TRAITEMENT AUTOMATIQUE DES LANGUES NATURELLES : EVOLUTION ET PERSPECTIVES

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

INTRODUCTION

Les domaines d'application de l'informatique se sont considérablement étendus au cours des dix dernières années et cette tendance continue. les applications de l'informatique arrivent ainsi à toucher tous les secteurs d'activité; qu'il s'agisse de la recherche scientifique et technique, de l'information et de la documentation automatique (banques de données), du développent en bureau d'étude (conception assisté par ordinateur), de la gestion de production, du personnel et des stocks, de la facturation, de la production automatisée, de la diffusion d'information par la presse, d'exploration de l'espace (conception, lancement et contrôle des fusées) de la modélisation économique ou sociale, de la formation et de l'enseignement assisté par ordinateur EAO... etc. "L'information, par son développement et son impact sur des domaines variés, constitue un fait de société, ses effets se résument dans l'expression : informatisation de société (BRETON, 87).

Parmi les perspectives ouvertes par l'informatique, il en est une particulièrement fascinante: La conception de "systèmes" et de machines qui comprendraient notre propre langue. En fait, un tel objectif est très mal défini (KAYSER,85) et ses applications sont souvent mal connues. Au vu de nombreux résultats incomplets et des tâches inachevées, on constate, en effet, que le traitement et "l'indexation automatique de documents écrits en langue naturelle dont de nombreux informaticiens et spécialistes de la transmission de l'information rêvent maintenant depuis près de trente ans n'en est encore qu'au stade de la petite enfance" (METZGER, 88). De nombreuse méthodes et approches de nature très variées et diverses ont été mises en œuvre : Chacune de ces méthodes s'applique à un ensemble de problèmes très limités, aucune de ces méthodes ne représente une solution universelle.

Traditionnellement, le traitement automatique des langues naturelles (TALN) est l'un des domaines essentiels de l'intelligence artificielle (IA). Les domaines abordés par L'IA. sont, en fait, multiples. Il existe néanmoins un ensemble de points communs entre tous les systèmes d'IA qui permettent de mieux cerner les caractéristiques majeures de cette discipline.

Un programme d'IA se caractérise (HATON 89) d'abord par le fait qu'il manipule des informations symboliques beaucoup plus que des nombres, comme c'est le cas en informatique "classique". Ces informations représentant des concepts, des règles, des objets, des faits identiques à ceux habituellement pris en compte par l'être humain lorsqu'il raisonne. Cela n'exclut pas bien entendu l'utilisation de procédures de traitement numérique, mais l'exploitation des résultats se fera en général de façon symbolique.

Une seconde notion en IA est celle de méthodes heuristiques (qui aident à trouver la solution), par opposition aux méthodes algorithmiques classiques. Un algorithme, codé dans un certain langage de programmation pour fournir un programme, consiste en une description exhaustive de séquence d'opérations à mener pour résoudre un problème donné. Une heuristique est une méthode de résolution qui emprunte des voies non déterministes et dont le succès n'est pas garanti, mais qui, lorsqu'elle "marche", permet souvent une grande économie de temps de calcul. En cas d'insuccès, Il est nécessaire de revenir en arrière et d'essayer une autre solution.

Une autre caractéristique de l'IA est de s'accommoder de situations où les données et les informations traités sont incomplètes, inexactes, voire conflictuelles.

Une notion majeure de l'IA est celle de connaissance, et notamment les systèmes experts, constituant un pan majeur de l'édifice actuel de l'IA.

Enfin, l'IA est par essence pluridisciplinaire. La réalisation d'un système fait largement appel aux techniques avancées de l'informatique. Mais l'IA puise également ses sources dans d'autres disciplines : logique et psychologie cognitive, linguistique, ergonomie, philosophie et sans doute un jour neurosciences et biologie.

Les caractéristiques propres à l'IA qui viennent d'être mentionnées se retrouvent dans un ensemble d'applications. Nous allons les présenter brièvement.

1- Traitement automatique des langues naturelles écrites ;
2- Interprétation d'images et vision par ordinateur ;
3- Démonstration automatique de théorèmes ;
4- Traitement automatique de la parole ;

5- Systèmes experts ;
6- Robotique ;
7- Jeux.

1- ASPECTS HISTORIQUES

Dès l'apparition des ordinateurs, des tentatives ont été faites en vue du traitement de la langue naturelle. Les recherches en cryptographie (ou science des messages codés) imposées par la seconde Guerre mondiale avaient montré que certaines manipulations purement statistiques pouvaient faire apparaître des propriétés à comprendre des messages transmis en langue naturelle. Avec les difficultés rencontrées, il apparaît maintenant bien déraisonnable que, par des méthodes analogues, l'ordinateur pourrait faire œuvre utile dans ce domaine aussi complexe que la traduction d'une langue vers une autre. "Ces premiers efforts se sont donc soldés par un échec, principalement parce qu'ils partaient d'une vision trop simpliste, trop réductrice, de ce que sont réellement les langues naturelles: elles ne peuvent être traitées automatiquement en séparant artificiellement la forme et l'usage. La non-considération des caractéristiques essentielles partagées, sous une forme ou sous une autre, par toutes les langues, explique l'échec de ces premières tentatives. Ces caractéristiques impliquent généralement qu'il n'y a pas de correspondance exacte entre l'ensemble des mots ou des phrases et l'ensemble des "sens" (SABAH, 90).

Les premières tentatives de traitement des langues naturelles écrites ont concerné, dès 1946, la traduction automatique (TA). L'ordinateur était alors considéré comme un simple dictionnaire électronique. Deux autres, Warren et Donald Booth pensaient alors que les méthodes utilisées pour découvrir les codes secrets (basées sur les tables de fréquence de lettres et de mots), pouvaient s'appliquer à la traduction. Ils ne visaient donc pas une quelconque compréhension du message contenu dans les textes, mais pensaient traiter les séquences de lettres ou de mots de manière purement formelle. Les difficultés qu'ils entrevoyaient étaient l'incorporation de dictionnaires complets des deux langues, le sens multiple de la plupart des mots (polysémie), ainsi que le fait que l'ordre des mots est différent d'une langue à l'autre, même si elles ont des structures proches. en 1952 a lieu au MIT (Massachusetts Intitule of Technology), avec des précurseurs tels que Bar-Hillel, la première conférence sur la TA. A Cette époque, la TA se réduit à une traduction mot à mot et à une réorganisation syntaxique (c'est-à-dire respectant la grammaire) de la phrase traduite. Cette énorme sous-estimation du problème se fera cruellement sentir et affectera l'histoire de la TA jusqu'à une époque récente.

L'intérêt pour la traduction automatique commence à diminuer très fortement vers les années 1960 après une quinzaine d'années d'efforts infructueux. Au cours de ses tournées de conférences, Y. Bar Hillel montrait que la traduction nécessitait la mise en machine de connaissances contextuelles et encyclopédiques, c'est-à-dire le recours à de véritables

bases de connaissance, ce qu'il jugeait "évidemment chimérique" (BONNET et HATON, 84) ; son diagnostic, fort peu apprécié à l'époque, car il eut pour conséquence de couper brutalement les vivres aux recherches sur ce sujet. En fait, sur la demande du gouvernement américain, une équipe d'experts (l'Automatic Langage Processing Advisory Council : ALPAC) fait une enquête et rédige un rapport dans lequel il apparaît que la traduction automatique en l'état de connaissance de l'époque coûte environ deux fois plus chers que la traduction humaine et donne des résultats nettement moins bons. Ce diagnostic amène un arrêt de la plus grande part des financements publics aux USA puis en Europe. L'US-Air force, aux USA et la DRET (Direction des Recherches et Etudes Techniques du Ministère de la Défense) en France continueront cependant à financer des équipes réduites.

La leçon tirée de ces premiers efforts fut que, "traduire sans comprendre", était impossible. Il fallait donc trouver des moyens de représenter la signification d'une phrase ou d'un texte, lever les ambiguïtés possibles de certains mots, ce qui nécessitait d'utiliser le contexte et peut-être de disposer d'un modèle du monde. "Il fallait aussi prendre en considération la différence entre la reconnaissance et la génération, car beaucoup d'échecs en traduction automatique proviennent essentiellement de l'absence de symétrie entre la réception et la production" (LE GUERN). La recherche reprendra donc au moins dix ans plus tard, après l'enregistrement de progrès dans la compréhension de toutes ces notions.

Les années 60 représentent la période pré-sémantique du traitement du langage naturel; à partir notamment des travaux de N. Chomsky sur les grammaires formelles et les grammaires transformationnelles (1957-1969) qui débouchent sur les traitements purement syntaxiques. Les premiers systèmes apparaissent alors aux Etats-Unis : Base-ball (interface avec une base de données sur les matches d'une saison), Student (résolution d'exercices d'algèbre), Sir ou encore Eliza (simulation du dialogue entre un psychiatre et son patient). Tout ces systèmes sont fondés sur la simple recherche de mots clés dans la phrase à analyser et sont donc très limités.

Dès son apparition, la théorie chomskienne s'est développée, parallèlement, mais de façon indépendante aux autres recherches du TALN, comme une étude mathématique, et non pas linguistique, et a influencé fortement la science informatique dans la réalisation de langages de programmation informatiques (langages artificiels). Mais cette théorie devient aussi un outil théorique et pratique important pour l'analyse et la compréhension des langues naturelles et influence radicalement toute la recherche linguistique, y compris les travaux en IA sur le TALN.

La contribution de Chomsky fut double. D'une part, il a introduit la théorie des langages formels et a défini une hiérarchie de classes de grammaire et de langages devenue depuis classique, en informatique comme en linguistique; D'autre part, il a proposé le modèle transformationnel comme alternative aux grammaires régulières et aux grammaires non

contextuelles qu'il affirmait insuffisantes pour la description du langage naturel soutenait que la capacité générative faible des grammaires régulières ne suffisait pas à la description du langage naturel et que les grammaires non contextuelles étaient insuffisantes en capacité générative forte (MILLER, 90).

Effectivement, la théorie chomskienne, malgré ses nombreux avantages, ne présente pas un modèle universel, car elle n'envisage pas de solutions globales et détaillées pour aborder toutes les dimensions de la langue en TALN.

En fait la théorie chomskienne donne quelques résultats en morphologie et en syntaxe mais n'a permis à ce jour de fabriquer un modèle formel de la morphologie et de la syntaxe d'aucune langue naturelle (de nombreux linguistes considèrent d'ailleurs aujourd'hui que le formalisme de Zellig Harris est plus simple et plus puissant). La théorie chomskienne des langues naturelles achoppe par ailleurs lourdement sur la formalisation du sens. Cet échec selon ces auteurs pose un problème gênant puisque les ambiguïtés restant à la fin des phases d'analyse morphologique et syntaxique ne pourront pas être levées : dans un cas extrême, une phrase de 17 mots peut par exemple donner lieu à 572 interprétations syntaxiques (CARRE 91).

Au vu des limites des premiers résultats, qui étaient obtenus par des traitements exclusivement morphologiques et syntaxiques, certains chercheurs ont alors tenté d'aborder des traitements qui prennent en compte la sémantique des énoncés.

A cet égard, le modèle des grammaires des cas de Charles Fillmore présente deux avantages fondamentaux qui expliquent leur influence sur les travaux d'intelligence artificielle. Le premier est d'offrir un modèle de la structure profonde d'une phrase où la sémantique joue un rôle essentiel. Le second avantage de ces grammaires est de tendre vers un mécanisme d'analyse purement sémantique fondé sur les restrictions de sélections issues des ossatures des verbes.

Des rélexions importantes sur la représentation des connaissances voient aussi le jour, principalement à l'initiative de Ross Quillian (QUILLIAN 66) qui préconise l'utilisation des réseaux sémantiques pour représenter le sens des mots et des phrases en explicitant les relations des divers concepts entre eux grâce à des liens qui indiquent les sens de ces relations. Un réseau sémantique est une sorte d'immense tableau dont chaque case contient un élément (typiquement un mot), relié à certains autres par des liens de sens formalisés (équivalence, filiation...).

Les années 70 marquent le début de la période syntaxico-sémantique. L'importance du contexte et le rôle essentiel d'une bonne connaissance du domaine traité pour comprendre un texte sont ainsi mis en avant à cette époque. Les caractéristiques linguistiques

importantes qui posent des problèmes difficiles dans les programmes de l'époque sont principalement les *anaphores* et *l'implicite*. Parmi les travaux élaborés, on peut citer notamment les systèmes Lunar (de Willam Woods à BBN), SHRDLU (de Terry Winograd au MIT), Margri (de Shank à Yale). Lunar effectue une analyse syntaxique puis une interprétation sémantique de l'arbre syntaxique. William Woods (BONNET et HATON, 84) applique sa technique des ATNs (les réseaux de transition augmentés) à un système d'interrogation de bases de données en langue naturelle baptisé LUNAR. La base comporte des données d'échantillons de roches lunaires ramenés par les astronautes de la NASA.

L'analyse des questions se fait en trois étapes :
a- une analyse syntaxique produisant le ou les arbres syntaxiques possibles de la requête,
b- une interprétation sémantique de l'arbre syntaxique produit, aboutissant à une requête formelle,
c- l'exécution de la requête formelle commandant la recherche dans la base de données et la réponse en langue naturelle.

A la grammaire est associé un vocabulaire d'environ 3500 mots. L'analyseur est capable de comprendre de façon limitée des références pronominales et des constructions enchâssées.

Quant au système de SHRDLU, il utilise en fait des connaissances sémantiques en cours d'analyse ; ce système a eu le grand mérite de montrer la nécessité d'une interaction entre les diverses sources de connaissances disponibles et la possibilité de lever des ambiguïtés à un nivau d'analyse par le recours à un autre niveau. en 1973, Wilks propose aussi un important système de traduction de l'anglais vers le français.

En effet les recherches dans le domaine du TALN ne cessent guère d'évoluer, M. Minsky tente alors d'élaborer un cadre général de représentation des connaissances, les frames, cependant que R. Schank s'efforce d'identifier clairement les diverses connaissances nécessaires dans un système interprétant la langue naturelle.

L'apparition des frames (Minsky) et des scripts (Schank) correspondant à la théorie de la dépendance conceptuelle où une phrase simple peut se représenter par un schéma, marque le début de la période pragmatique, prenant en compte le rapport entre ce qui est énoncé et la réalité de l'univers d'application en ce même instant. Cette période se poursuit jusqu'à la période actuelle qui se caractérise par l'interaction de tous les niveaux de connaissances disponibles, en vue d'une analyse souple tolérant les fautes dans les énoncés.

2- LES OUTILS TECHNIQUES ET CONCEPTUELS DU TALN

Le TALN fait intervenir des domaines d'investigation très variés que l'on peut regrouper autour des cinq axes suivants:

2.1- LA LINGUISTIQUE

Parmi les domaines les plus recherchés pour le traitement automatique des langues naturelles, il faudra donner la place la plus importante à l'analyse linguistique. La linguistique joue un rôle primordial dans l'analyse et les applications du TALN ; Sans une analyse linguistique préalable et approfondie, toutes les méthodes envisagées dans ce domaine sont condamnées à l'echec. La linguistique théorique doit fournir des descriptions entièrement explicites organisées dans des théories cohérentes du savoir linguistique.

Les linguistes enfin cherchent à identifier les phénomènes caractéristiques de la langue dans toute leur étendue et à en donner des descriptions plus ou moins formelles. Il s'agit pour eux de mettre en évidence les règles qui donnent une description de la structure ou des fonctions des phrases d'une langue. Ce sont ces règles qui correspondent à la compétence linguistique d'un sujet. Il faut alors tenir compte de toutes les dimensions et composantes qui interviennent dans la langue.

2.2- L'INFORMATIQUE

L'informatique au sens plus étroit du terme se préoccupe de développer des outils logiciels pour le TALN par ordinateur. Il peut s'agir d'écrire des programmes de traitement dans des langages classiques (PASCAL, MODULA) ou spécialisés dans le traitement symbolique (LISP, PROLOG). Mais on peut également entreprendre de définir de véritables langages spécialisés orientés vers les applications linguistiques, tels que les DCG (Définite Clause Grammars) ou PART II; en fait les premiers travaux de A. Colmerauer sur le langage Prolog inspirèrent la communauté des chercheurs en programmation logique, et en particulier F. Pereira et D. Waren qui créèrent les grammaires à clauses définies (DCG) pour développer des formalismes basés sur une variété d'unifications, tels que les grammaires d'extraposition, les grammaires à causes et les grammaires de discontinuité. Enfin l'informatique théorique permet d'optimiser les algorithmes et programmes de traitement (MILLET 90).

Toutes les applications en TALN, nécessitent au préalable un processus d'élaboration qui est de plus en plus scindée en deux phases principales, chacune ayant une finalité propre :

- **La conception organisationnelle** qui propose des solutions conceptuelles et décrit niveau général en termes de macro-organisation ;

- **la conception opérationnelle** qui précise le niveau organisationnel en fonction de choix techniques appropriés ; elle fait en général l'objet de plusieurs scénarios dont nous allons essayer de présenter le choix de l'outil informatique dans le traitement automatique de la langue naturelle est une tâche délicate qui nécessite de diposer des outils adéquats, tant au niveau d'analyse que matériel et logiciel (langages, environnements de programmation et outils évolués de développement).

Du fait des caractéristiques propres des systèmes d'IA (manipulation de symboles et raisonnement, récursivité, etc...) tous les langages de programmation existant ne se prêtent pas bien à l'écriture d'un système d'IA, et en particulier d'un système de TALN.

Dans l'histoire de la conception de langages informatiques, on distingue usuellement quatre époques principales, dites "générations". Ce paragraphe utilise essentiellement les documents suivants : (GARRIER 91), (BONNET et HATON 86), (CHATAIN 87).

- Les langages de la première génération se limitent aux codes machines binaires ou décimaux-codés binaires, qui restent la seule langue "comprise" par les ordinateurs puisqu'elle correspond aux deux seuls états d'un circuit électrique : activé (1) ou non (0) ou d'un support magnétisé/non-magnétisé. Leur manipulation délicate les rendaient presque inaccessibles aux non-professionnels.

- Le langage-type de la seconde génération sont les "assembleurs" ensembles de codes némoniques plus faciles à retenir que les précédents, mais encore très proches de ceux des machines.
- Avec la troisième génération apparaissent les langages évolués ou symboliques (Cobol, Fortan, Basic, Pascal, Ada, Modula-2 C, etc..) les codes du langages sont remplacés par des mots proches des termes naturels.

Les premiers langages étaient spécialisés : Cobol vers la gestion des entreprises (Common Business Oriented Language) ; Fortran vers les calculs scientifiques (FORmula TRANslation-IBM, 1957) ; Basic vers l'initiation (Beginner All-purpose symbolic Instruction Code) ; Pascal, créé par Nikllaus With (Université de Zurich), avait également un objectif pédagogique : enseigner une méthode de structuration des programmes.

- Les langages de quatrième génération comme SQL (Langage d'interrogation structurée) sont encore plus proches du langage naturel. Ils sont "déclaratifs" et non procéduraux. D'une manière générale, les instructions sont moins nombreuses et plus puissantes. Le but est de rendre l'utilisateur final indépendant de l'informaticien pour ce qui concerne la consultation des fichiers des grandes bases de données relationnelles.
- Les langages de cinquième génération sont particulièrement adoptés pour résoudre les problèmes dont l'objectif est imposé quoique la procédure permettant d'y aboutir soit inconnue. Les relations connues (prédicats) entre objets (arguments) constituent la base des connaissances à partir de laquelle le système construit son raisonnement.

Les langages du TALN : Prolog et Lisp.

Parmi les langages les plus anciens qui ont été élaborés pour les applications de l'IA (TALN, système expert, etc...) on peut accorder une place privilégiée au **Prolog** et au **Lisp.** Cependant, l'étendue du domaine et les habitudes de travail des programmeurs ont conduit à utiliser également avec succès les langages classiques et algorithmiques.

Pour pouvoir comparer et choisir le langage le mieux adapté pour le traitement automatique des SN, on présente ici un extrait caractéristique des deux langages :
Lisp (LISt Processing), conçu par John McCarthy aux Etats-Unis dans les années cinquante au MIT, et dont de nombreux dialectes existent, les plus communs étant InterLisp et MacLisp. Il est un langage de traitement de listes, dont il tire son nom, bien adapté aux manipulations symboliques. Au cours de son développement, ce langage a bénéficié des environnements de programmation très riches (éditeurs, visualisation graphique, aides à la programmation etc...).

Tout objet manipulé par Lisp porte le nom d'expression symbolique (S-Expression en anglais) "qui est un" ou une "liste". Un atome est constitué d'une suite de caractères, c'est l'élément insécable du langage. Une liste est une suite ordonnée d'atomes ou de listes, entourés de parenthèses. la liste vide, (), appelée Nil joue un rôle important en Lisp. En effet, T(true) et Nil sont les valeurs renvoyées par Lisp lors de l'évaluation d'une condition logique (un prédicat).

La "boucle fondamentale" de Lisp se décompose en "saisie", "évaluation" et "affichage". L'interactivité entre l'opérateur et l'application est permanente.

Il permet de créer tous les foncteurs (ou "opérateurs") imaginables, nécessaires, au traitement d'un problème quelconque. Un foncteur créé par un utilisateur de Lisp est utilisable dans les mêmes conditions que les foncteurs standards.

Prolog, langage de PROgrammation en LOGique, conçu en 1972 par Alain Colmerauer appartient au groupe de langages dits déclaratifs. Son fonctionnement est basé sur la manipulation symbolique. la puissance de Prolog réside dans sa faculté de déduire des faits à partir d'autres faits. Prolog déduit donc ses conclusions à l'aide d'heuristiques et son système (lexique, syntaxe et sémantique) est structuré selon les modèles de la logique des prédicats. Les langages déclaratifs conviennent parfaitement au traitement automatique des langues naturelles.
Les premières étapes du développement de Prolog ont été principalement assurées à Edimbourg, avant que les japonais le choisissent pour leur ambitieux projet d'ordinateur "de 5ème génération". Un programme de Prolog est une suite de "clauses"; chaque clause peut être considérée à la fois comme génération des règles de réécriture et comme une formule représentant le sens. En effet, la clause :
SN(X,Z) ⟶ DET (X,Y) NOM(Y,Z) [une façon de monter qu'il y a un groupe nominal situé entre le mot X et le mot Z, c'est de montrer qu'il y a un déterminant entre le mot X ou et le mot Y, et un nom entre le mot Y et le mot Z] a exactement le même effet que la règle de réécriture :
SN ⟶ DET, NOM; mais la clause :

ONCLE (X,Z)⟶ FRERE (X,Y) PERE(Y,Z) [une façon de montrer que X est l'oncle de Z, c'est de montrer que X est le frère de Y, et que Y est le père de Z] exprime un sens de mot "oncle". Prolog est donc utilisé non seulement pour programmer les analyseurs syntaxiques et sémantiques, mais aussi pour représenter les connaissances nécessaires à la compréhension (extrait de KAYSER 85).

Le problème de choix entre Lisp et Prolog est un faux problème car les deux langages sont très proches de avec l'autre et chacun présente un certain nombre d'avantages et d'inconvénients. Le problème du choix est avant tout un problème de goût et de l'habitude du programmeur avec le langage utilisé.

Lisp et Prolog sont considérés comme des langages de la 5ème génération, il sont déclaratifs ou symboliques. Ces langages mettent l'accent sur les relations qui existent entre les faits.

Lisp et Prolog ne fournissant pas les mêmes fonctions, ils ne sont donc pas exclusifs mais plutôt complémentaires.

L'idée ne viendrait à personne de comparer Cobol à Fortran, ils ont tous deux un domaine d'application très différent l'un de l'autre.

Lisp est un langage de plus bas niveau que Prolog, et ce de fait il est suffisamment flexible pour incorporer les vertus des langages de plus haut niveau, style Prolog.

Lisp est un langage de construction de systèmes, un langage d'implantation alors que Prolog est essentiellement un langage de niveau utilisateur, un langage d'application.

C'est ainsi qu'un système Prolog pourrait être écrit en Lisp alors que l'inverse serait difficilement envisageable.

Lisp, de part son ancienneté, dispose de beaucoup d'implantations, d'environnements très développés, d'une solide expérience, de calculateurs spécialisés (machine-Lisp) et d'une normalisation en cours (CommonLisp). Après de multiples essais, Prolog pour sa part, possède les vertus de sa jeunesse (formalisme rigoureux, engouement dû à sa nouveauté..), mais aussi les défauts (environnement encore assez faible, pas encore de calculateurs spécialisés, manque de normalisation...).

Les langages mélangeant Prolog et Lisp sont peut être la réponse au problème du choix, mais les limites fonctionnelles, tant au niveau du temps qu'au niveau des moyens, ne nous permettent pas, pour l'instant, d'envisager simultanément ces deux langages. De ce fait,

nous avons préféré choisir le Prolog comme l'outil informatique qui se prête cependant mieux à l'établissement d'un système de règles.

Il se trouve que le langage Prolog a été conçu, à l'origine, pour la description de systèmes de réécriture (plus généraux que les systèmes hors contexte) : une clause Prolog = une règle de réécriture. Un interpréteur Prolog est à la fois général et non déterministe (solutions multiples) : on peut considérer, en outre, que son mode de fonctionnement est une généralisation de ce qu'on appelle habituellement une analyse descendante. Un analyseur syntaxique Prolog peut donc se présenter sous la forme d'un ensemble de clauses, où chaque clause représente une règle de réécriture. On peut, considérer l'interpréteur comme l'analyseur et le programme Prolog comme la grammaire, elle même (METZGER 88).

Par ailleurs le langage Prolog a d'autres avantages sur les langages de programmation dits "algorithmiques" (ou "procéduraux"), la programmation Prolog "permet l'écriture de programmes où ne transparaît que très peu la façon dont les opérations s'enchaînent (et, pourtant, tout y est) ; les algorithmes fondamentaux qui régissent l'exécution d'un programme Prolog (unification) sont implicites. De ce fait, Prolog rend plus aisée, pour l'homme, la lecture et la manipulation des programmes" (METZGER 88).

Parmi les quelques version commercialisées de Prolog, on peur distinguer le Turbo Prolog de Borland, version 2 qui est un outil perfomant permettant des programmations concises et rapides pour les applications de la langue naturelle.

Turbo Prolog qui est un langage destiné à la programmation sur les ordinateurs IBM Pc et compatibles et certains Mac d'Apple, n'est pas un Prolog standard, comme défini par Colmerauer (Marseille) et Clocksin et Mellish (dit aussi Prolog d'Edinbourg), mais il dispose de la majorité des particularités de Prolog ainsi que les fonctions les plus courantes dans les langages traditionnels.

Le turbo est un langage dit déclaratif, en se sens qu'on ne décrit pas comme dans les langages traditionnels (dits procéduraux) une méthode (ou algorithme) de résolution, mais que l'on déclare les propriétés du problème, le système se chargeant lui-même d'appliquer une méthode de résolution.
Un programme en Turbo prolog est articulé en quatre sections, identifiée chacun par un mot clé et se présente ainsi (HUDAULT 91) :

1- Domaines, pour les déclarations des types ;
2- Predicates, pour les déclarations des prédicats ;
3- Goal, pour but ou suite de buts ;
4- Clauses, pour les faits, les règles et les questions.

Les deux premières parties (domaines et predicates) correspondent aux déclarations du Pascal. la section clauses contient la base de connaissances. Toutes les sections sont optionnelles.

En Turbo prolog, les clauses sont de trois sortes ; faits, règles et questions. les faits sont toujours vrais, sans aucune condition ; les règles déclarant des choses vraies sous certaines conditions.En utilisant les questions, l'utilisateur peut demander au programme ce qui est vrai.Chaque clause de prolog est constituée d'une tête et d'un corps. Les faits sont des clauses dont le corps est vide, alors que les questions sont uniquement constituées d'un corps. Les règles ont une tête et un corps (non vide). Les faits et les règles peuvent se regroupés dans des bases de faits et de règles.

- Prolog et l'analyseur morpho-syntaxique

Ce paragraphe utilise, pour une bonne part, un extrait du document de (TOWNSEND 88).

Dans un système de prolog, l'analyseur est le composant principal de programme c'est à dire le cœur du programme qui permet de créer des systèmes d'interfaces en langage naturel (notées ILN) contenant des règles syntaxiques du langage étudié avec parfois quelques règles simples de sémantique. Le Prolog peut disposer, selon les objectifs de trois différents types d'analyseurs.

- Les analyseurs de bruits de fond ;
- Les analyseurs à graphes ;
- Les analyseurs morpho-syntaxiques ou à base de règles grammaticales.

Analyseur de bruits de fond
Ce type d'analyseur permet une analyse la plus simpliste. Il balaie la phrase pour retrouver simplement certains mots clés ou balises. Le reste est simplement ignoré ; ce sont les bruits de fond. Les balises activent ensuite les fonctions prévues dans l'ordinateur.

Ce type d'analyseur est correct pour des environnements restreints impliquant un vocabulaire limité et peu de commandes associées. L'un des inconvénients est que l'analyse ne se réalise pas correctement sur une phrase contenant une demande ambiguë. Le programme par exemple reçoit la syntaxe correcte mais, sans analyse sémantique, il ne peut exécuter la commande convenable : la non-prise en compte du contexte provoque une erreur.

Analyse par graphes d'états
Une machine à graphes voit la phrase comme une suite d'automates finis et traite en se déplaçant dans l'espace défini, passant d'un nœud à un autre. Le programme débute en

reconnaissant un nœud ou un état donné. Il lit ensuite le premier mot qui lui permet de parcourir un chemin vers un autre nœud en déclenchant une action donnée. Le traitement continue ainsi vers un nœud final ou une conclusion qui assurent la fin du programme.

Les analyseurs par automates d'états fonctionnent parfaitement lorsque le nombre de phrases type est limité. Pour les domaines plus importants, on aura un trop grand nombre de clauses pour faire passer l'automate d'un état au suivant. Pour obtenir des systèmes efficaces, il faut limiter le nombre de mots et de types de phrases acceptés. Cela implique une étude sérieuse du domaine et de sa limitation.

Analyseur morpho-syntaxique
Dans une perspective du TALN, l'élaboration d'un analyseur morpho-syntaxique en Prolog, nécessite au préalable la conception d'un programme scindé en deux structures principales étroitement liées : une "base de connaissances" permettant d'enregistrer des faits et des règles, et un mécanisme dit "processus de résolution" capable de calculer la validité d'une requête soumise à l'analyse, à partir des données enregistrées dans la base.

Il faut distinguer nettement une base de connaissance d'une base de données classique dont on ne peut extraire que les informations qui y ont été explicitement rangées (HATON 89°. De ce fait J.P. Metzger propose en TALN, la constitution d'une véritable base de connaissance considérée comme un outil très complexe et performant qui permet à la fois l'organisation des données nécessaires et l'analyse et l'interprétation de savoir (voir la connaissance) et de compétence.

On entend ici par connaissance toutes les formes de savoir de l'homme : objets qui forment le monde réel, faits et événements, concepts plus vastes correspondant à des groupements ou des généralisations d'objets de base, relations entre concepts, heuristiques et stratégiques de savoir faire ou encore procédures de raisonnement.

Dans la phase de l'application, le programme lit une phrase en entrée, vérifie avant tout si les mots de la phrases se trouvent dans le lexique et si la présentation syntaxique de la phrase correspond bien à l'une des règles de réécriture implantées dans le programme. A cet effet, lorsque la phrase est entrée, le programme transforme la phrase en une liste de symboles, qui parcourt cette liste en vérifiant que chaque mot de la phrase appartient au lexique et construit simultanément une liste numérique qui correspond à la nature morphologique de chaque mot. La liste numérique ainsi construite est unifiée aux diverses formes de règles et de phrases possibles (actuellement restreintes), et la structure syntaxique de la phrase sera affichée à l'écran.

Autrement dit, pour la reconnaissance de diverses catégories de phrases et de syntagmes, chaque forme du discours doit, au préalable, se soumettre à une analyse morphologique

complète. Pour ce faire, le programme consulte d'abord le lexique ; si la forme recherchée se trouve dans lexique, il attribue à la forme, sa catégorie lexico-morphologique et ses traits morpho-syntaxiques respectifs, et continue ainsi l'analyse de la forme suivante.

Dans le cas contraire; le programme doit consulter les règles grammaticales de reconnaissance qui représentent, selon un modèle bien précis, toutes les informations nécessaires de la formation des mots qui sont formés d'une base et des accessoires grammaticaux.

L'analyseur morpho-syntaxique est probablement le plus universel pour les traitements et interfaces en langages naturels ; il est le plus puissant, mais aussi le plus complexe.

Prolog est prévu d'origine pour tenter d'utiliser la logique afin de formaliser les règles grammaticales. Les règles de ce langage peuvent être des expressions écrites connues sous le nom de formes grammaticales. Ces formes grammaticales peuvent être employées pour interpréter les expressions de tous les langages, humains ou non. Naturellement, le langage doit être structuré selon ces règles pour être étudié.

Pour élaborer un analyseur morpho-syntaxique en Prolog, il faut d'abord commencer par couper les phrases en composants. On commence par le Syntagme Nominal et le Syntagme verbal, puis il faut poursuivre l'analyse en cherchant les constituants de chaque groupe, noms, adjectifs, prépositions, verbes, et autres éléments. L'opération se fait d'une façon descendante c'est à dire de haut en bas et commence par couper la phrase en deux éléments : SN, SV. Ensuite, les groupes (ou les syntagmes) sont étudiés pour leurs constituants propres.

Au sein de l'analyseur on peut introduire les règles morpho-syntaxiques, dites les règles de réécriture ou de production qui permettent le cheminement de l'analyseur pour la reconnaissance automatique des formes morpho-syntaxiques différentes. Les règles de réécriture peuvent être facilement traduites en langage Prolog. Voici on présente un exemple :

Les règles de réécriture	**Les règles en Prolog**
N'' ⟶ D'+N'	N''(_X,_Y) : -D'(_X,_t) & N'(_t,_Y).
Ex. : les deux chats de la concierge	
D' ⟶ D_DEF+D_NUM, les+deux	D'(_X,_Y) : -D_DEF(_X,_t) & D_NUM(_t,_Y)
D' ⟶ D_DEF : les.	D'(_X,_Y) : -D_DEF(_X,_Y).
D' ⟶ D_NUM deux	D'(_X,_Y) : D_NUM(_X,_Y).
N' ⟶ N+SP(+SP)	N'(_X,_Y) : N(_X,_t) & SP(_t,_Y)
Ex; chats de la concierge.	

L'avantage d'un tel système est l'analyse et la reconnaissance de la structure hiérarchique de la phrase aux niveaux de la syntaxe et de la morphologie des composants. Les phrases non reconnues sont rejetées. L'analyse est facilement implantée en Turbo Prolog. L'inconvénient majeur tient au nombre important de clauses à définir pour décrire toutes les formes grammaticales possibles et obtenir un système parfait. ces prédicats de définition constituent un dictionnaire des règles syntaxiques de la langue.

2.3- LES MATHEMATIQUES

L'étude mathématique des formelles des propriétés outils de traitement et des théories linguistiques est aussi nécessaire et fondamentale pour les progrès du TALN. "Le rôle de plus en plus important joué par les mathématiques dans l'élaboration des théories linguistiques a suscité le développement d'une discipline nouvelle, désignée sous le nom de "linguistique mathématique". En fait, les deux disciplines, linguistiques automatique et linguistique mathématique, se sont développées simultanément, l'une s'appuyant sur l'autre, la première autorisant l'accès à un volume considérable de données, la seconde apportant le secours d'une méthode, support des trois démarches fondamentales.
a- formalisation; b) classifications; c) recherche et construction de modèles" (GOUJON 75).

L'approche mathématique, recouvre une grande variété de méthodes qui présentent toutes l'avantage d'éviter au chercheur une exploration exhaustive et intelligente de la langue. Elle a été largement explorée dans les années 1950-1970. Ses limites intrinsèques font que c'est aujourd'hui principalement l'approche linguistique qui est utilisée. en fait selon de nombreux chercheurs, ni les méthodes mathématiques ni les approches statistiques ne sont guère applicables sans une analyse linguistique préalable et approfondie dans le domaine des TALN. Pour savoir plus sur la linguistique mathématique voir aussi les recherches de J.P Desclés.

2.4- LA LOGIQUE

Le recours à la logique comme mécanisme de représentation déductive est aussi nécessaire pour le TALN. "Il convient de distinguer la logique formelle (qui établit la validité d'un raisonnement sur la seule base de sa forme) de la logique naturelle (qui permet de raisonner à l'aide de la langue seule). Cette dernière part principalement de prémisses conformes à l'opinion générale et cherche à en tirer des conséquences acceptables par l'interlocuteur (GRIZE 86).

La logique formelle a deux en jubranches importantes et qui se recoupent :

a- La logique des propositions

En logique des propositions, on énonce des formules qui sont soit vraies (V), soit fausses (F). Une proposition étant une déclaration, un jugement ou l'expression d'une pensée. Ex. le malade a de la fièvre.

A partir de propositions, il est possible de construire d'autres propositions plus complexes en utilisant les connectives logiques fréquemment employées : ET, OU NON, IMPLIQUE, EQUIVALENT, représentés respectivement par des signes $\cap$, $\cup$, $\neg$, $\supset$, =. Ex; le malade a de la fièvre Et il est anémié.

La logique des propositions utilise, pour la déduction de formules, la règle d'inférence appelée "Modus ponens". Celle-ci permet à partir d'assertions connues (vraies) d'en déduire de nouvelles qui en dépendent SI ==> Q ET SI P est vrai. Cela s'exprime plus formellement par : (A ET (A ==>B))==> B.

b- la logique des prédicats.

Le calcul propositionnel n'est pas suffisant pour représenter des connaissances plus complexes. Dès lors on a fait recours à un second type de système de logique plus puissant : la logique des prédicats. Elle permet d'exprimer non des propositions vraies ou fausses mais de particulariser ou de généraliser des objets (VOYER 87). Nous rappelons ici que la logique des prédicats est une extension du calcul propositionnel traditionnel auquel on intègre la notion de variable et de quantificateur (existentiel et universel).

Par rapport à la logique des propositions, la logique des prédicats introduit deux notions importantes qui sont les variables et les quantificateurs :

- Un prédicat est une expression contenant une ou plusieurs variables et qui est susceptible de devenir une proposition vraie ou fausse si on attribue à ses variables certaines valeurs déterminées.
- Les quantificateurs quant à ceux, permettent d'indiquer la portée de toute assertion. Elle peut s'appliquer à tous les éléments de l'univers du discours (Il s'agit alors du quantificateur "pour Tout" ou "Quel Que Soit", noté également ") ou à certains éléments seulement (il s'agit du quantificateur "Il Existe"$).

La logique des prédicats est encore appelée logique du Premier Ordre. Plusieurs formalismes de représentation des connaissances ont été bâtis sur la logique du premier Ordre. C'est le cas notamment de PROLOG de SNARK et de TANGO.

La compétence logique, enfin, reconstruit l'implicite du discours en actualisant les présupposés et sous-entendus permis par le contexte. elle utilise à la fois des informations de type linguistique (signifiants) et extra linguistique pour les combiner grâce aux mécanismes de la logique naturelle (interfaces).

La formalisation des opérations de structuration logico-sémantique repose sur deux concepts difficiles à cerner : la notion de prédicat linguistique et celle d'inférence (SAYER 87).

Pour savoir plus sur le traitement et la structure logico-sémantique de la langue voir aussi (DUPONT 83 et 90).

2.5 - LES SCIENCES COGNITIVES

Depuis une vingtaine d'années le développement du concept de "cognition" s'est avéré aussi irrésistible en psychologie que le concept de "behavior" au début de ce siècle. "Par «sciences cognitives», il faut entendre tous les courants en psychologie cognitive comme en Intelligence Artificielle qui se réfèrent du systèmes de traitement de l'information pour décrire le système cognitif en fonctionnement. Selon ce paradigme, tous système intelligent, naturel ou artificiel, c'est à concevoir comme une machine à traitement d'information, ou à manipulation de symboles dont la finalité est de résoudre des problèmes au sein d'un environnement avec lequel il interagit" (DEMAILLY et LE MOIGNE 86). L'ambition des sciences cognitives est d'élaborer une véritable systémique, son programme de recherche consiste à étudier l'organisation et le traitement de l'information quels que soient les système matériels considérés.

Les recherches en sciences cognitives et en intelligence artificielle sur la représentation du savoir sont donc très importantes, notamment pour les aspects sémantiques et textuels du TALN. (voir aussi la thèse de P.Lavorel)

3- LES DOMAINES D'APPLICATIONS DU TALN

Actuellement les applications du traitement automatique des langues naturelles sont multiples. Parmi les principales, ayant donné lieu à des développement plus ou moins avancés, on trouve principalement l'indexation et la traduction automatique :

3.1- L'INDEXATION AUTOMATIQUE DE DOCUMENTS

L'informatisation des systèmes documentaires impose la nécessité d'une réflexion théorique sur les opérations qui en constiuent les composantes. elle ne change pas fondamentalement la nature de ces opérations : l'indexation automatique consiste, comme l'indexation manuelle, à sélectionner dans chaque document les éléments qui permettront à l'utilisateur de le retrouver dans le fonds documentaire quand il en aura besoin. Ces éléments, ce sont ***les descripteurs***. Dans un système d'information automatisé, les descripteurs sont ***les syntagmes nominaux*** des documents constituants le corpus. La solution adoptée pour l'indexation automatique est donc la constitution d'un analyseur morpho-syntaxique permettant d'extraire les syntagmes nominaux (LE GUERN 91).

Les syntagmes nominaux sont en fait les plus petites unités informatives des discours susceptibles de servir de base à une relation référentielle autonome (voir aussi le chapitre5).

L'indexation automatique consistera donc à identifier dans le document des mots ou des expressions qui, une fois isolés, sont sensés à eux seuls décrire le contenu du document concerné et convenablement stockés dans une base de données, doivent permettre de retrouver ultérieurement les informations pertinentes à une question donnée. L'identification de ses parties résulte de critères qui peuvent être les suivants :

a- Critères statistiques;
b- Référence à un univers construit à priori;
c- Critères linguistiques.

L'approche linguistique est féconde et permet d'espérer une solution solide au problème de l'indexation automatique (BOUCHE 88).

La fiabilité et le bon déroulement de l'indexation automatique de documents écrits en langue naturelle impose la préalable résolution de deux problèmes majeurs, étroitement liés:

a- "Celui de représentation du "contenu" du texte qui doit être intégrée dans une base de données ;

b- Celui du passage de la forme originale du texte à cette représentation ; cette forme originale ne constituant pas, cela semble maintenant admis par tous, une représentation de "l'information véhiculée" par le texte adaptée à la recherche d'information (METZGER 88).

Les recherches entreprises en vue d'automatiser certaines procédures documentaires ou d'apporter une aide au concepteur de système, semblent déboucher sur des résultats de plus en plus concrets. Des problèmes difficiles comme celui des descripteurs complexes ou des mots nouveaux semblent sur le point d'être résolus. On peut donc espérer que de nouveaux logiciels documentaires permettront de gérer plus facilement de petits fonds locaux, notamment sur des machines à traitement de textes (LAINE 81).

3.2- LA TRADUCTION AUTOMATIQUE

Les applications dans le domaine du traitement automatique des langues naturelles (ou TAO, traduction assistée par ordinateur) peuvent être rangées en deux grandes catégories:

- celles qui nécessitent une *analyse automatique* de textes en vue d'une reconnaissance partielle ou complète des unités le constituant : mots, phrases, concepts. On trouvera ici, par exemple, la correction orthographique, la documentation et la traduction automatique, l'interrogation en langues naturelles de bases de données scientifiques et techniques ou de bases de données à la formation assistée par ordinateur;
- celles qui comportent une *génération automatique* de formes linguistiques, pour un système de traduction automatique, par exemple, ou bien formuler des réponses à des questions.

Le processus de traduction d'un texte d'une langue à une autre comprend plusieurs étapes: la lecture du texte source, sa compréhension, la préparation documentaire ("constitution d'un lexique spécialisé") le transfert dans la langue cible et la mise en forme du texte dans cette dernière, la révision des erreurs de traduction, des lourdeurs stylistique et de l'orthographe, la saisie et la mise en forme typographique du texte final. On voit que, dans ce processus, une part importante du travail est consacrée à des tâches autre que le transfert proprement dit.

Les applications de traduction automatique ou assistée par ordinateur (TAO) incluent tout ou une partie de ces différentes étapes et, pour chacune d'elles, poussent plus ou moins loin l'automatisation. Comme dans tous les champs d'application des industries de la langue, une automatisation complète ne sera possible que dans des cas très particuliers (CARRE 91), de toute manière comme le dit Michel Le Guern, "Il y a une impossibilité absolue d'envisager une traduction automatique des textes littéraires".

Selon certains auteurs "à l'heure actuelle aucun système de traduction ne "comprendre" un texte en langue naturelle. Cela amène la question de savoir ce que "comprendre" implique chez l'être humain. Malgré certains progrès en psychologie cognitive, nous ne savons pas grand chose sur ce qui se passe en nous, lorsque nous saisissons le sens d'un texte" (HORMANN 78).

La traduction automatique est un travail complexe, aux facettes multiples, dont quelques unes peuvent être réalisées ou soutenues par l'ordinateur, à condition qu'on connaisse et reconnaisse honnêtement ce que la TA est susceptible de produire et ce qu'il vaut mieux confier au traducteur humain.
Il est donc nécessaire de préciser que toute tentative en TAO doit inclure et prendre en considération un certain nombre de contraintes qui sont en réalité les défis majeurs pour la mise en œuvre d'une traduction automatique authentique et performante :

- "Les premiers échecs en traduction automatique ont mis en évidence que ni la traduction mot à mot ni l'analyse grammaticale de structure de phrase ne sont suffisantes pour comprendre le langage naturel : il faut en réalité mettre en jeu un processus de raisonnement faisant intervenir des informations et des connaissances nombreuses et variées sur la langue, le contexte, les interlocuteurs, etc. (lexique, syntaxe, sémantique, pragmatique)" (HATON 89).

- "Quand le traducteur humain fait une traduction, il utilise non seulement sa connaissance des langues mais aussi une certaine connaissance du monde. On pourrait penser qu'en traduction assistée par ordinateur, un traitement purement syntaxique, qui établit une certaine correspondance d'expressions et de structures linguistiques entre les langues

respectives, n'est pas suffisant, qu'on a besoin aussi d'un traitement sémantique, qui prenne en considération les rapports avec quelque chose d'extra linguistique (STAHL 83);

- Les termes et les mots sont souvent polysémiques, il faut savoir le contexte et la situation de texte et faire le choix lexique en fonction de ces facteurs;
-Il faut tenir compte de la différence entre "signification lexicale" et "signification" textuelle", il faut donc une traduction en fonction du texte entier" (SCHMID 92).

Aujourd'hui malgré de nombreux handicaps et des problèmes méthodologiques et techniques la traduction automatique comme une activité industrielle et une application privilégiée d'études plus générales sur l'analyse et la génération automatique des langues naturelle ait trouvé une place raisonnable en informatique. Sans oublier les travaux des précurseurs américains, la plupart des pays industriels comme la France et le Japon contribuent un budget considérable pour la TA leur activité industrielle. De très nombreux systèmes de TAO sont actuellement à des stades de développement du moins avancés. A Grenoble, par exemple, le centre d'études pour la traduction automatique (C.E.T.A) d'abord et le groupe d'études pour la traduction automatique (G.E.T.A) en suite, dès sa formation, avait mené de nombreux projets de recherches très importants en TAO. Parmi les projets le plus récent qui a été initialement développé à l'université de Grenoble (au G.E.T.A) on peut citer l'ARIAN qui est actuellement en cours de l'industrialisation par la société SITE. Pour savoir plus sur les activités de GETA voir les travaux de Bernard Vauquois (VAUQUOIS, 75).

3.3- D'AUTRES DOMAINES D'APPLICATIONS DU TALN SONT :

- L'élaboration automatique de résumés de textes ;

- La génération automatique de textes (construction d'une forme syntaxique ayant un sens), problème dual de la compréhension est tout aussi difficile ;

- L'aide à la préparation de documents (détection et éventuellement correction des fautes, des tournures incorrectes, des incohérences), notamment en vue de leur édition/ parmi les applications d'analyse automatique du langage naturel, les correcteurs d'orthographe sont certainement les outils les plus répandus ;
- Le contrôle de systèmes informatiques ou automatiques (commande de robots) : accès à des bases de données (interface homme/machine en langage naturel), dialogue avec un Système expert ou un robot, enseignement assisté par ordinateur (EAO)
- La résolution de problème en langue naturelle (programmation d'ordinateur, à la limite).

4- LES METHODE DE CONCEPTION

Les tendance actuelles montrent que le traitement automatique de la langue naturelle est désormais une activité industrielle; mais en même temps les recherches dans le domaine

sont importantes car les réalisations sont encore très limitées ; les progrès sont et resteront assez lents, mais ils sont continus, à la fois sur le plan des améliorations techniques (analyseurs morpho-syntaxiques, raisonneurs déductifs, stratégies de raisonnement) et sur le plan théorique (méthodes de représentation des connaissances, modélisation des processus d'inférence). On est arrivé à un stade où l'on distingue bien ce qui est possible de ce qui ne le sera que dans fort longtemps (KAYSER 85).

Selon certains auteurs, l'utilisation de méthodes et techniques utilisées en intelligence artificielle ne paraît guère possible ou, en tout cas, tout à fait prématurée dans le domaine du TALN et de la documentation. "Comment peut-on, en effet, faire appel à des concepts comme ceux de réseaux sémantiques, de scénario, de schéma, de règle d'inférence... alors qu'on est incapable de faire une analyse morpho-syntaxique correcte et complète d'un texte écrit librement en langue naturelle ? (METZGER 88).

Il est donce important de préciser que parmi toutes les méthodes et techniques utilisées en intelligence artificielle, l'analyse morpho-syntaxique est aujourd'hui, dans la plupart des centres de recherches universitaires et scientifiques, considérée comme une solution et une nécessité préalable pour le traitement automatique des langues naturelles.

A Lyon le groupe SYDO* dès sa constitution en 1975, dans le cadre de ses activités de recherches en "conception de systèmes d'information spécialisés", s'attache à définir de nouvelles méthodes d'analyse automatique de corpus textuels et de recherches d'informations assistée par ordinateur et met l'accent avant toute solution en intelligence artificielle sur une analyse morpho-syntaxique, on ne peut intégrer les données fournies par un document qu'en les traduisant à la main dans le formalisme de système d'intelligence artificielle sur lequel on travaille. L'analyse morpho-syntaxique est un passage obligé pour l'intégration automatique d'un document en langue naturelle dans un système d'IA".
L'équipe SYDO était composée à ses débuts de Michel Le Guern, Richard Bouché, Jean Paul Metzger, Alain Berrendonner, Sylvie Lainé, et Jacques Rouault.
Dans les applications automatiques en langues naturelles on ne peut pas parler du **"Traitement"** proprement dit, sans passer au préalable par une phase de **"conception"**.
Le traitement automatique des langues naturelles suppose avant tout l'intervention de trois processus essentiels qui se présentent dans l'ordre chronologique suivant :

1. Conception
2. Réalisation
3. Evolution.

à suivre....

Références Bibliographiques

BONNET (Alain), HATON (Jean Paul), TRUONG, NGOC (Jean MICHEL). Systèmes experts : vers la maîtrise technique. Paris, Iner 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'ADBS 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). 1991. Maîtrise de l'Intelligence Artificielle. Marabout Alleur. Belgique.

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

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

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

HARMANN (Hans). 1978. Meinen und Versienen Grundüge 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°1.

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. Sémantic Memory. Bolt, 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 (Olivier). 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 linguistique 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 Metz en juin 1983. in : La recherche française par ordinateur, publiés par : C. CHARPENTEIER et J. DAVID, Slatkine-Champion, Genève-Paris 1985.

TOWNSEND (Carl), 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. Moteurs de systèmes experts. Eyrolles 1987.