Traitement Automatique des Langues Naturelles : Evolution et Perspectives

Seyed Mohammed MAHMOUDI Pierre DUPONT,
Respectivement : Membre du Laboratoire d'Information Cognitive (LJC) Université Lumière Lyon
Responsable du LIC, Professeur à l'Université Lumière Lyon 2,
5, *Avenu Pierre Mendés-France -69500 BRCM Tel : 78.77.23.28*

Fm- a **conception** consiste à définir les objets et les règles qui constitueront le système de traitement logique de fonctionnement. Dans le cas de traitement automatique des langues naturelles, cette phase correspond à ce que l'on appelle traditionnellement l'analyse morpho-syntaxique.

Le processus de conception peut-être présenté et discuté comme une activité d'exploration et de définition des problèmes, activité que mènent, au moyen de conversation et transaction, plusieurs acteurs en interaction dans des situations caractérisées par l'ambiguïté, le conflit et l'incertitude stratégique. Dans les situations complexes caractérisées par la présence de plusieurs acteurs en interaction coopérative et/ou conflictuelle, une part importante des activités et stratégies des concepteurs semble être engagée dans une démarche exploratoire qui tend, en premier lieu, non à résoudre, mais à définir le problème [DEMAILLYet LE MOIGNE, 86].

La conception est donc un processus heuristique visant à recueillir des informations sur la structure du problème et pas seulement une technique formelle apte à sélectionner les moyens optimaux pour réaliser des fins préétablies.

Un concepteur engagé dans un effort de conception ne peut toujours (sinon, presque jamais) être considéré comme un joueur solitaire ou comme un mathématicien. Dans la plupart des situations concrètes de la vie réelle, la conception est un processus d'exploration et de recherche collective qui se déroule au moyen de conversations et

* Cet article est la deuxième partie de l'article portant le mêmetitre et publié dans RIST vol. 7 n°2, I 997

transactions entre les acteurs de l'intervention. C'est là ce qui se produit notamment dans cet espace décisionnel-type que constituent les organisations. Les problèmes-clefs dans un processus de conception sont souvent la coordination et la résolution collectives des problèmes [SCHELLING, 80].

Le rôle du concepteur est donc la concertation et l'organisation des structures complexes et ses implications pour l'organisation des processus de leur conception |SIMON, 911. La conception est en t'ait une tâche très complexe, il ne faut pas confier cette tâche à des seuls informaticiens, son programme nécessite aussi l'intervention de nombreux domaines et acteurs spécialisés qui varieront en fonction des objectifs et des applications envisagées.

Dans un processus de conception plusieurs étapes sont nécessaires pour que les programmes de traitement se réalisent convenablement : Il faudra en fait définir les objectifs et les moyens nécessaires, décrire et analyser le système, exposer les problèmes et les difficultés et construire, enfin, les règles de fonctionnement qui débouchent sur des solutions logico-techniques des problèmes.

Ainsi, lorsque la dernière phase de la conception est achevée, la réalisation de techniques de fonctionnement deviendra possible. **La réalisation** correspond à la mise en oeuvre de-là solution logique : c'est une phase d'implémentation ,technique traditionnellement appelé analyse organique et programmation.

Au cours de la phase de réalisation, un certain nombre de modifications y pourrait survenir, si le système rencontre des nouveaux besoins. C'est alors **la phase de l'évolution** qui s'impose. Cette phase se traduit nécessairement par une opération de maintenance du système implémenté.

Conception Réalisation Evolution

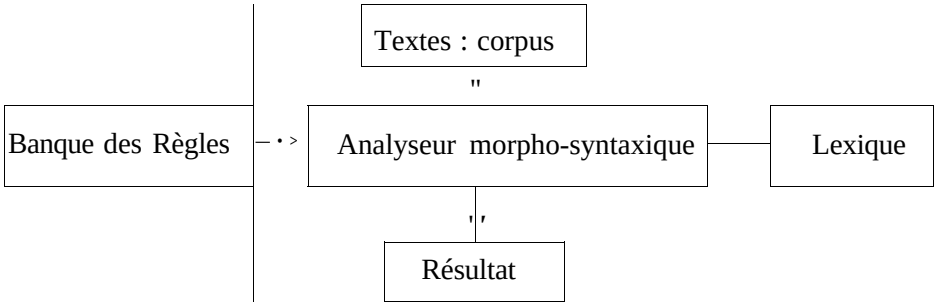
4.1.- LUS ETAPES DE LA CONCEPTION DE L'ANALYSEUR MORPHO-SYNTAXIQUE

La tâche essentielle du concepteur en TALN consiste à mettre en oeuvre une conception cohérente, homogène et complète, qui tout en actualisant le temps et l'espace, envisagerait la construction de modèle symbolique à l'aide duquel on inférera en suite le réel. La conception est donc la clé principale de toute application automatique en 1A et

surtout en traitement des langues naturelles; sans une conception préalable et approfondie on ne pourra pas mettre en oeuvre une réalisation correcte correspondant aux objectifs déterminés. Une bonne conception est une condition nécessaire de bonne réalisation et de bonne évolution.

Face à cette réflexion, nous ne prétendons pas pouvoir épuiser le sujet, la mise en oeuvre d'un processus de conception dans le contexte du TALN est une tâche très complexe qui nécessite l'intervention de plusieurs domaines théoriques et technqiues et dépendra surtout des différents facteurs et contraintes pratiques. Par conséquent, toute possiblité d'automatisation d'un système de représentation de l'information dépend non seulement de l'objectif visé mais aussi de l'automatisabilité des procédures d'analyse [METZGER, 88]. IL ne faut pas donc tenter de donner les solutions à n'importe quel prix, "il n'y a pas de solutions" comme le dit Montsegur. "Il n'y a que des problèmes, et la sagesse ne consiste pas à les résoudre, mais à les poser comme il faut".

L'architecture globale du système



4.1.1.- Définir les objectifs du traitement

Dans le processus de conception, la première chose à faire, c'est évidemment de définir les objets et les besoins. Parler de conception sans objectifs peut apparaître comme une contradiction dans les termes. Il semble "évident" que le vrai concept de rationalité implique des objectifs vers lesquels la pensée et l'action sont dirigées" [SIMON, 91].

4.1.2.- Choix du corpus

Le traitement automatique de la langue naturelle nécessite avant tout la définition d'un univers de travail, c'est-à-dire le choix d'un corpus particulier qui correspond aux applications envisagées.

La détermination d'un corpus adapté pour la reconnaissance automatique des SN en français, est une tâche très difficile.

Le corpus doit réunir un ensemble de textes variés mais homogènes, du point de vue de la forme et du style (littéraire, familière, etc.). "A ce titre, chaque construction est intéressante et des constructions rares peuvent être d'une importance capitale pour l'élaboration d'une théorie" [ROUAULT, 87].

4.1.3.- Définir les méthodes d'analyse et de conception

L'informatisation de tout système en 1A impose au préalable la nécessité d'une réflexion théorique et "méthodologique" sur l'ensemble des opérations qui en constituent les composantes" |LE GUERN, 91]. Plus la complexité de l'automatisation croît, plus il devient nécessaire d'utiliser une méthode d'analyse et de conception.

Le recours à une approche structurée, incluant les notions de "général vers particulier" et un modèle conceptuel de définition et de manipulation est une nécessité primordiale pour toute application automatique en langues naturelles. La modélisation dans la phase de conception est la conjonction du modèle et du raisonnement sur le modèle; ce raisonnement est une simulation du modèle, tout en respectant les contraintes algorithmiques dues à l'utilisation automatique ultérieure.

Faire de la conception c'est avant tout faire de l'analyse. L'analyse d'un système consiste à rassembler et à interpréter des faits, à diagnostiquer des problèmes et à utiliser les éléments recueillis pour améliorer le système. Alors que l'analyse spécifie ce que le système doit accomplir, la conception indique comment atteindre l'objectif, tout en minimisant les ambiguïtés. La conception consiste aussi à décrire les données à entrer, à traiter ou à stocker, chaque donnée et chaque calcul est analysé et détaillé. L'analyse en fait "propose", le concepteur "dispose".

Dans le cadre du TALN, l'analyse correspond à la "délinéarisation" de l'entrée linguistique, "c'est-à-dire, l'utilisation de syntaxe et d'autre sources de connaissance pour déterminer les fonctions des mots dans les phrases entrées pour créer une structure de données, comme un *arbre de dérivation,* qui peut être utilisée pour atteindre la "signification" de la phase. Un analyseur peut être vu comme un applicateur d'image syntaxiques significatives. L'ensemble des images syntaxiques utilisées est déterminé par la grammaire du langage d'entrée. En théorie, en appliquant une grammaire compréhensive un analyseur peut décider ce qui est et ce qui n'est pas une phrase grammaticale et peut construire une structure de données correspondant à la structure syntactique de toutes les phrases grammaticales qu'il Irouve. Tous les systèmes informatiques de traitement du langage naturel contiennent un composant d'analyse de quelque sorte, mais l'application pratique des grammaires au traitement du langage naturel s'est révélée difficile" [FEIGENBAUM, 86].

Le choix d'une méthode d'analyse et de conception est une partie intégrante du processus de la conception. La question du choix de la représentation dans le contexte du traitement et la compréhension de la langue naturelle est un domaine extrêmement complexe et crucial. 11 semble nécessaire de comprendre plusieurs sortes de méthodes et de représentation pour comparer et choisir en t'ait la meilleure solution.

Définir une méthodologie d'analyse et de conception pour élaborer un analyseur morpho-syntaxique, c'est, du même coup, choisir une *grammaire de référence,* c'est-à-dire un système générateur de règles remplissant une tâche centrale à la théorie linguistique. La grammaire de référence est en fait un schéma conceptuel formel pour la spécification de phrases autorisées dans le langage indiquant les règles pour la combinaison de mots dans des phrases et des clauses. Les grammaires de références sont très variées et différentes. De ce fait, lorsqu'on veut choisir une grammaire pour une application spécifique, "il convient de savoir précisément quel en est le but" [BERRENDONNER, 83] et les applications visées; il conviendra aussi de :

— définir le champs d'analyse (l'analyse de texte écrits ou de textes oraux),
— déterminer la nature de l'information traitée, c'est-à-dire définir une typologie de corpus (scientifique, littéraire, etc.),
— choisir une stratégie d'analyse, c'est-à-dire choisir une analyse morphologique ou syntaxique, logico-sémantique ou pragmatique, une analyse descendante ou ascendante, profonde ou superficielle, etc....

Parmi les différentes méthodes et grammaires de références qui ont été utilisées dans des programmes de TALN, nous allons essayer de présenter brièvement les plus importantes. Chacune de ces grammaires s'applique à une partie des problèmes, l'ensemble constitue une base de référence pour la constitution de notre modèle linguistique. Aucune de ces méthodes ne représente une solution universelle.

4.1,3.1.- Grammaire formelle

L'une des contributions les plus importantes à l'étude du langage a été la théorie des langages formels introduite par Noam Chomsky dans les années cinquante. La théorie s'est développée comme une étude mathématique, et non pas linguistique, et a influencé fortement la science informatique dans la réalisation de langage de programmation informatique (langage artificiel). Néanmoins, elle est utile en liaison avec les systèmes de compréhension du langage naturel, à la fois comme un outil théorique et pratique [FEIGENBAUM, 86].

Pour présenter le langage formel, défini comme un ensemble de chaînes de longueurs finies formées à partir d'un vocabulaire fini de symboles, et la grammaire du langage formel qui est spécifiée en terme de concepts, tels que, les catégories syntactiques, les

symboles terminaux du langage, les règles de réécriture ou production et le symbole de départ (axiome), Chomsky a délimité quatre types de grammaires et les a numérotés de 0 à 3. Voici un extrait des grammaires formelles tiré pour l'essentiel du document [FEIGENBAUM, 86].

Type 0. La grammaire de type 0, n'a aucune restriction sur la forme que les règles de réécriture peuvent prendre, elle est en fait la règle la plus générale des grammaires. Il a été montré qu'un langage peut être généré par un type de grammaire 0 si et seulement si il peut être reconnu par une machine de Turing, c'est-à-dire, si nous pouvons construire une machine de Turing qui s'arrêtera dans un état ACCEPTE pour exactement les phrases d'entrées qui peuvent être générées par le langage.

Type 1. Les grammaires de type 1, sont aussi appelées des grammaires contextuelles, car, elles sont sensibles au contexte. Une grammaire de type 0 est aussi de 1 si la forme des règles de réécriture est aussi restreinte de telle façon, que pour chaque production Xè Y de la grammaire, le côté droit Y, contienne au moins autant de symboles que le côté X voici un exemple de cette grammaire avec un symbole de départ S et les terminaux a, b, et c :

S———*- uSBC
S———• cxBC
CB———* BC
oB———• u.|3
^B———* PP
PC———*- Pc
cC———*- ce

Type 2. Les grammaires hors contexte ou libre de contexte sont des grammaires clans lesquelles chaque réécriture doit avoir seulement un symbole unique non terminal à son côté gauche. Pour les détails voir plus loin.

Type 3. Dans les grammaires de type 3, les règles de la grammaire sont de la forme suivante : X—• aY ou X—•a, où X et Y sont des variables uniques et a est un symbole terminal unique, la grammaire est une grammaire régulière de type 3. Une grammaire régulière peut générer l'ensemble de chaines d'un ou plus d'un a suivi par un ou plus de b (sans garantie d'un nombre égal de a et b) :

S———•>- aS
S———»• uT
T———^ 0
T———• pT

Dans les grammaires formelles, de type 0, 1, 2 et 3 à cause des formes de plus en plus restreintes des productions, chaque type est un sous-ensemble convenable du type supérieur à lui dans la hiérarchie. Une hiérarchie correspondante existe dans les langages formels. On peut montrer qu'il y a des langages qui sont libres de contexte (type 2) mais non réguliers (type 3), sensibles au contexte (type 1) mais pas libres de contexte, et de type 0 mais non pas sensibles au contexte.

Pour des grammaires régulières et libres de contexte, il y a des algorithmes pratiques d'analyse pour déterminer si, oui ou non, une chaine donnée est un élément d'un langage et si on peut lui attribuer une structure syntaxique sous la forme d'un-arbre de dérivation. Les grammaires libres de contexte (hors contexte) ont une application considérable dans les langages de programmation. Les langages naturels, ne sont pas généralement libres de contextes [CHOMSKY, 63], et ils contiennent aussi des caractères qui peuvent être traités plus convenablement, sinon en totalité, par une grammaire plus puissante.

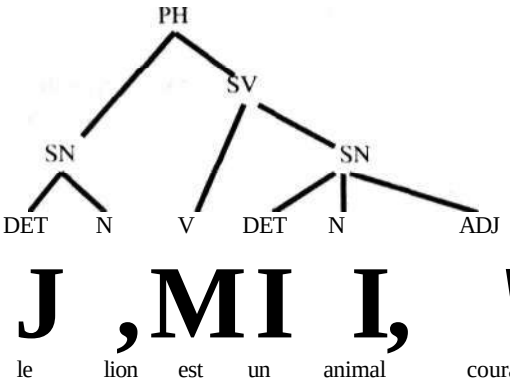
Grammaire hors contexte

Parmi les grammaires présentées dans les paragraphes ci-dessous, il faut donner une place privilégiée aux grammaires hors contexte qui sont bien maîtrisées par les linguistes et les informaticiens. Dans le cadre de notre traitement nous sommes fort bien intéressés par l'utilisation de cette grammaire pour la reconnaissance des Syntagmes Nominaux. Cette grammaire qui est de type 2, constitue un outil efficace dans une optique d'analyse syntaxique du textes écrits en langue naturelle. Cette grammaire est une description mathématique de la syntaxe, qui permet de traiter le langage sans s'intéresser au contexte qu'il évoque, d'où son nom.

A l'aide d'une grammaire hors contexte qui fournit les règles de composition, on peut analyser une phrase afin de déterminer les différents constituants syntaxiques. Cette grammaire peut surtout être construite pour la reconnaissance des Syntagmes Nominaux. En utilisant la grammaire hors contexte, chaque dérivation peut être représentée comme un arbre. Dans ce contexte les règles hors contexte peut avoir l'aspect suivant :

Ph———• SN + SV
SN———»• DET + NOM
SN———• DET + ADJ + NOM
SN———*- DET + NOM + ADJ
SV———•Verbe + SN
DET———*- un|le|la|ce|cet

Par exemple la phase "le lion est un animal courageux" pourra être analysée par telles règles pour produire un arbre de dérivation :

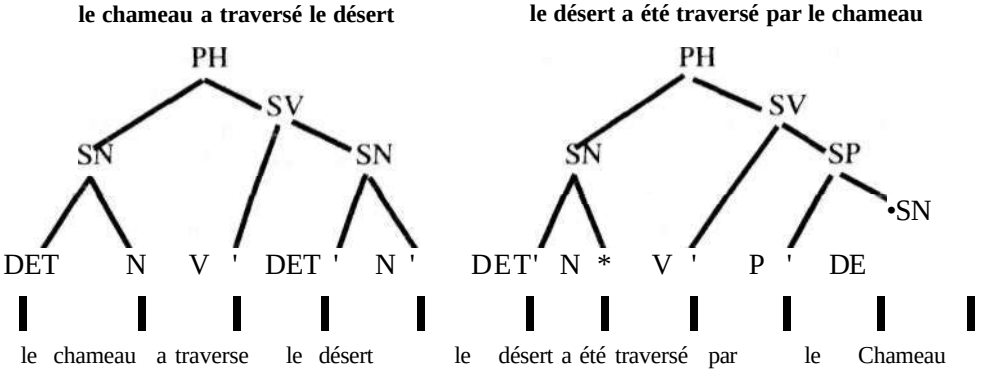


L'orsqu'on analyse une phrase, on peut se demander dès le départ, s'il faut analyser mot après mot les relations entre les mots et construire progressivement la structure (un arbre de dérivation par exemple) représentant la phrase, "ou bien est-il préférable de faire une hypothèse sur la structure et chercher à la vérifier en regardant comment la phrase cadre à l'intérieur?" [ANTOMARCHI et Castiel, 86]. Dans le premier cas, la procédure est simple: parcourir la chaine courante de gauche à droite jusqu'à ce que l'on rencontre un symbole non terminal, remplacer ce non terminal en utilisant une règle, répéter l'opération jusqu'à épuisement de tous les non terminaux.
Dans le deuxième cas, lorsque l'analyse est guidée par les buts (chaînage arrière), le système fait l'hypothèse qu'il a une phrase syntaxiquement correcte et sélectionne les règles engendreant de telles structures (SN,SV——*- P). Puis il cherche les règles lui donnant les constituants de la phrase, et ainsi de suite.

Une grammaire est dite "hors contexte" parce que le membre gauche de chaque règle est composé uniquement du symbole à remplacer : rien d'autre ne peut influencer le remplacement. Toutes les chaines ne contenant que des terminaux et construites à l'aide de la procédure associée à une grammaire particulière sont des phrases bien formées dans cette grammaire. L'ensemble de toutes les phrases bien formées possibles constitue le langage spécifié par la grammaire [WINSTON, 88].

4.1.3.2.- Grammaires transformationnelles

Vers 1957, Chomsky a proposé l'idée de grammaire transformationnclle. Cette grammaire met en jeu une variété d'opérations formelles de "transformation" et contribue pour la construction d'une représentation syntaxique profonde de chaque phrase, directement rattachée à sa signification, mais restant aussi indépendante que possible des différentes formes superficielles (active, passive) qu'elle peut adopter. Dans cette méthode, une phrase pourra s'analyser en termes d'autres phrases. Par exemple, un passif sera analysé à partir d'un actif, ce qui n'est pas prévu dans les méthodes distributionnelles. Nous pouvons ainsi écrire :



Alors qu'avec l'analyse distributionnelle, une phrase ne peut s'analyser que par décomposition en éléments de substitutions, une transformation est (ou induit) une *relation* entre deux phrases ou deux structures de phrases. Les transformations mettent en jeu les opérations formelles de permutation, d'insertion et d'effacement [CARRE, 91].

Dans certaines langues comme l'anglais dont la structure syntaxique n'est ni régulière ni libre de contexte, la nécessité d'une analyse à haut niveau, que fournit la grammaire transformationnelle est très fructueuse. Les langages transformationnels, qui ne sont pas d'un type formel contraint laissent en fait toutes possibilités descriptives aux théories linguistiques.

4.1.3.3.- Les réseaux de transition récursifs

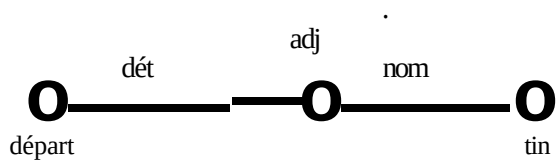
Ce paragraphe utilise pour l'essentiel les documents : [BONNET et HATON, 84] et [WINSTON, 88].

Les réseaux de transition récursifs (RTN, pour "Recursive Transition Network") sont un moyen de spécifier une grammaire. Ils sont issus des automates d'état fini avec quelques adjonctions indispensables pouvant prendre en compte l'aspect récursif de certaines définitions. Ces réseaux sont équivalents (en puissance d'expression) aux grammaires à contexte libre utilisées en informatique classique.

Dans un analyseur classique construit autour des règles de grammaires hors contexte, une procédure simple spécifie comment commencer et arrêter les noeuds, les règles de grammaire spécifient où rattacher chaque noeud.

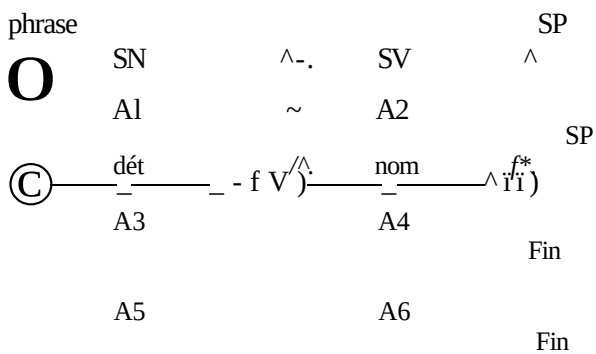
Mais les règles hors contexte ne sont que des instruments pour exprimer comment les membres de phrase se relient les uns aux autres et aux mots. Moyennant des automates d'état fini, les réseaux de transition récursifs constituent un autre mécanisme, mathématiquement isomorphe.

Un automate d'état fini consiste en un ensemble de noeuds représentant des états et d'arcs qui relient ces noeuds; le rôle des arcs est d'indiquer comment on peut passer d'un état à un autre. Il y a un état Départ et un on plusieurs états Fin. Les arcs sont étiquetés des mots ou catégories de mots du langage, spécifiant que l'on peut emprunter l'arc si l'on rencontre le mot en question ou s'il satisfait la catégorie spécifiée. On dit que l'automate accepte la séquence de mots à analyser si en partant de Départ avec le début de la phrase, on peut atteindre l'un des états Fin à la fin de la phrase. L'automate de la figure ci-dessous accepte la phrase "le joli petit cheval" ou simplement "le cheval" mais pas "petit cheval".



Un automte d'état fini pour un groupe nominal très simple

Lorsque les phrases contiennent un nombre arbitraire de groupes prépositionnels, on ne peut pas représenter récursivement les phrases avec l'automate ci-dessous présenté. Il faudra donc pouvoir spécifier sur les arcs, non seulement des symboles terminaux, mais aussi des symboles non terminaux tels que les SN ou les SP (Syntagme Prépositionnel) définis eux-mêmes par un autre automate pouvant faire référence à l'automate dont il fait partie, et on arrive ainsi à la notion de RTN. Voici un exemple :

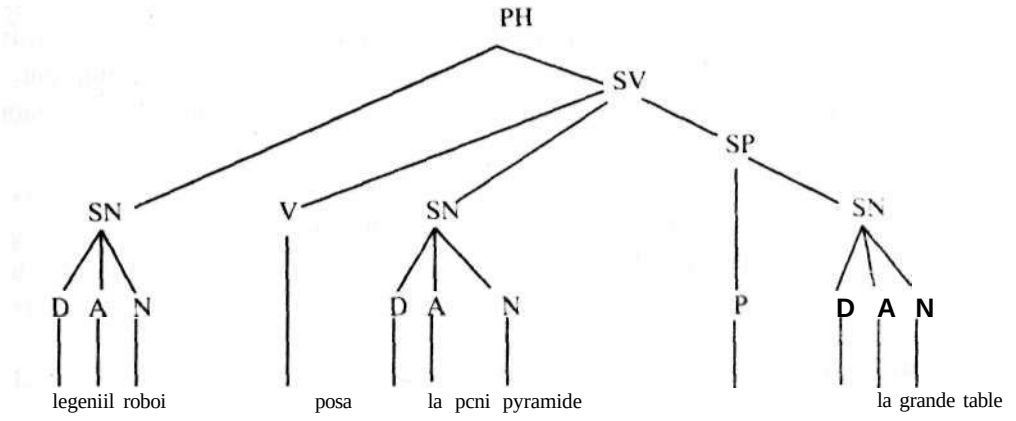


Grammaire à réseaux de transition récursif. Chaque nœud de l'arbre syntaxique est appelé P et chaque flèche correspond à un arc (A).

Comme le schéma l'illustre, pour traverser le réseau phrase, moyennant l'interpréteur du réseau de transition, par exemple, nous devons d'abord traverser le réseau syntagme nominal. Pour traverser le réseau syntagme nominal, le premier mot doit être un déterminant. Ainsi nous pouvons entrer dans la boucle de syntagme prépositionnel (SP) et entrer dans le réseau de transition syntagme prépositionnel. Quelque soit le chemin parcouru, le mot suivant doit être une préposition. Lorsque la phrase ne contient plus d'autres mots, ayant déterminé un syntagme verbal (SV), nous retournons au réseau phrase où nous nous trouvons aussi dans un état de réussite.

Cette procédure constitue une forme d'analyse descendante. Cette dénomination résulte du fait que l'analyse part de la création du noeud phrase situé en haut de l'arbre syntaxique et évolue vers le bas selon l'examen circonstanciel des mots de la phrase.

Lorsque la phrase toute entière est analysée avec succès, selon l'exemple suivant, on peut aboutir à l'arbre syntaxique présenté ci-dessus:



Maigre ses nombreux avantages, la melhode d'analyse des reseaux de transition recursifs ne permet pas cependant de prendre tacilemeni en compte tous les phénomènes Je la langue naturelle et en particulier certains aspects contextuels. De ce fait, les réseaux de transition augmentés (Al'Ns) développé par Williams Woods (1970) peuvent offrir un formalisme plus approprié pour gérer les aspects contextuels de la langue naturelle. Voici une brève description des ATNs :

4.1.3.4.- Les **réseaux de transition augmentés**

Les réseaux de transition .augmentés (ou ATN pour Augincntcd Transition Network) ont été développés comme une représentation modifiable de grammaires pour les langues naturelles qui permettent de spécifier une grammaire.

Etant donné que les grammaires hors contexte et à réseaux de transition récursils peuvent taire apparaitre toutes les contraintes syntaxiques, il faudra donc compléter ces grammaires à l'aide de transformations ou autre formalisme, classiquement choisi en intelligence artificielle, comme des grammaires à réseaux de transition augmentés qui utilisent la recherche en profondeur.

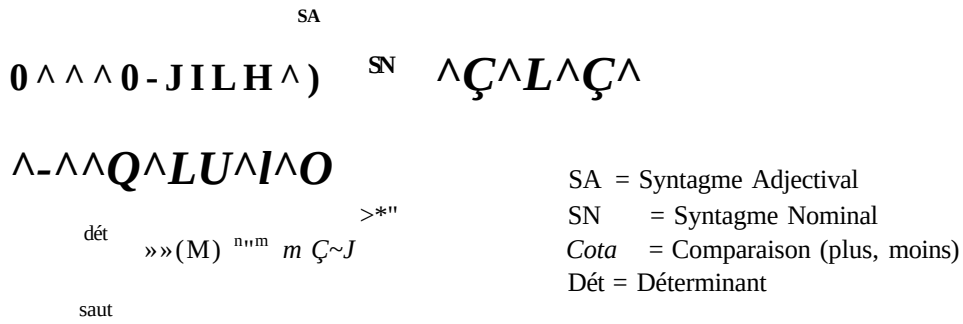
Les ATN sont représentés habituellement par un graphe : ils ont la capacité de prendre des notes en cours d'opération et de se référer à ces notes pour prendre êtes décisions ultérieures qui comportent des noeuds décrivant un état et des arcs permettant de passer

d'un noeud à l'autre. Sont en particulier distingués un état initial et un ou plusieurs états finaux.. Chaque arc correspond à un type de mot ou groupe de mots donné : article, adjectif, nom, groupe nominal... et peut être emprunté si l'on rencontre un mot de cette nature. L'ATN accepte une séquence de mots si, entre le début et la fin d'une phrase, il passe de l'état"départ" à l'un des états "fin". Les ATN ont des propriétés récursives : un arc peut spécifier un groupe de mots (groupe nominal...) qui sera alors défini par un autre ATN, pouvant faire référence au premier. Divers tests et opérations peuvent être rattachés aux arcs : les premiers préciseront des conditions à vérifier pour pouvoir emprunter l'arc et les seconds des actions à entreprendre lorsque l'arc a été emprunté. Certains arcs particuliers ("saut") font directement passer d'un état à un autre sans mot intermédiaire" |ANTOMARCHI et Castiel, 86].

L'ATN ci-dessous permet de comprendre les questions suivantes :

- Quelle est la ville la plus proche ?
- Quelle est la plus proche ville ?

Ces questions seront aussi traitées sous leur forme plurielle.



4.1.3.5.- Les analyseurs déterministes (extrait de [WINSTON, 88]).

LLS analyseurs déterministes "en américain WASP pour Wait And See Parser) qui sont plas complexes que les autres analyseurs, offrent une alternative prometteuse aux analyseurs hors contexte et aux analyseurs ATN. Ils rassemblent la connaissance syntaxique et remplacent le retour arrière par une prospection. La structure des analyseurs déterministes nous donne une indication sur la nature du langage.

Dans la structure d'un analyseur déterministe, les phrases sont tout d'abord traitées par un pré-analyseur qui met les syntagmes nominaux dans des noeuds des syntagmes nominaux. Le travail de ce dernier consiste à identifier les SN et à remplacer les terme du SN par un noeud SN dans l'arbre syntaxique. Les phrases pré-traitées traversent alors un tampon qui constitue la base de données utilisée par les règles condition-action associées

au nœud courant de l'arbre syntaxique, noeud situé au sommet de la pile des noeuds. L'arbre syntaxique est construit en fonction des actions précisées dans les règles condition-action. La partie condition de chaque règle est habilitée à étudier le contenu du tampon et le noeud courant. Les mots et les syntagmes nominaux circulent automatiquement dans le tampon chaque fois qu'une partie condition d'une règle d'analyse est vide. La partie action de chaque règle, quant à elle, spécifie une action parmi les suivantes :

•Créer un nouveau noeud et le mettre sur la pile.
•Terminer le noeud au sommet de la pile et le déposer dans la première case du tampon, en décalant vers la droite les éléments déjà présents.
•Rattacher le premier élément du tampon au noeud du sommet de pile et décaler vers la gauche les autres éléments du tampon pour combler la place laissée vacante par le départ du premier élément.
•Echanger les deux premiers éléments du tampon.

La procédure d'analyse est donc très simple dans cette méthode. Contrairement à l'analyseur ATN, toute la connaissance concernant le départ, l'arrêt et l'attachement est contenue dans les règles.

En faisant une comparaison entre un analyseur déterministe et un analyseur ATN, qui s'effectue au niveau de la compétence et non de la performance, on peut tirer les conclusions suivantes :

- Les analyseurs déterministes et les analyseurs ATN fournissent des réponses différentes aux questions relatives au commencement, à l'arrêt et au rattachement. Pour les analyseurs ATN, les réponses sont d'une manière implicite et rigide dépendantes de la façon dont la procédure d'analyse utilise les réseaux. Pour les analyses déterministes, les réponses sont d'une manière explicite et plus souple dépendantes des règles faisant intervenir la création, le dépôt et le rattachement ;

- L'analyseur déterministe dispose d'un tampon de prévision à trois éléments, tandis que les analyseurs ATN ne considèrent que le prochain élément ;

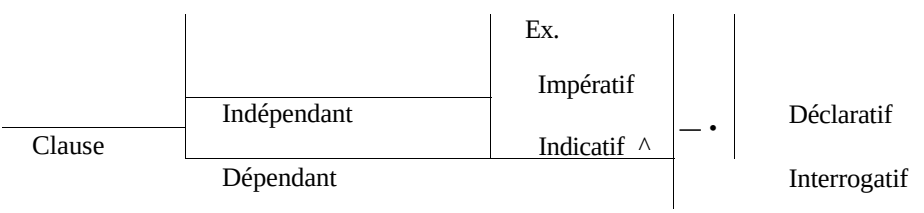
- Dans la grammaire ATN la connaissance syntaxique est exprimée dans une procédure d'interprétation, dans des noeuds et des arcs, et dans des procédures sur arcs relativement obscures; dans les grammaires déterministes, la connaissance syntaxique est exprimée dans une procédure d'interprétation et dans des règles condition-action relativement claires et transparentes. Mais, tandis que la partie inteprète d'un analyseur ATN est plus simple à exprimer que celle d'un analyseur déterministe, les procédures sur les arcs ont tendance à être bien plus complexes que les règles condition-action d'un analyseur

4.1.3.6.- **Grammaire systémique** (extrait de [FEIGENBAUM, 86].
La grammaire systémique développée par Michael Holliday et d'autres à l'université de Londres, est une théorie à l'intérieur de laquelle la structure linguistique telle qu'elle est liée à la fonction de l'utilisation du langage, souvent appelée **pragmatique,** est étudiée. Selon Holliday. une vision d'une structure linguistique qui ne porte aucune attention aux demandes tonctionnelles laites sur le langage manque de perspicacité, étant donné qu'elle n'offre aucun principe pour expliquer pourquoi la structure est organisée d'une façon plutôt que d'une autre. Ce point de vue, s'oppose à celui de la grammaire trans-formationnelle, qui s'est intéressée à la structure syntaxique d'un discours séparé de son utilisation prévue.

Dans cette perspective, Gholliday distingue trois fonctions générales du langage et un modèle de grammaire qui a quatre catégories primitives. Voici un extrait :

- Les fonctions du langage sont les suivantes :
  •**La** fonction d'idéation sera l'expression du contenu.
  •**La** fonction inter-personnellc s'applique au but d'une déclaration.
  •**La** fonction textuelle reflète la nécessité de la cohérence dans l'utilisation du langage (ex. comment une phrase donnée est liée aux phrases précédentes).

- Le modèle de grammaire contient quatre catégories primitives :
  •**Les** unités de langage qui forment une hiérarchie (phrases, mots, groupes des mots, les morphèmes) ;
  •**La** structure des unités. Chaque unité est par exemple composée d'une ou plusieurs unités
  du rang inférieur, et chacune de ces composants remplit un rôle particulier ;
  •**La** classification des unités est déterminée par les rôles à remplir au niveau supérieur. Les
  classes de groupes anglais, par exemple, sont le verbe qui sert d'attribut, le nominal, qui
  peut être sujet ou complément ; et l'adverbial, qui remplit la fonction d'adjonction ;
  •**Le** système. Le système est une liste de choix représentant les options permises à un élocuteur.



4.1.3.7.- Grammaire de cas

Les systèmes de cas, tels qu'ils sont utilisés à la fois dans la linguistique moderne et en intelligence artificielle sont descendants du concept de cas, tel qu'il apparaît en grammaire traditionnelle. Traditionnellement, le cas d'un nom est dénoté par une terminaison d'inflexion indiquant le rôle du nom dans la phrase. Le latin, par exemple, a au moins six cas : nominatif, accusatif, génitif, datif, ablatif, et vocatif.

Avec la grammaire de cas, on peut définir les aspects de la structure de surface de la phrase. Dans le cadre transformationnel et comme révision, Filmore (1968) propose une nouvelle approche pour la structure profonde de la phrase. L'idée centrale est que la proposition intégrée dans une phrase simple a une structure profonde constituant en un verbe, et une ou plusieurs phrases nominatives. Chaque phrase nominative est associée avec le verbe dans une relation particulière appelée "relation syntaxique d'ordre sémantique" ou des cas (extrait de [FEIGENBAUM, 88]).

4.1.3.8.- Grammaire d'unification, (extrait de [SABAH, 88]).

11 y a une autre grammaire, qui donne au lexique une importance primordiale. Cette grammaire développe une analyse fondée principalement sur les caractéristiques syntaxiques et sémantiques des mots de la phrase. Cette optique est très voisine de celle des réseaux sémantiques qui cherchent à représenter le sens des phrases.

Le principe de base est d'utiliser le même formalisme pour représenter les éléments du dictionnaire, les règles de grammaire et les structures internes des phrases. Le modèle utilisé est celui de schéma de la relation de "Frame".

{ (attribut = valeur) }

Où chaque couple est considéré comme une description partielle de ce constituant, indépendant des autres. L'ordre de ces couples n'est pas pertinent et, la notion est la même, que l'attribut soit une fonction, la description d'un trait ou une réalisation lexicale.

Un schéma est représenté dans une "boite", symbolisée par des crochets "[ ]". L'empilement des boites les unes dans les autres traduit alors la structure, de façon analogue à un parenthèsagc. Si, à un niveau donné, plusieurs solutions sont possibles, on les représentera par des boîtes réunies par une accolade. Cette dernière notion, ainsi que le fait que l'ordre des éléments à l'intérieur d'un schéma n'est pas pertinent, établit une distinction entre cette théorie et les grammaires catégorielles.

Quelques.

1- Entrées lexicales - On pourra, par exemple, trouver dans un dictionnaire les schémas suivant pour le mot "table"

Catégorie	B	Nom
Nombre	=	Singulier
Genre	=	Féminin
Lex	=	Table

Si le dictionnaire contient l'ensemble des formes fléchies, il y figurera également des descriptions de verbes conjugués, comme par exemple "ferma" :

Catégorie	=	Verbe
Nombre	=	Passé simple
Type	=	Action
Racine	=	Fermer
Lex	=	Ferma

Avec cette méthode, on peut aussi représenter les cas ambigus. Par exemple la description de "la", qui peut être un article, un pronom ou un nom (la note de musique), sera représenté de la façon suivante ; une accolade regroupe alors ces trois possibilités :

Catégorie	Article
Nombre	Singulier
Type	Défini
Genre	h'éminin
Lex	la

Catégorie	l'ronon
Nombre	Singulier
Type	Personnel
Personne	Troisième
Lex	la

Catégorie	Nom
Nombre	= Singulie"
Genre	Masculin
Lex	la

2 - Règles de grammaire - Pour représenter la description des règles de grammaire, on utilise aussi le même formalisme, comme pour les cas précédants. Une phrase est un ensemble de mots et de groupe, dont la structure change selon le contexte. Ainsi une phrase simple peut être représentée comme : GN + GV + GN ou GN + GN + GV etc. Chaque groupe est découpé en catégorie, ces catégories seront représentées selon les schémas précédants . Voici un exemple :

```
Catégorie   =   Phrase
JEorme      =   (GN GV GN

                    CAT     =  GN
                    Forme   =  (DET* ADJ NOM* ADJ)
                    DET
                    NOM
                   ——ADJ ————————————————————
          GN =      —                          —
                    CAT     =  VERBE
                    Forme   =  (VERBE)
                   UŒRBE    =  [CAT=  VERBE)   ___

                    CAT     =  VERBE
          GV =      Forme   =  (VERBE)
                   UŒRBE    =  [CAT=  VERBE]

Catégorie   =  phrase
Forme       =  (GN GN GV)   ]
```

Dans cette grammaire la représentation interne des phrases est aussi prévue. Cette représentation pourrait donner accès au sens de la structure de la phrase (selon G. Sabah). Nous rappelons que ce formalisme est équivalent à une grammaire indépendante du contexte (type 2 de Chomsky), de ce t'ait, nous accepterions l'idée du sens avec un peu de réserve.

4.2.- Choisir **une stratégie** d'analyse

Dans toutes les applications informatiques utilisant les langues, un certain nombre de processus de base sont nécessaires : il faudra "comprendre effectivement" un texte pour effectuer une tâche quelconque sur ce texte [SABAH, 90], on doit ensuite utiliser des procédures de "raisonnement et d'analyse". C'est-à-dire choisir une stratégie d'analyse. Il ne s'agit pas de choisir une seule stratégie, le processus de reconnaissance des formes syntaxiques suppose en fait une interaction de morphologie, syntaxe, et du lexique, c'est-à-dire la nécessité de différents aspects qui fonctionnent en même temps.

On présente, très brièvement, les principales analyses qui pourraient intervenir, de près et de loin, dans le processus de la conception d'un analyseur morpho-syntaxique :

L'analyse morphologique

Les premières connaissances nécessaires à un logiciel qui analyse et traite les langues naturelles écrites sont des connaissances morphologiques. En effet, dans la phrase

initialement fournie au système, un même mot peut apparaître sous des formes diverses. Afin de reconnaître qu'il s'agit du même mot, une solution souvent utilisée consiste à reconstruire la forme de référence à chaque occurrence du mot. On utilise alors des connaissances au sujet de la forme canonique d'un mot : de ses racines et de ses terminaisons possibles ; des règles qui régissent leurs associations possibles.

L'analyse syntaxique

Quel que soit un texte, les mot dont il se compose sont assemblés entre eux. La syntaxe, ou grammaire dans un sens dévié, correspond à la partie de la linguistique qui traite de la connaissance ou de la science des assemblages de mots, un mot n'étant jamais qu'un ensemble d'informations particulières. Les mots sont susceptibles d'être agencés entre eux selon des règles de construction logiques et formelles, dont les nouvelles grammaires structurales, descriptives, génératives, transformationnelles ou catégorielles tentent de retrouver et de définir les lois ou structures internes, immanentes aux différentes langues et aux langages. La démarche n'est pas sans intérêt et a permis de mettre au point, entre autres exemples, les **analyseurs morpho-syntaxiques** intégrés à un nombre croissant d'applications informatiques en traitement de texte, en traduction automatique et en recherches documentaires informatisées [E. Roubaud, 67].

L'analyse syntaxique, effectuée par des **analyseurs syntaxiques,** consiste à explorer les textes préalablement traités sur un plan morphologique et lexical pour en extraire les règles de construction et d'agencement des mots, de manière à pouvoir analyser la structure des phrases. Diverses méthodes sont utilisées par les analyseurs syntaxiques pour rechercher ces différentes façons de regrouper les mots. Les uns utilisent des démarches formelles, élaborées à partir de 1955 par des linguistes et des logiciens, comme *les grammaires transformationnelles, descriptives, catégorielles,* conçues à l'origine pour aborder avec rigueur les difficultés posées par la traduction automatique, ou encore de toutes nouvelles grammaires, dite à *réseaux de transition augmentés ("augmented transition network") ou fonctionnelles lexicales ("lexical function grammars"),* plus proches du formalisme informatique. D'autres utilisent des démarches probabilistes et statistiques, pour étudier en priorité les formes de regroupement de mots les plus vraisemblables ou les plus probables. Aucune grammaire formelle ne parvient certes à résoudre toutes les difficultés grammaticales et logiques que pose une langue donnée. Ainsi la syntaxe permet de définir des règles de réécriture ou plutôt de combinaison et de ré-agencement de mots et, permet aussi, d'élaborer des programmes informatiques de reconstruction et de recréation de textes (VUILLEMIN, 87].

L'analyse lexicale

L'analyse lexicale consiste, toujours en termes informatiques, à déterminer à partir d'un dictionnaire préétabli les catégories lexicales des différents lemmes identifiés, autrement

dit à les classer, selon les terminologies utilisées, en formes fonctionnelles et non fonctionnelles, grammaticales ou lexicales, en formes nominales, verbales, adjectivales et autres, etc. Les découpages sont varies, selon les buts recherchés, mais le principe général est identique, avec, comme résultat, qu'un texte ainsi traité se présente progressivement comme une suite ou séquence de mots, chacun accompagné d'une certaine comme d'informations nouvelles, établies à partir de dictionnaires successifs, et sur sa forme et sur sa nature, et susceptibles de se prêter à son tour à de nouveaux traitements encore élaborés.

L'analyse sémantique

Par l'intermédiaire d'une nouvelle catégorie de programmes de traitement appelés *analyseurs sémantiques,* l'analyse sémantique peut traiter des ensembles de données morphologiques, lexicales et grammaticales pour transformer la forme syntaxique des mots ainsi obtenue en une forme logique, à partir de laquelle ces analyseurs sémantiques pourront appliquer des raisonnements et en déduire des inférences. Le formalisme linguistique initial est transformé en un formalisme logique. Là encore, les démarches ou procédures de représentation, ou langages de représentations, sont très nombreux, sans exclure, dans la mesure où ces analyseurs sémantiques s'efforcent de multiplier en général les approches d'un même énoncé pour en déduire des conclusions, en principe veaisemblables sur ses significations possibles et, nature profondément métaphorique du langage humain, qui fait que même les expressions les plus usuelles, les plus banales, sont encombrées de métaphores (de comparaisons implicites entre les différents sens d'un mot) dont beaucoup sont devenues inconscientes, et impossibles à comprendre en dehors d'une analyse de leur contexte.

L'analyse pragmatique

Cette étape correspond au dernier niveau d'analyse des programmes de compréhension des textes et des TALN (La séquence de quatre premières étapes, qui fait passer de la phrase au "sens", est souvent appelée par les informaticiens : "phase de compréhension"). L'analyse sémantique précédente ayant permis d'aboutir à certaines inférences ou conclusions sur le sens latent d'un énoncé, à partir, en somme d'une analyse intrinsèque de ses significations, les analyseurs pragmatiques sont d'autres programmes de traitement qui vont essayer de comparer ces résultats ou inférences à ce que l'on peut déduire au contraire d'une analyse extrinsèque du contexte de cet énoncé. En effet, toute phrase écrite par un auteur ou dite par un locuteur est prise dans un réseau de relations et de correspondances avec ce qui a été écrit ou dit et avec ce qui sera dit ou écrit, implicitement ou explicitement. Seule l'étude de ces contextes peut lever ces ambiguïtés, sous réserve de pouvoir définir d'autre règles d'analyse, dites pragmatiques, déductives et inférentielles, capables de parvenir à ce résultat [VUILLEMIN, 87].

BONNET (Alain), HATON (Jean Paul), TRUONG, NGOC (Jean MICHEL), Systèmes experts : vers la maîtrise technique. Paris, hier Editions, 1986.

BOUCHE (Rihard), 1988. Valeur référentielle et langage d'indexation dans les systèmes d'information documentaires. Communication faite le 28 Novembre 1988 au Colloque "Archives et Temps Réel", organisé à Lille par le CREDO (Université Lille **III);** l'ABDS Nord, et les Archives du Nord.

BRETON (Philippe), 1987. Une histoire de l'informatique. La Découverte, Paris 1987.

CARRE (R), DEGREMONT (J.F), GROSS (M), PIERREL (J.M), SABAH (G), Langage Humain et Machine, Presses du CNRS Paris 1991.

CHATAIN (J.N), DUSSQUCHOY (A), 1987, Systèmes experts : méthodes et outils. Paris Eyrolles 1987.

DUPONT (Pierre), 1990. Eléments logico-sémantiques l'analyse de la proposition. Publié chez P. Lang (Sciences pour la communication); Bern 1990.

CARRIER (Claude). 1990. Maîtrise de l'intelligence Artificielle. Marabout Alleur. Belgique.

GOUJON (P). 1975. Mathématique de ia base pour les linguistes. Hermann. 1975 Paris.

GRIZE (Jean-Biaise). 1986. Logique naturelle et vraisemblance. Actes du colloque logique naturelle et argumentation. Royaumont 1986.

HATON (J.P). 1989. L'Intelligence Artificielle. Que-sais-je PUF 1989 Paris.

HARMANN (Hans). 1978.Meinen und Versienen Grundiige einer Psychologischen Semantik. Frankfurt ann Main, Synrkamp 1978. traduit et cité par [SCHMID, 92].

HUDAULT (Bénédicte): L'intelligence artificielle à travers Turbo-Prolog. 1991. Paris, Editions Marketing.

KAYSER (Daniel). 1985. Des machines qui comprennent notre langue in "La Recherche"n°17, octobre 1985.

LAINE (Sylvie). 1982. Extraction et sélection des descripteurs complexes dans un ensemble de textes pour leur indexation automatique. Thèse de Docteur-Ingénieur.
Université Claude BERNARD Lyon I.

LE GUERN (Michel), BERRENDONNER (A), BOUCHE (R), ROUAULT (J). 1980. Pour une méthode d'interaction pondérée des composants morphologique et syntaxique en analyse automatique du français. T.A. informatique 1980, n°l.

METZGER (Jean Paul). 1988, Syntagmes nominaux et information textuelle: reconnaissance automatique et représentation. Thèse d'Etat Es Sciences l'Université Claude Bernard, Lyon 1.

MILLER (Philippe). TORRIS (Thérèse), 1990. Formalismes syntaxiques pour le traitement automatique du langage naturel. Hermès Paris 1990.

QUILLIAN (Ross). 1966. SEMANTIQUE Memory. Boit. Beranek and New man Inc. Octobre 1966.

SABAH (G). 1990. L'intelligence Artificielle et le langage: Représentation des connaissances. Paris. HERMES 1988-1990, tomes 1 et 2.

SAYER (Oliver), 1987. Introduction d'une base de données textuelles SYDO. Mémoire de DEA: Conception de système d'informations spécialisées.

SCHMID (Anne-Marie), 1992. in BULAG (n°18): Conférence faite au Département de l'inguistique de l'Université de Besançon le 17 mai 1992.

STAHL (Gérold). 1983. Moins de traitement sémantique et plus de préédition en traduction assistée par ordinateur. Actes du colloque organisé par l'Université de Mets en juin 1983. in: La recherche française par ordinateur, publis par: C. CHARPENTEIER et J. DAVID, Slatkine-Champion, Genève-Paris 1985.

TOWNSEND (Cari), 1988. Turbo Prolog: applications. Sybex. Paris.

VAUQUOIS (Bernard). 1975. La traduction automatique à Grenoble. Documents de linguistique quantitative, n°2 Dunod 1975.

VOYER (Robert). 1987. Moteus de systèmes experts. Eyrolles 1987.