6.2. un exemple de session

Nous montrons dans cette partie une session d'indexation d'un compte rendu médical.

Script started on Thu Sep 8 10:05:12 1988
{1}pro
C-Prolog version 1.5
| ?- [loader_general].
/users/lgi/rechdoc/berrut/prototype/prolog/utilitaires reconsulted 5732 bytes 0.966667 sec.
/users/lgi/rechdoc/berrut/prototype/prolog/DICTIONNAIRE/dico reconsulted 96752 bytes 16.45 sec.
/users/lgi/rechdoc/berrut/prototype/prolog/DICTIONNAIRE/ambiguite reconsulted 7020 bytes 1.60001 sec.
/users/lgi/rechdoc/berrut/prototype/prolog/DICTIONNAIRE/primit_dico reconsulted 3892 bytes 0.816667 sec.

figure 24 : les différents modules de RIME



/users/lgi/rechdoc/berrut/prototype/prolog/DICTIONNAIRE/verif_dico
reconsulted 8936 bytes 1.58334 sec.
/users/lgi/rechdoc/berrut/prototype/prolog/SEMANTIQUE/gram reconsulted
6716 bytes 1.06667 sec.
/users/lgi/rechdoc/berrut/prototype/prolog/SEMANTIQUE/interdit
reconsulted 1060 bytes 0.200001 sec.
/users/lgi/rechdoc/berrut/prototype/prolog/SEMANTIQUE/reduct
reconsulted 1544 bytes 0.600012 sec.
/users/lgi/rechdoc/berrut/prototype/prolog/SEMANTIQUE/semantique
reconsulted 3440 bytes 0.816681 sec.
/users/lgi/rechdoc/berrut/prototype/prolog/MORPHOLOGIE/parser
reconsulted 2184 bytes 0.616679 sec.
/users/lgi/rechdoc/berrut/prototype/prolog/MORPHOLOGIE/morphologie
reconsulted 1080 bytes 0.350007 sec.
/users/lgi/rechdoc/berrut/prototype/prolog/SYNTAXE/grammaire
reconsulted 19876 bytes 4.13335 sec.
/users/lgi/rechdoc/berrut/prototype/prolog/SYNTAXE/syntaxe reconsulted
384 bytes 0.200004 sec.
/users/lgi/rechdoc/berrut/prototype/prolog/SYNTAXE/simplification
reconsulted 6036 bytes 1.35001 sec.
/users/lgi/rechdoc/berrut/prototype/prolog/COOPERATION/cooperation
reconsulted 3600 bytes 1.00003 sec.
/users/lgi/rechdoc/berrut/prototype/prolog/COOPERATION/transfert.morp
ho.syntaxe reconsulted 1092 bytes 0.266673 sec.
/users/lgi/rechdoc/berrut/prototype/prolog/COOPERATION/transfert.syntax
e.semantique reconsulted 1436 bytes 0.400022 sec.
loader_general consulted 170780 bytes 33.0833 sec.

yes
| ?- **cooperation.**

```
 ******************************
*                              *
 *  WELCOME TO RIME            *
*                              *
 * all rights reserved(@)      *
*                              *
 ******************************
```

(@) envoyer vos dons a l'association pour la restauration de la Chapelle de Beaumont, 01330 Villars-les-Dombes.

Donner le nom du compte rendu a traiter :
!!! terminer par un point suivi par un retour chariot !!! -
I: exemple.

Voila le compte rendu traite :

---

opacit{ pulmonaire en projection du lobe pulmonaire sup{rieur droit avec signe de n{crose.

Pour continuer, taper sur la touche retour chariot
I:

Donner le nom du fichier qui contiendra le resultat de la morphologie du fichier exemple :
!!! terminer par un point suivi par un retour chariot !!! -
I: resm.

Appel de la morphologie ...

Resultat de la morphologie :

[
['opacit{', [subc, [fem, sing]], [[a_pr_val, 'densit{', 'augment{'], signe]],
[pulmonaire, [adjq, [masc/fem, sing]], [poumon, const_org]],
['en projection du', [prep, '[]'], ['en projection de', creux]],
[lobe, [subc, [masc, sing]], [lobe, detail]],
[pulmonaire, [adjq, [masc/fem, sing]], [poumon, const_org]],
['sup{rieur', [adjq, [masc, sing]], ['sup{rieur', position]],
[droit, [adjq, [masc, sing]], [droit, position]],
[avec, [prep, '[]'], [avec, creux]],
[signe, [subc, [masc, sing]], [signe, creux]],
[de, [prep, '[]'], [de, creux]],

['n{crose', [subc, [fem, sing]], ['n{crose', lesion]]
].

Pour continuer, taper sur la touche retour chariot
|:

Donner le nom du fichier qui contiendra le resultat de la transformation morphologie syntaxe du fichier resm :
!!! terminer par un point suivi par un retour chariot !!! -
|: resms.

Appel de la transformation morphologie-syntaxe ...

Resultat de la transformation morphologie-syntaxe :

[
['opacit{', [subc, [fem, sing]]],
[pulmonaire, [adjq, [masc/fem, sing]]],
['en projection du', [prep, '[]']],
[lobe, [subc, [masc, sing]]],
[pulmonaire, [adjq, [masc/fem, sing]]],
['sup{rieur', [adjq, [masc, sing]]],
[droit, [adjq, [masc, sing]]],
[avec, [prep, '[]']],
[signe, [subc, [masc, sing]]],
[de, [prep, '[]']],
['n{crose', [subc, [fem, sing]]]
].

Pour continuer, taper sur la touche retour chariot
|:

Donner le nom du fichier qui contiendra le resultat de la syntaxe du fichier resms :

!!! terminer par un point suivi par un retour chariot !!! -
I: ressy.

Appel de la syntaxe ...

Resultat de la syntaxe :

[
[gn, ['opacit{', [subc, [fem, sing]]], [pulmonaire, [adjq, [masc/fem, sing]]]],
[prep_non_de, ['en projection du', [prep, '[]']]],
[gn, [lobe, [subc, [masc, sing]]], [pulmonaire, [adjq, [masc/fem, sing]]],
['sup{rieur', [adjq, [masc, sing]]], [droit, [adjq, [masc, sing]]]],
[prep_non_de, [avec, [prep, '[]']]],
[gn, [signe, [subc, [masc, sing]]]],
[prep_de, [de, [prep, '[]']]],
[gn, ['n{crose', [subc, [fem, sing]]]]
].

Pour continuer, taper sur la touche retour chariot
I:

Donner le nom du fichier qui contiendra le resultat de la transformation syntaxe semantique du fichier ressy :
!!! terminer par un point suivi par un retour chariot !!! -
I: resss.

Appel de la transformation syntaxe semantique ...

Resultat de la transformation syntaxe semantique :

[
[gn, ['opacit{', [[a_pr_val, 'densit{', 'augment{'], signe]], [pulmonaire, [poumon, const_org]]],
[prep_non_de, ['en projection du', ['en projection de', creux]]],

[gn, [lobe, [lobe, detail]], [pulmonaire, [poumon, const_org]], ['sup{rieur',
['sup{rieur', position]], [droit, [droit, position]]],
[prep_non_de, [avec, [avec, creux]]],
[gn, [signe, [signe, creux]]],
[prep_de, [de, [de, creux]]],
[gn, ['n{crose', ['n{crose', lesion]]],
].

Pour continuer, taper sur la touche retour chariot
|:

Donner le nom du fichier qui contiendra le resultat de la semantique du fichier resss :
!!! terminer par un point suivi par un retour chariot !!! -
|: resse.

Appel de la semantique ...

Resultat de la semantique :

--> simplifier : [[[a_pr_val, densit{, augment{], signe], [poumon, const_org]]
<-- reduit : [[p_sur, [a_pr_val, densit{, augment{], poumon], signe]
--> simplifier : [[lobe, detail], [poumon, const_org], [sup{rieur, position], [droit, position]]
<-- reduit : [[a_pr_val_loc, [partie_de, lobe, poumon], [et, sup{rieur, droit]], loc]
---> simplifier : [[signe, creux], [n{crose, lesion]]
<-- reduit : [n{crose, lesion]
---> simplifier : [[[p_sur, [a_pr_val, densit{, augment{], poumon], signe], [[en_projection_de, signe, loc], signe], [[a_pr_val_loc, [partie_de, lobe, poumon], [et, sup{rieur, droit]], loc]
, [avec, creux], [n{crose, lesion]]
<-- reduit : [[permet_de_deduire, [en_projection_de, [p_sur, [a_pr_val, densit{, augment{], poumon], [a_pr_val_loc, [partie_de, lobe, poumon], [et, sup{rieur, droit]], n{crose'], cr]

```
----------------------------
Pour continuer, taper sur la touche retour chariot
|:
----------------------------


Vive la Dombes libre !
yes
| ?-
[ Prolog execution halted ]
43.7u 6.0s 3:51 21% 95+499k 115+35io 4pf+0w
(2)exit
script done on Thu Sep 8 10:09:10 1988
```

Chapitre 7
Conclusion

L'indexation est l'opération clé dans un système de recherche d'informations : elle sert de pont entre le langage des documents et celui des questions. C'est sur sa qualité que repose l'efficacité d'un système.

L'information traitée étant généralement un contenu exprimé en langue naturelle, l'opération d'indexation hérite des complexités d'interprétation de la langue naturelle, et relève par conséquent de l'étude des liaisons syntaxe-sémantique.

D'un point de vue informatique, ces problèmes linguistiques ont suscité beaucoup d'intérêt et de travaux de recherche. Mais ils restent très ardus et très discutés entre d'une part les tenants des modèles linguistiques syntaxiques, et d'autre part les tenants des modèles sémantiques et pragmatiques. Les premiers prônent des outils généraux et simples, mais difficilement capables de réellement représenter l'information véhiculée dans les documents. Les seconds offrent des outils plus lourds, mais par contre capables de compréhension. Cependant, ils posent le problème classique de cohérence et de complétude par rapport au domaine effectivement couvert par le corpus traité.

Parmi les systèmes de recherche d'informations existants, certains exploitent à divers niveaux des éléments linguistiques purement syntaxiques. Ces systèmes permettent une première approche de la représentation du contenu des documents, sans pour autant aborder de façon très fine le problème de la compréhension du document. Actuellement certains travaux, que nous avons

cités dans le chapitre 2, portent sur les niveaux sémantique et pragmatique nécessaires dans un processus de compréhension d'un corpus.

Dans la méthode d'indexation proposée ici, nous avons tenté d'apporter notre contribution à la résolution de ce problème en nous intéressant tout d'abord à la définition d'un modèle sémantique permettant la représentation du contenu du corpus traité. Ce travail a été mené en collaboration avec les médecins du CHU de Grenoble, et a permis l'élaboration du modèle exposé au chapitre 3, et qui est basé sur les principes de la dépendance conceptuelle de Schank.

Nous avons ensuite recensé les phénomènes linguistiques (*les tâches*) qui apparaissent dans le corpus traité, et dont le traitement s'avère nécessaire pour permettre la traduction des textes en langue naturelle dans le modèle sémantique. Nous avons pour cela mis en évidence trois groupes de tâches linguistiques :

- les tâches infra-structurelles, qui considèrent les mots en tant qu'entités individuelles ;
- les tâches intra-structurelles, qui permettent le traitement syntaxique et sémantique des groupes de mots (syntagmes) ;
- les tâches inter-structurelles, qui permettent le traitement syntaxique et sémantique des dépendances (dues à des effacements, à des anaphores, ...) entre les groupes de mots traités au niveau intra-structurel.

Nous avons enfin mis en œuvre une **indexation automatique** permettant la génération du modèle sémantique, tout en traitant toutes les tâches linguistiques recensées. Cette indexation a été partitionnée en 4 processus :

- **trois processus linguistiques** (morphologie, syntaxe, sémantique) **indépendants**, et se répartissant les différentes tâches linguistiques recensées ;
- **un processus de coopération** garantissant **l'indépendance** et **l'enchaînement** des trois processus linguistiques, ainsi que la transmission de leurs résultats.

Il est clair qu'on ne pourra vraiment juger de l'efficacité du modèle sémantique qu'au travers de son utilisation à l'interrogation. Nous pourrons alors vérifier si le niveau de représentation des connaissances que nous avons adopté, ainsi que leur mode de représentation arborescent, s'avèrent efficaces. Ce processus est en cours de réalisation dans le groupe. Il est vraisemblable qu'à l'usage quelques défaut apparaîtront ; les fonctionnalités prévues dans notre prototype pour faciliter la modification du système seront alors très utiles.

Les logiciels présentés sont encore largement expérimentaux : notre premier souci a été d'évaluer tout d'abord les performances qualitatives de la méthode. Des progrès significatifs peuvent être réalisés dans l'immédiat notamment en permettant une compilation des programes PROLOG de RIME, et en utilisant une version du langage PROLOG permettant une gestion plus pratique des clauses (interface avec un SGBD par exemple).

Sur le plan qualitatif, il est par ailleurs clair que le prototype actuel doit être complété de manière à offrir toutes les fonctionnalités théoriques prévues. Nous espérons arriver à cet état de prototypage en juillet 1989.

Bibliographie

[ATP77]

Table ronde CNRS

Ambiguïtés de la langue écrite

1977.

[BERR85]

C. BERRUT

Résolution des ambiguïtés grammaticales. Une première approche dans le cadre d'un analyseur de surface de la langue naturelle

Rapport de DEA, Université Joseph Fourier, Grenoble, juillet 1985.

[BERR & CINQ88a]

C. BERRUT, P. CINQUIN

Natural language understanding of medical reports

IFIP-IMIA International Working Conference on Computerized natural language processing for knowledge representation, Genève, 1988.

[BERR & CINQ88b]

C. BERRUT, P. CINQUIN

Radiological images databases

Lecture notes in medical informatics, 1988.

[BERR & PALM86]

C. BERRUT, P. PALMER

Solving grammatical ambiguities within a surface syntactical parser for automatic indexing

ACM-SIGIR86, Pise, 1986.

[BOOK86]

A. BOOKSTEIN

Performance of self-taught documents

ACM-SIGIR, Pise, 1986.

[BORS, WHER & SCHE84]

F. BORST, E. WHERLI, J.R. SCHERRER

MEDIAL, a natural language processing system for medical records

Medical Informatics Europe84, Spriger Verlag ed. pp 128-133, 1984.

[BOSC, COUR & ROBI85]

P.BOSC, M.COURANT, S.ROBIN

Recherche d'informations basée sur la comparaison d'objets

RIAO85, pp 273-291, Grenoble, 1985.

[BRUA85]

M.F. BRUANDET

Modèle partiel de connaissances pour un système de recherche d'informations

RIAO, pp 101-114, Grenoble, 1985.

[BRUA89]

M.F. BRUANDET

Outline of a knowledge base model for an intelligent information retrieval system

Information Processing and Management, Vol 25(1), 1989.

[CATA82]

N. CATACH

Orthographe et lexicologie

Nathan U., 1982.

[CHIA & al86]

Y. CHIARAMELLA, B. DEFUDE, D. KERKOUBA, M.F. BRUANDET

IOTA : a prototype of an information retrieval system

AVM-SIGIR, Pise, 1986.

[CHIA, BERR & CINQ87]

Y. CHIARAMELLA, C. BERRUT, P. CINQUIN

A conceptual model for medical reports in a multimedia environment

Conférence AI and Neurosciences, Demongeot ed., Manchester University Press, février 1987.

[CROF & LEWIS87]

W.B. CROFT, D.D. LEWIS

An approach to natural language processing for document retrieval

ACM-SIGIR, New Orleans, 1987.

[CHOM65]

N. CHOMSKY

Aspects of the theory of syntax

the MIT Press, Cambridge, MA, 1965.

[CULL78]

R.E. CULLINGFORD

Script application : computer understanding of newspaper stories

Yale University, Computer Science Dept, Research report n°156, 1978.

[DEFU86]

B.DEFUDE

Etude et réalisation d'un système intelligent de recherche d'informations : le prototype IOTA

Thèse de l'INPG, Grenoble, 1986.

[DEJA & GARB86]

D. DEJACO, G. GARBOLINI

An information retrieval system based on artificial intelligence techniques

ACM-SIGIR, Pise, 1986.

[DEJO79]

G.F. DeJONG

Skimming stories in real time : an experiment in integrated understanding

Yale University, Computer Science Dept, research report n° 158, 1979.

[DILL & GRAY83]

M. DILLON, A.S. GRAY

FASIT : a fully automatic syntactically based indexing system

Journal of the ASIS, 34, 1983.

[DILL & MACD83]

M. DILLON, L.K. McDONALD

Fully automatic book indexing

Journal of documentation, pp 135-154, 1983.

[DOSZ86]

T. DOSZKOCS

IR, AI and UFOS : or IR-relevance, natural language problems, artful intelligence and user-friendly online systems

ACM-SIGIR, Pise, 1986.

[ERLI82]

R. DACHELET, B. NORMIER

Etude de la langue des comptes rendus d'hospitalisation en vue d'une analyse automatique

Société ERLI, Octobre 1982.

[FILL68]

C.J. FILLMORE

The Case for Case, Universals

Linguistic Theory, 1-88, E. BACH and R.T. HARMS, Molt Rinehart and Wiston, New York.

[FLUH77]

C. FLUHR

Algorithmes à apprentissage et traitement automatique des langues

Thèse d'Etat, Paris, 1977.

[FLUH82]

C. FLUHR

Intelligence artificielle et problèmes linguistiques dans les systèmes d'informatique juridique

Actes du sixième symposium sur l'informatique juridique en Europe, Thessaloniki, 1981.

[FODO74]

J. FODOR, T. BEVER, M. GARRETT

The psychology of language

Mc Graw-Hill Book Co., New York, 1974.

[FUCH83]

C. FUCHS

Une version transformationnelle de l'ellipse

L'ellipse grammaticale, Histoire, Epistomologie et Langage, Tome 5, fascicule 1, 1983.

[GREV80]

M. GREVISSE

Le bon usage

DUCULOT, Gembloux, 1980, 11ème édition.

[GOLL72]

GOLLEM

Manuel d'utilisation

SIEMENS, 1972.

[KERK84]

D. KERKOUBA

Une méthode d'indexation automatique des documents fondée sur l'exploitation de leurs propriétés structurelles. Application à un corpus technique.

Thèse de l'INPG, 1984.

[KNUT & BEND70]

D. KNUTH, P. BENDIX

Simple word problems in universal algebras

Computational problems in abstract algebras, Ed. Leech J., Pergamon Press, pp 263-297, 1970.

[HUE 81]

G.HUET

A complete proof of correctness of hte Knuth-Bendix completion algorithm

Journal Comp. Sys. Sc., vol 23, n°1, pp 11-21, Août 1981.

[JOUA & LESC86]

J.P. JOUANNAUD, P. LESCANNE

La réécriture

TSI juin 1986.

[MARC79]

M. MARCUS

Diagnosis as a notion of grammar

Proceedings on Theorical Issues in Natural Language Processing, Cambridge, MA, pp. 6-10, 1975.

[MARS78]

W. MARSLEN-WILSON, L. TYLER, M. SEIDENBERG

Sentence processing and the clause boundary

W. Levelt and G. Flores d'Arcais, eds., Studies in the Perception of language, J. Wiley and Sons, Chichester, U.K., pp. 723-728.

[MILN82]

J.C. MILNER

Ordre et raison de la langue

Editions du Seuil, Paris, 1982.

[MUNO87]

G. MUNOZ BACA

Stockage et exploitation de dossiers médicaux multimédia au moyen d'une base de données généralisée.

Thèse de l'Université J. Fourier, Grenoble, 1987.

[NIE88]

J. NIE

An outline of a general model for Information Retrieval Systems

ACM-SIGIR88, Grenoble, 1988.

[PAC & PRA69]

M. PACAK, A.W. PRATT

Identification and transformation of terminal morphemes in medical English

Methods of Information in Medicine, Vol. 8, n°2, pp. 84-90.

[PALM81]

P. PALMER

Etude de l'organisation d'un dictionnaire pour l'analyse du français

Rapport de DEA, Université Joseph Fourier, Grenoble, 1981.

[PALM & BERR85]

P. PALMER, C. BERRUT

Etude d'un analyseur de surface de la langue naturelle pour un système de recherche documentaire

Proceedings of the 13th annual CAIS Conference, Montreal, 1985.

[PALM88]

P. PALMER

Outils de traitement linguistique adaptés à l'indexation automatique de textes libres

thèse de l'Université Joseph Fourier, à paraître.

[RAMO87]

H. RAMOS

Une application de la réécriture à la compréhension de la langue naturelle : le traitement de comptes rendus médicaux

Rapport de DEA, Université Joseph Fourier, Grenoble, septembre 1987.

[RIES75]

C. RIESBECK

Conceptual analysis

R.Schank, ed., Conceptual Information Processing, North-Holland Publishing co., Amsterdam, pp. 83-156.

[RIJS79]

C.J. Van RIJSBERGEN

Information retrieval

Second edition, Buttersworth London, 1979.

[RODE85]

V. ROSENTHAL, M.DEFORMEL

Traitement automatique des anaphores : peut-on sortir d'un univers fermé?

TA transformations n°1, 1985.

[SAGE78]

N. SAGER

Natural language information formatting : the automatic conversion of texts to a structured data base

Advanced in computers, vol 17, 1978.

[SALT71]

G. SALTON

The SMART project

Prentice Hall, 1971.

[SALT80]

G. SALTON

Automatic information retrieval

Computer, vol 13, n°9, 1980.

[SALT, FOX & WU83]

G. SALTON, E.A. FOX, H. WU

Extended Boolean Information Retrieval

Communications of the ACM, 26:11, pp 1022-1036, 1983

[SALT & MCGI83]

G. SALTON, M.J. McGILL

Introduction to modern information retrieval

McGraw Hill Book company, New York, 1983.

[SALT85]

G. SALTON

A note on information retrieval models and theories

RIAO, Grenoble, 1985.

[SCHA72]

R. SCHANK

Conceptual dependency : a theory of natural language understanding

Cognitive Psychology, vol. 3, pp. 552-631.

[SCHA & ABEL77]

R. SCHANK, R.P. ABELSON

Scripts, Plans, Goals, and Understanding

Lawrence Erlbaum Press, Hillsdale, N.J., 1977

[SCHA80]

R. SCHANK

Language and memory

Cognitive Science, vol. 4, pp. 243-284.

[SCHA & BIRN80]

R. SCHANK, L. BIRNBAUM

Memory, meaning, and syntax

Research Report 189, Yale University, Novembre 1980.

[SCHA81]

R. SCHANK

Representing Meaning : An Artificial Intelligence Perspective

Cognitive Science Technical Report 1, Yale University, Avril 1981.

[SMEA87]

A. SMEATON

Using parsing of natural language as part of document retrieval

PhD Thesis, Glasgow, 1987.

[SMEA & RIJS88]

A. SMEATON, C.J. VAN RIJSBERGEN

Experiments on incorporating syntactic proccessing of user queries into a document retrieval strategy

ACM-SIGIR, Grenoble, 1988.

[SMIT80]

L.C. SMITH

Artificial intelligence applications in information systems

Ann. Rev. Inform. Sci. Technol., 15, pp 67-115, 1980.

[SPAR72]

K. SPARCK JONES

Progress in documentation, automatic indexing

Journal of Documentation, 12/1972.

[SPAR79]

K. SPARCK JONES

Problems in the representation of meaning in Information Retrieval

ASLIB Informatics Group & British Computer Society, Information Retrieval Group, 1979.

[THUR86]

G. THURMAIR

A common architecture for different text processing techniques in an information retrieval environment

ACM-SIGIR, Pise, 1986.

[WHER83]

E. WHERLI

Syntactic analysis in medical data processing

MED-INFO, IFIP-IMIA, North Holland, 1983

[WINO72]

T. WINOGRAD

Understanding Natural Language

Academic Press, New York, 1972.

[WINO77]

T. WINOGRAD

On some contested suppositions of generative linguistics about the scientific study of language

Cognition, vol. 5, pp. 151-179.

[WOOD70]

W. WOODS

Transition networks grammars for natural language analysis

Communications of the ACM, vol. 13, pp. 591-606.

[YAN86]

J. YAN

Interprétation sémantique de comptes rendus médicaux

Rapport de DEA, Université Joseph Fourier, Grenoble, septembre 1986.

[ZWEI & al87]

P. ZWEIGENBAUM, B. BACHIMOND, J. BOUAUD, M. CAVAZZA, L. DORE, J.F. BOISVIEUX, A. AURENGO

HELENE : compréhension automatique de comptes rendus médicaux

SITEF, Toulouse, 1987.

annexe 1 : un compte rendu médical

Monsieur Hyde
Le 12.09.85

TOMODENSITOMETRIE : (homme de 47 ans - Adressé pour bilan d'épanchement pleural droit avec température et signes de compression médiastinale - FIBROSCOPIE : compression extrinsèque sur la bronche lobaire moyenne mais aspect endoscopique normal - ETUDE CYTOLOGIQUE DU LIQUIDE PLEURAL pas de cellule maligne)
Les coupes ont été pratiquées depuis le sommet thoracique jusqu'à la région cœliaque.

Les constatations sont les suivantes:
- processus expansif et invasif de la loge médiastinale antérieure, depuis l'étage supra-aortique jusqu'à l'étage cardiaque. Plusieurs critères sémiologiques peuvent être précisés,
  - densité tissulaire et structure hétérogène faisant craindre la présence de larges zones de nécrose,
  - déformation du versant antérieur de la veine cave supérieure,
  - déplacement modéré mais indiscutable du médiastin à gauche et en arrière.

- important épanchement pleural de la grande cavité droite avec par place épaississement nodulaire du feuillet pleural faisant craindre une extension pleurale et/ou sous-pleurale de la tumeur médiastinale antérieure.
- sous l'épanchement, collapsus incomplet et incarcération du poumon droit (lobe moyen et lobe inférieur).
- pas d'anomalie décelable à l'étage sous-diaphragmatique.

EN CONCLUSION,
processus expansif et invasif du médiastin antérieur, de grandes dimensions, de densité tissulaire, associé à des images suggérant une greffe pleurale et/ou sous-pleurale à distance. Cet aspect TDM suggère volontiers le diagnostic de thymome lymphoépithélial invasif avec greffe pleurale.

Docteur Jekill.

annexe 2 : la grammaire du modèle sémantique

CR ::= [OP, CR, CR]
| CONSTAT
| DIAGNOSTIC
| [permet_de_déduire, CONSTAT, DIAGNOSTIC]

CONSTAT ::= [OP, CONSTAT, CONSTAT]
| [dû-à, CONSTAT, DIAGNOSTIC]
| [dû-à, CONSTAT, CONSTAT]
| [EN_REL_TOPO_AVEC, CONSTAT, CONSTAT]
| [montre_par, SIGNE, EXAMEN]
| SIGNE

DIAGNOSTIC ::= [OP, DIAGNOSTIC, DIAGNOSTIC]
| [dû-à, DIAGNOSTIC, DIAGNOSTIC]
| [EN_REL_TOPO_AVEC, DIAGNOSTIC,
DIAGNOSTIC]
| LESION

SIGNE ::= [OP, SIGNE, SIGNE]
| [a_pr_val, SIGNE, QUAL]
| [p_sur, SIGNE, LOC]
| [EN_REL_TOPO_AVEC, SIGNE, LOC]
| [p_sur, SIGNE, FCT]
| [a_pr_val, SIGNE, DEGRE]
| signe

LESION ::= [OP, LESION, LESION]
| [a_pr_val, LESION, QUAL]
| [p_sur, LESION, LOC]
| [p_sur, LESION, FCT]
| [EN_REL_TOPO_AVEC, LESION, LOC]
| [a_pr_val, LESION, DEGRE]
| lésion

EXAMEN ::= [OP, EXAMEN, EXAMEN]
| examen

LOC ::= [OP, LOC, LOC]
| [EN_REL_TOPO_AVEC, LOC, LOC]
| [a_pr_val_loc, LOC, POS]
| [a_pr_val, LOC, DEGRE]
| CONST_ORG

QUAL ::= [OP, QUAL, QUAL]
| [a_pr_val, QUAL, QUAL]
| [p_sur, CAR_PHY, LOC]
| [p_sur, CAR_PHY, FCT]
| [a_pr_val, CAR_PHY, VAL]
| VAL

VAL ::= [a_pr_val, VAL_QUAL, VAL_QUAN]
| [a_pr_val, VAL_QUAL, DEGRE]
| VAL_QUAN
| VAL_QUAL

VAL_QUAL ::= VAL_QUAL_ABS
| VAL_QUAL_REL

VAL_QUAN ::= [OP, VAL_QUAN, VAL_QUAN]
| val_quan

VAL_QUAL_REL ::= [par_rap, VAL_QUAL_ABS, CONST_ORG]
| [par_rap, VAL_QUAL_ABS, FCT]

CONST_ORG ::= [partie_de, DETAIL, CONST_ORG]
| mettre possibilité de nombres
| const_org

VAL_QUAL_ABS ::= val_qual_abs

POS ::= [OP, POS, POS]
| mettre possibilité de nombres
| position

FCT ::= [OP, FCT, FCT]
| mettre possibilité de nombres
| fonction

CAR_PHY ::= car_phy

DETAIL ::= détail

DEGRE ::= degré

OP ::= et | ou

EN_REL_TOPO_AVEC ::=
gauche | droit | haut | bas | avant | arrière
| intra | extra
| contact | près | loin | cerné

| fistule | séquestre | envahissement
| vers | à_partir_de | jusqu'à
| (groupe ???)
| contraste_entre | contraste_avec | en_projection_de |
plan_de_clivage

annexe 3 : liste des catégories grammaticales et de leurs variables grammaticales

catégorie grammaticale			variables grammaticales			
ABRV	ABRéViation	ouverte				
ADJC	ADJectif Cardinal	fermée	GNR	NBR		
ADJD	ADJectif Démonstratif	fermée	GNR	NBR		
ADJE	ADJectif interrogatif ou Exclamatif	fermée	GNR	NBR		
ADJI	ADJectif Indéfini	fermée	GNR	NBR		
ADJO	ADJectif Ordinal	fermée	GNR	NBR		
ADJP	ADJectif Possessif	fermée	GNR	NBR		
ADJQ	ADJectif Qualificatif	ouverte	GNR	NBR		
ADJR	ADJectif Relatif	fermée	GNR	NBR		
ADVB	ADVerBe	ouverte				
ADVN	ADVerbe de négation Ne	fermée				
ADVP	ADVerbe de négation Pas	fermée				
ARTC	ARTicle Contracté	fermée	GNR	NBR		
ARTD	ARTicle Défini	fermée	GNR	NBR		
ARTI	ARTicle Indéfini	fermée	GNR	NBR		
CONJ	CONJonction	fermée				
DPNT	Deux PoiNTs	fermée				
GUIL	GUILlemets	fermée				
INTJ	INTerJection	fermée				
LOCC	LOCution Conjonctive	fermée				
LOCP	LOCution Prépositionnelle	fermée				
NOMB	NOMBre	fermée				
PADV	Pronom ADVerbial	fermée				
PARE	PAREnthèse	fermée				
PNTF	PoNcTuation Forte	fermée				
PRDM	PRonom DéMonstratif	fermée	GNR	NBR		
PREP	PREPosition	fermée				
PRIN	PRonom INdéfini	fermée	GNR	NBR		
PRPC	PRonom Personnel Complément	fermée	GNR	NBR		
PRPO	PRonom POssessif	fermée	GNR	NBR		
PRPS	PRonom Personnel Sujet	fermée	GNR	NBR		
PRPV	PRonom Personnel préVerbal	fermée	GNR	NBR		
PRRL	PRonom ReLatif	fermée	GNR	NBR		
SUBC	SUBstantif Commun	ouverte	GNR	NBR		
SUBP	SUBstantif Propre	ouverte	GNR	NBR		
VBCJ	VerBe ConJugué	ouverte	MOD	TMP	PRS	NBR
VBIF	VerBe InFinitif	ouverte				
VBPA	VerBe participe PAssé	ouverte	GNR	NBR		
VBPR	VerBe participe PRésent	ouverte				
VIRG	VIRGule	fermée				
XACJ	auXiliaire Avoir ConJugué	fermée	MOD	TMP	PRS	NBR
XAIF	auXiliaire Avoir InFinitif	fermée				
XAPA	auXiliaire Avoir participe PAssé	fermée	GNR	NBR		
XAPR	auXiliaire Avoir participe PRésent	fermée				
XECJ	auXiliaire Etre ConJugué	fermée	MOD	TMP	PRS	NBR
XEIF	auXiliaire Etre InFinitf	fermée				
XEPA	auXiliaire Etre participe PAssé	fermée	GNR	NBR		
XEPR	auXiliaire Etre participe PRésent	fermée				

annexe 4 : liste des préfixes admis dans la composition par particule

Particules latines:

ab (s)
a (c, d, f, etc)
ambi
archi
ba, be (s)
bi (s)
circ (on, um)
cis
co (l, m, n, r)
contr (a, o)
dé (s)
déci
di
di (s)
duo
e (x)
extra *hors de*
extra *extrémement*
in (l, m, r) *dans*
in (l, m, r) *privatif*
infra
inter
intr (a, o)
juxta
mes, mis
multi
ob (c, f, p)
oct (a, o)
omni
pén (e)
per
post
pré
pro
quadr (i, a, u)
quinqu (a)
quint
rétr (o)
semi
septe (m, n)
sub
super
supra
sus
trans (tré)
tri
ultra
uni

Particules grecques:

a (n) *privatif*
amphi
an (a)
ante
anti
apo
arch (é, i)
cat (a, e)
déca
di
di (a)
dys
é (c)
ect (o)
en
end (o)
épi
eu
exo
hém (i)
hyper
hypo
méta
par (a)
péri
sy (m, n)

Particules françaises:

à (à, au)
après
arrière
avant
ca
chez
contr (e)
demi
dessous
dessus
en
entr (e)
ex *qui a cessé d'être*
for (fau, four)
hors
mi (demi, semi)
moins
non
outre
par
plus
pour
pui (s)
quasi
r (e, é)
sans
sous
sur
trop
vi (ce)

AUTORISATION DE SOUTENANCE

DOCTORAT 3ème CYCLE, DOCTORAT-INGENIEUR, DOCTORAT USTMG

Vu les dispositions de l'Arrêté du 16 avril 1974,

Vu les dispositions de l'Arrêté du 5 juillet 1984,

Vu les rapports de M. Patrick BOSC

M. Patrice POGNAN

M.lle Catherine BERRUT est autorisée

à présenter une thèse en vue de l'obtention du titre de docteur de l'Université Joseph FOURIER - Grenoble I -

Grenoble, le 2 DEC. 1988

Le Président de l'Université Scientifique Technologique et Médicale

UNIVERSITE JOSEPH FOURIER SCOLARITE

J.J PAYAN