

90/456

Université de Nancy I

Centre de Recherche en
Informatique de Nancy
INRIA Lorraine

Sc N 90 /
393 A

Contribution à la commande orale
d'un robot doté d'un système de vision

par

Joëlle KLEIN



Thèse soutenue publiquement le 23 octobre 1990
pour l'obtention du doctorat de l'université de Nancy I
(mention informatique)

Jury

Président : J.P. Haton

Rapporteurs : J. Caelen
D. Coulon

Examineurs : M.C. Haton
J.M. Pierrel

BIBLIOTHEQUE SCIENCES NANCY 1



D

095 145242 6

Université de Nancy I

Centre de Recherche en
Informatique de Nancy
INRIA Lorraine

Contribution à la commande orale
d'un robot doté d'un système de vision

par

Joëlle KLEIN



Thèse soutenue publiquement le 23 octobre 1990
pour l'obtention du doctorat de l'université de Nancy I
(mention informatique)

Jury

Président : J.P. Haton

Rapporteurs : J. Caelen
D. Coulon

Examineurs : M.C. Haton
J.M. Pierrel

Remerciements

Je remercie très vivement Monsieur Jean-Paul Haton qui me fait l'honneur de présider ce jury pour m'avoir accueillie dans son équipe et pour la confiance qu'il m'a témoignée.

Je tiens à exprimer toute ma gratitude à Monsieur Jean-Marie Pierrel qui m'a dirigée dans ma recherche et qui, par ses encouragements m'a permis de faire aboutir ce travail.

Je remercie tout particulièrement Messieurs Jean Caelen et Daniel Coulon qui, avec grande gentillesse, ont accepté de juger ce travail dans des délais très brefs et malgré les charges qui leur incombent.

Un merci chaleureux à Madame Marie-Christine Haton qui a aimablement accepté de participer à ce jury pour sa présence amicale à tous les moments importants de ma vie universitaire.

Je tiens à remercier encore une fois tous les membres du jury pour la gentillesse qu'ils m'ont témoignée, ainsi que tous les membres du laboratoire et de l'IUT qui ont contribué, par leur sympathie, à rendre ces années de thèse très agréables.

Table des matières

Introduction	9
Chapitre 1	
La communication homme-machine.....	11
1.1. Les différents modes de communication.....	12
1.1.1. Les langages de programmation.....	12
1.1.2. Les logiciels.....	12
1.1.3. La communication "naturelle".....	13
1.1.3.1. La communication par mots clés.....	14
1.1.3.2. La communication en langage artificiel.....	14
1.1.3.3. La communication en langage naturel limitée à un domaine particulier.....	15
1.2. La communication orale en langage naturel.....	16
1.2.1. Caractéristiques générales du langage naturel.....	17
1.2.1.1. Avantages et inconvénients de l'utilisation du langage naturel.....	17
1.2.1.2. Notion de sous-langage.....	19
1.2.2. Influence du langage naturel sur la production de parole.....	20
1.3. Les méthodes analytiques du traitement automatique de la parole.....	22
1.3.1. Décodage acoustico-phonétique.....	25
1.3.2. La recherche lexicale.....	27
1.4. Les grandes approches de traitement de la langue naturelle.....	30
1.4.1. Les grammaires formelles.....	30
1.4.1.1. Les grammaires syntaxiques.....	30
1.4.1.2. Les grammaires transformationnelles.....	32
1.4.1.3. Les grammaires sémantiques.....	33
1.4.2. L'approche sémantique.....	35
1.4.2.1. Les réseaux sémantiques.....	35
1.4.2.2. Les grammaires de cas.....	36
1.4.2.3. La définition de traits sémantiques dans le lexique.....	37
1.4.3. Les grammaires systémiques.....	38
Chapitre 2	
Le dialogue.....	39
2.1. Généralités sur le dialogue.....	39
2.1.1. Les caractéristiques du dialogue.....	39
2.1.1.1. La concision de forme.....	39
2.1.1.2. La concision de fond.....	41
2.1.2. Les règles du dialogue.....	42
2.2. Le dialogue dans un système de communication homme-machine.....	43

2.2.1. Les qualités du module de dialogue.....	43
2.2.2. Les différents rôles du module de dialogue.....	45
2.3. L'interprétation.....	46
2.3.1. Les règles de production.....	49
2.3.2. Les frames ou schémas.....	49
2.3.3. Les scripts ou scénarios.....	51
2.3.4. Les plans et les thèmes.....	52
2.3.5. Extensions.....	55
2.4. Les sources de connaissance.....	57
2.4.1. Les sources de connaissances statiques.....	58
2.4.2. Les sources de connaissances dynamiques.....	59
2.5. Quelques réalisations.....	59
2.5.1. Systèmes de communication classiques.....	60
2.5.2. Systèmes multi-média.....	62
2.6. Dialogue avec un robot.....	64

Chapitre 3

Cahier des charges.....67

3.1. Motivations et buts.....	67
3.1.1. Intégration de la reconnaissance d'images.....	68
3.1.2. Exemples d'applications.....	69
3.2. Limites du sous-langage.....	71
3.3. Compétences du système.....	74
3.4. Indépendance entre connaissance et raisonnement.....	77
3.5. Les principales sources de connaissance statiques.....	78

Chapitre 4

Présentation générale du système.....81

4.1. Modèle de la tâche.....	81
4.1.1. Le modèle de l'univers.....	81
4.1.2. Représentation de l'univers.....	83
4.1.2.1. Occurrences des objets.....	83
4.1.2.2. Positions des objets.....	85
4.1.3. Informations pragmatiques.....	86
4.2. Modèle de la langue.....	89
4.2.1. Choix de la grammaire.....	89
4.2.2. Arbre des cas.....	91
4.2.3. Les grammaires locales.....	94
4.2.4. Construction de groupes complexes.....	97
4.2.4.1. Coordination de groupes.....	98
4.2.4.2. Qualification de groupes.....	99
4.3. Les analyseurs du modèle de la langue.....	100
4.3.1. Analyseur acoustico-phonétique.....	100

Table des matières

4.3.2. Analyseur lexical.....	102
4.3.3. Analyseur syntaxico-sémantique.....	103
4.3.4. Module d'interprétation.....	103
4.4. Modèle du dialogue.....	103
4.4.1. Gestion du dialogue.....	104
4.4.2. Stratégies du dialogue.....	105
4.4.3. Organisation du système.....	105
4.4.3.1. Le lexique.....	105
4.4.3.2. La représentation de l'univers.....	108
4.4.3.3. Les grammaires.....	108
4.4.3.4. Architecture du système.....	108

Chapitre 5

Reconnaissance de phrases.....111

5.1. Du phonème au mot.....	111
5.1.1. Comparaison de deux formes.....	113
5.1.2. Calcul du score de ressemblance.....	115
5.1.3. Sélection dynamique des mots.....	121
5.2. Du mot au groupe.....	121
5.2.1. Construction d'un groupe.....	121
5.2.2. Calcul du score du groupe.....	123
5.2.3. Représentation interne du groupe.....	124
5.3. Du groupe à la phrase.....	125
5.3.1. Du groupe au squelette de la phrase.....	125
5.3.2. Du squelette à la phrase complète.....	126
5.3.2.1. Les différentes étapes de coordination et qualification.....	126
5.3.2.2. Exemples de construction de groupes complexes.....	128
5.3.2.3. Calcul du score du groupe complet.....	129
5.4. Sélection de phrases.....	129
5.4.1. Sélection de phrases dont le score est supérieur à un seuil.....	130
5.4.2. Sélection des N meilleures phrases.....	131
5.4.3. Sélection de phrases adaptée aux résultats.....	131
5.4.4. Conclusion.....	132

Chapitre 6

Interprétation et gestion du dialogue.....133

6.1. Construction d'énoncés complets.....	133
6.1.1. Résolution des ellipses.....	133
6.1.1.1. Cas particulier de l'ellipse du prédicat.....	134
6.1.1.2. Valeur par défaut et recherche dans l'historique.....	135
6.1.1.3. Résolution pragmatique.....	138
6.1.2. Résolution des références.....	139
6.1.2.1. Référence à un concept.....	139
6.1.2.2. Référence à une occurrence.....	144
6.2. Interprétation des commandes.....	147
6.2.1. Confrontation au modèle de la tâche.....	147
6.2.2. Résolution des groupes complexes.....	147
6.2.3. Résolution des ambiguïtés.....	150
6.2.3.1. Recherche des compléments.....	150

Table des matières

6.2.3.2. Recherche dans l'histoire	151
6.2.3.3. Confrontation entre le modèle de la tâche et l'univers	151
6.2.3.4. Mise en évidence de la pensée du locuteur	152
6.2.4. Résolution des cas d'échecs	155
6.2.5. Gestion des contestations	156

Chapitre 7

Résultats et implémentation	159
7.1. Présentation générale	159
7.1.1. Génération de phonèmes	159
7.1.2. Résultats détaillés de la reconnaissance	160
7.1.2.1. Structure du mot	161
7.1.2.2. Structure du groupe	161
7.1.2.3. Structure de la phrase	162
7.2. Résultats de la reconnaissance	163
7.2.1. Exemples de phrases simples et correctes	164
7.2.2. Validation du module de reconnaissance de mots	164
7.2.3. Résistance aux erreurs syntaxiques	165
7.2.4. Reconnaissance d'expressions complexes	166
7.2.5. "Développement" d'expressions	168
7.3. Résultats de l'interprétation	170
7.3.1. Groupes simples	170
7.3.2. Groupes complexes	171
7.3.3. Groupes ambigus	171
7.3.4. Groupes elliptiques	172
7.4. Implémentations et perspectives	173
Conclusion	175
Bibliographie	177
Annexe	189

Index des figures

Figure 1.1.	: Modèle de communication entre deux interlocuteurs	p. 11
Figure 1.2.	: Une représentation possible du dialogue entre un être humain et une machine	p. 17
Figure 1.3.	: Une classification des phonèmes du français	p. 24
Figure 1.4.	: Spectrogramme de la commande : "Prends le livre sur le bureau"	p. 26
Figure 1.5.	: Matrice de confusion	p. 28
Figure 1.6.	: Exemple de grammaire formelle	p. 30
Figure 1.7.	: Arbre de dérivation de la phrase : "le robot prend le livre rouge"	p. 31
Figure 1.8.	: Un modèle de la théorie standard d'après [Sabah-88]	p. 32
Figure 1.9.	: Grammaire sémantique simplifiée	p. 33
Figure 1.10.	: Arbre de dérivation	p. 34
Figure 1.11.	: Exemple de réseau sémantique	p. 35
Figure 2.1.	: Exemple de dialogue	p. 43
Figure 2.2.	: Dialogue d'après [Vilnat-85]	p. 44
Figure 2.3.	: Les deux types d'interprétation d'après [Sabah-88]	p. 47
Figure 2.4.	: Schéma de "pâte" d'après [Colnet-86]	p. 50
Figure 2.5.	: Plan pour se nourrir	p. 53
Figure 2.6.	: Les primitives servant à définir les sous-buts généraux	p. 54
Figure 2.7.	: Réalisation possible de savoir(x)	p. 54
Figure 2.8.	: Exemple du MOP "emprunt" d'après [Lehnert-83]	p. 56
Figure 3.1.	: Exemple de dialogue avec confirmation systématique	p. 73
Figure 3.2.	: Exemple de dialogue sans confirmation	p. 74
Figure 3.3.	: Exemple de question posée par le système	p. 75
Figure 3.4.	: Exemple de description d'un objet par le système	p. 76
Figure 3.5.	: Exemple de demande de précision	p. 76
Figure 3.6.	: Exemple d'objet inconsistant	p. 76
Figure 3.7.	: Exemple d'inconsistance de la commande	p. 76
Figure 3.8.	: Exemple d'inconsistance de l'état de l'objet	p. 77
Figure 4.1.	: Prototype de l'objet "table"	p. 82
Figure 4.2.	: Prototype de l'objet "porte"	p. 82
Figure 4.3.	: Prototype de l'objet "lampe"	p. 82
Figure 4.4.	: Création d'une occurrence	p. 84
Figure 4.5.	: Exemple de définition invalide	p. 84
Figure 4.6.	: Extrait d'un fichier représentant l'univers	p. 85
Figure 4.7.	: Exemple de définition des positions	p. 86
Figure 4.8.	: Classes de validité des prépositions	p. 86
Figure 4.9.	: Connaissances pragmatiques associées au prédicat "prendre"	p. 88
Figure 4.10.	: Arbre des cas des prédicats	p. 92
Figure 4.11.	: Arbre des cas et des contraintes syntaxiques des verbes	p. 93
Figure 4.12.	: Les grammaires locales	p. 96
Figure 4.13.	: Les grammaires locales associées aux cas	p. 97

Figure 4.14.	: Exemples de règles de coordination	p. 99
Figure 4.15.	: Exemples de règles de transcription	p. 102
Figure 4.16.	: Exemples de mots des sous-lexiques	p. 107
Figure 4.17.	: Architecture générale du système	p. 110
Figure 5.1.	: Comparaison de deux formes par programmation dynamique	p. 115
Figure 5.2.	: Matrice de ressemblance	p. 117
Figure 5.3.	: Construction d'un groupe nominal	p. 122
Figure 5.4.	: Représentation interne d'un groupe nominal	p. 124
Figure 5.5.	: Représentation interne d'un groupe prépositionnel	p. 125
Figure 6.1.	: Description de deux méthodes de résolution d'ellipse du cas destination	p. 138
Figure 6.2.	: Exemples de références conceptuelles du cas agent	p. 139
Figure 6.3.	: Exemple de règle de résolution d'ellipse	p. 140
Figure 6.4.	: Exemples de résolution de groupes	p. 147
Figure 6.5.	: Résolution d'un groupe simple	p. 148
Figure 6.6.	: Résolution d'un groupe complexe	p. 148
Figure 6.7.	: Application de la règle de développement	p. 149
Figure 6.8.	: Résolution d'ambiguïté par confrontation avec le modèle de la tâche et l'univers	p. 152
Figure 6.9.	: Arbre d'instanciation	p. 153
Figure 6.10.	: Contraintes de discrimination	p. 154
Figure 6.11.	: Exemples de cas d'instanciation	p. 155
Figure A.1.	: Codification des phonèmes	p. 190
Figure A.2.	: Grammaires locales	p. 191
Figure A.3.	: Règles de coordination et transformations associées	p. 193

Introduction

De tout temps l'homme a cherché à construire des machines destinées à le servir. Il lui a fallu plusieurs millénaires pour passer des pièges primitifs aux robots utilisés de nos jours. Mais ces derniers, même s'ils ont réussi à conquérir la plupart des secteurs industriels, de la construction automobile à la médecine en passant par l'électronique, restent encore, comme leurs ancêtres, de simples machines conçues pour réaliser, presque indéfiniment, un acte, un geste ou une mesure prévus par l'homme. L'introduction de l'intelligence artificielle dans les programmes de commande de robots pourrait apporter un confort d'utilisation, par exemple permettre de varier les tâches en fonction du contexte (plans d'actions), prise de décision du robot en cas d'incidents sur une chaîne de montage, commande orale ...

Dans cette thèse, nous nous intéressons à la commande de robot. Nous supposons que le robot possède un dispositif de vision évolué, qu'il est capable de se mouvoir dans son univers et qu'il sait exécuter des commandes formelles. Notre but est de pouvoir commander ce robot de manière orale et dans un langage qui soit le plus naturel possible. Pour cela notre travail a consisté à définir une grammaire permettant de reconnaître et d'interpréter des expressions complexes de désignation d'objets, en tenant compte des problèmes liés à l'utilisation d'un langage aussi naturel que possible, c'est-à-dire les problèmes d'ellipses d'anaphores ... Le système que nous avons développé accepte en donnée une suite de phonèmes représentant la commande, cette commande est analysée de manière à construire, grâce à la grammaire utilisée, une représentation syntaxico-sémantique de l'énoncé. L'interprétation se fait dans un univers simulé représentant la connaissance que possède le robot de son monde, celle-ci étant théoriquement obtenue par le système de vision dont il dispose.

Dans le premier chapitre, nous présentons les diverses méthodes de communication qui existent entre l'homme et la machine ainsi que les méthodes de traitement de la parole et les différentes approches du langage naturel.

Dans le deuxième chapitre, nous nous intéressons plus particulièrement à l'aspect dialogue de la communication.

Le chapitre 3 contient, quant à lui, les objectifs du cahier des charges que nous avons essayé d'atteindre.

Les trois chapitres suivants présentent le système développé. Dans un premier temps, nous faisons une présentation générale du système, puis nous présentons plus particulièrement, dans le chapitre 5, les différentes étapes de la reconnaissance des commandes ; dans le chapitre 6, nous présentons l'interprétation et la gestion du dialogue ainsi qu'une réflexion sur les stratégies visant à résoudre les ambiguïtés d'interprétation.

Dans le chapitre 7, nous présentons les résultats obtenus et les perspectives de réalisation.

Chapitre 1 La communication homme-machine

La communication est un échange de messages codés entre deux intervenants, la conséquence directe en est que les deux intervenants doivent connaître le même code ou tout au moins, l'émetteur du message doit être capable de coder le message et le récepteur de le décoder. Nous pouvons modéliser la communication entre deux interlocuteurs par la figure 1.1.

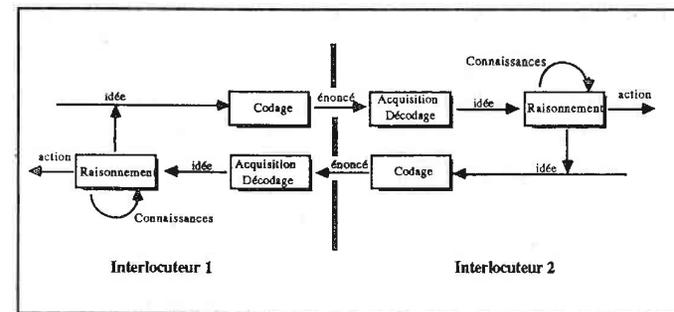


Figure 1.1. Modèle de communication entre deux interlocuteurs humains

La communication homme-machine, implique que l'un des deux intervenants est un ordinateur. Dans les systèmes de communication élémentaires, l'homme est l'émetteur du message et l'ordinateur, le récepteur et interprète de celui-ci. Cependant, les travaux actuels tendent à mettre les deux intervenants sur le même plan, c'est-à-dire que l'homme et la machine jouent chacun à leur tour les rôles de concepteur-émetteur et de récepteur-interprète.

Dans les paragraphes suivants, nous allons présenter les différents modes de communication existants, leur intérêt, leurs limites et le public visé, puis nous parlerons plus particulièrement de la communication en langage naturel, ensuite, des problèmes posés au niveau du traitement automatique de la parole (TAP)

lorsque cette communication se fait de manière orale. Enfin, nous présenterons les différents modèles proposés pour représenter la langue naturelle.

1.1. Les différents modes de communication

Les systèmes de communication possibles entre l'homme et l'ordinateur sont très variés, dans les paragraphes qui suivent, nous allons les présenter de manière hiérarchique, des plus spécifiques aux plus généraux.

1.1.1. Les langages de programmation

La communication entre l'ordinateur et l'homme fit son apparition avec les premiers programmes interactifs écrits à l'aide de langages de programmation allant du langage de bas niveau (ex : assembleur) aux langages les plus évolués (ex : Les langages orientés objet LOO [Masini-89]). Ce type de communication avec un ordinateur nécessite un apprentissage tant du langage lui-même (lexique, syntaxe et sémantique) que du mode de fonctionnement de l'ordinateur (structure des données...). Il est donc réservé à un public de spécialistes. Le but essentiel des langages de programmation est le développement d'applications informatiques, ils s'adressent donc aux concepteurs de logiciels et non aux utilisateurs.

1.1.2. Les logiciels

A un niveau supérieur, nous pouvons parler des logiciels écrits par des informaticiens et possédant une interface les rendant utilisables par des non spécialistes en informatique, mais nécessitant tout de même la manipulation d'outils informatiques (clavier, écran, disquettes...).

Les travaux effectués par les équipes de recherche en génie logiciel tendent à améliorer la convivialité de ces logiciels, en en diminuant le temps d'apprentissage par l'intégration d'outils moins spécifiques que les terminaux classiques (ex : souris, crayons optiques, touches fonctions, menus ...). Ce type de communication s'adresse à un public très large à condition que l'application traitée reste simple (ex : guichets bancaires automatiques, minitel, balances pour la pesée des fruits et légumes dans les super-marchés ...).

Cependant, pour des applications plus importantes, un apprentissage non négligeable est nécessaire pour rendre le logiciel suffisamment performant et pour pouvoir l'utiliser au maximum de ses possibilités, il concerne donc un public plus averti disposant de moyens et de temps d'apprentissage.

1.1.3. La communication "naturelle"

Dans la dernière catégorie, nous considérons les modes de communication dits de haut-niveau, résultats des travaux de recherche menés en intelligence artificielle (IA). Ils permettent à l'utilisateur de communiquer, de manière naturelle sans scénario prédéfini, par l'intermédiaire d'images ou de signes, ou par l'intermédiaire du langage naturel (écrit ou parlé).

Ce type de communication fait encore actuellement l'objet de nombreuses recherches au niveau :

- de la reconnaissance des formes (reconnaissance de parole, d'images...),
- de l'interprétation de ces formes, en tenant compte de l'imperfection de la phase de reconnaissance (dialogue de confirmation, contestation ...),
- du traitement de l'information (plans d'actions pour un robot, génération de réponses pour une application de type renseignements administratifs ...).

L'objectif qui paraît le plus intéressant, parce qu'il représente le type de communication le plus naturel, est de pouvoir communiquer en langage naturel. La difficulté du problème a entraîné plusieurs approches successives et plus ou moins simplificatrices :

- la communication par mots clés,
 - la communication en langage artificiel,
 - la communication en langage naturel limitée à un domaine particulier,
- ainsi que l'étude de l'adaptation à des domaines ouverts (savoir encyclopédique ...) qui si elles ne débouchent pas sur des applications pratiques permettent de comprendre certains mécanismes de la communication.

1.1.3.1. La communication par mots clés

L'analyse de phrases par mots-clés consiste à rechercher sur le message, des mots connus du système et considérés comme porteurs de sens. Ceci permet à l'utilisateur de s'exprimer de manière naturelle à condition d'utiliser les mots-clés dans ses phrases. Cependant, il faut connaître le mode d'interprétation du système, en effet, le système ne reconnaissant que les mots-clés, toute modification de sens apportée par des mots grammaticaux (ex : les adverbes tels que "beaucoup", "peu" ...) entraînerait une mauvaise interprétation. La liberté apparente du mode d'expression est due à l'utilisation possible d'un grand vocabulaire dans la mesure où "l'enrobage" de la phrase n'est pas analysé, cependant, toutes les nuances qui peuvent apparaître dans un message en langage naturel ne peuvent en aucun cas être prises en compte.

Le système ELIZA [WEI-66] est un exemple de communication écrite par mots-clés. L'analyseur détecte des mots dans l'énoncé de l'utilisateur et, à partir de phrases standard plus ou moins générales, il construit des questions pour poursuivre un dialogue que l'on pourrait qualifier de "dialogue de sourd" dans la mesure où aucune interprétation de la phrase n'est faite. Le système construit à partir des mots reconnus une suite de relances du dialogue. Les résultats ont pu paraître spectaculaires par l'apparence naturelle du dialogue. Cependant on ne peut en aucun cas considérer que le système assure une quelconque compréhension de la phrase. La communication par mots-clés s'avère très efficace pour des applications très fermées mais ne peut pas être utilisée dans un cadre général.

Ce type de communication est donc réservé a priori à des applications simples utilisant peu de mots, nécessitant l'apprentissage du vocabulaire de base et la connaissance de notions sur le fonctionnement de l'analyseur afin d'éviter des contre-sens malencontreux.

1.1.3.2. La communication en langage artificiel

La reconnaissance ne se fait plus seulement à partir des mots, mais aussi en fonction de la syntaxe de l'énoncé. Ceci permet une interprétation du message mais impose à l'utilisateur de se limiter à des règles syntaxiques rigides et à un vocabulaire restreint.

La syntaxe étant rigide, la reconnaissance se fait de manière assez simple à l'aide d'automates, et peut être optimisée par prédiction des éléments à venir en fonction des éléments déjà reconnus.

Ce type de communication est adapté à la commande de robots, ou à l'interrogation de bases de données par des spécialistes, il ne nécessite aucun traitement sémantique dans la mesure où il existe une correspondance directe entre la structure syntaxique et la commande à exécuter.

Exemples de systèmes de communication orale utilisant un langage artificiel :

- le système MYRTILLE I [Pierrel-75]
développé par Jean-Marie Pierrel au CRIN
- le système KEAL [Mercier-77], [Siroux-85] développé au CNET¹
- le système PARTNER [Morin-87]
qui utilise un langage artificiel à consonance naturelle
- le système DIAPASON [Alinat-87] inspiré de MYRTILLE I.

1.1.3.3. La communication en langage naturel limitée à un domaine particulier

Ce type de communication, qui utilise un langage familier (langage naturel), est ouvert au grand public. Il accepte les énoncés en langage naturel avec un minimum de contraintes, ce qui permet une phase d'apprentissage très réduite (ex : le système DIAL de renseignements administratifs développé au CRIN [Roussanaly-88]).

L'intérêt de ce type de communication évolué réside dans le fait qu'il met l'ordinateur à la portée d'un public non averti, et le transforme ainsi en un "interlocuteur associé" et n'est donc plus considéré comme un simple exécutant.

Les diverses applications se regroupent en deux grandes classes :

- les applications où le dialogue est utilisé en tant qu'assistance au système :
 - la traduction automatique,

¹ Centre National d'Etudes des Télécommunications

- le résumé automatique,
- la dictée automatique [Charpillet-85].

- les applications où le dialogue est le "cœur" du système :

- la réalisation d'interfaces en langue naturelle (interrogation de bases de données, utilisation de systèmes experts, aide aux handicapés [Haton-85])
- l'indexation automatique de documents et la recherche documentaire,
- la commande de robots.

Ce type de communication ouvre la porte à un véritable dialogue entre l'homme et la machine, où chacun à son tour peut prendre l'initiative, ce qui replace l'interlocuteur humain dans un contexte naturel et familier. Pour obtenir une plus grande convivialité, l'intégration de la reconnaissance de la parole s'avère être d'un grand intérêt.

1.2. La communication orale en langage naturel

Si le processus d'interprétation d'un message par un être humain semble faire partie d'un tout¹, la complexité de celui-ci et le grand nombre de connaissances requises pour le mener à bien a entraîné les chercheurs en reconnaissance de parole à le décomposer en plusieurs modules distincts :

- le décodage acoustico-phonétique qui permet de passer du signal à une suite de sons élémentaires (phones, phonèmes, diphones ...),
- l'analyse syntaxico-sémantique qui permet à partir d'une suite de sons élémentaires de construire des mots puis des phrases,
- l'interprétation qui donne un sens à la phrase,
- le raisonnement et la génération de réponse.

¹Les études des psycho-linguistes dans ce domaine sont très nombreuses et les conclusions très variées. On pourra consulter une synthèse de ces travaux dans [Segui-88].

La figure 1.2 nous donne une représentation possible d'un système de dialogue entre un homme et une machine. Certains systèmes fonctionnent de la manière ascendante présentée dans ce schéma, mais il est évident que ce n'est pas le cas général, la tendance actuelle veut que les différents modules communiquent entre eux de manière à ce que chacun d'entre eux puisse remettre en cause les conclusions des autres, ce qui se rapproche du comportement humain.

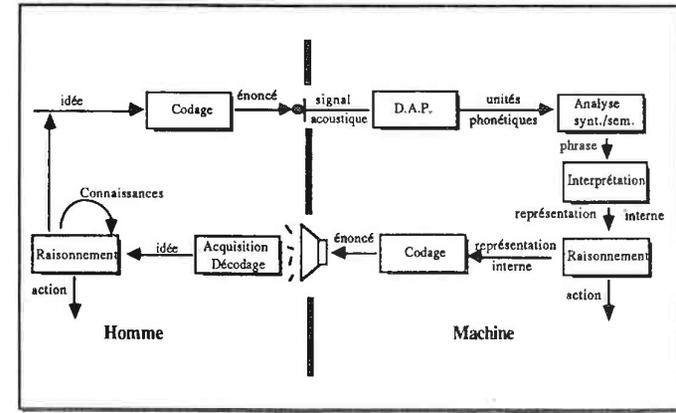


Figure 1.2. Une représentation possible du dialogue entre un être humain et une machine

Dans les paragraphes suivants, nous allons voir quelles sont les caractéristiques générales de la communication orale en langage naturel, puis l'influence du langage naturel sur la production de parole.

1.2.1. Caractéristiques générales du langage naturel

1.2.1.1. Avantages et inconvénients de l'utilisation du langage naturel

Les avantages de l'utilisation du langage naturel sont dus à ses principales caractéristiques :

- c'est un langage familier, il ne nécessite pas d'apprentissage,
- c'est un langage souple qui permet d'énoncer avec concision l'information et d'exprimer la connaissance à divers niveaux d'abstraction, en utilisant :

- des ellipses : - *Prends le livre*
- ... *et le cahier.*
- des anaphores : - *Prends le livre*
- *Pose le sur la table*
- et des connaissances implicites : - *Ouvre la porte*

les connaissances implicites liées à cet énoncé sont par exemple :

- saisir la poignée
- tourner la poignée
- pousser (tirer) la porte.

Il permet de désigner des objets par des expressions plutôt que par des étiquettes :

le petit cube rouge près de la pyramide plutôt que *cube27*

- c'est un langage robuste, en effet la compréhension ne se dégrade pas trop brutalement même si le message se dégrade beaucoup (non respect des règles syntaxiques usuelles notamment).

Ex : on comprend aussi bien les deux énoncés suivants :

- *Pose le cube sur la table*
- *Le cube pose le sur la table*

Le principal inconvénient du langage naturel est sa grande complexité, tant au niveau syntaxique (nombreuses formes grammaticales admettant un grand nombre d'exceptions, variations morphologiques des mots ...) qu'au niveau sémantique (polysémie, références, métaphores¹ ...).

Il est inconcevable à l'heure actuelle de modéliser le langage naturel dans son ensemble, car il faudrait être capable de représenter toutes les règles grammaticales

¹Métaphore: procédé par lequel on transporte la signification propre d'un mot à une autre signification qui ne lui convient qu'en vertu d'une comparaison sous-entendue. Ex: la lumière de l'esprit

et leurs exceptions, tous les contextes possibles et les interprétations des mots polysémiques associées.

Ex : *le chat dévore la souris*
Jean dévore un livre

Pour concevoir un système de compréhension du langage naturel, il faut donc limiter au maximum les informations à traiter (tout en gardant les caractéristiques du langage naturel) de manière à pouvoir les modéliser. Nous en arrivons donc à la définition de la notion de sous-langage adapté à l'application visée, ce qui fait l'objet du paragraphe suivant.

Un autre problème lié à l'utilisation du langage naturel est la résolution d'ellipses, d'anaphores dont les références pronominales qui ne sont pas toujours simples.

Ex : *Pierre casse un carreau / je le vois le casser*

Nous parlerons de ces problèmes dans le chapitre suivant.

1.2.1.2. Notion de sous-langage¹

Des études expérimentales sur des corpus réels [GRECO-85] ont montré que l'utilisation que l'on fait du langage naturel est très dépendante de :

- l'univers de l'application, ses principales influences étant la réduction du lexique et la réduction des cas de polysémie

Ex : *Il faut débusquer la taupe*

ne pose aucun problème d'interprétation si l'on connaît le contexte de l'énonciation (jardinage ou espionnage) alors que cette phrase reste ambiguë hors de tout contexte.

¹ Dans la littérature, on pourra trouver d'autres appellations telles que: micro-langage, langage spécialisé, langage technique ou langage opératif ...[Véronis-88], [Falzon-84], [Deville-86].

Parmi les influences du domaine, on peut noter aussi les règles de grammaires déviantes, la fréquence élevée de certaines constructions et l'utilisation de symboles (ou mots) spéciaux [Deville-87bis].

- la situation de communication qui, entre deux interlocuteurs humains, peut revêtir de nombreuses formes (commande, narration, débat philosophique ...), se trouve limitée à des schémas de type commande ou requête lorsque l'un des deux interlocuteurs est une machine et que la communication est orientée vers la tâche à réaliser.

Kittredge [Kittredge-79] a déduit, à partir de l'étude de 11 sous-langages, des caractéristiques générales que l'on peut résumer ainsi :

- les sous-langages peuvent être décrits par moins de règles grammaticales que le langage naturel,
- ces règles ne forment pas un sous-ensemble de celles du langage naturel, il existe des règles spécifiques aux sous-langages,
- chaque sous-langage possède des règles spécifiques permettant de le distinguer des autres sous-langages,
- il existe des similarités entre les versions d'un sous-langage exprimé dans plusieurs langues différentes.

Pour une tâche bien spécifique, il est donc possible de définir un sous-langage du langage naturel offrant toutes les possibilités voulues et cependant suffisamment restreint pour être modélisable. Les langages artificiels à consonance naturelle et les sous-langages sont les seuls moyens de communication permettant d'aborder un dialogue naturel avec l'ordinateur, c'est pour cette raison que les recherches actuelles se dirigent dans cette direction¹.

1.2.2. Influence du langage naturel sur la production de parole

La caractéristique orale de ce type de communication entraîne une grande variabilité sur la forme des mots. En effet, le langage naturel parlé respecte les accords grammaticaux entre les différents éléments de la phrase, contrairement au

¹Dans [Véronis-88] on trouve les notions de sous-langages naturel et artificiel, le premier correspondant à un sous-langage pré-existant alors que le second correspond à un sous-langage possédant certaines caractéristiques (complétude, convivialité...) construit en vue du dialogue homme-machine.

langage artificiel où l'ensemble des structures doivent être modélisables par une grammaire [Calliope-89] ou la communication par mots isolés où les phénomènes articulatoires disparaissent.

Ex : en langage naturel on dira : *ouvre l'enveloppe*
en mots isolés on dira : *ouvre - la - enveloppe*

Il faut donc tenir compte des altérations dues aux flexions de déclinaison et de conjugaison. Quand le vocabulaire traité devient important, il est inconcevable d'y stocker toutes les formes fléchies de tous les mots. Il faudra introduire des règles phonologiques intégrant les radicaux des mots et l'ensemble des désinences associées pour générer les diverses formes fléchies (en particulier, pour la conjugaison des verbes réguliers et pour les pluriels des noms communs). De plus, il faudra tenir compte des altérations dues à la coarticulation et aux accents régionaux. Les problèmes de coarticulation, s'ils existent à l'intérieur d'un mot, sont plus flagrants entre deux mots :

Ex : coarticulation --> *Le mois d'juin* ou *la p'ti'table*
accent régional --> *La bedite* pour *La petite*

La continuité de la parole naturelle implique des liaisons entre les mots. Le traitement de ces liaisons est un problème qui n'est pas simple, en effet, en fonction du sens de la phrase, une liaison entre deux mots sera présente ou non :

Ex1 : *Un savant anglais*

S'il y a une liaison entre savant et anglais, c'est qu'on parle d'une personne de nationalité anglaise très érudite, sinon on parle d'un savant de nationalité anglaise.

Ex2 : *Fais les entrer* et *fais les entrées*

Dans le premier cas, il y a une liaison moins marquée que dans le deuxième cas.

Dans un système évolué de compréhension de phrases, les informations quant à la présence d'une liaison pourront être un élément important pour soulever une ambiguïté.

Pour plus d'informations concernant la présence de liaisons dans un énoncé, on pourra consulter [Dell-85].

Un autre phénomène dû à la continuité du discours est l'absence de segmentation nette entre les mots et les groupes de la phrase. Les mots sont

enchaînés les uns aux autres (cf les liaisons), ce qui augmente la difficulté du traitement de reconnaissance des mots et même d'interprétation de la phrase.

Ex : *La porte et l'apporte*

seront représentés phonétiquement de la même manière

La prise en compte d'éléments prosodiques permet de définir des césures dans une phrase, et donc de trouver la séparation entre deux mots et ainsi dans certains cas de lever des ambiguïtés.

Ex : *La belle / ferme / le voile*
La belle ferme / le / voile
/ représente une césure dans l'énoncé.

Si le module de reconnaissance de phrases ne possède pas de traitement prosodique, la seule solution possible est d'envisager tous les cas et d'attendre la confrontation de la phrase avec le modèle du monde pour lever les ambiguïtés. L'introduction d'informations prosodiques ne permettrait cependant pas d'exclure l'une des solutions, mais elle pourrait tout au plus définir une priorité d'interprétation.

On peut aussi noter l'apparition de phénomènes tels que l'hésitation (ex : euh!) et la reprise (ex : la petite, euh non, la grande porte).

1.3. Les méthodes analytiques du traitement automatique de la parole

La reconnaissance de parole s'appuie sur deux techniques différentes :

- la méthode globale,
- la méthode analytique.

La méthode globale consiste à considérer un mot, une expression ou même une phrase entière comme une seule entité et à la comparer directement avec le signal. Les techniques de programmation dynamique [Bellman-57] [Sakoe-78] qui permettent de calculer la "distance" minimale entre deux formes vont déterminer quel mot (expression ou phrase) a été prononcé.

Cependant, si cette méthode s'avère efficace sur un vocabulaire restreint, elle est inexploitable en reconnaissance de parole continue sur un vocabulaire important, car elle suppose la mise en mémoire de toutes les formes énonçables possibles, donc des formes tenant compte des transformations phonologiques dues au langage continu. On se rend donc bien compte que la mise en mémoire de tant d'informations deviendrait vite prohibitive et que les temps de recherche ainsi que la durée d'apprentissage se trouveraient grandement accrus.

La méthode analytique consiste à considérer le message comme une suite d'unités phonétiques. Par conséquent, reconnaître une phrase revient à déterminer la suite d'unités phonétiques qui la composent.

Les deux principaux problèmes issus de cette technique sont d'établir les liens qui existent entre le signal et les unités phonétiques de base, d'une part et d'autre part les liens qui unissent les suites d'unités phonétiques aux mots.

Les unités phonétiques choisies doivent réaliser un bon compromis entre la facilité de segmentation et celle de reconnaissance. Tous les choix intermédiaires entre un découpage minimal en syllabes et un découpage plus fin en phonèmes sont possibles. Cependant, si la segmentation en syllabes, par l'étude des courbes d'énergie est un problème simple, l'identification en est un très complexe car le nombre de syllabes est très élevé (de l'ordre de plusieurs milliers pour la langue française) et leur variation acoustique très large. Le phonème est en général considéré comme le compromis le plus équilibré, cependant le choix de l'unité phonétique de base ne doit pas être totalement restrictif, en effet, un découpage préliminaire en syllabes avant le passage aux phonèmes, ou une recombinaison des phonèmes en syllabes (ou encore la combinaison des deux), peut permettre de vérifier un maximum d'informations et ainsi d'améliorer les taux de reconnaissance.

La figure 1.3 nous donne une classification des phonèmes du français utilisés en reconnaissance de parole. Cette liste est incomplète, en effet par exemple, il n'y a qu'un seul phonème /a/ pour représenter le son " a " des mots suivants : pattes, pâte, pas Cependant les limites des décodeurs acoustico-phonétiques actuels ne permettant pas de faire correctement une distinction aussi fine, cet ensemble sera considéré comme suffisamment représentatif, d'autant plus que ces

distinctions tendent à disparaître dans la langue orale d'aujourd'hui.

VOYELLES			CONSONNES		
Phon	exemples	classe	Phon	exemples	classe
a	patte, bas,	orales	p	pas	plosives sourdes
i	pâte		t	tas	
y	il		k	cas	
ɔ	nu		b	bon	plosives sonores
o	bol		d	dans	
ə	eau		g	gars	
œ	le		v	vie	fricatives sonores
e	peu, peur		z	zéro	
é	blé		ʒ	je	
u	merci		f	feu	fricatives sourdes
ɑ̃	ou	s	sous		
õ	an	ʃ	chat		
ẽ	on	nasales	n	nous	nasales
œ̃	lin		m	ma	
	un		ɲ	agneau	
SEMI-VOYELLES			l	la	liquides
			r	rue	
ʷ	huit				
w	oui				
j	yeux, baille				

Figure 1.3. Une classification des phonèmes du français

Remarque : on trouvera en annexe la liste des codifications utilisées pour représenter les phonèmes.

Nous avons vu que les deux principaux problèmes liés à l'utilisation d'une méthode analytique consistaient en

- le passage du signal aux phonèmes, ce qu'on nomme communément le décodage acoustico-phonétique (DAP),
- le passage de la suite de phonèmes à la suite de mots, ou recherche lexicale.

Ces deux étapes font l'objet des paragraphes suivants.

1.3.1. Décodage acoustico-phonétique

Le décodage acoustico-phonétique consiste à passer du signal à une suite de phonèmes. Il permet donc de discrétiser le signal dont les frontières sont imprécises (à cause des phénomènes de transition et de coarticulation), le résultat obtenu, bien que plus facilement utilisable par les niveaux supérieurs n'en reste pas moins incertain.

Les nombreux problèmes rencontrés en DAP sont dus aux caractéristiques de la parole naturelle :

- la grande quantité d'informations à traiter qui nécessite la mise en œuvre de techniques telles que FFT¹ ou LPC² pour ne conserver que les informations pertinentes,
- les variations intra-locuteur qui peuvent entraîner des formes phonétiques différentes pour une même phrase prononcée plusieurs fois par un locuteur ; de la même façon, un phonème prononcé dans plusieurs contextes aura des réalisations différentes,
- les variations inter-locuteur qui se manifestent, d'une part, par les caractéristiques propres de chaque voix (hauteur, timbre...) dues en grande partie à des phénomènes morphologiques et d'autre part, par les accents et les intonations variant d'une région à une autre.

¹ Transformée de Fourier

² Coefficients de prédiction linéaire

Les premières approches du DAP consistaient à comparer le signal de parole à des formes de référence. Cette technique nécessite une phase d'apprentissage qui, si elle est possible en reconnaissance de phrases mono-locuteur, devient difficilement applicable en mode multi-locuteur car le nombre de références nécessaires entraînerait des temps d'apprentissage très longs, et les temps de reconnaissance seraient fortement augmentés du fait du grand nombre de comparaisons à effectuer.

Les techniques actuelles utilisent une approche intelligente mettant en œuvre des raisonnements complexes basés sur des connaissances diversifiées. Dans le système APHODEX [Carbonell-86] développé au CRIN, une approche de type système expert a été utilisée. L'expertise est fournie par F. Lonchamp, phonéticien capable de décoder un spectrogramme, représentation du signal en fonction de trois paramètres : le temps, la fréquence et l'intensité (figure 1.4).

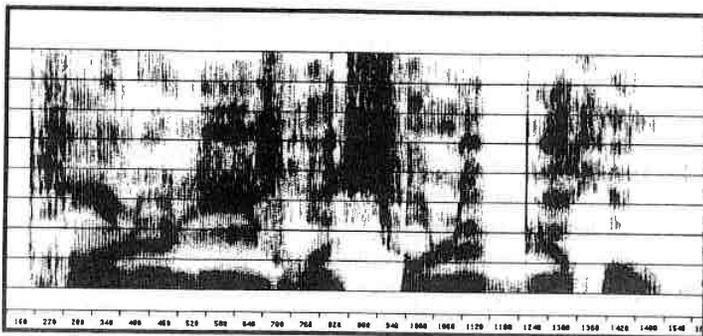


figure 1.4. Spectrogramme de la commande :
"Prends le livre sur le bureau"

Bien que les résultats obtenus par APHODEX soient intéressants [Haton-90], l'état actuel du système ne permet pas son utilisation dans un système de dialogue. C'est pourquoi, dans un premier temps, nous avons utilisé dans notre système de dialogue un module de simulation de la tâche du DAP (cf Chapitre 4).

1.3.2 La recherche lexicale

Elle consiste à rechercher des mots dans l'énoncé du locuteur, on peut distinguer deux grandes classes de problèmes :

- la vérification de mots (Word Verification) qui consiste à vérifier la présence d'un mot hypothèse à une position précise dans l'énoncé, Les techniques alors mises en œuvre sont de trois types [Pierrel-87] :

- Les techniques heuristiques consistent à mettre en correspondance la forme de référence et la plage d'énoncé sélectionnée en tenant compte d'éventuelles insertions, élisions ou substitutions de phonèmes, soit en cherchant le meilleur sous-chemin commun entre les deux formes, soit en calculant par des algorithmes de programmation dynamique [Haton-74] une distance entre les deux formes ou enfin, en combinant ces deux méthodes [Pierrel-79]. C'est ici que doivent intervenir les données concernant les caractéristiques du décodeur acoustico-phonétique. Par exemple le coût de substitution d'un phonème par un autre dépend des caractéristiques propres de ces phonèmes (voisinage) et des limites du décodeur : confondre P et T est moins grave que confondre P et ON et ceci d'autant plus si on sait que le décodeur ne sait pas faire la différence entre P et T. Une manière de représenter ces caractéristiques est l'utilisation d'une matrice de confusion dont un exemple (figure 1.5) est donné dans [Mangeol-88]. Pour chaque couple de phonèmes, on indique grâce à une valeur définie de manière pragmatique le coût de leur confusion, deux phonèmes voisins auront un taux de confusion proche de 0 et deux phonèmes très différents auront un taux proche de 10. Un traitement similaire peut être fait pour les insertions et les élisions en attribuant à chaque phonème un coût d'élision et un coût d'insertion.
- Les techniques stochastiques utilisent une représentation des différentes formes phonétiques d'un mot à l'aide d'un graphe où chaque transition est affectée d'une probabilité [White-78]. Les modèles de Markov (HMM) sont une application de ces techniques et ont été utilisées en DAP [Miclet-88], [Mariani-89].
- Les techniques d'analyse par synthèse consistent à synthétiser le mot hypothèse et à comparer directement la production acoustique obtenue à celle de l'énoncé (en appliquant des algorithmes de programmation

dynamique), mais son principal inconvénient reste son important coût en temps de calcul [Klatt-75]

i	: (e 2)	(ai 3)	(y 1)	(œ 6)	(& 6)	(gh 4)	(l 4)
e	: (i 2)	(ai 0)	(œ 4)	(& 5)	(a 4)	(l 5)	(y 6)
ai	: (i 3)	(e 0)	(œ 4)	(& 5)	(a 4)	(r 4)	
y	: (i 1)	(œ 2)	(& 2)	(o 4)	(u 3)	(l 4)	
œ	: (& 0)	(a 1)	(o 2)	(m 5)			
&	: (e 3)	(y 3)	(œ 0)	(a 5)	(o 3)	(u 5)	
a	: (œ 2)	(& 4)	(o 5)	(in 1)	(ã 1)	(r 5)	
o	: (œ 1)	(& 3)	(u 1)	(in 3)	(ã 3)	(õ 1)	(r 5)
u	: (y 3)	(œ 2)	(& 3)	(o 1)	(õ 4)	(r 4)	
in	: (œ 2)	(a 1)	(ã 1)	(õ 1)	(r 4)		
ã	: (œ 3)	(a 1)	(o 3)	(in 1)	(õ 1)	(r 4)	
õ	: (œ 3)	(o 1)	(in 1)	(ã 1)	(n 5)		
w	: (a 1)	(in 2)	(ã 0)	(õ 3)	(r 0)		
j	: (i 1)	(s 8)	(z 3)	(gh 3)	(v 3)	(l 4)	
s	: (j 6)	(z 1)	(ch 1)	(gh 5)	(f 2)	(v 5)	
z	: (s 1)	(ch 4)	(gh 1)	(f 4)	(v 1)		
ch	: (s 1)	(z 4)	(gh 1)	(f 2)	(v 4)	(r 4)	
gh	: (s 4)	(z 1)	(ch 1)	(f 4)	(v 1)		
f	: (s 2)	(z 5)	(ch 2)	(gh 5)	(v 1)	(r 4)	(p 6)
v	: (s 5)	(z 1)	(ch 5)	(gh 1)	(f 5)	(p 4)	(b 1)
n	: (œ 5)	(in 4)	(ã 4)	(õ 4)	(v 0)	(m 1)(nj 1)	(l 3) (r 5)
m	: (œ 5)	(in 4)	(ã 4)	(õ 4)	(n 0)	(nj 1)(l 4)	(r 5)
nj	: (n 1)	(m 1)					
l	: (y 4)	(j 5)	(n 4)	(m 3)	(r 2)	(d 5)	
r	: (a 3)	(in 3)	(ã 3)	(w 1)	(z 4)	(gh 4)	(n 5) (f 4) (m 5) (l 2)
p	: (f 6)	(v 6)	(t 1)	(k 1)	(b 1)	(d 4)	(g 4)
t	: (p 1)	(k 1)	(b 4)	(d 1)	(g 4)		
k	: (p 1)	(t 1)	(b 4)	(d 4)	(g 1)		
b	: (f 7)	(v 1)	(n 5)	(m 5)	(p 1)	(t 4)	(k 4) (d 1) (g 1)
d	: (n 5)	(m 5)	(p 4)	(t 1)	(k 4)	(b 1)	(g 1)
g	: (n 5)	(m 5)	(p 4)	(t 4)	(k 1)	(b 1)	(d 1)

Figure 1.5. Matrice de confusion

remarque : la matrice est représentée par l'ensemble des phonèmes, chaque phonème ph1 étant associé à une liste de couples composés d'un phonème acoustiquement proche ph2 et le coût de la confusion entre ph1 et ph2.

- la détection de mots (Word Spotting) utilisée lorsque l'on ne dispose pas d'hypothèses mots. Elle est employée dans trois contextes particuliers :

- la construction d'un treillis de mots à partir de la forme phonétique de la phrase sans analyse linguistique préalable. Cette solution ne peut être envisageable que pour un vocabulaire restreint, en effet, il est inconcevable de vérifier tous les mots possibles sur l'ensemble du signal pour un grand vocabulaire, le coût en temps en serait beaucoup trop important.
- la localisation de points d'ancrage, qui consiste à détecter des mots clés dans l'énoncé afin de pouvoir orienter la suite de la reconnaissance.
- la vérification d'hypothèses émises par les niveaux linguistiques, qui consiste à rechercher sur l'énoncé des mots respectant certaines contraintes (syntaxiques, sémantiques ...) sans information sur leur position dans la phrase.

Pour la détection de mots, le problème se pose différemment. Il faut vérifier la présence de mots hypothèses dans le signal sans connaissance a priori de leur position éventuelle. Pour éviter un trop grand nombre de comparaisons fines, il convient de procéder en deux étapes, la première consistant à déterminer une zone sur laquelle le mot a une forte probabilité de se trouver et la seconde effectuant la comparaison de cette zone avec le mot de référence (par une des méthodes vues ci-dessus).

La première étape de détection de zone peut être résolue par des algorithmes mettant en correspondance des patrons phonétiques [Mangeol-88] ou des cohortes de syllabes [Causse-76], détectant des points d'ancrage phonétiques, ou encore réalisant une matrice de coïncidence entre la forme de référence et l'énoncé [Vives-85]

1.4. Les grandes approches de traitement de la langue naturelle

Dans les paragraphes précédents, nous avons vu comment retrouver des mots sur le signal. Pour construire la phrase complète, il n'est pas pensable de tester tous les mots du vocabulaire et de combiner tous ceux qui ont été reconnus pour obtenir l'ensemble des phrases possibles ; d'une part, le temps de recherche deviendrait rapidement prohibitif et, d'autre part, une grande partie des combinaisons de mots ainsi obtenues ne seraient pas porteuses de sens.

Pour recomposer les énoncés, l'analyseur doit être guidé par des règles de construction de phrases. Dans les paragraphes suivants, nous allons présenter quelques approches de la langue naturelle parmi les plus représentatives ainsi que quelques théories qui en ont découlé.

1.4.1. Les grammaires formelles

1.4.1.1. Les grammaires syntaxiques

Les premières approches du langage par les informaticiens étaient purement syntaxiques, les phrases étaient représentées à l'aide de grammaires formelles [Chomsky-71], c'est-à-dire des grammaires dont les terminaux correspondent à des classes grammaticales. La figure 1.6 nous montre un exemple simplifié d'une grammaire formelle de la langue française.

PHRASE	--->	GN + GV
GN	--->	ART + NOM + (ADJ)
GV	--->	VERBE + GN
VERBE	--->	pose / prends / ...
ART	--->	le / la / un / ...
NOM	--->	robot / livre / chaise / table / ...
ADJ	--->	petit / rouge / ...

Figure 1.6. Exemple de grammaire formelle

Une telle grammaire permet de reconnaître une phrase telle que :

" Le robot prend le livre rouge "

qui va être représentée par l'arbre de dérivation donné à la figure 1.7

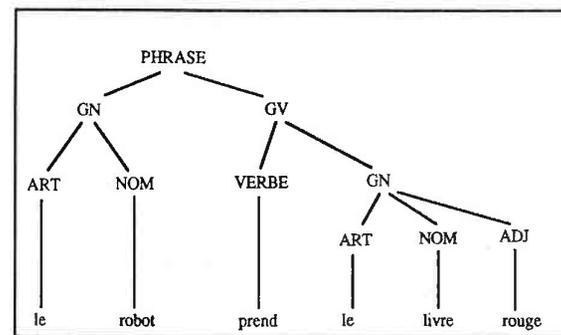


Figure 1.7. Arbre de dérivation de la phrase :
"le robot prend le livre rouge"

Ce type de grammaire purement syntaxique a l'avantage principal d'utiliser des techniques d'analyse très bien connues et faciles à mettre en œuvre. Cependant, l'utilisation de telles grammaires entraîne une rigidité qui contraint à l'utilisation d'un langage artificiel, ce type de grammaire est donc à exclure pour un système dont l'objectif est d'accepter des énoncés les plus naturels possibles.

De plus, une analyse syntaxique, ne tenant pas compte du sens, considère les deux énoncés suivants comme des phrases correctes :

- le chat mange la souris
- le livre mange le téléphone.

Pour diminuer le nombre de phrases acceptables en fonction de leur sens, il est devenu nécessaire de tenir compte d'informations sémantiques. La sémantique représente les contraintes autres que positionnelles entre les éléments d'une phrase dues à leur signification et telles que ces éléments soient compatibles entre eux.

1.4.1.2. Les grammaires transformationnelles

L'intégration de connaissances sémantiques a donné naissance aux grammaires transformationnelles. Ces grammaires, purement génératives, constituent une théorie complète du langage. Chomsky a proposé une organisation des connaissances syntaxiques en trois volets :

- Une grammaire formelle permet d'engendrer la structure profonde des phrases simples, déclaratives et à la forme active représentées par leur arbre de dérivation.
- Un ensemble de règles de transformations permettant à partir d'un arbre de dérivation d'obtenir toutes les formes possibles d'une phrase (ex : forme passive, pluriel ...)
- Des règles morpho-phonémiques permettant de passer de la phrase à la suite de caractères ou de phonèmes la représentant.

L'arbre de dérivation représente la structure profonde de la phrase qui est interprétée par un composant sémantique qui en donne le sens.

La figure 1.11 représente un modèle de la théorie standard des grammaires transformationnelles.

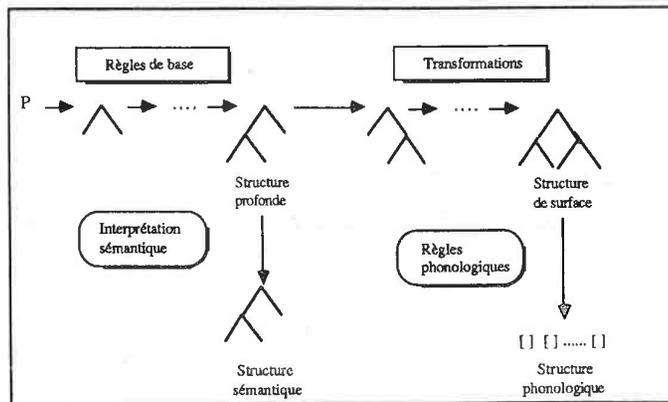


Figure 1.11. Un modèle de la théorie standard d'après [Sabah-88]

Les flèches verticales correspondent aux fonctions interprétatives et les flèches horizontales aux fonctions génératives.

La théorie étendue reconnaît l'influence des transformations sémantiques. Gross [Gross-86][Gross-86bis] a effectué, pour le français, une classification détaillée des traits sélectifs (ex : transitif, humain ...) qui sont ajoutés à la grammaire afin d'éviter les phrases dépourvues de sens (ex : le carreau casse Jean). La principale difficulté technique d'utilisation des grammaires transformationnelles en reconnaissance est la définition des règles réciproques permettant, à partir du signal, de retrouver la structure profonde de la phrase.

1.4.1.3. Les grammaires sémantiques

Elles sont une autre approche sémantique du langage. Elles utilisent les techniques connues des grammaires formelles, mais à la différence des grammaires purement syntaxiques, les non-terminaux de ces grammaires ne sont plus uniquement des classes syntaxiques, mais des classes grammaticales et sémantiques à la fois. La plupart des systèmes utilisent les grammaires sémantiques.

La figure 1.9 nous donne un exemple de grammaire sémantique simplifiée.

COMM	-->	AGENT + VERB + OBJET
VERB	-->	prend / pose / ouvre / ...
OBJET	-->	livre / porte / table / ...
AGENT	-->	robot / Pierre / ...

figure 1.9. Grammaire sémantique simplifiée

Ainsi la phrase " le robot prend le livre " vue dans le paragraphe précédent va être représentée par l'arbre de dérivation donné à la figure 1.10.

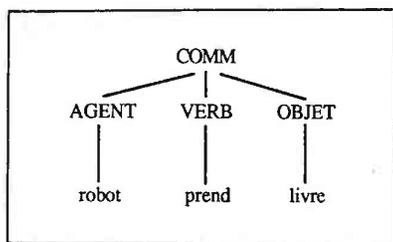


Figure 1.10 Arbre de dérivation

Cette grammaire sémantique permet de rejeter une phrase telle que : " *le livre prend Pierre* " ce qui n'est pas le cas des grammaires formelles qui ne font pas la distinction entre les différents rôles (au sens sémantique) que peuvent jouer les mots du lexique.

Un exemple d'utilisation de grammaires sémantiques plus concret dans le domaine du diagnostic médical est donné dans [Coulon-86] où les non terminaux sémantiques sont des concepts tels que : malade, symptôme, date ...

PARTNER [Morin-87] est une application de commande orale d'un système de courrier électronique utilisant des grammaires sémantiques qui ont pour originalité d'utiliser des terminaux variables permettant une adaptation simplifiée du système dans une autre langue.

L'avantage de ces grammaires est qu'elles sont très souples et faciles à mettre en œuvre (elles bénéficient des études faites sur les grammaires formelles). L'inconvénient majeur vient du fait de l'intégration des classes sémantiques dans la grammaire, ce qui implique d'une part la nécessité d'une connaissance a priori de ces classes, et d'autre part la grande dépendance de la grammaire vis-à-vis de l'application, ce qui en diminue la portabilité.

Nous pouvons citer d'autres théories issues de ces études :

- la théorie standard étendue [Chomsky-71bis],
- la théorie des traces [Chomsky-75],
- les grammaires syntagmatiques généralisées [Gazdar-79],
- les grammaires logiques [Pereira-80] issues des grammaires de métamorphoses [Colmerauer-78],
- ...

1.4.2. L'approche sémantique

1.4.2.1. Les réseaux sémantiques

Les réseaux sémantiques sont très utilisés en représentation des connaissances [Kayser-86]. Ils sont composés de nœuds représentant des concepts et d'arcs valués représentant les liens entre les concepts. Dans un système de dialogue ils peuvent intervenir à deux niveaux :

- lors de la reconnaissance de phrases en permettant de prédire les éléments à venir,
- lors de l'interprétation de phrases en servant de modèle à la phrase reconnue.

La figure 1.8 nous donne un exemple de représentation de la connaissance contenue dans l'énoncé :

Obaq aime les pommes, d'après [Norihiro-81]

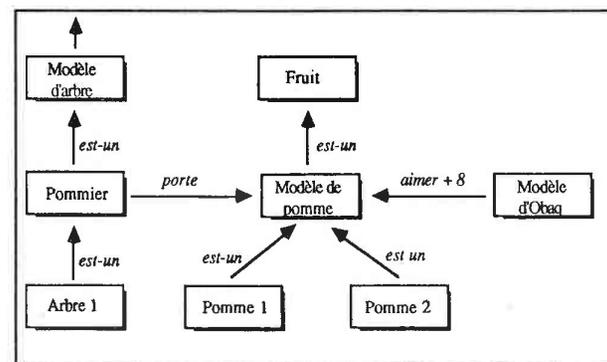


Figure 1.11 Exemple de réseau sémantique

L'inconvénient principal des réseaux sémantiques vient du fait que, comme pour les grammaires sémantiques, il faut connaître a priori l'ensemble des relations entre les concepts, ce qui implique une redéfinition complète du réseau lors d'un changement d'application.

1.4.2.2. Les grammaires de cas

La sémantique prenant une part de plus en plus importante dans les théories linguistiques, les grammaires de cas, fondées sur la théorie de Fillmore [Fillmore-68] firent leur apparition. Fillmore soutient que l'on peut identifier un ensemble de cas sémantiques permettant de mettre en évidence, à la manière des cas syntaxiques, les relations de sens qui existent entre les différents groupes d'une phrase simple et le verbe qui la gouverne. Chaque verbe est représenté par une (ou plusieurs) structure(s) casuelle(s) (ossature(s)), c'est-à-dire l'ensemble des cas (obligatoires ou optionnels) indiquant pour chaque argument le type de lien qui le lie au verbe.

Les grammaires de cas permettent de résoudre des problèmes non résolus par les grammaires transformationnelles tout en conservant un caractère général que ne possèdent pas les grammaires sémantiques. Un cas simple est la conjonction des sujets, prenons pour exemple les trois énoncés suivants :

- Pierre remplit le vase (1)
- L'eau remplit le vase (2)
- *1 Pierre et l'eau remplissent le vase (3)

Dans le cas des grammaires transformationnelles, (1) et (2) auront la même structure profonde, cependant la conjonction entre Pierre et l'eau n'est pas autorisée car les deux groupes ne possèdent pas les mêmes traits sémantiques. Les grammaires de cas permettent de tenir compte de ces différences et la structure profonde obtenue permet de faire apparaître nettement le sens de la phrase.

Cependant l'ensemble des cas choisis a des conséquences importantes sur le mode d'interprétation des énoncés ; par exemple si l'on affecte au verbe ACHETER l'ensemble des cas : (AGENT, OBJET, BÉNÉFICIAIRE), l'énoncé : *Pierre achète des fleurs à M^{lle} Rose* présuppose que les fleurs sont destinées à M^{lle} Rose, mais si l'on remplace le cas BÉNÉFICIAIRE par le cas PATIENT, le même énoncé présupposera que M^{lle} Rose est la vendeuse à qui Pierre achète ses fleurs.

¹ L'astérisque indique une phrase incorrecte

Dans le système de renseignements administratifs développé au CRIN, G. Deville [Deville-86] utilise une approche fonctionnelle du langage inspirée des grammaires de cas.

Il existe des théories voisines de la théorie des grammaires de cas, nous citerons par exemple : [Grimes-72], [Simmons-74], [Schank-72] et [Bogurov-83].

Les grammaires de cas ont pour avantage de fournir un modèle de la structure profonde de la phrase en associant à un verbe sa structure casuelle, ce qui n'est pas sans rappeler la notion de prédicat logique appliqué à ses arguments qui trouve une application directe dans les langages de type commande. De plus l'analyse d'une phrase se fait à partir de restrictions sémantiques tout en autorisant l'inclusion de contraintes syntaxiques. Les grammaires de cas sont l'un des rares modèles permettant le traitement automatique de phrases syntaxiquement incorrectes, ce qui est primordial dans un système de dialogue oral en langage naturel.

1.4.2.3. La définition de traits sémantiques dans le lexique

Lorsque l'analyse du langage utilise une approche sémantique, il est nécessaire de pouvoir indiquer pour chaque mot du vocabulaire les rôles sémantiques qu'il est susceptible de jouer.

La définition de traits sémantiques dans le lexique permet de regrouper des sous-ensembles de mots en classes sémantiques. Cette méthode peut être associée à une analyse syntaxique de la phrase pour en diminuer les combinaisons possibles, ou encore aux grammaires de cas. Le système Myrtille II [Pierrel-81] utilise la définition de traits sémantiques dans le lexique associée à une analyse syntaxique faite à l'aide de RNP¹.

La définition de traits sémantiques dans le lexique est un complément nécessaire à toute analyse sémantique, elle a pour avantage de diminuer le temps

¹ RNP: réseaux à noeuds procéduraux proches des ATN de Woods

de recherche des mots dans l'énoncé par la sélection du sous-lexique correspondant au rôle sémantique recherché.

1.4.3. Les grammaires systémiques

Une nouvelle approche du langage consiste à considérer celui-ci comme une activité sociale liée à son contexte d'utilisation. C'est le but des grammaires systémiques où une phrase ne sera plus seulement interprétée en fonction de sa syntaxe et des propriétés sémantiques des différents groupes qui la composent, mais elle sera remise dans le contexte où elle a été énoncée. Les grammaires systémiques sont purement descriptives, elles font l'étude de l'organisation fonctionnelle du langage et des liens qui existent entre la forme d'un énoncé et le contexte d'une part, et des liens entre la forme d'un énoncé et les situations dans lesquelles il peut apparaître, d'autre part.

Par exemple, les conventions sociales permettent de détecter un ordre dans la question : "*Pouvez-vous me passer le sel ?*". De la même manière, la situation d'énonciation permet de détecter une menace, une promesse ou une constatation dans la phrase : "*Papa va rentrer*".

D'après la théorie de Halliday (voir [Sabah-88]), il y a quatre systèmes qui agissent dans la détermination de la structure de la phrase. Soit, par exemple la phrase : "*Pierre prend un livre*", les quatre systèmes sont instanciés de la manière suivante :

- système du monde : *phrase principale, déclarative, affirmative.*
- système d'actants : *agent = Pierre, objet = livre.*
- système thématique : *thème = Pierre, rhème = il prend le livre*
- système informatif : *information connue = Pierre prend quelque-chose
information nouvelle = c'est un livre*

Ces grammaires sont une bonne approche pour l'interprétation de dialogue ou la compréhension de textes car elles prennent en compte tout le contexte de l'énonciation, elles permettent donc de tenir compte des différents points de vue et des connaissances contextuelles, elles constituent une approche sémantique et pragmatique de la langue. Ce type de grammaire est un outil important dans des systèmes de compréhension de textes mais semble trop riche pour des applications de type commande.

Chapitre 2 Le dialogue

Dans un premier temps, nous avons vu les problèmes de la communication entre un homme et une machine. Dans ce chapitre, nous allons aborder plus spécifiquement les problèmes liés au dialogue, dans le cas général, puis dans le cas particulier où un des deux interlocuteurs est une machine.

2.1. Généralités sur le dialogue

Le dialogue entre personnes peut être considéré comme une activité sociale, les études faites par des linguistes et des psycho-linguistes ont permis de mettre en évidence les principales caractéristiques des dialogues ainsi que des règles qui les régissent tant sur le fond que sur la forme.

2.1.1. Les caractéristiques du dialogue

Les phrases énoncées au cours d'un dialogue ont pour principale caractéristique leur concision due à la culture commune des interlocuteurs. On peut distinguer deux classes de concision :

- la concision de forme,
- la concision de fond.

2.1.1.1. La concision de forme

La concision de forme est obtenue par l'application de règles syntaxiques telles que les références anaphoriques, l'énoncé de phrases elliptiques... Ces différentes formes d'énonciation peuvent être source d'ambiguïtés qu'il faudra tenter de lever grâce à des règles syntaxico-sémantiques ou à des connaissances sur le monde.

Ex : *En posant le vase sur le meuble il le cassa*

L'ambiguïté sur le pronom "le" de la deuxième partie de l'énoncé peut être résolue par la connaissance des règles régissant l'univers grâce auxquelles on pourra déduire qu'un choc entre un vase et un meuble entraînera la détérioration du vase plutôt que celle du meuble. Cependant on se rend compte que ce genre d'information impliquerait une modélisation du monde très complexe qui ne pourrait être effective que pour un domaine très particulier.

Un autre moyen de lever l'ambiguïté de la phrase donnée en exemple est l'utilisation de règles syntaxico-sémantiques. Si l'on fait un parallèle avec les grammaires de cas vues au chapitre précédent, on peut remarquer que "le vase" joue un rôle sémantique d'objet dans la première partie de l'énoncé et que c'est ce même rôle que tient le référent "le" dans la deuxième partie. La correspondance est donc intuitive sans avoir pour cela à connaître les règles régissant le monde. Si le locuteur avait voulu faire comprendre que s'est le meuble qui a été cassé, il aurait énoncé sa phrase de la manière suivante :

"Il cassa le meuble en y posant le vase".

Il est cependant d'autres cas où les règles syntaxico-sémantiques ne suffisent pas pour lever l'ambiguïté et dans ce cas les connaissances sur le modèle du monde et sur la situation du dialogue deviennent nécessaires :

Ex : Il prend une pomme, ouvre la bouche et la croque.

Dans ce cas le référent "la" se rapporte à la pomme qui a bien un rôle d'objet mais qui n'est pas le dernier cité, le dernier étant la bouche. Les connaissances pragmatiques sur l'univers vont nous permettre de déterminer que c'est la pomme qui est croquée et non pas la bouche.

Nous verrons dans le chapitre 4 les deux types de références (référence conceptuelle et référence occurrence) que nous avons traitées et les moyens syntaxico-sémantiques que l'on peut mettre en œuvre pour les résoudre ainsi que les connaissances pragmatiques nécessaires à une bonne résolution.

La concision de forme peut être obtenue aussi par l'énoncé de phrases elliptiques. Les éléments manquants se trouvant dans l'énoncé précédent.

Ex : Prends le livre / et le cahier

2.1.1.2. La concision de fond

La concision de fond peut être obtenue par l'utilisation de figures de style (les tropes) dans lesquelles on emploie les mots avec un sens différent de leur sens habituel. Les figures les plus courantes sont les métaphores et les métonymies.

Ex : Il vit du fruit de son travail (métaphore)
Il vit de son travail (métonymie)

J. Pitrat [Pitrat-85] nous donne un bon exemple de métaphore pour le commentaire de parties d'échec :

" Les pièces noires sont agglomérées comme un troupeau de moutons sur lequel le loup s'apprête à fondre"

Cette phrase fait une analogie entre un animal et une pièce d'un jeu d'échec. L'expression "troupeau de moutons" apporte de l'information par les trois concepts dont elle est porteuse :

- ces animaux ne sont pas agressifs,
- ces animaux ne savent pas se défendre,
- cet ensemble d'animaux est mal structuré.

La concision peut être elle aussi obtenue par l'utilisation d'énoncés elliptiques, mais dans ce cas les parties manquantes peuvent être reconstruites à partir d'inférences, représentées par des valeurs par défaut associées au monde, ou par des connaissances sur les circonstances du dialogue.

Considérons par exemple l'énoncé suivant : "Jean aime la télévision", la télévision étant un objet dont la principale fonction est d'être regardée, l'inférence qui sera faite est :

" Jean aime regarder la télévision".

Mais si le dialogue nous a permis d'apprendre que Jean travaille à la télévision, une autre inférence possible sera :

"Jean aime travailler à la télévision".

Si les connaissances courantes ne permettent pas de faire l'inférence correcte, l'énoncé devra contenir toutes les informations typiques :

"Jean aime réparer les télévisions".

2.1.2. Les règles du dialogue

A l'issue des travaux menés par des linguistes et des psycho-linguistes, Grice [Grice-75] a mis en évidence des règles portant sur le fond et la forme des dialogues. Selon sa théorie, la communication implique la représentation et la reconnaissance des intentions du locuteur. Les règles de Grice sont suffisamment explicites et universelles, la transmission d'un message implique le respect de ces règles. Cependant si une des règles est violée, c'est cette violation qui est porteuse du sens et non plus le texte lui-même.

Les règles sont regroupées en quatre classes de maximes :

- les maximes de quantité : l'énoncé ne doit comporter ni trop ni trop peu d'informations,
- les maximes de qualité : le locuteur ne doit pas énoncer une phrase qu'il croit fausse et ne doit pas donner d'information qui ne puisse être démontrée,
- les maximes de relation : l'information donnée par le locuteur doit être pertinente,
- les maximes de manière : elles invitent le locuteur à éviter les expressions obscures et ambiguës, à être bref et à donner les informations dans le bon ordre.

La violation d'une de ces maximes correspond à une intention bien précise du locuteur qui est de vouloir faire passer une information sans la dire explicitement. Par exemple, si l'on demande à un enseignant ce qu'il pense du travail de son élève, et que sa réponse est : "il ne bavarde jamais et il est très gentil avec ses camarades", les maximes de relation et de quantité sont violées. Du fait de cette violation, il essaye de faire comprendre qu'il n'est pas satisfait du travail de son élève. La violation est faite au niveau de ce qui est dit mais pas au niveau de ce qui est sous-entendu.

Ces travaux rejoignent ceux d' A. Borillo [Borillo-77], linguiste qui a montré l'importance de la prise en compte de la situation réelle d'énonciation pour comprendre et participer à un dialogue.

2.2. Le dialogue dans un système de communication homme-machine

Nous nous plaçons ici dans le cadre d'un dialogue général entre un homme et une machine. Nous allons dans un premier temps présenter les qualités principales que doit posséder un module de dialogue entre un homme et une machine, puis nous parlerons des différents rôles que doit jouer un tel module, ensuite nous détaillerons ses rôles fondamentaux que sont l'interprétation et le raisonnement et enfin nous établirons au vu de toutes les fonctionnalités qui lui sont demandées, la liste des sources de connaissances qui sont nécessaires pour obtenir un module de dialogue suffisamment évolué.

2.2.1. Les qualités du module de dialogue

La principale caractéristique que doit posséder un module de dialogue est le respect d'un échange naturel avec l'interlocuteur humain. Il doit donc en particulier lui laisser l'initiative tout en le guidant dans ses recherches d'informations. L'initiative étant laissée à l'utilisateur, la machine doit être capable de prendre en compte toutes les informations fournies par celui-ci même si elles n'interviennent pas dans la réalisation du but courant, et non pas simplement extraire les informations pertinentes sinon elle peut être amenée à poser une question dont la réponse est une information déjà fournie dans une phase précédente du dialogue. Si l'on ne tient pas compte de ces informations, on peut être amené à rencontrer des dialogues du type de celui de la figure 2.1.

S : Bonjour, que désirez-vous?
U : Je voudrais le numéro de téléphone de M. Dupond dont je ne connais pas le lieu de résidence.
S : Où habite M. Dupond?
U : Je vous ai dit que je ne le savais pas!

figure 2.1. Exemple de dialogue

Le module de dialogue doit être capable de relier plusieurs énoncés entre eux pour en dégager le thème général et les informations nécessaires à leur interprétation afin d'en déduire le plan éventuel du locuteur.

Il doit être capable de détecter les changements de sujet aussi bien que la reprise d'un thème déjà abordé, comme dans l'exemple de dialogue de la figure 2.2 tiré de [Vilnat-85].

Il doit être capable de reconnaître les fonctions possibles de l'énoncé sur la base d'informations contextuelles plutôt que syntaxiques. Les phrases sont utilisées pour réaliser des actes de langage, elles ont en général une forme adaptée à leur fonction (interrogative -> question, impérative -> ordre ...), mais parfois les actes sont indirects (ex : je t'ordonne de venir). Des expériences ergonomiques [Clark-79], [Luzzati-85] ont montré que la formulation des questions avait une influence importante sur le type des réponses apportées. L'application informatique des actes de langage a été développée dans un système de renseignements sur des horaires de train par Allen, Cohen et Perrault [Cohen-79], [Allen-80] où les actes de langages sont classés selon deux catégories : les actes de dialogue orientés vers le but à atteindre qui ont la particularité d'être prédictibles et les actes de contrôle du dialogue portant sur le dialogue lui-même.

U : Je voudrais le numéro de téléphone d'un restaurant d'extrême-orient.
S : Préférez-vous un restaurant vietnamien, japonais ou chinois?
U : Cela m'est égal
S : Dans quel quartier?
U : Je voudrais d'abord savoir si vous avez le numéro d'un radio-taxi.
S : Oui, nous avons Allo Taxi
U : Dans ce cas, j'aimerais un restaurant dans le 5^{ème} arrondissement
S : Vous avez Le Mandarin

Figure 2.2. Dialogue d'après [Vilnat-85]

Pour chaque décision prise, il doit pouvoir expliquer le raisonnement qui l'y a mené en s'adaptant au niveau du locuteur (spécialiste ou non).

Il doit aussi pouvoir réagir en cas d'échec d'interprétation.

Enfin, il doit connaître ses limites pour éviter des recherches inutiles en cas de contradiction avec ses connaissances ou d'une demande dépassant la limite de ses compétences.

2.2.2. Les différents rôles du module de dialogue

On peut distinguer cinq grands rôles attribués au module de gestion du dialogue :

- La gestion du canal de communication, dont les trois principales fonctions sont :

- les demandes de répétition quand l'analyseur de phrases a été incapable de détecter le moindre mot dans l'énoncé dans le cas où le locuteur a parlé trop doucement ou trop fort. Le module de dialogue doit être capable de définir la cause de la non reconnaissance de manière à conseiller le locuteur sur sa façon de parler (ex : Parlez plus fort, articulez, parlez plus lentement...),
- la relance du dialogue quand l'utilisateur ne réagit pas,
- l'émission de messages de mise en attente et de maintien du dialogue quand le temps de réponse de la machine est trop important (ex : veuillez patienter votre demande est en cours de traitement.).

- L'interprétation de l'énoncé dont les fonctions sont :

- la résolution des ellipses et des anaphores par l'utilisation de connaissances syntaxico-sémantiques aussi bien que par celle de connaissances pragmatiques,
- la détermination du type de la phrase et en particulier savoir si elle porte sur la tâche ou sur le dialogue, s'il s'agit d'une question, d'une contestation, d'un apport d'information....
- la compréhension du sens de l'énoncé, c'est à dire la détermination du sens de la phrase en fonction des connaissances disponibles sur le monde et sur l'environnement du dialogue. Il s'agit là de l'interprétation

de la phrase dans le contexte de la tâche et du dialogue. Nous verrons dans le paragraphe 2.2.3. les différentes interprétations possibles d'un énoncé et les moyens mis en œuvre pour extraire les informations implicites contenues dans une phrase.

- La gestion du dialogue, c'est-à-dire le traitement des énoncés portant sur le dialogue lui-même, comme les demandes de répétition, les contestations, les confirmations ...

- Le raisonnement appliqué à un énoncé portant sur la tâche. Il s'agit de vérifier la cohérence des propos tenus par le locuteur et de réagir en conséquence soit par une réponse adéquate, soit par la réalisation d'une action.

- La prédiction vers les autres niveaux d'analyse, soit par l'émission d'hypothèses syntaxico-sémantique (si le module de dialogue pose une question précise, il s'attend à un type de réponse bien particulier qu'il est capable de définir à l'avance), par des restrictions d'ordre lexical (par exemple lors d'une demande portant sur la couleur d'un objet particulier, il peut restreindre la liste des adjectifs attendus à celle indiquant une couleur).

Après avoir présenté les différents rôles du module de dialogue, nous allons détailler dans les paragraphes suivants ceux dont l'importance est primordiale et dont la réalisation demande la manipulation de nombreuses connaissances, ce sont les modules d'interprétation et de raisonnement.

2.3. L'interprétation

Le module de dialogue reçoit une représentation syntaxico-sémantique de l'énoncé, fournie par l'analyseur de phrases. Le but du module de dialogue est d'interpréter cet énoncé. Pour Sabah [Sabah-88], la représentation syntaxico-sémantique de l'énoncé représente le sens littéral de la phrase et l'étape postérieure à cette première compréhension consiste en l'interprétation qui peut être de deux types :

- réalisation d'inférences en vue d'explicitier les connaissances communes aux locuteurs (celles-ci restant implicites),
- étude de l'influence du contexte sur le sens de la phrase.

Il schématise ces deux types d'interprétation par la figure 2.3.

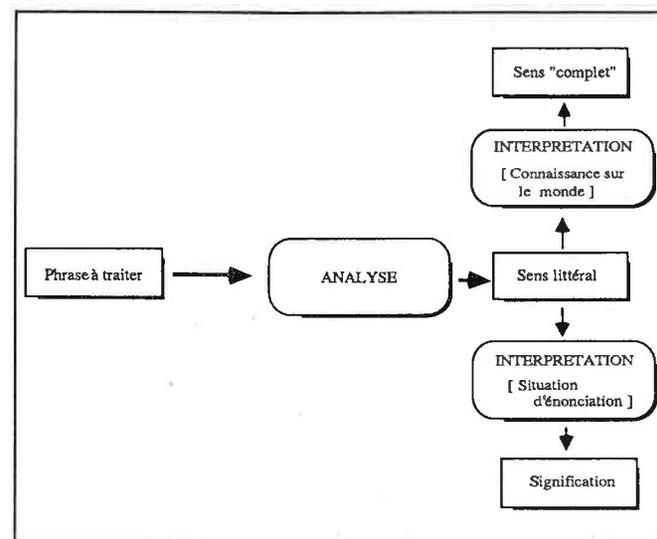


Figure 2.3. Les deux types d'interprétation d'après [Sabah-88]

L'importance des connaissances pragmatiques pour interpréter un énoncé est évidente, le problème consiste dans le choix d'une représentation de ces connaissances et du niveau de compréhension souhaité qui va influencer sur la quantité des informations à stocker ainsi que sur leur structure. L'interprétation d'une phrase ou d'un texte dépend de l'utilisation ultérieure que l'on veut en faire. Dans certains cas, il est nécessaire d'obtenir le sens "complet" de l'énoncé, par exemple pour l'explication de texte ou la compréhension de textes, dans d'autres cas, la signification en contexte est suffisante comme par exemple pour le résumé de textes.

En fonction des buts poursuivis, la représentation des connaissances pragmatiques nécessaires à une bonne interprétation va différer tant sur la forme que sur le contenu.

La pragmatique a deux rôles essentiels dans l'interprétation de phrases :

- la levée d'ambiguïtés,
- la représentation du sens de la phrase dans le contexte.

Dans le § 2.1.1.1, nous avons vu un type d'ambiguïté dû à l'utilisation de références anaphoriques, certaines pouvant être levées à l'aide de connaissances syntactico-sémantiques et d'autres à l'aide de connaissances pragmatiques. Cependant il existe d'autres types d'ambiguïtés pour lesquelles les connaissances pragmatiques sont nécessaires et parfois même insuffisantes encore.

Ex : Ferme la porte

L'énoncé en lui-même n'est pas ambigu, cependant son interprétation peut l'être s'il existe plusieurs portes dans l'univers considéré. Mais si parmi toutes ces portes, une seule est ouverte, la pragmatique va nous permettre de décider que c'est d'elle qu'il s'agit, en effet pour qu'une porte puisse être fermée, il faut a priori qu'elle soit déjà ouverte sinon l'action demandée est impossible à réaliser. La sélection de l'objet, dans ce cas, se fait parce que toute autre sélection est impossible.

Dans certains cas d'ambiguïtés, plusieurs solutions sont possibles mais certaines sont plus probables que d'autres :

Ex : J'écris un livre sur mon bureau

"sur mon bureau" peut être interprété comme le lieu où le livre est écrit mais aussi comme le sujet sur lequel porte l'ouvrage. Si cette deuxième interprétation est moins probable que la première, elle n'en reste pas moins possible, en effet si le bureau en question a une histoire intéressante, il n'est pas exclu qu'il puisse faire l'objet d'un livre.

Certaines ambiguïtés peuvent être dues à l'ambiguïté de sens des mots eux-mêmes tels que le verbe "acheter". Acheter quelque-chose à quelqu'un peut signifier que ce quelqu'un est le vendeur de l'objet ou le bénéficiaire de l'achat. Dans ce cas, les connaissances pragmatiques sont impuissantes à lever l'ambiguïté, il faut faire appel au contexte de l'énonciation afin de connaître les liens existants entre les différentes personnes.

Ces exemples montrent bien la difficulté de la mise en œuvre des connaissances pragmatiques et contextuelles. Dans les paragraphes suivants, nous allons

présenter quelques modes classiques de représentation de ces connaissances permettant de prendre en compte certains des problèmes soulevés ci-dessus.

2.3.1. Les règles de production

Les règles de production permettent de représenter les connaissances pragmatiques sous forme déclarative. Un système à règles de production est composé :

- d'un ensemble de règles,
- du contexte,
- d'un interprète ou moteur d'inférences qui sélectionne les règles applicables dans le contexte, résout les conflits quand plusieurs règles sont activables simultanément et exécute les actions modifiant le contexte.

Ex : Soit la règle : **Fermer (x) ---> Ouvert (x)**
et le contexte : **Ouvert (porte1)**
Fermé (porte2)

L'application de la règle dans ce contexte permettra la sélection de l'objet porte1 et lèvera ainsi l'ambiguïté qui aurait pu apparaître.

Les règles de production ont pour avantage d'utiliser un formalisme très facile à comprendre et de permettre des modifications (ajout, suppression de règles...) qui ne remettent pas en cause ce qui est déjà fait. Le principal inconvénient des règles de production est qu'elles ne permettent pas de représenter certaines connaissances qui deviennent nécessaires lorsque le niveau de compréhension s'affine.

2.3.2. Les frames ou schémas

L'idée des schémas est venue à la suite d'études en psychologie cognitive et a été appliquée en particulier en informatique pour traiter les phénomènes de la vision [Minsky-75]. Lorsqu'une personne entre dans une pièce, elle n'en détaille pas chaque constituant pour en avoir une connaissance complète. Avant d'entrer elle a déjà une représentation a priori de ce que peut contenir la pièce, il lui suffit donc de noter les détails qui diffèrent de son modèle hypothétique. Pour chaque

concept nous possédons une représentation type qui nous permet de l'identifier rapidement lorsque nous le rencontrons ce qui nous permet de le reconnaître même si certains détails sont différents de ce que nous attendons.

Un schéma est donc composé d'un ensemble d'attributs et pour chaque schéma, il existe un prototype représentatif de tous les éléments de sa classe.

Pour un attribut, on dispose de plusieurs types d'informations :

- les **valeurs possibles** (type, intervalle ...),
- un **serveur** : procédure ou règle déclenchée lorsque l'on a besoin d'une valeur et que celle-ci n'est pas définie. Un serveur peut être une valeur par défaut, une question à poser, un calcul ... Les serveurs les plus courants sont : if-needed, to-establish,
- un **démon** : procédure ou règle qui est déclenchée lors de l'attribution d'une valeur pour en vérifier la cohérence par exemple (c'est un réflexe du système). les démons les plus courants sont : if-added, if-modified.

La figure 2.4 nous donne un exemple de schéma décrivant l'objet "pâte".

```
Pâte ( couleur ( valeur (jaune) )
      ( matière ( valeur (farine œufs sel)
                  ( if-added (démon-couleur) ) )
      ( prix (if-needed (calcule-prix (matière))))

(définir démon-couleur (c)
 (cas ( c = "épinard" ) => valeur (couleur) <-- "vert"
      ( c = "tomate" ) => valeur (couleur) <-- "rouge"))
```

Figure 2.4. Schéma de "Pâte" d'après [Colnet-86]

Ce mécanisme est simple à mettre en œuvre sur un domaine limité mais devient plus délicat lorsque le nombre de connaissances à traiter est important. En effet,

les démons peuvent déclencher d'autres démons ce qui, dans des systèmes complexes, peut devenir incontrôlable et peut entraîner des conséquences catastrophiques en cas de modification.

Dans un système de commande de robot muni d'un dispositif de vision, ce type de représentation des connaissances est intéressant pour limiter les temps d'analyse d'images, en effet, une classe d'objet va être représenté par un frame possédant des propriétés physiques (couleur, position, taille ...) qui seront évaluées par le système de vision, une solution intéressante est de disposer de serveurs qui ne seront activés que si la valeur de la propriété est nécessaire à un moment donné, ce qui évite une analyse systématique de toutes les propriétés des objets.

2.3.3. Les scripts ou scénarios

D'après Schank, l'homme, dans sa vie quotidienne, obéit à une grande quantité d'événements stéréotypés pour lesquels les actions s'enchaînent les unes aux autres sans que les divers choix possibles à un moment donné ne soient envisagés. Ils correspondent à ce que l'on appelle nos habitudes /qui n'a jamais pris la direction de sa maison en sortant du travail alors qu'exceptionnellement, il fallait choisir une autre direction pour effectuer une course quelconque?

Un script servira donc à représenter des séquences d'actions stéréotypées (prendre le métro, faire le ménage ...), il sera composé de trois parties :

- les acteurs et les objets présents lors de sa réalisation,
- le déclencheur,
- la succession normale des actions (souvent donnée sous forme de graphe pour permettre des répétitions, des courts-circuits ...)

Les scripts ayant pour but premier de rendre compte de situations stéréotypées, ils s'adaptent mal aux actions imprévisibles. L'intérêt des scripts réside dans le fait qu'ils permettent de compléter le sens des énoncés, en effet à partir de quelques informations permettant de définir le script dans lequel on se trouve et les actions les plus importantes, il est possible de reconstituer tout le scénario probable correspondant et donc de pouvoir répondre à des questions portant sur des faits non expliqués.

Les scripts sont donc très utiles pour la compréhension de textes, pour le résumé (pour lequel il suffira d'affecter à chaque action du script un coefficient d'importance). Dans un système de commandes de robot, les scripts peuvent être utilisés pour décrire les actions stéréotypées (ex : ouvrir une porte, prendre un objet ...), le déclencheur étant l'état dans lequel doit se trouver le robot pour exécuter l'action (ex : être près de la porte ...).

2.3.4. Les plans et les thèmes

Comme nous l'avons vu dans le paragraphe précédent, les scripts s'adaptent mal aux actions imprévisibles. Pour les circonstances exceptionnelles, l'utilisation de plans est bien adaptée. Un plan va permettre de comprendre les liens entre différentes actions destinées à réaliser un but [Pitrat-85].

L'utilisation de buts pour la définition des plans apporte une grande généralité, en effet, toutes les actions possibles réalisant chaque but peuvent être envisagées, la conséquence en est que le texte doit contenir plus d'informations quant aux choix effectués, contrairement aux scripts où les actions s'enchaînent naturellement. Cependant, il n'est pas toujours facile de reconnaître un plan dans un texte, pour cela il est nécessaire d'introduire la notion de thème, un thème (par exemple l'amour, la joie ...) permet de définir des buts implicites qui lui sont fortement liés.

Notons que les termes de plan et de but utilisés sont ceux adoptés par les chercheurs du domaine du traitement du langage naturel. En planification d'actions industrielles, notamment en robotique, le terme de plan recouvre la succession des actions permettant de réaliser un but spécifié à partir d'un état initial. Des générateurs de plans sont conçus pour cela. Le but est alors la spécification des caractéristiques qui doivent être vérifiées dans un état considéré comme "état final" pour le problème [Hendler-90].

La compréhension repose donc sur trois mécanismes particuliers : les plans, les buts et les thèmes. Les plans permettent de définir grâce à un ensemble de buts et de sous-but, comment arriver au résultat souhaité. La figure 2.5 nous donne un exemple de plan pour réaliser le but "se nourrir".

INTENTION	:	se nourrir
BUT	:	trouver de la nourriture
BUT	:	aller à l'endroit où elle se trouve
BUT	:	se l'approprier
ACTION	:	préparer la nourriture
ACTION	:	la manger

Figure 2.5. Plan pour se nourrir.

Les thèmes nous permettront de connaître certains buts implicites, par exemple le thème de la survie contiendra le but implicite : "se nourrir" qui pourra être réalisé grâce au plan défini ci-dessus.

Il existe trois sortes de thèmes :

- les thèmes généraux qui correspondent aux souhaits de l'être humain moyen dans le contexte du récit (ex : être heureux, être riche ...)
- les thèmes interpersonnels (ex : l'amitié, l'amour, l'autorité)
- les thèmes de rôles (ex : avocat, professeur ...).

Il existe aussi des liens entre les scripts et les plans, en effet, les plans peuvent être utilisés dans des scripts (pour prendre en compte les événements exceptionnels) et les plans peuvent faire appel à des scripts (ex : préparer la nourriture dans le plan "se nourrir").

Pour réaliser un but général, un plan peut faire appel à deux types de sous-but :

- les sous-but spécifiques qui sont des séquences d'actions simples dépendant de l'objet sur lequel elles portent (ex : ouvrir(x) sera réalisé différemment si x est un livre, une porte ou une boîte de conserves)
- les sous-but généraux indépendants des objets sur lesquels ils portent, ils caractérisent des changements d'états, ils sont en nombre restreint, Schank [Schank-77] a défini les 5 primitives données par la figure 2.6, cependant un des problèmes non résolus est que l'on n'est pas capable de définir l'ensemble des primitives nécessaires et suffisantes.

- | | |
|---|---------------------------|
| - | Savoir (x) |
| - | Contrôler (x) |
| - | Etre proche (x) |
| - | Contrôler socialement (x) |
| - | Intermédiaire (x) |

Figure 2.6. Les primitives servant à définir des sous-buts généraux

A chaque sous-but primitif est associé un ensemble de réalisations possibles, une réalisation étant caractérisée par :

- son nom,
- l'action correspondante,
- la liste des préconditions contrôlables,
- la liste des préconditions incontrôlables,
- la liste des préconditions intermédiaires et
- les résultats de la réalisation.

la figure 2.7 nous donne un exemple de réalisation possible pour le but Savoir(x).

-Nom :	Demande
-action :	A demande x à B
-conditions	
contrôlables :	A est proche de B
incontrôlables :	B sait x
intermédiaires :	B veut dire x à A
-Résultats :	B dit x à A, A sait x

Figure 2.7. Réalisation possible de Savoir (x)

Nous avons pu remarquer qu'il existe plusieurs représentations possibles pour les connaissances pragmatiques et que loin d'être concurrentes, elles semblent au contraire très bien se compléter et reflètent des mécanismes de compréhension existant chez l'être humain. Une compréhension complète d'un texte nécessite l'intégration de tous ces mécanismes, cependant, malgré tous les efforts qui peuvent être faits, l'interprétation ne pourra être générale et universelle, elle ne pourra s'appliquer qu'à certains types de problèmes et à un niveau de compréhension prédéfini.

Par exemple, les plans d'action, dans un système de commandes de robot, peuvent se combiner aisément avec les scénarios. Les plans permettent, dans ce cas, d'atteindre les buts qui constituent les déclencheurs des scénarios. Nous avons vu dans le paragraphe précédent que l'action "ouvrir une porte" peut être décrite à partir d'un scénario et que le déclencheur de cette action est l'état dans lequel doit se trouver le robot pour la réaliser c'est-à-dire "être près de la porte". Pour atteindre ce but, il lui faut appliquer un plan général "aller près de la porte" qui ne peut pas être considéré comme un scénario car sa réalisation dépend fortement de l'état du monde (la position des objets, celle du robot ...) Winograd [Winograd-71] a utilisé les plans dans un système de commande de robots dans un univers restreint de manipulation des cubes.

2.3.5. Extensions

Des travaux plus récents [Lehnert-83] tendent à définir des structures de représentation des connaissances moins lourdes que les scénarios et dont les traitements associés sont moins indépendants des connaissances que les plans. Les MOP's (Memory Organization Packets) intègrent les buts et les intentions des personnages dans des situations données et séparent les connaissances en sous-ensembles plus facilement manipulables. La figure 2.8 donne l'exemple du MOP "emprunt".

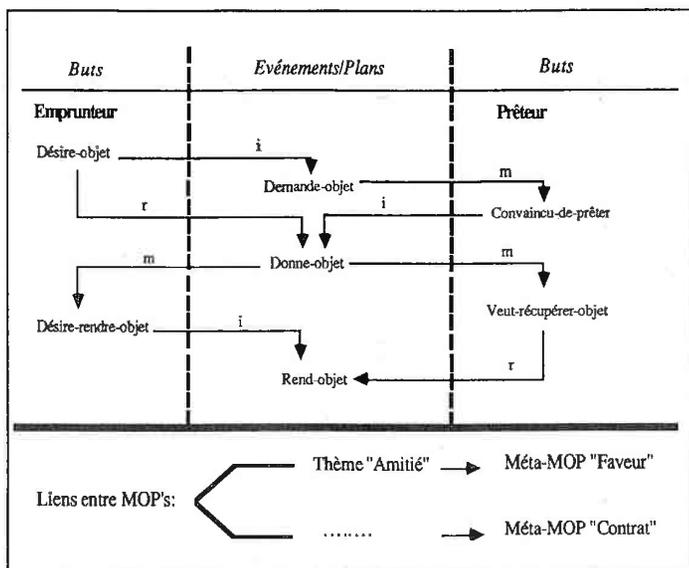


Figure 2.8. Exemple du MOP "emprunt" D'après [Lehnert-83]

Il existe trois types de liens (i, m et r) représentant les Intentions, les Motivations et les Réalisations entre les buts des personnages. Les Méta-MOP's ont la même structure que les MOP's, cependant ils sont plus généraux donc ils contiennent des informations plus vagues.

Pour rendre compte des aspects émotionnels, une structure comparable aux MOP's est utilisée, il s'agit des TAU's (Thematic Affect Units). Elles permettent de rendre compte des événements imprévus et de décrire les réactions affectives des personnages face à ces événements en fonction de leur point de vue de la situation (ex : victime ou voleur, malade ou médecin ...). Cette représentation des connaissances est très intéressante pour la compréhension de textes ou de dialogues entre des interlocuteurs humains ou dans un système de dialogue homme-machine "général", cependant dans un système de commande de robot cela paraît beaucoup trop riche car l'ensemble des connaissances ainsi représentées joue un rôle très faible dans le genre d'application qui nous intéresse.

2.4. Les sources de connaissance

Nous avons vu dans le paragraphe 2.2.2 les différents rôles que doit remplir le module de dialogue. Au regard de ces fonctionnalités, nous pouvons définir l'ensemble des connaissances nécessaires à la bonne réalisation de chacune d'entre elles.

- La gestion du canal de communication

Elle nécessite des connaissances d'ordre acoustico-phonétiques afin de fournir des renseignements sur les caractéristiques du signal dans le but de permettre au module de dialogue de diagnostiquer au mieux les causes de mauvaise reconnaissance de l'énoncé (ex : parlez plus fort, moins vite...).

- L'interprétation de l'énoncé

Comme nous l'avons vu, elle se décompose en plusieurs étapes et chacune d'elles nécessite ses propres sources de connaissance.

• La reconnaissance de la phrase à partir du signal

Il s'agit de trouver l'ensemble des mots qui la composent et les liens qui les unissent. Les premières connaissances mises en œuvre sont les connaissances acoustico-phonétiques qui transforment le signal en une suite de sons élémentaires (les phonèmes). Après ce premier traitement, interviennent les connaissances syntaxiques et lexicales (détermination de l'ordre des mots) et les connaissances sémantiques (relations entre les mots et les groupes).

Ces différentes connaissances interagissent les unes avec les autres afin de construire une représentation syntaxico-sémantique de l'énoncé.

• La résolution des ellipses et des anaphores pour lesquelles une représentation de l'historique du dialogue est nécessaire pour y rechercher à l'aide des connaissances syntaxico-sémantiques les référents et les éléments manquants. Lorsque ces connaissances s'avèrent insuffisantes, il convient d'introduire des connaissances pragmatiques sur le monde et la tâche.

• La détermination du type de l'énoncé peut utiliser des connaissances prosodiques (détection des interrogatives), des connaissances

syntaxiques (la position des constituants pouvant fournir certains renseignements sur la nature de la phrase ex : si le verbe est en tête de phrase, il s'agit sans doute d'un ordre) et enfin le modèle du dialogue et de la tâche.

• La compréhension du sens de l'énoncé se fait en fonction du contexte, donc à partir de l'univers de la tâche, des connaissances pragmatiques et de l'historique du dialogue.

- La gestion du dialogue, quant à elle se fait principalement à l'aide du modèle et de l'historique du dialogue.

- Le raisonnement repose sur le modèle de la tâche, celui de l'univers, l'historique du dialogue et une représentation des solutions partielles ou temporaires. Le modèle du locuteur peut être utile pour déterminer ses droits d'accès dans le cas d'une base de connaissance par exemple et la limite de ses pouvoirs dans le cas de commande de robot.

- La prédiction nécessite essentiellement le modèle du dialogue, des connaissances syntaxico-sémantiques et lexicales.

Les mêmes sources de connaissances peuvent intervenir à plusieurs niveaux il est donc difficile de les classer de manière hiérarchique en fonction de l'ordre dans lequel elles interviennent. Cependant, on peut remarquer qu'elles se décomposent en deux classes distinctes :

- les sources de connaissances statiques (ou prédéfinies), fortement dépendantes de la tâche et de l'univers, qui ne sont pas modifiées au cours des échanges,
- les sources de connaissances dynamiques (ou évolutives) qui elles évoluent au cours du dialogue.

2.4.1. Les sources de connaissances statiques

Parmi les sources de connaissances statiques, on peut citer :

- le modèle de la tâche représenté par des connaissances sémantico-pragmatiques (valeurs par défaut, plans, scénarios,...),

- la définition du langage qui intègre les connaissances acoustico-phonétiques, syntaxico-sémantiques et lexicales,
- le modèle du dialogue qui utilise la définition du langage et des stratégies de contrôle et de gestion du dialogue,
- le modèle du locuteur tant au niveau acoustico-phonétique que pragmatique.

2.4.2. Les sources de connaissances dynamiques

Les sources de connaissances dynamiques permettent de rendre compte de l'état général de la tâche et du dialogue, elles correspondent à :

- l'univers de la tâche qui indique l'état de celle-ci après chaque échange entre l'homme et la machine,
- l'historique qui contient l'ensemble des messages échangés,
- les solutions partielles qui, une fois complétées, permettent d'atteindre le but décrit par l'utilisateur.

2.5. Quelques réalisations

Dans les premiers systèmes de communication, le module de dialogue servait uniquement à démontrer la compréhension d'une phrase afin de valider les résultats du module de reconnaissance. Le but des recherches était la compréhension de phrases, mais aucune étude du dialogue lui-même n'était faite. La fin des années 70 a vu apparaître des systèmes intégrant un module de dialogue en langage naturel (Gus [Bobrow-77], Mycin [Shortliffe-78]). Le but des ces dialogues était en fait de mettre en valeur une méthode de représentation des connaissances.

Puis apparurent des études sur le dialogue lui-même. Les systèmes réalisés alors avaient pour but de s'adapter au locuteur plutôt que de l'obliger à se plier aux caractéristiques du système [Allen-80], [Vilnat-85], [Roussanly-88].

Dans ce paragraphe, nous allons présenter dans un premier temps des systèmes de communication classiques (oraux ou écrits) appartenant aussi bien à chacune des trois catégories citées ci-dessus, et dans un deuxième temps, nous présenterons des systèmes que nous pouvons qualifier de multi-média car ils font intervenir un mode de communication supplémentaire comme la vision.

2.5.1. Systèmes de communication classiques

Un des problèmes qui a suscité de nombreux travaux est la résolution d'ellipses et d'anaphores.

- En 1975, le système SRI-DC [Walker-78] apporte une solution à l'interprétation de certaines formes d'ellipses, comme par exemple :

- les américains ont-ils des porte-avions?

- et les russes?

ou - et des sous-marins?

La représentation de l'univers d'application est faite à l'aide d'un vaste réseau sémantique .

- CADI : Constructeur Automatique de Dialogue Intelligent [Nouhen-81] se consacre à la résolution d'ellipses et d'anaphores simples. Sa limite principale est de ne tenir compte que d'énoncés syntaxiquement corrects, cependant il gère un dialogue à initiative mixte et fait des prédictions vers le module de reconnaissance.

- Le système présenté dans [Dinard-87] concernant l'apprentissage de l'allemand, se concentre sur la résolution d'ellipses grammaticales et extra-grammaticales ce qui permet de prendre en compte des énoncés syntaxiquement faux.

D'autres équipes de recherche ont axé leurs travaux vers l'interprétation des énoncés et en particulier elles ont essayé de déterminer le type d'un énoncé, (rechercher par exemple s'il porte sur la tâche ou le dialogue s'il est absurde, incohérent ou faux) et les conséquences des différents types d'énoncés sur les "actions" associées :

- Dans [Veronis-87], l'utilisation d'une extension de la représentation objet (la logique multi-sorte) permet de traiter différemment des énoncés faux et des énoncés absurdes dans un contexte de raisonnement géométrique ce qui permet de donner à l'élève un diagnostic précis de ses erreurs.

- [Sadek-87] nous présente un système d'interrogation orale de bases de données. Le dialogue est géré par un module paramétrable, indépendant

de la tâche et servant d'intermédiaire entre l'utilisateur et l'application. Il utilise des mécanismes de génération de plans (scénarios, frames...) Les actions concernent aussi bien la tâche que le dialogue. La représentation de l'historique se fait à l'aide d'espaces de croyances qui apportent une originalité par rapport à [Bunt-80] qui est la prévision des actes de contrôle du dialogue (ACD) grâce à ces espaces de croyances.

- Le système d'interrogation sur les pages jaunes de l'annuaire [Vilnat-85] utilise la notion de suivi de thème, il permet de faire la différence entre une déviation et un changement de thème. Il intègre un module d'auto-justification et une représentation dynamique de l'utilisateur, lui permettant de s'adapter au niveau d'explication requis par celui-ci.

- [Hayes-83] nous présente un système de gestion de fichier en langage naturel faisant la distinction entre énoncés intra-phrasiques et extra-phrasiques. Le point étudié ici concerne les phrases de correction (ou contestation). En fonction du contexte, une phrase de contestation peut entraîner des réactions différentes de la part du système. Par exemple, si elle concerne une correction, elle va entraîner une modification de la tâche à exécuter, mais si c'est une annulation, il faudra faire un retour arrière si la commande a déjà été exécutée de manière à revenir à l'état initial.

D'autres systèmes plus généraux tendent vers la réalisation d'une tâche (sans pour autant négliger tous les aspects vus ci-dessus) :

- Le système GUS [Bobrow-77], est un système de réservation de places d'avions qui ne gère pas explicitement le dialogue, le locuteur peut dire ce qu'il veut, mais dans les interventions ne sont pris en compte que les renseignements que le système recherche. Ceci est dû à l'utilisation de frames stéréotypes. Le but du dialogue étant de remplir les stéréotypes, il suit une certaine trame dirigée vers ce but.

- Le système DIAL développé au CRIN est un système de renseignements administratifs tels qu'on peut les trouver dans les pages roses de l'annuaire, il utilise les grammaires de cas et un module de raisonnement à base de règles lui permettant de détecter des incohérences dans les propos du locuteur.

Une autre catégorie de systèmes que l'on peut qualifier de systèmes cognitifs cherchent à valider une structure de représentation des connaissances. Le système le plus connu est :

- BORIS [Lehnert-83] qui est un système de compréhension d'histoires portant sur le divorce. Boris est capable de répondre à des questions portant sur le texte qui lui est fourni. Sa compréhension du texte se fait au niveau le plus profond. Il utilise dans sa représentations des connaissances des MOP's, des Méta-MOP's et des TAU's (voir § 2.2.3.5.) qui sont stockées dans une mémoire narrative. Cette interprétation en profondeur nécessite une grande quantité d'informations, c'est pourquoi ce système fonctionne sur de petites histoires simples portant sur le divorce et ceci dans un contexte très particulier, mais est difficilement généralisable. Boris peut être considéré comme un modèle cognitif, car ses structures de connaissances reflètent des théories psychologiques, ces structures sont issues d'études expérimentales sur le comportement de l'homme lors de la compréhension d'un texte, et elles ont suscité un grand intérêt chez les psychologues.

2.5.2. Systèmes multi-média

Certains systèmes de compréhension de textes ou de gestion de dialogues utilisent la vision comme support de connaissance :

- NAOS [Neumann-84] est un système d'analyse visuelle d'une scène de rue dans laquelle se trouvent des bâtiments, des passants et des véhicules... qui dialogue avec une personne connaissant la rue mais ne voyant pas la scène qui s'y déroule. Le but de ce système est de raconter ce qui se passe dans la rue et de répondre aux questions de l'utilisateur. Ce système intègre la notion de temps (avant, après...), il utilise une représentation casuelle des connaissances et des scénarios pour représenter les événements.

-[Norihiro-81] nous présente l'intégration de la vision dans un système de compréhension de textes. Un texte illustré par une suite d'images doit être interprété, le support visuel permet de résoudre certaines ambiguïtés

de la langue naturelle, de compléter l'information et de vérifier la compréhension du texte. Les images ne seront pas entièrement analysées, seuls les objets cités dans le texte feront l'objet d'une recherche active dans l'image. Les connaissances sur le monde sont représentées par un réseau sémantique à structure hiérarchique (les noeuds représentent les objets et les liens représentent les relations sémantiques entre les objets). L'analyseur syntaxique utilise les ATN, il construit des arbres représentant toutes les possibilités d'imbrication des groupes, l'analyse de l'image permet de choisir l'arbre le plus probable.

Les systèmes de compréhension du langage naturel sont un apport non négligeable à la robotique. Actuellement peu de travaux ont été réalisés dans ce domaine :

- Le système RAPT [Corner-83], [Yin-83] est une première approche de ce type d'application, il permet de programmer un robot assembleur. La tâche est représentée par un réseau de relations dans lequel les noeuds correspondent à la position des objets et les arcs représentent les relations entre ces positions. Un ensemble de règles de réécriture permet de simplifier le réseau. On fournit au robot la description des positions des objets, c'est-à-dire la position de départ, les positions intermédiaires et la position finale de manière à ce qu'il puisse déterminer comment faire le montage. Les différentes étapes sont interprétées grâce au langage RAPT dans lequel, chaque objet est décrit physiquement (faces planes, trous...), les relations entre les objets sont décrites à chaque étape et un moteur d'inférences permet de déduire les étapes intermédiaires nécessaires pour passer d'une position à une autre, sa tâche consiste à simplifier le réseau de relations. Bien que la description des positions ne se fasse pas en langage naturel, il s'agit ici d'une interface plus naturelle, donc plus conviviale que la programmation classique. L'intégration d'un module de vision permet la vérification des informations fournies par l'intermédiaire de RAPT.

2.6. Dialogue avec un robot

Nous pouvons distinguer deux types de systèmes de dialogue entre un homme et une machine : les systèmes portant sur une tâche non évolutive (par exemple l'interrogation d'une base de connaissances), et ceux portant sur une tâche évolutive (par exemple la commande de robot). Nous nous plaçons ici dans ce deuxième cas, la machine est un robot dont l'existence a pour but principal l'exécution des ordres donnés par l'être humain.

Nous allons reprendre point par point les fonctionnalités du module de dialogue et étudier quelles sont les particularités et restrictions engendrées par le type de l'application.

- L'interprétation de l'énoncé

Les grands principes de la reconnaissance de phrases ne changent pas, cependant le langage de commande étant très spécifique, les règles syntaxico-sémantiques sont en nombre plus réduit que dans un système d'interprétation d'énoncés généraux. Les phrases sont construites à partir des prédicats portant sur la tâche, et la spécificité de ceux-ci entraîne des constructions fortement typées tant au niveau syntaxique qu'au niveau sémantique.

La résolution des ellipses se trouve simplifiée dans la mesure où celles-ci sont construites à partir des énoncés précédents, les ellipses dues aux figures de style disparaissent car l'univers de la tâche est relativement simple et surtout très concret. La résolution des anaphores reste un problème entier, la seule simplification de ce problème réside dans le fait que les connaissances pragmatiques sont en nombre restreint comparé à une application générale, donc de ce fait elles sont facilement représentables dans leur ensemble.

La détermination du type de la phrase, du fait de la restriction des règles syntaxico-sémantiques impliquée par le type de communication, se réduit à l'identification de phrases interrogatives, impératives ou informatives lorsqu'elles portent sur la tâche et à des énoncés de confirmation, d'interrogation ou de contestation lorsqu'elles portent sur le dialogue.

La compréhension du sens de l'énoncé, quant à elle, est grandement simplifiée, en effet, pour chaque commande, il suffit de reconnaître le prédicat et les objets concernés, et la correspondance entre la forme syntaxico-sémantique et le sens est univoque. Dans ce type d'application, on se rapproche de la compréhension de langages artificiels où chaque énoncé a une réalisation spécifique. Ici, les énoncés

peuvent prendre de nombreuses formes, mais dès que les constituants sont identifiés, le sens est connu.

- La gestion du dialogue et du canal de communication

La spécificité de l'application n'entraîne aucune modification des modules de gestion du dialogue et du canal de communication.

- Le raisonnement

Dans le cas de commande de robot, le raisonnement va servir, d'une part, à vérifier la validité de la demande de l'utilisateur (par exemple vérifier si un prédicat peut s'appliquer à un objet, ou si la commande ne va pas à l'encontre du but fixé, et d'autre part, à établir des plans d'actions (à partir de plans ou de scénarios) pour réaliser le but du locuteur, chaque commande pouvant être considérée comme un but à atteindre.

- Les prédictions

Les différents types de prédictions possibles sont en nombre limité. Elles concernent :

- les commandes
- les confirmations ou contestations éventuelles
- les réponses aux questions posées par le système

Chapitre 3

Cahier des charges

Dans ce chapitre, nous allons présenter les motivations de ce travail ainsi que les différents types d'applications envisageables, puis nous parlerons des restrictions que nous avons apportées pour que de tels systèmes soient réalisables, dans ce but nous préciserons les limites du dialogue et les compétences que doit posséder le système, ensuite nous aborderons le problème de la généralisation du système, c'est-à-dire sa transportabilité vers d'autres domaines d'application et enfin, nous détaillerons les diverses sources de connaissances dont dispose un robot et celles qui lui sont nécessaires pour gérer correctement sa tâche.

3.1. Motivations et buts

En 1984, sous l'initiative d'Olivier Faugeras, six laboratoires se sont associés pour définir un projet commun : le développement d'un système de vision pour un robot mobile se déplaçant dans un univers d'intérieur, le projet ORASIS [Mohr-88]. Les six laboratoires sont le CERFIA de Toulouse, le LIFIA de Grenoble, les équipes de vision de l'INRIA Rocquencourt et Sophia-Antipolis et le CRIN/INRIA de Nancy [Thirion-89] et l'IRISA de Rennes. Cette action est intégrée dans le pôle vision du GRECO-PRC "Communication Homme-Machine". L'idée de départ qui a guidé notre travail était de fournir à ce robot une interface de communication permettant de lui demander de réaliser quelques actions simples sur l'univers dans lequel il évolue.

La connaissance que possède le robot de son environnement est obtenue par l'analyse d'images à partir d'un modèle théorique du monde. Chaque objet est identifié grâce à ses caractéristiques et si le robot utilise un identifiant pour chaque objet (par exemple : cube23 représente le cube qui a été reconnu et qui a pour caractéristiques d'être petit et rouge), ce nom reste inconnu de l'utilisateur. Pour communiquer avec le robot, l'utilisateur se verra contraint de donner le nom du concept représentant l'objet (par exemple le cube) et si cela s'avère insuffisant, il pourra indiquer les caractéristiques de celui-ci pour le différencier des autres (par

exemple le petit cube rouge). Cette contrainte suffit en elle-même à justifier l'utilisation de la langue naturelle. En effet, un langage de commande est a priori assez pauvre sémantiquement aussi bien que syntaxiquement, on pourrait donc considérer comme suffisant l'utilisation d'un langage artificiel à syntaxe prédéfinie. Cependant, comme nous l'avons vu, l'impossibilité dans ce type d'application d'identifier les objets par un nom individuel interdit ce choix simplificateur.

L'identification des objets se fait donc par des expressions et non des noms individuels, les problèmes qui en découlent sont:

- la reconnaissance et l'interprétation d'expressions complexes et
- la résolution d'ambiguïtés lorsque plusieurs objets de même type possèdent des caractéristiques communes et que la description faite ne permet pas de les différencier.

L'utilisation de la langue naturelle pour décrire les objets de l'univers puis, par extension, les commandes, entraîne aussi l'apparition de tous les problèmes dus aux caractéristiques du langage naturel que nous avons vus dans le chapitre précédent, c'est-à-dire l'analyse d'énoncés elliptiques, de références pronominales et en particulier les risques de violation de la syntaxe relativement rigide du langage de commande.

Une autre conséquence directe de l'utilisation de la langue naturelle, liée aux problèmes d'ambiguïtés, est la nécessité d'introduire un méta-langage, c'est-à-dire un langage portant sur le dialogue et non pas sur la tâche, de manière à pouvoir traiter par exemple : les demandes de confirmation ou d'informations complémentaires, les contestations... Cela entraîne une gestion du dialogue en tant que tel et un des buts sera de permettre ce dialogue et non pas de le diriger. L'utilisateur devra pouvoir prendre l'initiative (ce qui est primordial dans un système de commande!), le robot ne devra intervenir que s'il ne peut pas exécuter une commande qui lui est demandée par manque de renseignements ou suite à une ambiguïté ou à une incohérence par exemple.

3.1.1. Intégration de la reconnaissance d'images

La reconnaissance d'images est un problème difficile et encore non résolu de manière satisfaisante sur des images complexes (univers de bureau par exemple). Dans notre système, les résultats de la reconnaissance seront donc simulés, cependant, on pourrait envisager d'interfacer le module de dialogue et le module

de vision, ce qui permettrait au robot de poser des questions à l'utilisateur pendant la phase de reconnaissance des objets. Dans cette optique, nous nous approchons du raisonnement humain dans lequel l'interprétation en général, qu'elle soit d'images, de sons, de mouvements... fait intervenir le maximum de connaissances disponibles permettant d'aboutir à la bonne identification. Le fait que dans ce cas le robot pourrait poser des questions en cours d'interprétation permet d'envisager un apprentissage de la part de la machine, ce qui correspond à un axe de recherches très important dans le domaine de l'intelligence artificielle.

3.1.2. Exemples d'applications

Sans aller jusqu'à ce niveau d'intégration, nous pouvons trouver des applications plus simples directement envisageables pour ce type de langage de commande.

On peut par exemple imaginer une maison entièrement robotisée que l'on pourrait commander à distance par téléphone. On pourrait lui demander d'allumer la télévision, d'ouvrir et de fermer les volets pour faire croire à une présence alors que les occupants sont en vacances, ou encore d'allumer le chauffage juste avant de rentrer et il serait très agréable de pouvoir demander:

" Allume le radiateur du salon " plutôt que " Allumer radiateur 2 "

Dans ce type d'application, si la connaissance d'une grande partie de l'univers peut être prédéfinie, un système de vision permettra au robot de localiser les obstacles pouvant gêner ses déplacements ou les objets qu'il doit manipuler et qui sont susceptibles de changer de place. Les problèmes dus aux ambiguïtés sont grandement simplifiés car la plupart des objets commandés (four, télévision, chauffage...) s'ils ne sont pas forcément uniques sont descriptibles à l'aide d'expressions relativement simples, de plus les commandes associées sont simples et ne permettent qu'une seule interprétation dépendante de l'objet sur lequel elles portent. Ce type d'application est un parfait exemple pour lequel un langage artificiel serait largement suffisant, cependant, lui donner une dimension naturelle la met à la portée d'un public plus large, le public susceptible d'être intéressé par ce produit.

Une maison robotisée telle que nous l'avons décrite, pourrait être aussi envisagée comme un environnement mis à la disposition de personnes

handicapées qui pourraient ainsi acquérir une certaine autonomie et un confort amélioré.

Un autre type d'application, ne nécessitant pas de module de vision est la gestion d'un répondeur téléphonique interrogeable à distance auquel on pourrait demander de passer des messages, d'en effacer certains ...

L'intérêt d'intégrer un système de vision à un robot est de pouvoir lui laisser une grande autonomie, ce qui est indispensable quand l'univers dans lequel il évolue est inaccessible à l'homme. Un tel univers pourrait être l'enceinte d'une centrale nucléaire où des robots commandés à distance pourraient intervenir dans des zones contaminées, un autre contexte possible est celui de l'exploration spatiale où un robot peut être amené à évoluer sur une planète inconnue. On peut envisager aussi, un véhicule entièrement autonome capable de résoudre seul les problèmes dus à la circulation (présence d'autres véhicules ou de piétons, feux de signalisation ...) et obéissant au conducteur dont les ordres seraient du type : tourne à droite après le bâtiment jaune, arrête-toi le long du trottoir ...

Si l'on reprend l'exemple de la centrale nucléaire, le robot sera très utile en cas d'incidents, cependant dans ce cas la situation peut être très dangereuse, et l'urgence de l'intervention et la panique qui peuvent en découler, peuvent avoir des conséquences importantes sur l'élocution du "pilote". C'est pourquoi l'intégration du langage naturel et par conséquent de la liberté d'expression qui en découle permet au robot d'interpréter des ordres dont la construction syntaxique peut s'avérer très particulière. Cependant, il faut aussi tenir compte que dans de telles circonstances, la rapidité d'exécution d'une commande est primordiale, et si l'on gagne du temps par le simple fait d'accepter des énoncés dont la construction syntaxique est "folklorique", la perte de temps associée à l'analyse de telles phrases reste très importante, il faudra donc avoir comme souci permanent l'optimisation du temps de réponse.

Si comme nous l'avons vu, nous autorisons des constructions syntaxiques assez "floues", nous n'en sommes pas moins obligés de définir les limites des énoncés acceptables par le système. Ceci fait l'objet du paragraphe suivant.

3.2. Limites du sous-langage

Comme nous l'avons vu dans le paragraphe précédent, un langage de type commande, comparé au langage naturel, est très pauvre tant au point de vue syntaxique que sémantique. Cependant la composante naturelle que nous désirons y introduire entraîne une plus grande variabilité syntaxique, et pour mener à bien l'analyse d'un tel sous-langage, il faut déterminer quelles sont les contraintes qui restent imposées malgré la liberté apparente.

Une première contrainte sera de n'autoriser que des phrases ne comportant qu'un seul prédicat, c'est-à-dire dans le cas d'un langage de type commande, des phrases ne correspondant qu'à un seul ordre.

Ex : *Ouvre la porte*

Pour obtenir une description naturelle des objets, il faut être capable d'accepter des expressions de type groupe nominal décrivant un objet avec ses principales caractéristiques :

Ex : *la petite armoire*

Cependant, nous avons vu que les caractéristiques d'un objet peuvent s'avérer insuffisantes pour le définir complètement, il peut donc être nécessaire de compléter la description de l'objet par sa localisation dans l'univers. Il est bien entendu inconcevable d'utiliser ses coordonnées dans l'espace, il doit être localisé de manière relative, par rapport aux autres objets. Pour cela, il faut accepter des groupes prépositionnels ou relatifs dans la description :

Ex : *le livre sur le bureau*

Par souci de limiter les contraintes, toutes les formes de coordination et d'imbrication sont possibles à un niveau quelconque de profondeur bien qu'il soit évident que l'imbrication des groupes nominaux et des groupes prépositionnels ne peut dépasser une certaine limite sans entraîner une incompréhension totale :

Ex : *Le livre sur le bureau et près de la lampe
sur le dossier à-côté du livre...*

Ceci permet, d'une part de tenir compte des objets dont la discrimination nécessite une description complexe et, d'autre part, de prendre en compte des

expressions tarabiscotées qui peuvent paraître claires à celui qui les énonce et très obscures à celui qui les entend, mais l'analyseur étant capable de construire toutes les combinaisons possibles et d'en extraire les seules plausibles, il ne sera pas gêné et pourra poursuivre le traitement.

Nous avons dit, au début de ce paragraphe, que nous n'autorisons que les phrases ne comportant qu'une seule commande, cependant, nous élargissons cette contrainte en acceptant que cette commande puisse porter sur plusieurs objets :

Ex: *- Prends le livre et le cahier sur le bureau*
- Prends le livre sur le bureau et le cahier sur la table

Par contre, nous n'autoriserons pas des énoncés tels que :

Ex: *Pose le livre sur le bureau et le cahier sur la table*

On peut s'étonner de la similitude entre les deux derniers exemples et le fait que l'un des deux est un énoncé autorisé et l'autre pas. La justification de ce choix tient dans le fait que dans l'énoncé accepté, la commande PRENDRE s'applique à deux objets différents : le livre et le cahier, tandis que dans l'énoncé suivant, nous sommes en présence de deux commandes différentes : POSER un objet sur un bureau et POSER un objet sur une table! Dans le premier cas les groupes prépositionnels ("sur le bureau" et "sur la table") ne servent qu'à décrire les objets alors que dans le second cas, ils modifient la commande (ce sont des qualificatifs de la commande).

Une contrainte générale que l'on peut en déduire est que chaque qualificatif (ou cas, voir chapitre suivant), ne peut être instancié qu'une seule fois, soit par des groupes simples soit par des groupes coordonnés ou imbriqués.

Nous avons donc décrit la forme des commandes "complètes" qui sont autorisées dans notre application, cependant, si nous voulons que la caractéristique naturelle soit respectée, il faut aussi tenir compte des énoncés elliptiques, des références anaphoriques... Voici donc quelques exemples d'énoncés que l'on pourra rencontrer :

Ex: *- Prends le livre sur le bureau*
- Pose le sur la table

- Prends le livre sur le bureau
- Et le cahier sur la table

- Ouvre le tiroir du bureau
- Mets le livre dedans

Ces exemples montrent des cas de références et d'ellipses se rapportant à l'énoncé précédent, mais on peut aussi envisager des cas plus complexes où le référent se trouve dans l'énoncé courant:

Ex: *- Prends le classeur noir et le rouge*
- Prends le petit et le grand classeur

...

Il faut aussi envisager deux types de discours en fonction du comportement du robot. Si celui-ci demande confirmation avant d'exécuter une commande, la continuité du dialogue est rompue par un sous-dialogue de confirmation-réponse et cela entraîne donc la disparition de la plupart des énoncés elliptiques et des références au niveau du dialogue principal, mais ils resteront présents dans le sous-dialogue, alors que dans le cas où le robot ne demande pas de confirmation, le dialogue est plus continu et la caractéristique naturelle est mieux respectée. Les figures 3.1 et 3.2 nous donnent un exemple de chaque type de dialogue.

U: Prends le livre rouge.
S: Est-ce que je dois prendre le livre rouge (livre15) ?
U: Oui, c'est bien ça.
S: C'est fait, à vous.
U: Prends le cahier.

Figure 3.1. Exemple de dialogue avec confirmation systématique

U: Prends le livre rouge.
S: C'est fait.
U: et le cahier maintenant

Figure 3.2. Exemple de dialogue sans confirmation

3.3. Compétences du système

La commande du robot s'effectuant de manière orale et en langage naturel, certains énoncés peuvent subir des altérations les rendant syntaxiquement incorrects mais restant cependant tout à fait interprétables. L'analyseur doit être capable de retrouver le sens de tels énoncés, c'est pourquoi il doit disposer d'une grammaire souple autorisant des écarts par rapport aux formes grammaticales de référence. Le robot doit pouvoir comprendre chacun des énoncés suivants:

Ex: - *Ferme la porte*
- *La porte, ferme-la*
- *Le livre, pose le sur la table*
- *Sur la table, pose le livre*
- *Poser livre sur table*
- ...

Il doit être capable de comprendre tout énoncé porteur de sens (sémantiquement correct) même si le support syntaxique est quant-à lui totalement incorrect, cela lui permet d'interpréter des phrases classiques du langage naturel aussi bien que des phrases tenant davantage du langage artificiel.

Nous avons vu dans le paragraphe précédent qu'une référence peut se rapporter à un élément de la phrase ou d'une phrase précédente. Il faut cependant remarquer que tous les référents ne jouent pas le même rôle. Prenons deux exemples :

Ex1: - *Prends le livre*
- *Pose le*
Ex2: - *Prends le livre rouge*
- *et le noir*

Dans le premier exemple, le pronom "le" se rapporte à l'objet concerné par la phrase précédente, c'est-à-dire à une occurrence bien particulière associée au concept LIVRE. Dans le second exemple le groupe "le noir" se rapporte au concept LIVRE évoqué dans l'énoncé précédent.

Au moment de l'interprétation d'une phrase contenant des références, il faut être capable de définir si on est en présence d'une référence à un concept ou d'une référence à une occurrence d'objet. Nous verrons dans le chapitre 4 comment nous avons tenté de résoudre ce problème.

Un autre but de cette application sera de faire réagir le robot en cas d'ambiguïtés ou d'inconsistance dans les propos tenus. Dans le cas d'ambiguïtés, le robot ne doit pas seulement indiquer qu'il y a un problème, mais il doit donner au locuteur les moyens de le résoudre le plus simplement possible (figure 3.3).

U: Prends le livre sur le bureau
S: Le livre fermé ?
U: non, l'autre

Figure 3.3. Exemple de question posée par le système

Le robot doit être capable de trouver des caractéristiques discriminantes entre les objets sur lesquels porte le conflit, de plus pour limiter les problèmes dus à la reconnaissance de phrases, il est amené à faire un choix arbitraire entre les différents objets possibles dans l'attente d'une confirmation ou d'une contestation.

Dans certains cas les caractéristiques propres des objets ne suffisent pas à les différencier les uns des autres, il faut donc que le robot pousse plus loin son raisonnement de manière à être capable de décrire l'un des objets de manière unique (figure 3.4).

U: Prends le livre
S: Le livre rouge sur le bureau?
U: Oui

Figure 3.4. Exemple de description d'un objet par le système

Dans d'autres cas, pour lever l'ambiguïté sur les objets cités, il peut être suffisant de lever l'ambiguïté sur les objets servant à leur description (figure 3.5).

U: Prends le livre sur le bureau
S: Sur le bureau qui est près de la fenêtre?
U: Oui

Figure 3.5. Exemple de demande de précision

Dans le cas d'inconsistance, le robot doit être capable de préciser sur quelle partie de la phrase elle porte (figures 3.6, 3.7, 3.8) :

U: Prends le livre sur le bureau
S: Il n'y a pas de livre sur le bureau

Figure 3.6. Exemple d'objet inconsistant

U: Allume le livre
S: On ne peut pas allumer un livre

Figure 3.7. Exemple d'inconsistance de la commande

U: Ferme la porte
S: La porte est déjà fermée

Figure 3.8. Exemple d'inconsistance de l'état de l'objet

En vue d'optimiser le traitement de reconnaissance des phrases, le robot doit être capable de faire des prédictions quant aux énoncés à venir. Le module de prédiction fournit au module de reconnaissance de phrases des hypothèses concernant la structure syntactico-sémantique de la phrase attendue, ou encore des hypothèses sur le sous-lexique envisagé.

3.4. Indépendance entre connaissance et raisonnement

Nous avons vu dans le paragraphe 3.1.2 que les applications possibles d'un système de commande de robot peuvent être variées tout en respectant la spécificité du système. On peut envisager deux classes d'applications, celles concernant un même robot évoluant dans plusieurs univers différents et celles concernant des robots totalement différents (ex: robot ménager et robot dans une centrale).

Le système, pour être portable d'une application à une autre doit être construit de manière modulaire, il faut séparer les connaissances particulières et dépendantes des applications des traitements généraux applicables quelle que soit l'application choisie.

La source de connaissance caractérisant l'application est le lexique. En effet, c'est lui qui contient les mots représentatifs (les noms des objets manipulables, les actions possibles sur ces objets...). En fait, ce n'est pas le lexique complet qui caractérise l'application, car il existe des sous-lexiques communs à toutes les applications, ce sont les lexiques dits grammaticaux, ils contiennent tous les mots outils du sous-langage étudié (par exemple les prépositions, les conjonctions, les articles ...). Le sous-lexique spécifique de l'application contiendra les informations syntaxiques, sémantiques et pragmatiques.

Le modèle du monde est lui aussi très lié à l'application, cependant, celui-ci concerne davantage les étapes antérieures et postérieures au dialogue, c'est-à-dire la reconnaissance des objets par le module de vision et les actions associées aux commandes.

Pour permettre un passage simple d'une application à une autre, il pourra être envisagé de créer un constructeur automatique de lexique, permettant à une personne connaissant l'application de définir un nouvel univers s'adaptant à un autre domaine. Ceci n'est possible que pour la partie spécifiquement dialogue, l'adaptation à un autre domaine de module de vision et de l'interface de manipulation des objets nécessite un traitement important qui dépasse le cadre de nos objectifs.

Pour des applications différentes concernant le même robot, seuls les objets sur lesquels portent les actions vont changer. Le robot étant construit pour réaliser certaines tâches particulières, celles-ci ne subiront pas de modification d'une application à l'autre. Par exemple dans un univers de bureau, on pourra demander au robot d'ouvrir ou de fermer des portes alors que dans un univers de centrale nucléaire on lui demandera d'effectuer ces opérations sur des sas, même si l'action diffère légèrement en fonction de l'objet sur lequel elle porte, le principe de base reste le même.

3.5. Les principales sources de connaissance statiques

Comme nous l'avons vu, une partie des connaissances que possède le robot est simulée pour palier aux imperfections des systèmes actuels et se placer ainsi dans un cadre où l'expérimentation est possible. En effet, il ne paraît pas très intéressant de vouloir dialoguer avec un robot muni d'un système de vision si celui-ci ne sait reconnaître que des lignes dans son univers. Nous allons donc décrire les diverses sources de connaissance dont dispose le robot.

- Le modèle de la tâche

Il est composé de l'univers et des informations pragmatiques valides sur celui-ci. L'univers est créé par le module de vision qui nécessite la

définition d'un modèle de l'univers qui permet au robot de reconnaître les objets de son environnement. Le modèle de l'univers correspond au monde théorique où chaque objet est décrit par le concept qui lui est associé ainsi que par les propriétés qu'il doit posséder et les valeurs qu'elles peuvent prendre (ex: état --> allumé / éteint ou état --> ouvert : fermé...). La représentation de l'univers dans une application de type commande de robot est une source de connaissance dynamique car elle est appelée à être modifiée. nous la classons dans les sources de connaissances statiques car, le système de vision étant simulé, l'univers est défini au début de l'application, de plus les actions étant simulées l'état de l'univers ne change pas. L'univers comprend la description de tous les objets du monde où évolue le robot. chaque objet est décrit par le concept auquel il est rattaché et par les propriétés qui lui sont propres, ainsi que par sa position relative par rapport aux autres objets.

Les informations pragmatiques sont nécessaires à la bonne interprétation des énoncés et à la réalisation des commandes imposées par le locuteur. Par exemple, pour un objet, il précise quels sont les prédicats qui peuvent lui être appliqués, les conditions nécessaires à sa réalisation. ...

Ex: *A un objet de type "porte", on peut associer le prédicat "fermer" et une condition nécessaire pour pouvoir l'appliquer est que la propriété "état" de la porte considérée ait la valeur: "ouvert"*

- Le modèle de la langue

Il est composé des connaissances sur la langue et d'un ensemble d'analyseurs permettant de les exploiter. Les informations sur la langue définissent les constructions grammaticales autorisées, grâce à des connaissances syntaxico-sémantiques associées aux mots et à l'utilisation d'une grammaire du langage. Ces informations définissent aussi la somme des mots utilisés, c'est-à-dire leur représentation phonétique de manière à pouvoir les chercher dans l'énoncé. Les analyseurs utilisant ces connaissances sont :

- l'analyseur acoustico-phonétique qui permet de transformer le signal émis par le locuteur en une suite de phonèmes. Dans notre système, cet analyseur est simulé par un programme de transcription graphème-phonème ;

- l'analyseur lexical qui effectue la comparaison entre l'énoncé et la représentation phonétique d'un mot et donne en résultat le score associé à la reconnaissance ;

- l'analyseur syntaxico-sémantique qui construit les phrases à l'aide des mots reconnus par l'analyseur lexical et de la grammaire du langage.

- le modèle du dialogue

C'est lui qui détermine le comportement du système dans les diverses situations de communication rencontrées. Il permet de relancer le locuteur, il émet des messages de mise en attente, il gère les échanges et décide de l'abandon d'une interprétation lorsque les chances de réussite de l'analyse sont trop faibles, il développe des stratégies pour optimiser les échanges, la principale stratégie étant de pousser le locuteur à une confirmation de préférence à une contestation.

Nous avons rapidement présenté les diverses sources de connaissance de notre système qui seront développées aux chapitre 4, 5 et 6.

Chapitre 4 Présentation générale du système

Dans ce chapitre, nous allons présenter les différentes sources de connaissances "statiques" et la manière dont nous les avons représentées pour réaliser notre système puis nous présenterons l'organisation générale du système.

4.1. Modèle de la tâche

Le robot évolue dans un univers physique dont la connaissance est censée être fournie par le module de vision. Les connaissances sur l'univers sont de deux types:

- le modèle de l'univers,
- l'état de l'univers.

4.1.1. Le modèle de l'univers

Pour un robot disposant d'un module de vision, le modèle de l'univers est très complexe, il doit contenir les descriptions physiques des objets en termes de points, de segments, de surfaces etc... Pour le système de dialogue associé, une grande partie de ces connaissances est inutile. En effet, le système de dialogue se contente de raisonner sur les occurrences des objets et leurs propriétés. Le modèle qui l'intéresse consiste essentiellement en une description "linguistique" des objets de l'univers.

Comme nous l'avons déjà signalé, les systèmes de vision, à l'heure actuelle, ne fournissent pas encore ce type de résultats, c'est pourquoi nous avons décidé de fournir au robot ces connaissances de manière statique tout en les organisant de façon à pouvoir éventuellement y intégrer des appels à des modules de vision.

Pour chaque objet de l'univers, nous avons construit un prototype contenant une description de l'objet utile à un système de dialogue. A chaque classe d'objet est associée une liste de propriétés pour lesquelles un ensemble de valeurs

possibles est donné. Pour une propriété donnée, un objet peut posséder plusieurs valeurs (ex : couleur --> rouge et noir) à condition que les valeurs ne s'excluent pas mutuellement (ex : état --> ouvert et fermé). Pour tenir compte des cas d'exclusions, les valeurs possibles sont regroupées dans plusieurs sous-listes, une propriété ne pouvant posséder qu'une seule valeur de chaque sous-liste.

Les figures 4.1, 4.2 et 4.3 nous donnent respectivement les prototypes des objets de type "table", "porte" et "lampe".

Objet	:	table
Propriétés	:	taille ((petit, grand))
		forme ((carré, rectangulaire, ovale, rond))
		couleur ((noir), (brun), (rouge), (blanc)...)
	

figure 4.1. Prototype de l'objet "table"

Objet	:	porte
Propriétés	:	taille ((petit, grand))
		couleur ((noir), (gris), (bleu) ...)
		état ((ouvert, fermé))
	

Figure 4.2. Prototype de l'objet "porte"

Objet	:	lampe
Propriétés	:	taille ((petit, grand))
		couleur ((rouge), (gris), (blanc) ...)
		état ((allumé, éteint))
	

figure 4.3. Prototype de l'objet "lampe"

Le lien avec le système de vision peut être fait en définissant pour chaque propriété physique un serveur (cf Chapitre 2) faisant appel au module de vision pour l'évaluer lorsque sa valeur n'est pas connue et qu'elle est nécessaire.

Une interaction forte entre le système de dialogue et celui de vision permettrait de rendre le comportement du robot plus naturel. Dans l'état actuel du système, nous supposons que le robot connaît entièrement l'univers où il se trouve, ce qui présuppose qu'il en a fait une analyse globale donc très longue avant le dialogue. Si les deux systèmes sont couplés, le robot n'a pas besoin d'analyser globalement l'univers, il lui suffit de rechercher les objets dont on lui parle. Ceci peut permettre de résoudre simplement certains problèmes d'ambiguïtés que nous verrons au chapitre 6. En effet, si le locuteur possède le même champ de vision que le robot et qu'il lui demande d'exécuter une action, par exemple : Ferme la porte, le robot va commencer par chercher une porte dans son champ de vision, et s'il la trouve, il la fermera ; c'est seulement en cas d'échec qu'il cherchera une autre porte. Dans notre système nous n'avons pas cette notion de champ de vision, toutes les portes vont être traitées simultanément d'où l'apparition d'éventuelles ambiguïtés "artificielles".

4.1.2. Représentation de l'univers

Comme nous l'avons vu, dans un souci de simplification pour simuler le système de vision, nous avons donné au robot la description complète de l'univers dans lequel il évolue. Cet univers se décompose en deux parties:

- les occurrences des objets,
- les positions relatives des objets les uns par rapport aux autres.

4.1.2.1. Occurrences des objets

A chaque prototype d'objet est associée une liste d'occurrences représentant tous les objets de cette classe figurant dans l'univers. Chaque objet est identifié par une étiquette affectée par le robot et inconnue de l'utilisateur. La définition d'une occurrence est faite à partir du nom du concept et des valeurs des propriétés de l'objet. Le robot génère dans la liste des occurrences de l'objet concerné une occurrence de cet objet, c'est-à-dire un identifiant et construit la liste de ses propriétés. La figure 4.4 nous donne un exemple de définition d'un objet et de l'occurrence créée.

```

Définition      :   porte noire ouverte
occurrence     :   { Portel15
                    Propriétés :
                    - couleur  : noir
                    - état    : ouvert}

```

Figure 4.4. Création d'une occurrence

Les propriétés étant supposées définies par le système de vision à partir du modèle du monde, elles sont a priori correctes dans le sens où les valeurs sont choisies parmi un ensemble de valeurs possibles ; il est donc nécessaire, dans le cas d'une simulation du système de vision par une intervention humaine, de contrôler la validité des données c'est-à-dire vérifier que la valeur définie s'applique bien à une propriété de l'objet et qu'aucune propriété n'est évaluée par deux valeurs s'excluant mutuellement. La figure 4.5 nous donne un exemple de définition non valide et l'occurrence qui est créée en conséquence : lorsqu'une propriété ne s'applique pas, elle est tout simplement ignorée, et lorsque deux valeurs concurrentes sont fournies, on conserve uniquement la dernière.

```

Définition      :   porte rouge ouverte allumée fermée
occurrence     :   { portel26
                    propriétés :
                    couleur   : rouge
                    état     : fermé}

```

Figure 4.5. Exemple de définition invalide

La description de l'univers est donc faite à partir d'une suite de définitions d'objets par l'intermédiaire de leur nom et de la liste de leurs propriétés ; le système construit automatiquement la base de connaissances représentant

l'univers en rangeant les occurrences dans les listes adéquates. Chaque nouvelle définition correspond à la création d'une nouvelle occurrence, il est donc nécessaire de donner toutes les propriétés de l'objet lors de sa définition. La figure 4.6 montre un extrait du contenu d'un fichier décrivant un univers de bureau.

```

( bureau gris grand )
( fenêtre petite ouverte )
( fenêtre petite fermée )
( bureau gris grand )
( chaise noire )
( chaise noire )
( chaise rouge )
( table ronde grise noire)
( porte ouverte )
.....

```

Figure 4.6. Extrait d'un fichier représentant l'univers

4.1.2.2. Positions des objets

Notre système doit accepter des descriptions d'objets à partir de leurs caractéristiques mais aussi de leur position relative dans l'univers. Les positions sont un résultat du module de vision, donc pour les mêmes raisons que nous avons vues ci-dessus, cette partie est simulée et conservée intégralement en mémoire. Pour l'acquisition de ces connaissances, le concepteur doit connaître la codification interne des objets, en effet, les propriétés définies lors de la création des occurrences ne suffisent pas à discriminer tous les objets, il est donc nécessaire de les identifier par une étiquette.

Cependant il est impossible de créer toutes les relations qui lient les positions des objets, c'est pourquoi nous avons choisi de prendre la plus restrictive (ex : devant) sachant que si celle-ci est vérifiée, une position compatible mais moins restrictive sera elle aussi valide (ex : près de).

La figure 4.7. nous donne un exemple de la description des relations de position entre les objets et la figure 4.8 nous indique, pour quelques prépositions marquant une relation de position, l'ensemble des prépositions valides qui en sont déduites.

(près-de	bureau115	fenêtre116)
(devant	chaise119	bureau115)
(devant	chaise120	bureau118)
(près-de	table122	chaise121)
.....				

Figure 4.7. Exemple de définition des positions

devant	---->	près-de	à-côté-de
près-de	---->	à-côté-de	
à-côté-de	---->	près-de	
sur	---->	∅	
sous	---->	∅	
derrière	---->	près-de	à-côté-de
.....			

Figure 4.8. Classes de validité des prépositions

4.1.3. Informations pragmatiques

Le modèle de la tâche contient les informations pragmatiques permettant :

- de vérifier la pertinence de la commande,
- d'indiquer comment réaliser cette commande.

La vérification de la pertinence consiste à tester d'une part si la commande à effectuer s'applique bien au type de l'objet indiqué,

Ex : ouvre la porte ----> commande pertinente

allume la chaise ----> commande non pertinente
 et d'autre part si elle peut s'appliquer à l'occurrence de l'objet,
 Ex : ferme la porte ----> ne s'applique que si la porte est ouverte!

La réalisation d'une commande est quant-à elle un problème très lié avec les phénomènes traités en robotique. Pour notre système, nous nous contentons de vérifier la pertinence d'une commande et de la mettre sous forme "exécutable", c'est-à-dire que nous isolons le prédicat et nous évaluons ses arguments. Pour la réalisation des commandes, il suffit de définir des plans d'actions généraux (cf Chapitre 2 § 2.3.4) de manière à atteindre un état permettant l'exécution de la commande et ensuite d'appliquer le scénario correspondant. L'action à effectuer peut être éventuellement décomposée en plusieurs actions.

Ex : pose le livre sur la table
 ---> aller près du livre (but)
 prendre le livre (scénario)
 aller près de la table (but)
 poser le livre (scénario)

Les plans sont décrits de manière à faire évoluer le robot dans l'espace en cherchant le meilleur chemin et en évitant les éventuels obstacles. Comme pour la connaissance de l'univers, les actions du robot sont très liées au système de vision, c'est pourquoi nous ne les avons pas implémentées.

la vérification de la pertinence de la commande, par contre, doit être traitée dans le système de dialogue de manière à pouvoir fournir à l'utilisateur un diagnostic en cas d'impossibilité d'application.

Ex : Allumer ne s'applique pas aux objets de type table
 La porte est déjà fermée ...

Il est donc important d'organiser les connaissances pragmatiques nécessaires de manière à garder un maximum de compatibilité avec le module d'exécution de commandes.

Pour qu'une commande soit applicable, elle doit vérifier les deux conditions suivantes :

- elle s'applique au type de l'objet fourni en argument,
- l'objet vérifie les préconditions nécessaires à l'application de la commande.

Il y a donc trois types d'informations pragmatiques à conserver :

- l'ensemble des objets sur lesquels une commande est applicable,
- les préconditions requises,
- les effets obtenus sur l'objet manipulé.

Les informations pragmatiques seront liées au prédicat. Pour chaque prédicat on définit :

- la liste des objets sur lesquels il s'applique,
- les préconditions que doivent vérifier les objets,
- le scénario de l'action, composé de :
 - le (ou les) état(s) de déclenchement,
 - la suite d'actions à effectuer,
 - l'état de l'objet après l'action.

Dans notre système, comme nous l'avons dit ci-dessus, il n'y a pas de réalisation effective du scénario, pour la simulation il consistera simplement à énoncer l'état de déclenchement, l'action demandée après évaluation de ses arguments et à modifier l'état de l'objet. Pour une commodité d'utilisation, en particulier pour la résolution d'ellipses du prédicat, une partie des informations va être dupliquée. Pour chaque objet de l'univers on va indiquer l'ensemble des prédicats qui lui sont associés. La figure 4.9 donne l'ensemble des connaissances pragmatiques associées au prédicat "prendre".

Prédicat :	Prendre
Pragma :	
- lobj :	(livre, cahier, crayon, chaise, ...)
- précond :	état (obj) = posé
- scénario :	
déclencheur :	'je vais près de ' obj
actions :	commande instanciée
résultats :	état (obj) <- pris

Figure 4.9. Connaissances pragmatiques associées au prédicat "prendre"

4.2. Modèle de la langue

Pour qu'un système de commande de robot soit efficace, l'objectif principal est de véhiculer le plus fidèlement possible le sens de la phrase de l'interlocuteur vers le robot. Le plus important est donc de trouver la valeur sémantique de la phrase plutôt que d'en évaluer sa grammaticalité. Nous avons présenté dans le chapitre 2 les différentes approches de la langue naturelle, il nous appartient donc de choisir une représentation en fonction des caractéristiques de ces approches et du cahier des charges.

La caractéristique naturelle que nous voulons donner au système interdit l'utilisation de grammaires formelles purement syntaxiques trop rigides. La volonté d'adaptabilité du système à d'autres applications nous permet de rejeter les grammaires sémantiques qui bien que très faciles à utiliser dans notre type d'application, sont très peu portables, ce qui va à l'encontre de nos objectifs. Nous envisageons donc une approche sémantique permettant de laisser une grande liberté à l'interlocuteur. Dans les paragraphes suivants, nous allons voir les différentes étapes qui nous ont permis de définir une représentation syntactico-sémantique adaptée à nos besoins.

4.2.1. Choix de la grammaire

Dans un langage de type "commande", le verbe est l'élément important de la phrase, c'est lui qui délimite un environnement possible, il fournit la structure logique profonde de l'énoncé. Le verbe est porteur d'informations sémantiques et la compréhension de la phrase est fondée sur la reconnaissance des unités sémantiques associées au verbe. Un langage de commande peut être caractérisé par un prédicat et l'ensemble de ses arguments. Les grammaires de cas [Fillmore-68] ont une structure très proche, chaque cas pouvant représenter un argument du prédicat.

EX: Pour le verbe DONNER, on peut définir les cas suivants:

- l'agent : celui qui donne,
- le bénéficiaire : celui qui reçoit,
- l'objet : ce qui est donné,
- ...

Contrairement aux applications de compréhension de textes généraux pour lesquels l'ensemble de cas minimum ne peut pas être défini, pour une application

de type commande ou utilisant un langage opératif [Deville-87], il est facile de déterminer l'ensemble des cas nécessaires. Nous avons donc défini un ensemble de cas que nous considérons comme suffisant pour notre type d'application (car ils permettent de mettre en évidence tous les types de problèmes rencontrés dans la désignation d'objets). Les cas que nous utilisons sont les suivants :

- AGENT ex : **Pierre** pose le livre sur la table
- OBJET ex : Prends **le livre**
- DESTINATION ex : Pose le livre **sur la table**
- SOURCE ex : Sors le livre **du tiroir**
- BENEFICIAIRE ex : Donne le livre à **Paul**
- MANIERE ex : Pose **doucement** le vase sur la table

Nous pouvons déjà remarquer que tous les cas n'ont pas la même priorité dans une phrase. Par exemple, si un cas objet est absent d'une phrase, celle-ci risque d'être incompréhensible, par contre, si c'est un cas manière qui manque, cela ne nuira pas trop à l'interprétation. En effet, certains cas sont obligatoires pour la compréhension alors que d'autres sont facultatifs. Nous pouvons distinguer trois types de cas en fonction de leur priorité :

- **les cas obligatoires** : ce sont les cas qui doivent figurer explicitement dans l'énoncé sous une forme quelconque.

Ex : le cas OBJET

- Prends **le livre**
- ...Pose-le

- **les cas facultatifs** : ce sont les cas qui ne sont pas indispensables à l'interprétation et qui apportent des informations supplémentaires.

Ex : le cas MANIERE

- Pose **doucement** le vase sur la table

- **les cas semi-obligatoires** : ce sont les cas qui ne sont pas forcément présents dans l'énoncé, mais dont l'instanciation est nécessaire à l'interprétation. Dans certaines circonstances, l'absence d'un de ces cas sera palliée par une valeur par défaut (par exemple le cas agent d'une

phrase impérative), dans d'autres cas, il faudra faire une recherche dans l'historique du dialogue pour résoudre l'ellipse :

Ex : le cas DESTINATION

- Pose le livre **sur le bureau**
- ... et le cahier

Comprendre le sens de la phrase consiste à retrouver dans l'énoncé l'ensemble des cas nécessaires associés au verbe même si leur position ne respecte pas une syntaxe rigoureuse. La recherche des cas dans la phrase à partir du verbe permet d'interpréter les phrases suivantes de la même façon :

- *Donne le livre à Pierre*
- *A Pierre donne le livre*
- *Le livre à Pierre donne-le*

Il est évident que des cas comme OBJET ou AGENT vont se retrouver dans la plupart des prédicats que nous traitons, c'est pourquoi nous avons essayé de classer les prédicats en fonction des cas qui leur sont associés, nous obtenons ainsi un arbre des cas décrit dans le paragraphe suivant.

4.2.2. Arbre des cas

Dans un premier temps, nous allons uniquement nous intéresser aux prédicats, c'est-à-dire aux verbes correspondant à une commande dans l'application, et dans un deuxième temps, nous intégrerons ces prédicats dans l'ensemble des verbes qui peuvent apparaître au cours d'un échange.

Certains cas vont apparaître dans tous les prédicats, d'autres dans une partie d'entre eux seulement, c'est pour cette raison que nous avons établi notre classification sous forme d'un arbre, chaque nœud représentant une classe de prédicats avec ses propres cas et héritant des cas de la classe englobante. L'arbre des cas que nous avons défini a été construit à partir des prédicats que nous avons jugés utiles pour notre application. Cependant, il n'y a aucun problème pour rajouter un verbe dans une classe déjà définie, ni pour créer une nouvelle classe de prédicats regroupant un ensemble de cas non défini. Nous pouvons envisager de manière simple un générateur d'arbre des cas pour simplifier la tâche lors d'un changement d'application nécessitant l'intégration de nouvelles classes.

Nous avons déterminé 7 classes de prédicats :

- actes simples (ACTS) ex : prends le livre
- actes de déplacement (DEPL) ex : avance vers la fenêtre
- possessifs (POSS) ex : le livre appartient à Pierre
- actes avec destination (ACTB) ex : pose le livre sur le bureau
- actes avec source (ACTSR) ex : retire la clé de la serrure
- actes d'échanges (ACTECH) ex : donne moi le stylo rouge
- actes buts et sources (ACTBSR) ex : tourne le bouton de la gauche vers la droite

La figure 4.10 nous montre quels sont les cas associés à ces classes et comment nous avons pu établir une hiérarchie entre elles. Pour chaque cas nous avons indiqué son code de priorité : 'f' pour facultatif, 'o' pour obligatoire et 's' pour semi-obligatoire.

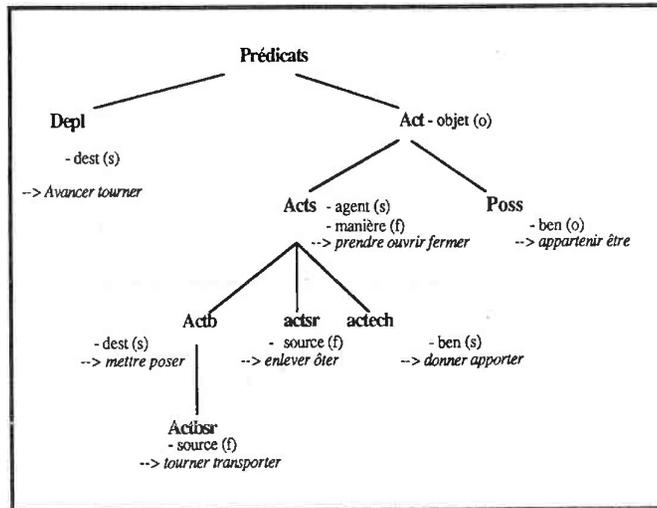


Figure 4.10. Arbre des cas des prédicats

Outre les prédicats concernant directement l'application, il existe d'autres verbes de type plus général, portant sur le dialogue ou purement descriptifs. (ex : être, dire, faire...). L'ensemble de tous ces verbes peut être réparti en 4 grandes classes :

- les prédicats,
- les copules (ex : être, faire...)
- les auxiliaires (ex : être, aller, pouvoir ...)
- les interrogatifs (ex : dire, demande...)

Pour les verbes autres que les prédicats, en plus des cas, nous définissons des contraintes syntaxiques telles que les propositions infinitives, transitives ... qui elles contiennent des prédicats.

La figure 4.11 nous donne une vue d'ensemble sur toutes les classes des verbes et décrit l'ensemble des cas des verbes qui ne sont pas de type prédicat.

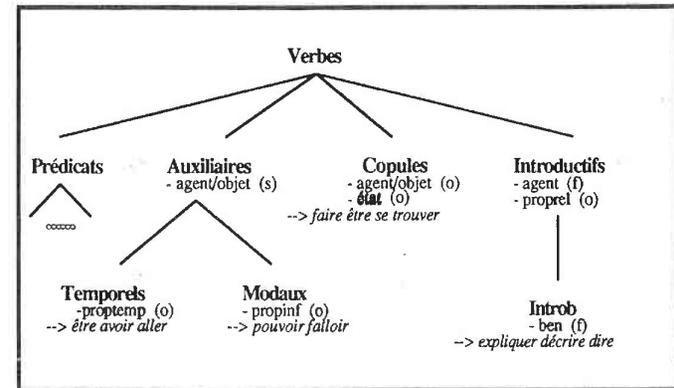


Figure 4.11. Arbre des cas et des contraintes syntaxiques des verbes

Nous avons vu que le verbe pilote la reconnaissance, c'est pour cela qu'il fait l'objet d'une recherche active dans l'énoncé. L'étape qui suit la recherche du verbe est celle de l'instanciation des cas. Pour ce faire, nous avons attaché à chaque cas un ensemble de grammaires susceptibles de le représenter. Ces grammaires que nous appelons "grammaires locales" (par opposition aux grammaires générales) font l'objet du paragraphe suivant.

4.2.3. Les grammaires locales

A chaque cas défini ci-dessus, nous avons attribué un ensemble de grammaires locales susceptibles de le représenter. Prenons un exemple :

Le cas OBJET peut être instancié de diverses manières :

- *Prends le livre rouge*
- *prends le noir*
- *Pose-le*

Chacun de ces trois groupes fait référence à un objet. Soit les codifications suivantes :

art	: article
adja	: adjectif antéposé
adjp	: adjectif postposé
nom	: nom
adjn	: adjectif numérique
pronp	: pronom personnel
+	: indique un élément obligatoirement présent
*	: indique un élément pouvant être répété.

Le premier groupe est un groupe nominal (GN) décrivant complètement l'objet. Une grammaire locale pour un GN peut être :

GN --> art adjn adja nom⁺ adjp*

- Ex : - *la petite table noire*
 - *les deux chaises*
 - *fenêtre*

Le deuxième groupe (le noir) peut être considéré comme un GN incomplet, en effet la forme complète en est par exemple : *le livre noir*. Ce groupe fait donc

référence à un GN, il en est une forme dérivée dont le nom est une ellipse. On peut caractériser ce groupe par :

GNR --> art adjn adj⁺ adj*

Ex : - *la petite*

- *la grande bleue*

Le troisième groupe (le) fait lui aussi référence à un groupe nominal et plus particulièrement à l'occurrence instanciant ce groupe, un tel groupe peut être décrit sous la forme :

GNREF --> pronp⁺ / art adjn⁺

Ex : *le, la, les deux, trois*

Nous pouvons donc distinguer deux types de références :

- les références à un concept,
- les références à une occurrence

dont la différence apparaît nettement au niveau syntaxique. Nous avons pris comme convention de définir deux types de grammaires dérivées d'une grammaire complète, la première suffixée par 'R' (ex : GNR, GPR) faisant référence de manière générale à un concept, et l'autre suffixée par 'REF' (ex : GNREF, GPREF) qui fait référence à une occurrence particulière. A titre d'exemple la figure 4.12 nous montre les grammaires locales des GN et des GP ainsi que celles de leurs dérivées.

Nous pouvons remarquer que les grammaires locales que nous avons définies décrivent la structuration des éléments importants de l'énoncé mais ne fournissent pas une description syntaxique complète (cf les relatives dans la figure 4.12.). En annexe on trouvera la description complète des grammaires locales. Les grammaires locales servent à décrire les cas, il est donc nécessaire que la grammaire tienne compte de la sémantique associée au cas. Par exemple si l'on recherche un cas objet sous la forme d'un groupe nominal, il faudra rechercher uniquement des noms d'objets. C'est pourquoi, pour chaque grammaire locale nous avons défini l'élément porteur de la sémantique (suivi d'un 's' dans la description de la grammaire). Pour pouvoir sélectionner les mots en fonction de leur valeur sémantique, il faut que le lexique possède une structure adaptée. La description complète du lexique sera faite au paragraphe 4.5.1.

GN	--->	art adja nom ^{+s} adjp*	Ex :	la petite table noire *
GNR	--->	art adj ⁺ adj* / prond ^{+s} rel ⁺ GP / prond ^{+s} rel ⁺ adj ⁺ adj*	Ex :	la petite blanche celle qui est sur la table celle qui est rouge
GNREF	--->	promp ⁺ / art adjn ⁺	Ex :	le, il , les deux
GP	--->	prep ^{+s} GN ⁺	Ex :	sur la petite table rouge
GPR	--->	prep ^{+s} GNR ⁺	Ex :	sur la petite
GPREF	--->	adv ^{+s}	Ex :	dessus
avec :				
GN	:	groupe nominal	GP	: groupe prépositionnel
GNR	:	forme dérivée de GN faisant référence à un concept	GPR	: forme dérivée de GP faisant référence à un concept
GNREF	:	forme dérivée de GN faisant référence à une occurrence	GPREF	: forme dérivée de GP faisant référence à une occurrence
art	:	article	rel	: pronom relatif
adj	:	adjectif	adv	: adverbe
nom	:	nom	+	: indique une présence obligatoire
prep	:	préposition	*	: indique un élément pouvant être répété
promp	:	pronom personnel	adja	: adjectif se plaçant avant un nom
prond	:	pronom démonstratif	adjp	: adjectif se plaçant après un nom
adjn	:	adjectif numérique	/	: ou
s	:	élément porteur de la sémantique	pronind	: pronom indéfini

Figure 4.12. Les grammaires locales

Les grammaires locales étant définies, il nous reste à décider pour chaque cas lesquelles d'entre elles peuvent l'instancier. La figure 4.13 nous donne cas par cas l'ensemble des grammaires associées et les valeurs sémantiques devant s'y retrouver.

OBJET	:	GN (<i>objet</i>) / GNR (<i>objet</i>) / GNREF(<i>objet</i>) ...
AGENT	:	GN (<i>agent</i>) / GNREF (<i>agent</i>) / GP (<i>agent, agent</i>)...
SOURCE	:	GP (<i>source, objet</i>) / GPR (<i>source, objet</i>) / GPREF (<i>source</i>) ...
DESTINATION	:	GP (<i>dest, objet</i>) / GPR (<i>dest, objet</i>) / GPREF (<i>dest</i>) ...
BENEFICIAIRE	:	GP (<i>ben, agent</i>) / GPR (<i>ben, agent</i>) / GNREF (<i>ben</i>) ...
MANIERE	:	GADV (<i>manière</i>) ...

Figure 4.13. Les grammaires locales associées aux cas

Nous avons présenté les cas utilisés dans notre grammaire ainsi que les formes que peuvent prendre les groupes les instanciant. Il nous reste à définir maintenant l'articulation entre tous ces éléments nous permettant d'obtenir des descriptions complètes d'objets.

4.2.4. Construction de groupes complexes

Nous avons vu dans le cahier des charges que nous interdisions les commandes comportant plusieurs prédicats. La nécessité d'autoriser la coordination de groupes à l'intérieur d'un cas pour décrire complètement celui-ci (ex : pose le livre sur le bureau et à côté de la lampe) allège la contrainte sur l'unicité du prédicat, en effet nous pouvons avoir des énoncés comportant une conjonction de coordination à l'intérieur du cas objet (ex : prends le livre et le cahier sur la table) qui est une fusion de deux commandes ayant le même prédicat. Une commande peut donc porter sur plusieurs objets, il faut noter cependant que les coordinations ne sont effectuées que sur des groupes voisins, c'est pourquoi l'analyseur reconnaîtra la commande : *Pose le livre et le classeur sur le bureau à côté de la lampe*, mais ne reconnaîtra pas : *Pose le livre sur le bureau et le classeur à côté de la lampe*. Dans le premier cas, il s'agit, comme nous l'avons dit ci-dessus, d'une fusion de deux commandes alors que dans le second cas, il s'agit

de la coordination de deux commandes, la deuxième comportant une ellipse du prédicat.

En résumé, nous autorisons les qualifications et les coordinations de groupes lorsqu'elles sont internes à un cas. Il est à noter que la coordination de groupes peut faire apparaître un phénomène de factorisation que l'on peut résoudre au niveau syntaxique.

Ex : - Prends le classeur noir et le rouge
- Prends le petit et le grand classeur

Les deux paragraphes suivants présentent la coordination de groupes ainsi que les règles qui la régissent, et la qualification de groupes.

4.2.4.1. Coordination de groupes

L'utilisation des grammaires de cas nous permet de prendre en compte de manière simple certains phénomènes linguistiques tels que la coordination de groupes. Les règles régissant la coordination de groupes sont liées d'une part à des contraintes syntaxiques, et d'autre part à des contraintes sémantiques mises en évidence par la structuration des éléments de la phrase en cas sémantiques. Deux groupes ne peuvent être coordonnés que s'ils sont susceptibles de jouer le même rôle sémantique, donc d'instancier le même cas. Le cas le plus simple que l'on peut rencontrer est celui de la coordination de deux groupes de même type et de même rôle sémantique :

Ex : *Le livre rouge et le petit cahier*

Livre et cahier sont deux noms représentant des objets, les deux groupes sont des GN, il n'y a donc aucun problème pour les coordonner.

De plus l'étape de coordination permet de résoudre syntaxiquement certaines références dues au phénomène de factorisation vue ci-dessus :

Ex : *Le petit livre rouge et le noir*
--> le petit livre rouge et le livre noir
Près de la fenêtre et de la table
--> près de la fenêtre et près de la table

Ceci nous permet de simplifier l'étape d'interprétation de l'énoncé. Cependant, toutes les coordinations ne peuvent pas se faire aveuglément, par exemple il est impossible de coordonner les deux groupes suivants :

Ex : * *Le livre et le noir*

Pour chaque couple de groupes susceptibles d'être coordonnés, nous avons défini un ensemble de règles syntaxiques et sémantiques que doivent respecter ces groupes, ainsi qu'un ensemble de transformations qu'ils subissent lors de la coordination. La figure 4.14 présente un sous-ensemble significatif de ces règles.

GN + GN :	conditions	--->	même rôle sémantique
	transformations	--->	-
	Ex :		<i>le livre et le classeur</i>
GN + GNR :	conditions	--->	le GN et le GNR contiennent un qualificatif caractéristiques d'une même propriété ou le GN contient un groupe imbriqué
			les deux groupes ont le même genre
	transformations	--->	transformer le GNR en groupe nominal en lui ajoutant le nom du GN
	Ex :		- <i>le livre rouge et le bleu</i> - <i>le livre qui est sur la table et le rouge</i>
GP + GP :	conditions	--->	même rôle sémantique
	transformations	--->	si la préposition du 2 ^{ème} GP est une forme contractée de celle du 1 ^{er} GP alors remplacer la 2 ^{ème} par la 1 ^{ère}
	Ex :		- <i>sur la table et près de la lampe</i> - <i>à côté de la porte et du radiateur</i>

Figure 4.14. Exemples de règles de coordination

4.2.4.2. Qualification de groupes

Lorsqu'un objet ne possède pas de caractéristiques permettant de l'identifier, il peut être nécessaire de compléter sa description par sa localisation dans l'univers.

Ex : *Le bureau gris près de la fenêtre*

Tout comme pour la coordination de groupes, la qualification respecte certaines règles syntaxiques :

- les GP ne peuvent qualifier que des GN, des GNR, des GP ou des GPR,
- les GPR ne peuvent qualifier que des GN, des GNR ou des GP,
- les GP ne peuvent qualifier que des groupes placés devant eux,
- un GP ne peut qualifier un groupe duquel il est séparé par un groupe d'un type autre que GP, exemples :

le livre bleu sur la table et le cahier sur le bureau ← qualification incorrecte

le livre bleu sur la table et le cahier sur le bureau ← qualification correcte

sauf si ce groupe n'est pas qualifié et est coordonné avec celui faisant office de séparateur, dans ce cas, le GP qualifie simultanément les deux groupes, il est mis en facteur.

le livre et le cahier sur le bureau

Dans ce cas particulier, deux solutions sont envisagées, le groupe est mis en facteur et il qualifie simultanément les deux GN, ou le groupe ne qualifie que le dernier GN.

L'analyseur syntactico-sémantique doit prendre en compte les différentes formes d'imbrication et de coordination possibles, et ce sont les informations pragmatiques qui permettront de déterminer la bonne solution, si elle existe et si elle est unique sinon il faudra recourir à une question pour résoudre le problème.

4.3. Les analyseurs du modèle de la langue

4.3.1. Analyseur acoustico-phonétique

Dans le chapitre 1, nous avons présenté le système de D.A.P. APHODEX développé au CRIN. Il aurait été intéressant de connecter notre système de

dialogue à APHODEX, cependant les temps de réponses relativement longs de celui-ci et les problèmes de connexion entre les différentes machines nous ont conduits à simuler l'étape de décodage tout en respectant ses principales caractéristiques et limites. Ce choix nous permet de traiter du texte écrit comportant des fautes, en effet, en donnant à l'analyseur la transcription phonétique des mots, nous évitons le problème de la détection de fautes d'orthographe ou d'accords. Notre système est capable d'interpréter de la même manière : *Prends le livre rouge* et *Prends leu livres rouge*. En général les fautes d'orthographe n'influent que très peu sur la forme phonétique du mot. On trouvera par exemple pour le mot téléphone : *tailéphone, téléfonne, téléléphone...* mais rarement *tétéphone*, ce type d'erreur étant essentiellement dû à des fautes de frappe.

Nous avons utilisé un programme de transcription graphème-phonème. Il est cependant impossible d'écrire un programme ne tenant compte que de la représentation graphique du mot et fournissant sa représentation phonétique, en effet, pour une même représentation lexicale deux mots peuvent avoir des réalisations phonétiques différentes.

Ex : *Sur une carte géographique l'est est à droite*
Au couvent les poules couvent

La prononciation d'un mot dépend de son sens et on se rend compte qu'un programme de transcription graphème-phonème nécessite de nombreuses sources de connaissances et que c'est un problème qui n'est pas trivial. Notre transcripateur ne traite pas ces cas particuliers, c'est pourquoi la phrase qui lui est fournie doit tenir compte de ce type de caractéristiques. Pour les deux phrases données en exemple, il faudra faire ressortir les variations de prononciation dans l'orthographe des mots.

Ex : *Sur une carte géographique, l'este est à droite*
Au couvent les poules couve

L'intérêt de cette simulation réside dans le fait que l'on se place dans le cadre du langage naturel oral où l'indéterminisme sur les terminaux est très important. Une entrée sous forme de mots aurait faussé les problèmes de reconnaissance. Nous avons préféré entrer le texte sous sa forme classique plutôt que sous sa forme phonétique par souci d'ergonomie, car il n'est pas très facile d'entrer un

message sous forme phonétique sans un caerin apprentissage, d'autant plus que la codification utilisée ne correspond pas aux codes phonétiques usuels.

Notre programme de transcription utilise des groupes de 1, 2 ou 3 lettres en tenant compte des contextes gauche et droit qui peuvent influencer la transcription. Il utilise en outre la classification des lettres en voyelles et consonnes ainsi que la longueur du mot à traduire (lorsque c'est nécessaire, essentiellement sur des mots courts). Pour simuler le comportement d'un module de DAP, il est possible d'insérer quelques erreurs dans le texte à traduire, par exemple remplacer un groupe de lettres par un groupe phonétiquement voisin (confusion de phonème), supprimer une ou plusieurs lettres (élision de phonème) ou au contraire en rajouter (insertion de phonèmes). La figure 4.15 montre quelques exemples de règles de transcription à partir des codes donnés en annexe.

' eau '	---->	o
' oui '	---->	w i
' et ' + ø	---->	ai
voy + ' s ' + voy	---->	z
i + ' lle '	---->	j &
' ou '	---->	u
' el ' + ø	---->	ai l

Figure 4.15. Exemples de règles de transcription

4.3.2. Analyseur lexical

L'analyse lexicale consiste à rechercher des mots dans l'énoncé. Ces mots sont des hypothèses faits par d'autres analyseurs et envoyés à l'analyseur lexical. La comparaison entre la forme de référence et l'énoncé se fait à partir de leur représentation phonétique. La comparaison est évaluée par le calcul d'un score de ressemblance entre les deux formes, si le score est suffisamment bon, le mot est sélectionné, sinon il est rejeté. Dans le chapitre 5 nous détaillons la comparaison de deux formes ainsi que le mode de calcul du score de reconnaissance et le mode de sélection des mots.

4.3.3. Analyseur syntaxico-sémantique

Sa tâche consiste à construire des hypothèses de type mot qui sont envoyées à l'analyseur lexical qui lui fournit en résultat l'ensemble des mots validés par la reconnaissance, puis à partir de ces mots et du modèle de la langue, il doit construire les différentes phrases reconnues. Son but est de construire les meilleures phrases possibles, c'est-à-dire qu'il doit utiliser les mots les mieux reconnus et optimiser les constructions de manière à avoir le moins de "trous" possible. Il doit fournir en résultat un ensemble de phrases reconnues classées de la meilleure à la moins bonne. Pour cela, lors de la construction, il doit évaluer la phrase par le calcul d'un score de reconnaissance. Les différentes étapes de la construction de phrases, le calcul du score et la sélection des meilleures phrases sont décrits dans le chapitre 5.

4.3.4. Module d'Interprétation

L'interprète doit évaluer les phrases reconnues, c'est-à-dire les confronter à l'univers pour en connaître la validité. L'interprétation se décompose en trois étapes :

- confrontation au modèle de la tâche,
- instanciation des cas,
- confrontation à l'univers.

L'interprète utilise toutes les connaissances susceptibles de l'aider à l'interprétation d'une phrase (connaissances pragmatiques, historique du dialogue...), il se contente uniquement d'utiliser ces connaissances mais ne développe pas de stratégie d'interpolation si son interprétation aboutit à un échec ou à une ambiguïté, il laisse ce travail au gestionnaire du dialogue. Il fournit à celui-ci un ensemble de phrases interprétées associées à un score d'interprétation permettant de les classer les unes par rapport aux autres. Nous verrons dans le chapitre 6 les différentes étapes de l'interprétation d'un énoncé.

4.4. Modèle du dialogue

C'est de lui que dépend toute l'organisation du système, en effet, c'est le modèle du dialogue qui gère les différents analyseurs. Il définit les stratégies à employer pour poursuivre le dialogue en cas d'ambiguïté, d'échec, de

contestation... Nous allons dans un premier temps présenter le mode de gestion du dialogue, puis nous parlerons des stratégies développées par le module de dialogue dans certaines circonstances particulières et enfin nous présenterons l'organisation générale du système qui en découle.

4.4.1. Gestion du dialogue

Le module de dialogue lance la reconnaissance d'une phrase en faisant certaines hypothèses. L'hypothèse de départ est l'attente d'une commande, c'est pourquoi il lance la recherche d'un prédicat après quoi il passe la main à l'analyseur syntaxico-sémantique qui est chargé d'effectuer la reconnaissance de la phrase à partir des prédicats trouvés. En cours de dialogue, il émet différents types d'hypothèses, lorsqu'une commande a été reconnue et correctement interprétée il s'attend à une confirmation, à une contestation ou à une nouvelle commande.

S'il s'agit d'une confirmation, il lui suffit de relancer le dialogue dans l'attente d'une nouvelle commande, s'il s'agit d'une contestation, il lui faut gérer le sous-dialogue permettant de mettre à jour les éléments qui sont modifiés par le locuteur. Une contestation peut apparaître dans plusieurs situations différentes :

- le système a reconnu et interprété une phrase de manière incorrecte,
- le locuteur revient sur ce qu'il a dit,
- le système a fait de mauvaises hypothèses pour résoudre une ellipse, une anaphore ou une ambiguïté.

Une contestation nécessite l'analyse de nouvelles informations mettant la commande à jour, cette analyse est gérée par le module de dialogue qui ne permettra qu'une suite limitée d'échanges dans le sous-dialogue de contestation/confirmation pour ne pas aboutir à une situation inextricable.

Lorsqu'il s'agit d'une nouvelle commande, le module de dialogue doit envisager une ellipse du prédicat (ex : Ferme la fenêtre / ... et la porte), c'est pourquoi il relance une analyse globale d'un énoncé prédicatif mais en faisant l'hypothèse que la commande peut se référer au prédicat de la commande précédente. Dans ce cas, si la recherche du prédicat ne donne pas de résultat, la reconnaissance de la phrase peut se poursuivre à partir de l'hypothèse faite.

Lorsque l'analyseur syntaxico-sémantique fournit le résultat de la reconnaissance de phrases, le gestionnaire de dialogue soumet les phrases ainsi obtenues à l'interprète qui rend à son tour un ensemble de phrases interprétées et

validées. Si l'interprétation ne donne pas satisfaction (ambiguïté ou échec), le système de dialogue développe des stratégies pour essayer de résoudre les problèmes. Ces stratégies ne sont développées que si on a la certitude d'arriver en peu de temps et d'interventions à une solution intéressante, en effet si tous les cas instanciés ont abouti à un échec, il est inutile d'essayer de résoudre le problème, il faut plutôt remettre en cause la reconnaissance et demander au locuteur de reformuler sa demande.

4.4.2. Stratégies du dialogue

Ces stratégies sont développées dans deux cas particuliers d'interprétation : les cas d'échecs et les cas d'ambiguïté.

Dans les cas d'échec, le système doit pouvoir fournir au locuteur un diagnostic de l'échec pour que celui-ci puisse se corriger plus facilement, par exemple si le locuteur demande au robot de prendre un livre rouge alors qu'il n'y en a pas, il faut lui répondre : *il n'y a pas de livre rouge* plutôt que : *commande incorrecte*. Dans la mesure du possible il faut aussi informer le locuteur d'une solution éventuelle de remplacement, par exemple s'il existe un livre noir la réponse pourra être : *il n'y a pas de livre rouge mais il y en a un noir*.

Pour les cas d'ambiguïtés, plutôt que de demander au locuteur de reformuler sa demande, il est préférable de laisser le système faire une hypothèse parmi les possibilités en conflit. Il est intéressant de choisir cette hypothèse avec le plus grand soin de manière à augmenter les chances de confirmation du locuteur et ainsi éviter une nouvelle analyse de contestation. Nous développerons dans le chapitre 6 les techniques permettant d'effectuer le diagnostic d'échec et de fournir au locuteur des informations pertinentes ainsi que la stratégie permettant de sélectionner parmi un ensemble de possibilités en conflit celle qui est la plus vraisemblable.

4.4.3. Organisation du système

4.4.3.1. Le lexique

Le lexique contient tous les renseignements qui doivent être accessibles à partir du mot, c'est-à-dire pour ce qui concerne la reconnaissance : la forme phonétique

du mot, ses rôles syntaxiques et sémantiques, les occurrences associées (pour les noms d'objets) et leur modèle.

Nous avons choisi de représenter les mots du lexique sous forme de structures apparentées aux frames. Pour un accès rapide aux diverses catégories de mots, le lexique est organisé en sous-lexiques grammaticaux (lexique des prépositions, des noms, des prédicats, des adjectifs...). Pour chaque sous-lexique, nous avons défini un ensemble de champs en relation avec le type des éléments. Ces champs correspondent aux caractéristiques qui peuvent se regrouper en quatre grandes catégories : PHONÉTIQUE, SYNTAXIQUE, SÉMANTIQUE, PRAGMATIQUE .

- Les caractéristiques phonétiques :

Elles fournissent pour un mot sa représentation phonétique, pour une compatibilité avec notre programme de simulation du DAP, la forme phonétique est construite à partir du mot orthographié grâce au programme de transcription graphème-phonème.

- Les caractéristiques syntaxiques :

Elles fournissent toutes les informations nécessaires à l'analyse syntaxique du mot, c'est-à-dire : le genre, le nombre, la classe syntaxique, la décomposition syntaxique (ex : du --> de le), les formes factorisées (ex : près de --> de, du)...

- Les caractéristiques sémantiques :

La principale information est le(s) rôle(s) sémantique(s) de chaque mot (ex : sur --> lieu, destination). Les caractéristiques sémantiques fournissent, lorsque cela est nécessaire, les éléments de la même classe d'équivalence (ex : devant --> près de, à côté de...). Pour les prédicats, elles donnent la classe qui permet d'accéder à l'ensemble des cas associés. De manière générale, elles fournissent le concept associé à chaque mot (ex : chevaux --> cheval, petite --> petit...).

- Les caractéristiques pragmatiques :

Elles ne concernent que les mots spécifiques à l'application, c'est-à-dire les noms d'objets et les prédicats. les noms d'objets sont le moyen d'accéder aux occurrences, même si celles-ci ne sont pas réellement dans le lexique, le nom de l'objet en est le point d'entrée. Les informations pragmatiques associées aux objets et se trouvant physiquement dans le lexique sont, d'une part, l'ensemble des prédicats applicables à l'objet, et d'autre part, le prototype de l'objet. Pour les prédicats, toutes les

informations pragmatiques vues dans le paragraphe 4.4 figurent dans le lexique.

TABLE	
- phon	: table, / t / a / b / l /
- nature	: nom
- type	: commun
- genre	: F
- nombre	: N % la forme ne permet pas de déterminer le nb %
- rôle	: objet
- prototype :	
- taille	: (petit , grand) % l'un ou l'autre %
- couleur	: rouge, noir, blanc, brun ...
- forme	: (rond, ovale, carré, rectangulaire)
PRES-DE	
- Phon	: prèdeu, / p / r / e / d / & /
- nature	: préposition
- rôle	: source, destination
- contract	: de, du, des
PRENDRE	
- phon	: / p / r / an / d / r /
- nature	: verbe
- racine	: prend
- type	: ACTS
- rôle	: prédicat
- auxiliaire	: avoir % utile à la génération %
- groupe	: 3
- pragmatique	
- précondition	: état (objet) = posé
- scénario :	
- déclenchement	: 'je vais près de' objet
- action	: phrase évaluée
- résultat	: état (objet) <-- pris

Figure 4.16. Exemples de mots des sous-lexiques

4.4.3.2. La représentation de l'univers

Comme nous l'avons vu dans le paragraphe précédent, le nom d'un objet permet d'accéder aux occurrences qui lui sont associées, cependant les occurrences sont une source de connaissances indépendante du lexique. Les occurrences sont stockées sous forme de liste d'identifiants (un identifiant pour chaque objet) auxquels sont associées les propriétés des objets ainsi représentés. De même les relations positionnelles entre les objets sont conservées dans la zone réservée à la représentation de l'univers, elles sont constituées d'une liste de triplets de la forme : (relation-de-position, objet1, objet2).

4.4.3.3. Les grammaires

L'ensemble de la grammaire est représenté par :

- l'arbre des cas que nous avons vu au paragraphe 4.2,
- l'ensemble des grammaires locales,
- les règles de coordination et de qualification,
- pour chaque cas, l'ensemble des grammaires locales pouvant le représenter.

L'organisation des connaissances de manière modulaire permet des modifications ponctuelles n'interférant pas les unes avec les autres. Par exemple, on peut rajouter de nouvelles formes syntaxiques utilisables par le système en intégrant de nouvelles grammaires locales, sans pour cela avoir à modifier le lexique ou l'univers, ni même l'arbre des cas. Il peut être aussi envisagé de réorganiser l'arbre des cas sans modifier les grammaires locales. De la même manière, un changement d'application consistera (dans le cas où le même type de langage est utilisé) à modifier uniquement les sous-lexiques des noms d'objets et des prédicats.

4.4.3.4. Architecture du système

Nous allons présenter les différentes étapes de l'analyse d'un énoncé, à partir de la suite de phonèmes obtenue par le "module de DAP" jusqu'à son interprétation.

La donnée initiale est le résultat du module de DAP qui est une suite de phonèmes représentant la phrase prononcée par le locuteur. Comme nous l'avons fait remarquer, dans un langage de type commande, c'est le prédicat qui détermine la forme de l'énoncé, c'est pourquoi la première étape consiste à rechercher un verbe dans l'énoncé. Lorsqu'un (ou plusieurs) verbe est trouvé, il faut définir l'ensemble des cas qui sont associés. Ces cas sont classés par priorité décroissante et recherchés dans l'énoncé sous les formes syntaxiques prédéfinies associées aux contraintes sémantiques des cas. A partir des groupes ainsi trouvés, l'analyseur syntaxico-sémantique reconstruit la phrase en deux passages : le premier consistant à instancier de manière minimale tous les cas possibles et le second en traitant les cas de coordination et de qualification. Après cette étape, on peut considérer que la phrase est reconnue. Il reste donc ensuite à en faire l'interprétation. Celle-ci se décompose en plusieurs étapes :

- la résolution des ellipses et des références conceptuelles, à l'aide de contraintes syntaxico-sémantiques et pragmatiques et à l'aide de l'historique du dialogue ;
- l'interprétation des cas, c'est-à-dire la recherche de la valeur de chaque cas nécessaire à l'interprétation. Il s'agit de résoudre les références aux concepts, de trouver les occurrences des objets cités... à l'aide d'informations pragmatiques et de la représentation de l'univers ;
- la résolution des ambiguïtés ou la résolution des cas dont l'instanciation est vide au moyen de stratégies que nous développerons dans le chapitre 6 ;
- la gestion de sous dialogues nécessaires à l'obtention d'informations supplémentaires lorsque les stratégies précédentes n'ont pas réussi à résoudre tous les problèmes, ou lorsqu'une demande de confirmation est nécessaire.

Ces deux grandes étapes, la reconnaissance et l'interprétation font respectivement l'objet des chapitres 5 et 6.

Nous pouvons représenter l'architecture de notre système par le schéma donné à la figure 4.17.

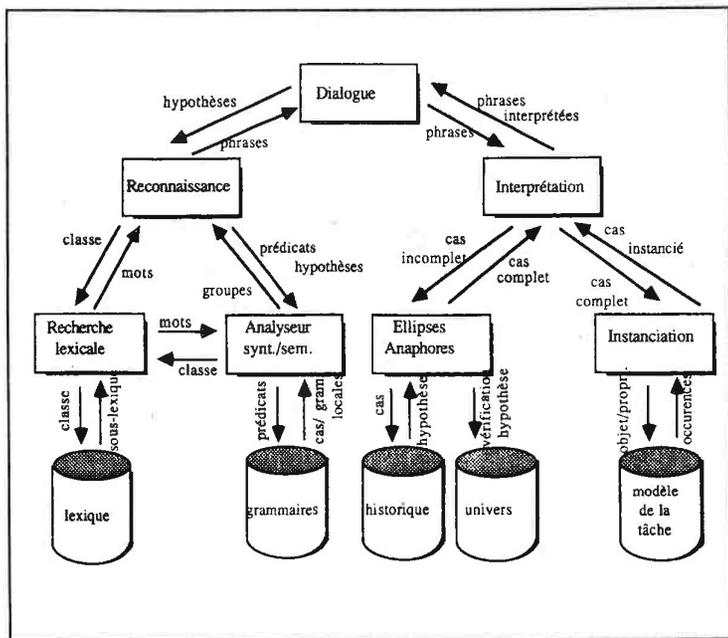


Figure 4.17. Architecture générale du système

Chapitre 5 Reconnaissance de phrases

Rappelons en quelques mots la démarche suivie. Pour interpréter un énoncé, l'interpréteur, dans le cas où il ne dispose d'aucune hypothèse, recherche un prédicat dans la phrase et suppose que l'on peut avoir une ellipse du prédicat précédent. A partir des prédicats trouvés et de celui de l'énoncé précédent, il détermine quels sont les cas sémantiques à satisfaire. Pour chaque cas sémantique, il dispose de grammaires locales décrivant les différentes formes syntaxiques possibles. A partir d'une grammaire :

- il sélectionne les différents sous-lexiques concernés (ex : lexique des adjectifs, des noms...),
- extrait de ces lexiques les mots vérifiant les contraintes sémantiques associées aux cas (ex : noms d'objets, prépositions de lieu...)
- recherche les mots dans l'énoncé.

Nous allons présenter comment nous avons réalisé la reconnaissance de phrases en la découpant en trois étapes : la reconnaissance de mots, la construction de groupes simples et enfin la construction de la phrase complète. Ensuite, nous présenterons les différentes stratégies de sélection des meilleures phrases.

5.1. Du phonème au mot

Notre approche s'appuie sur la méthode analytique (cf ch 1). L'énoncé est donc une liste de phonèmes fournie par l'analyseur acoustico-phonétique. Pour notre application, l'analyseur acoustico-phonétique est simulé par un module de transcription graphème-phonème. Le message sera donc entré sous forme écrite en tenant compte des altérations phonologiques dues au locuteur (accent, défaut de

prononciation ...), de celles dues aux particularités de la communication en langage naturel (liaisons...) et des limites des analyseurs actuels (problèmes de ségmentation, confusion, élision ...).

Exemple :

Soit la phrase " prends le bidon d'huile " prononcée par une personne à l'accent alsacien, l'énoncé aura la forme : " prends le piton tuile " et la suite de phonèmes reconnus sera :

/p/r/ an/ l/&/p/i/t/on/t/ui/l/

L'étape permettant le passage d'une liste de phonèmes à un mot consiste à comparer la liste de phonèmes composant le mot de référence à une sous-liste de phonèmes extraite de l'énoncé. Nous partons d'hypothèses émises par les niveaux supérieurs, donc nous nous plaçons dans le cadre de la vérification de mots (Word Verification). Cependant, le langage naturel étant par définition asyntaxique, un cas associé à un verbe peut se trouver à différentes places dans la phrase. Les mots recherchés pour définir un cas sémantique peuvent donc se trouver dans n'importe quelle plage de l'énoncé. De plus certains mots peuvent figurer dans plusieurs cas simultanément (ex : articles, adjectifs ...). C'est pourquoi la recherche des mots se fait aveuglément sur toute la phrase, les contraintes positionnelles liées à la vérification de mots disparaissent, on se rapproche donc de la détection de mots (Word Spotting) après limitation du vocabulaire.

Exemples :

- " Pose le livre sur la table " qui correspond à une construction syntaxique correcte

- " sur la table livre poser " qui a la même valeur sémantique que la phrase précédente mais qui ne respecte pas les règles syntaxiques de construction de phrases.

Pour diminuer le temps de recherche, on pourrait éventuellement :

- Tenir compte de certaines hypothèses positionnelles locales, par exemple si un article a été trouvé dans l'énoncé, pour construire un groupe nominal, il suffirait de restreindre la recherche des autres syntagmes sur la plage de

l'énoncé suivant immédiatement cet article. Cependant la gestion des zones explorées et non explorées s'alourdirait de telle manière que le temps gagné pour la recherche des mots serait perdu par la gestion de ces zones et celle des retours arrières en cas d'échec de reconnaissance.

Exemple : Si dans une phrase on a la suite de phonèmes :

/l/ a / p / o / r / t /

il est incontestable que l'analyseur trouvera le prédicat APPORTER qui en cas d'hypothèses positionnelles excluerait la recherche d'un nom sur la zone concernée alors que la bonne interprétation de cette partie est : LA PORTE.

- Arrêter la recherche quand un mot a été trouvé avec un score suffisamment bon. Cependant, dans ce cas il y a aussi un inconvénient majeur, c'est la possibilité d'apparition de plusieurs occurrences d'un même mot dans une phrase.

Exemples : *Tourne le petit bouton rouge et le petit bouton noir*

Pose la petite lampe sur la petite table noire

Le résultat de la comparaison entre un mot et la phrase sera une liste des occurrences de ce mot dans l'énoncé, une occurrence étant caractérisée par :

- un indice de début de mot (numéro du premier phonème reconnu)
- un indice de fin de mot (numéro du dernier phonème reconnu)
- un score de reconnaissance
- le mot reconnu

Exemples : (7 10 0.9 bouton) (17 20 1 bouton)

5.1.1. Comparaison de deux formes

Comme nous l'avons vu, nous ne disposons pas d'information quant à la position d'un mot dans l'énoncé. Il faut donc décomposer la recherche en deux étapes :

- recherche de tous les débuts possibles

- comparaison des phonèmes extraits à partir des débuts trouvés avec la forme de référence.

• Recherche des débuts

La recherche des débuts se fait séquentiellement sur la liste des phonèmes. Du fait des altérations possibles dues à la caractéristique naturelle de la langue, il paraît insuffisant de se contenter du premier phonème comme point d'ancrage. En effet, les zones les plus sensibles aux altérations sont les début et fin de mots en particulier à cause des liaisons possibles, des baisses d'intonation...

Pour décider si un mot candidat a une chance de se trouver sur une plage précise de l'énoncé, nous allons nous référer à ses deux premiers phonèmes. Lorsqu'un phonème du signal est égal au premier phonème du mot de référence la plage considérée est automatiquement acceptée. Si le premier phonème n'est pas reconnu mais que l'on a détecté le deuxième, il faut envisager le cas d'une altération en début de mot ; l'acceptation ou le rejet de ce début dépendra de la ressemblance entre le premier phonème de référence et le premier phonème candidat. Si ces deux phonèmes sont articulatoirement proches la proposition est acceptée sinon elle est rejetée.

Ex : recherche des débuts possibles du mot TABLE dans l'énoncé

/ m / a / p / a / d / e / ...

Un début possible est détecté sous le phonème /p/ car la taux de confusion entre /p/ et /t/ est très faible (cf matrice de confusion Ch 1 § 1.3.2.) et le phonème suivant /a/ est bien reconnu.

Si notre système de reconnaissance était connecté à un module de DAP, il serait plus intéressant de faire une première recherche en comparant les patrons phonétiques des mots, ce qui permet d'augmenter les chances de trouver le mot sur la partie de signal sélectionnée. En effet dans ce cas, la forme globale du signal ressemble au mot de référence alors que dans notre recherche, nous nous basons uniquement sur les deux premiers phonèmes, ce qui est un indice de ressemblance assez mince et diminue en fait très peu les temps de recherche par rapport à une comparaison directe sur tout le signal.

• Comparaison des deux formes

Nous utilisons le principe de la programmation dynamique [Boyer-] . La comparaison de la sous-liste de phonèmes avec le mot de référence peut se schématiser de la manière suivante :

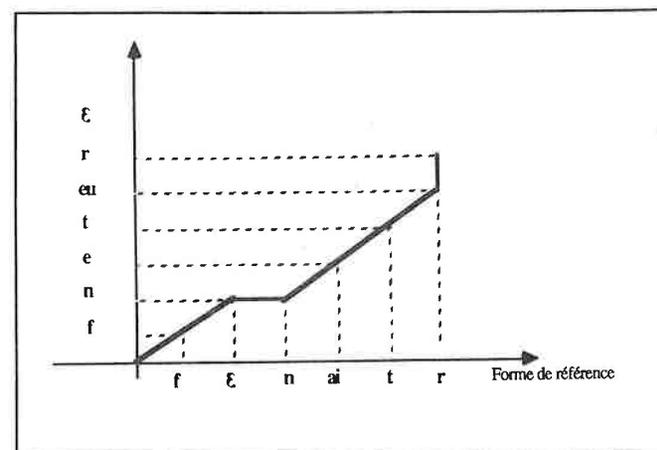


Figure 5.1. Comparaison de deux formes par programmation dynamique

5.1.2. Calcul du score de ressemblance

Pour déterminer si le mot a été correctement reconnu, nous calculons un score de ressemblance entre l'énoncé et la forme de référence. Ce score doit répondre à plusieurs critères :

- 1- il doit être proportionnel aux scores de reconnaissance des différents phonèmes,
- 2- il doit augmenter lorsque le nombre de phonèmes mis en correspondance augmente et de manière proportionnelle aux taux de ressemblance des phonèmes mis en correspondance,
- 3- il doit être pénalisé en cas d'insertion ou d'élision de phonèmes,

4- il doit permettre d'abandonner en cours de reconnaissance,

5- il doit être compris entre 0 et 1,

6- le score de reconnaissance d'un mot dont tous les phonèmes sont reconnus à 100%, doit être égal à 1 : SCORE (1, 1, 1, 1) = 1

7- de la même manière, on doit avoir : SCORE (0, 0, 0, 0) = 0

Nous allons expliquer comment au vu de ces critères nous avons établi notre formule de calcul de score.

• Réponse au 1^{er} critère

Le score doit dépendre des différents scores de reconnaissance des phonèmes constituant le mot de référence. Pour cela, il faut être capable de déterminer un taux de ressemblance entre deux phonèmes quelconques. Nous utiliserons des taux compris entre 0 et 1 (0 si les phonèmes ne peuvent pas être confondus, 1 si les phonèmes sont identiques et des valeurs intermédiaires pour des phonèmes plus ou moins voisins). B. Mangeol [Mangeol-88] a défini une matrice de confusion qui associe à chaque phonème un coût de remplacement par chacun des autres phonèmes. Ce coût sera d'autant plus faible si :

. une altération prévisible permet ce remplacement : /n /---> /d /

. les deux phonèmes sont proches articuloirement : /i /---> /e /

. les deux phonèmes sont parfois confondus à cause d'une imperfection du système. Ex : /e /---> /l / correspondra à un /e / non détecté par la segmentation en tant que noyau vocalique.

A partir de cette matrice de confusion, nous avons créé une matrice de ressemblance (figure 5.2.) où le taux de ressemblance est calculé par rapport à la valeur de confusion de la manière suivante : $\text{taux} = 1 - (\text{conf} / 10)$. La figure 5.2 nous donne les coefficients de la matrice de ressemblance multipliés par 10.

i	:	(e 8)	(ai 7)	(y 9)	(œ 4)	(& 4)	(z 6)	(l 6)		
e	:	(i 8)	(ai 10)	(y 4)	(œ 6)	(& 5)	(a 7)	(l 5)		
ai	:	(i 9)	(e 10)	(œ 6)	(& 5)	(a 6)	(r 6)			
y	:	(i 9)	(œ 8)	(& 8)	(o 6)	(u 7)	(l 6)			
œ	:	(& 10)	(a 9)	(o 6)	(m 6)					
&	:	(e 7)	(y 7)	(œ 10)	(a 8)	(o 7)	(u 5)			
a	:	(œ 8)	(& 8)	(o 5)	(in 9)	(ā 9)	(r 5)			
o	:	(œ 9)	(& 7)	(u 9)	(in 7)	(ā 7)	(ō 9)	(r 5)		
u	:	(y 7)	(œ 8)	(& 7)	(o 9)	(ō 6)	(r 6)			
in	:	(œ 8)	(a 9)	(ā 9)	(ō 9)	(r 6)				
ā	:	(œ 7)	(a 9)	(o 7)	(in 9)	(ō 9)	(r 6)			
ō	:	(œ 7)	(o 9)	(in 9)	(ā 9)	(n 5)				
w	:	(a 9)	(in 8)	(ā 10)	(ō 7)	(r 10)				
j	:	(i 9)	(s 2)	(z 7)	(gh 7)	(v 7)	(l 6)			
s	:	(j 4)	(z 9)	(ch 9)	(gh 5)	(f 8)	(v 5)			
z	:	(s 9)	(ch 6)	(gh 9)	(f 6)	(v 9)				
ch	:	(s 9)	(z 6)	(gh 9)	(f 8)	(v 6)	(r 6)			
gh	:	(s 6)	(z 9)	(ch 9)	(f 6)	(v 9)				
f	:	(s 8)	(z 5)	(ch 8)	(gh 5)	(v 9)	(r 6)	(p 4)		
v	:	(s 5)	(z 9)	(ch 5)	(gh 9)	(f 5)	(p 6)	(b 8)		
n	:	(œ 5)	(in 6)	(ā 6)	(ō 6)	(v 10)	(m 10)	(nj 9)	(l 7)	(r 5)
m	:	(œ 5)	(in 6)	(ā 6)	(ō 6)	(n 10)	(nj 9)	(l 6)	(r 5)	
nj	:	(n 9)	(m 9)							
l	:	(y 6)	(j 5)	(n 6)	(m 7)	(r 8)	(d 5)			
r	:	(a 7)	(in 7)	(ā 7)	(w 9)	(z 6)	(gh 6)	(n 5)	(m 5)	(l 8)
p	:	(f 4)	(v 4)	(t 9)	(k 9)	(b 9)	(d 6)	(g 6)		
t	:	(p 9)	(k 9)	(b 6)	(d 9)	(g 6)				
k	:	(p 9)	(t 9)	(b 6)	(d 6)	(g 9)				
b	:	(f 3)	(v 8)	(n 5)	(m 5)	(p 9)	(t 6)	(k 6)	(d 9)	(g 9)
d	:	(n 5)	(m 5)	(p 6)	(t 9)	(k 6)	(b 9)	(g 9)		
g	:	(n 5)	(m 5)	(p 6)	(t 6)	(k 9)	(b 9)	(d 9)		

Figure 5.2. Matrice de ressemblance

• Réponse au 2^{ème} critère

Les études sur la théorie des possibilités [Dubois-85] et en particulier l'approche par la règle de Dempster du raisonnement avec des prémisses incertaines, permettent de tenir compte du renforcement des possibilités : lorsque des informations fournies par plusieurs sources concordent, la possibilité du résultat s'en trouve augmentée.

Soit une proposition P et m1(P) et m2(P) deux possibilités non conflictuelles la possibilité de la proposition P est calculée de la manière suivante :

$$m12(P) = m1(P) + m2(P) - m1(P) * m2(P)$$

Dans la reconnaissance de mots, chaque nouveau phonème confirmant la présence du mot doit augmenter la possibilité de présence de ce mot. cependant, contrairement aux théories citées ci-dessus, la présence d'un seul phonème correct ne suffit pas pour reconnaître complètement le mot : les informations ne sont pas indépendantes! Il faut donc pondérer le taux d'accroissement. Posons :

- score_i : score de reconnaissance à la i^{ème} étape
- score_{i-1} : score de reconnaissance à la (i - 1)^{ème} étape
- score_{ph} : score de reconnaissance du phonème courant
- CP : coefficient de pondération

la formule devient :

$$\text{score}_i = (\text{score}_{i-1} + \text{score}_{\text{ph}}) - (\text{score}_{i-1} * \text{score}_{\text{ph}}) * \text{CP}$$

Un CP proche de 1 augmente le score de manière importante alors qu'un CP proche de 2 l'augmente de manière modérée. Il paraît cependant arbitraire de déterminer un CP statique. Il serait intéressant au contraire de déterminer un CP dynamique reflétant l'évolution de la reconnaissance du mot. une première solution envisagée est : $\text{CP} = 2 - \text{score}_{\text{ph}}$

Cependant, cette solution privilégie de manière trop importante le score de reconnaissance du phonème courant, elle ne tient pas compte de l'état général de la reconnaissance. La deuxième solution envisagée et retenue fait intervenir le score global de reconnaissance à l'étape i-1 et le score de reconnaissance du phonème courant :

$$\text{CP} = 2 - (\text{score}_{i-1} + \text{score}_{\text{ph}}) / 2$$

Cette formule a l'avantage de tenir compte de l'évolution du score global de reconnaissance. En effet, si le score du phonème est meilleur que le score global de reconnaissance, celui-ci augmentera de manière plus importante que si le nouveau phonème a tendance à infirmer la reconnaissance.

On obtient donc comme formule finale de calcul du score de reconnaissance d'un mot à une étape i :

$$\text{score}_i = \frac{(\text{score}_{i-1} + \text{score}_{\text{ph}}) - (\text{score}_{i-1} * \text{score}_{\text{ph}}) * (2 - (\text{score}_{i-1} + \text{score}_{\text{ph}}) / 2)}{1}$$

• Réponse au 3^{ème} critère

Les pénalités dues aux insertions ou élisions doivent dépendre de la longueur du mot de référence, en effet, une élision sur un mot de trois phonèmes est plus grave qu'une élision sur un mot de dix phonèmes par exemple. C'est pourquoi, lors de la comparaison de deux formes, nous calculons le nombre total d'élisions d'une part et le nombre total d'insertions d'autre part. Quand la comparaison est terminée, nous recalculons le score final du mot qui tient compte des élisions et des insertions de la manière suivante :

$$\text{score}_m = \text{score} - ((\text{nbe} * 0.3) + (\text{nbi} * 0.2)) / \text{long}$$

scorem	: score final de reconnaissance du mot
score	: score en fin de comparaison
nbe	: nombre d'élisions
nbi	: nombre d'insertions
long	: longueur du mot de référence.

Les coefficients 0.3 et 0.2 ont été choisis de manière pragmatique.

• Réponse au 4^{ème} critère

Il doit être possible d'abandonner la comparaison de deux formes si celles-ci diffèrent de manière trop importante, dans ce but nous allons utiliser plusieurs heuristiques :

- Le score de reconnaissance d'un mot, en cours de comparaison, ne doit pas descendre en-dessous d'un seuil fixé,
- le nombre d'erreurs d'un même type est limité,
- deux erreurs successives ne sont pas autorisées et enfin,
- le nombre total d'erreurs est limité.

Ceci nous permet de gagner du temps en ne poursuivant pas la reconnaissance d'un mot si celui-ci contient trop d'erreurs.

• Réponse au 5^{ème} critère

Vu les taux de ressemblance choisis et la formule de calcul de score, il est aisé de se rendre compte qu'un score sera toujours compris entre 0 et 1. Le score, après calcul des pénalités dues aux élisions et insertions, pourrait éventuellement, dans un cas extrême, devenir inférieur à 0, mais ceci ne pose aucun problème car pour être validé, un mot doit avoir un score supérieur à un seuil déterminé (qui dans tous les cas sera supérieur à 0.5).

• Réponses aux 6^{ème} et 7^{ème} critères

Il suffit d'appliquer la formule pour se rendre compte que ces critères sont respectés.

Nous obtenons donc une liste des mots reconnus dans l'énoncé. A chacun de ces mots est associé un score de reconnaissance, il reste cependant à déterminer un seuil d'acceptation de manière à ne sélectionner que les mots correctement reconnus.

5.1.3. Sélection dynamique des mots

Le problème qui se pose est de déterminer la valeur de ce seuil sachant que le système de reconnaissance doit pouvoir fonctionner :

- en mode multi-locuteur
- dans diverses situations de communication (bruitée ou non ...),

Ceci peut entraîner de grandes variations sur les taux de reconnaissance des phonèmes et donc sur les scores associés aux mots. Pour palier aux problèmes de bruits et d'altérations phonologiques dues au locuteur, il faut déterminer un seuil d'acceptation relativement faible. Mais dans ce cas nous sommes confrontés à un autre problème, en effet si les conditions de communication sont bonnes le seuil de tolérance va nous faire accepter un grand nombre de mots inutiles qui vont alourdir le traitement et qui pourront même entraîner une explosion combinatoire au moment de la construction des groupes. Il faut donc, en quelques sortes s'adapter au locuteur et aux conditions de communication. Il faut définir un seuil d'acceptation dynamique réévalué à chaque échange car, même si a priori le locuteur reste le même, les conditions peuvent être modifiées (bruit soudain, panique du locuteur en cas d'alerte entraînant une dégradation de son élocution ...). La sélection des mots va donc se décomposer en deux étapes :

- la première consiste à extraire de la phrase tous les mots dont le score de reconnaissance est supérieur au seuil statique prédéfini et en-dessous duquel on considère qu'un mot n'est pas valide,
- et la seconde va effectuer une deuxième sélection si le nombre d'occurrences trouvées est trop important. Le traitement consiste à calculer un nouveau seuil d'acceptation à partir des scores des mots trouvés et à ne conserver que les mots dont le score sera supérieur à ce seuil dynamique. Le seuil dynamique est calculé en faisant la moyenne entre le score du meilleur mot et le score du moins bon mot reconnu.

5.2. Du mot au groupe

5.2.1. Construction d'un groupe

Pour construire un groupe, nous disposons de deux sources d'information : une grammaire locale décrivant la syntaxe du groupe à composer et les mots

correspondant à chaque syntagme de la grammaire. Les contraintes à respecter sont :

- la syntaxe définie par la grammaire locale,
- les accords en genre et en nombre que l'on peut détecter,
- les contraintes positionnelles entre les mots : deux mots peuvent se suivre s'ils ont un taux de recouvrement faible (fusion du dernier phonème d'un mot avec le premier du mot suivant) ou s'ils ne sont pas trop éloignés l'un de l'autre (il est impossible d'insérer un autre mot entre les deux),

Il s'agit de trouver les meilleurs chemins possibles sur l'ensemble des mots trouvés, vérifiant les contraintes citées ci-dessus. Dans chaque liste, les mots sont classés par début croissant et pour un même début, par fin croissante. On peut schématiser l'ensemble des mots trouvés et les chemins sur ces mots de la manière suivante : Exemple : GN ---> art adj nom⁺ adj*

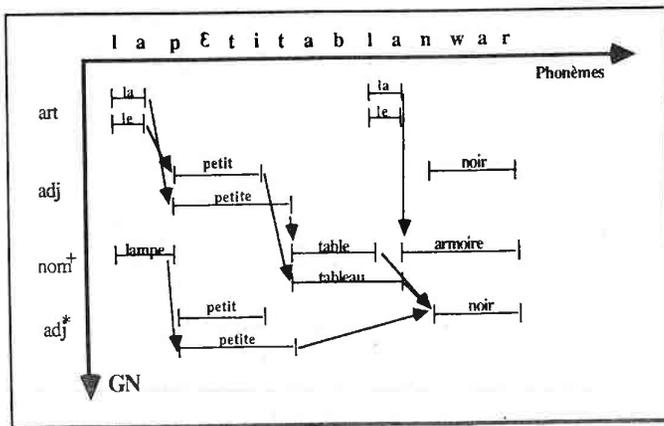


Figure 5.3. Construction d'un groupe nominal

Lorsqu'un syntagme n'est pas obligatoire, on prévoit la construction d'un groupe où ce syntagme ne figure pas. Dans l'exemple précédent, le seul syntagme obligatoire est le nom, on peut donc imaginer que le groupe minimal trouvé peut être réduit à un nom. Ceci nous permet d'obtenir toutes les combinaisons possibles autorisées par la grammaire afin de construire de manière optimale la

phrase. En effet, en construisant toutes les formes possibles, on obtient des groupes inclus dans d'autres groupes, mais à ce niveau il faut les conserver car nous ne pouvons savoir à l'avance quel groupe ou sous-groupe va être sélectionné au moment de la composition de la phrase.

Exemples : *Déplace le petit sac vers la porte*

La recherche d'un groupe nominal sur cette phrase va donner en résultat : " le petit sac vert ", si l'on ne sélectionne que cette possibilité, on se rend compte qu'une interprétation correcte de l'énoncé est impossible, la seule solution que l'on pourra trouver est : **Déplace le petit sac vert (et) la porte.**

Ce genre de problème ne se poserait pas dans les mêmes termes si l'interpréteur disposait d'un analyseur prosodique permettant de détecter les césures entre les groupes.

5.2.2. Calcul du score du groupe

Quand un groupe est construit, il faut calculer son score d'instanciation. Un mode de calcul simple consiste à faire intervenir le score de chaque mot de manière proportionnelle à la longueur du mot et le taux d'occupation du groupe entre le premier et le dernier phonème trouvés (le nombre de trous entre les mots). On obtient la formule :

$$\text{scoreg} = \text{taux-occ} * (\sum \text{scorem} * \text{longm}) / \text{longtm}$$

avec : $\text{taux-occ} = \text{longtm} / \text{longtot}$

d' où après simplification :

$$\text{scoreg} = (\sum \text{scorem} * \text{longm}) / \text{longtot}$$

scoreg	: score du groupe	taux-occ	: taux d'occupation du groupe
scorem	: score des mots intervenant dans la construction du groupe		
longm	: longueur, en nombre de phonèmes, de chaque mot		
longtm	: nombre total de phonèmes occupés par l'ensemble des mots		
longtot	: nombre total de phonèmes entre le début et la fin du groupe.		

Grâce au score ainsi calculé, l'analyseur sélectionne les groupes qui vont intervenir dans la construction de la phrase. Il dispose de trois stratégies paramétrables :

- choix des N meilleurs dans chaque plage (N fixé),
- sélection de tous les groupes dont le score est supérieur à un seuil donné,
- combinaison des deux stratégies précédentes, on arrête la sélection quand un des deux critères est vérifié : soit on a N groupes dans la plage, soit le score est inférieur au seuil limite.

remarque : une plage est définie lorsque l'on rencontre un groupe ne pouvant se "superposer" (entièrement ou en partie dans une limite définie) à une plage existante.

5.2.3. Représentation interne du groupe

Les renseignements fournis par l'analyseur après la construction d'un groupe sont de 2 types :

- généraux / syntaxiques

- . le type du groupe (gn, gp)
- . son score de reconnaissance
- . ses genre et nombre dans le mesure où ils ont pu être définis
- . la syntaxe du groupe qui peut dépendre d'un autre groupe

- lexicaux

- . la suite des mots composant le groupe.

Exemple : Informations fournies par l'analyseur pour le groupe : **La petite table**

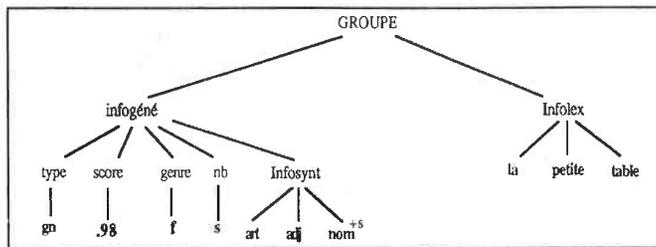


Figure 5.4. Représentation interne d'un groupe nominal.

Bien que nous ne les ayons pas faites figurer sur le schéma, les informations fournies par le module de reconnaissance de mot, c'est-à-dire

les numéros de phonèmes de début et de fin de mot ainsi que le score de reconnaissance, figurent dans la description de chacun des mots du groupe. Les codes utilisés pour qualifier les terminaux et les non-terminaux de la grammaire sont les suivants :

- + : indique un syntagme obligatoire
- s : indique le syntagme porteur de la valeur sémantique

Nous avons vu qu'un groupe peut être défini en fonction d'un autre groupe, nous obtenons donc une structure imbriquée. Pour conserver l'homogénéité des informations, nous regrouperons les informations syntaxiques d'une part et les informations lexicales d'autre part.

Exemple : soit le groupe prépositionnel : **sur la chaise**

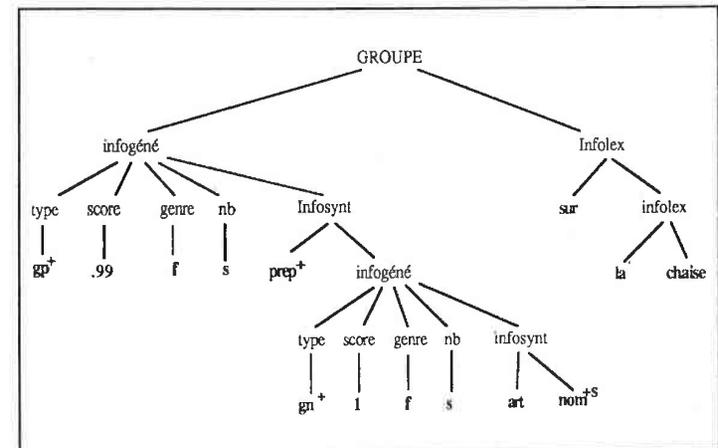


Figure 5.5. Représentation interne d'un groupe prépositionnel

5.3. Du groupe à la phrase

5.3.1. Du groupe au squelette de la phrase

Nous introduisons ici la notion de squelette de phrase, car cette étape de construction ne fournit pas la représentation finale de la phrase mais un simple dégrossissement en instanciant a priori tous les cas instanciables de manière

minimale. En effet, nous ne tenons compte que des groupes simples et pas encore des compléments associés à ce groupe. Voyons sur un exemple les motivations de ce choix.

Exemple :

Soit la phrase : **Pose le livre sur la table**

Comme nous l'avons vu dans le cahier des charges(cf ch 3), un objet doit pouvoir être décrit par un groupe nominal et éventuellement par des compléments indiquant la position de l'objet dans l'univers en fonction des autres objets. Si l'on tient compte de cette description possible au niveau de la construction du groupe, on obtient pour le cas objet de la phrase, l'instanciation : **Le livre sur la table**, et le cas destination ne pourra pas être instancié, et s'il n'y a aucun livre sur une quelconque table de l'univers, l'interprétation de la phrase sera un échec.

Pour la construction des cas, on pourrait utiliser la même stratégie que pour la construction des groupes, c'est-à-dire conserver les formes tronquées. Cependant si pour les groupes cette solution est envisageable, elle l'est beaucoup moins pour les cas complets car on risque, lorsque la description d'un objet devient complexe de provoquer une explosion combinatoire incontrôlable.

Les cas seront instanciés par ordre de priorité (cf ch 4), d'abord les cas obligatoires puis les cas semi-obligatoires et enfin les cas facultatifs. Comme pour la construction des groupes, l'analyseur prévoit, lorsqu'un cas n'est pas obligatoire, une instanciation vide de ce cas pour permettre ainsi toutes les combinaisons possibles sur la phrase.

5.3.2. Du squelette à la phrase complète

Dans ce paragraphe, nous allons présenter les différentes étapes de coordination et de qualification, nous donnerons ensuite des exemples de phrases obtenues à partir des énoncés reconnus et enfin nous définirons le mode de calcul du score du cas complet.

5.3.2.1. Les différentes étapes de coordination et qualification

Chaque cas de la phrase est instancié de manière minimale, c'est-à-dire par un seul groupe syntaxique. Comme nous l'avons vu dans le chapitre 4, les groupes

peuvent être soit coordonnés avec d'autres groupes, soit qualifiés par d'autres groupes. Nous avons explicité dans le paragraphe 4.2.2 les règles qui régissent les qualifications et les coordinations. Cependant, il ne faut pas perdre de vue le fait que les mots sur lesquels travaille l'analyseur syntaxico-sémantique sont des informations incertaines. Par exemple, ce n'est pas parce qu'on n'a pas trouvé de conjonction entre deux groupes que celle-ci n'existe pas. Pour pallier ces incertitudes, il faut envisager toutes les combinaisons possibles de groupes (dans le respect des contraintes syntaxiques et sémantiques).

Pour chaque phrase sélectionnée, on essaye de compléter les cas. Pour chaque cas, on regarde s'il y a un "trou" derrière lui, si oui on essaye de combler le trou par chacun des groupes susceptibles de s'y insérer. Toutes les solutions possibles sont conservées de manière à obtenir toutes les combinaisons de phrases possibles. La stratégie utilisée est globalement descendante, il serait cependant intéressant de la combiner avec une stratégie ascendante, comme dans le système DIRA [Caelen-88], de manière à pouvoir valider les résultats obtenus grâce à des prédictions concernant les groupes possibles.

Pour chaque groupe candidat, les diverses combinaisons suivantes sont essayées:

1. coordination avec le groupe le plus externe du cas

Ex : (**le livre (sur la table)**) et (**le cahier**)

2. coordination avec le groupe le plus interne du cas s'il existe

Ex : (**le livre** (**sur le bureau**) et (**près de la fenêtre**))

3. imbrication du groupe

Ex : (**le livre (sur le bureau (près de la fenêtre))**)

Dans le cas d'imbrication, il faut vérifier si le cas ne comporte pas de coordination, car dans ce cas, il faut envisager en plus le "développement" de l'expression :

Ex : **le livre et le cahier sur le bureau** --->

(**le livre (sur le bureau)**) et (**le cahier (sur le bureau)**))

Dans le paragraphe suivant nous donnons quelques exemples de groupes reconnus et de l'ensemble des phrases construites à partir de ces groupes.

5.3.2.2. Exemples de construction de groupes complexes

Cas instancié : cas objet --> le livre
 groupes candidats : sur le bureau près de la fenêtre
 phrases obtenues : (le livre ((sur le bureau) *conj* (près de la fenêtre)))
 (le livre (sur le bureau (près de la fenêtre)))

cas instancié : objet --> le livre rouge
 groupe candidat : (et) le bleu
 phrase obtenue : ((le livre rouge) et (le livre bleu))

Cas instancié : objet --> la petite
 groupes candidats : (et) la grande table rouge
 phrase obtenue : ((la petite table rouge) et (la grande table rouge))

Cas instanciés : objet --> le livre, dest --> sur la fenêtre, prédicat --> pose
 groupes candidats : près de la table
 phrases obtenues : (pose) (le livre) (sur la fenêtre (près de la table))
 (pose) (le livre) ((sur la fenêtre) (*conj*) (près de la table))

et pour le même énoncé, on aura aussi :

Cas instanciés : objet --> le livre, dest --> près de la table,
 prédicat --> poser
 groupes candidats : sur la fenêtre
 phrase obtenue : (pose) (le livre (*qui est* sur la fenêtre)) (près de la table)

Cas instancié : objet --> ((le livre) et (le cahier))
 groupes candidats : sur le bureau, près de la fenêtre
 phrases obtenues : - ((le livre (sur le bureau (*qui est* près de la fenêtre)))
 et (le cahier (sur le bureau

(*qui est* près de la fenêtre)))

- ((le livre)
 et (le cahier (sur le bureau
 (*qui est* près de la fenêtre)))

- ((le livre)
 et (le cahier ((sur le bureau) (*et*) (près de la fenêtre))))

- ((le livre ((sur le bureau) (*et*) (près de la fenêtre)))
 et (le cahier ((sur le bureau) (*et*) (près de la fenêtre))))

cas instancié : objet --> le livre
 groupes candidats : près de la lampe , (et) du téléphone
 phrase obtenue : (le livre ((près de la lampe) et (près du téléphone)))

...

5.3.2.3. Calcul du score du groupe complet

A chaque fois qu'un groupe est rattaché à un cas, il faut réévaluer le score d'instanciation de ce cas. Nous utilisons la formule :

$$\text{scorecas} = (@ \text{scorecas} * @ \text{longcas} + \text{scoreg} * \text{longg}) / \text{longtot}$$

scorecas : score du cas
 @scorecas : ancien score du cas
 @longcas : ancienne longueur du cas
 scoreg : score du groupe rajouté
 longg : longueur du groupe
 longtot : longueur totale du cas obtenu

5.4. Sélection de phrases

Pour diminuer les temps de traitement, nous effectuons une sélection à chaque étape de la reconnaissance. Les meilleurs squelettes sont sélectionnés pour être

complétés et les phrases obtenues sont classées par score de reconnaissance pour être ensuite interprétées. Il est donc nécessaire de calculer un score de reconnaissance pour chaque étape. Le score de reconnaissance d'un squelette de phrase ne dépend que des groupes reconnus, la formule retenue est :

$$\text{scoreph} = (\sum \text{scorecas} * \text{longcas}) / \text{lgtcas}$$

Pour une phrase complète, un paramètre supplémentaire intervient, c'est le taux d'occupation de la phrase par l'ensemble des cas retenus. Nous obtenons :

$$\text{scoreph} = \text{taux} * (\sum \text{scorecas} * \text{longcas}) / \text{lgtcas}$$

avec $\text{taux} = \text{lgtcas} / \text{longtotph}$

d'où la formule finale :

$$\text{scoreph} = (\sum \text{scorecas} * \text{longcas}) / \text{longtotph}$$

scoreph	: score de la phrase
scorecas	: score de chaque cas de la phrase
longcas	: longueur de chaque cas
longtotph	: longueur totale de la phrase
lgtcas	: longueur totale occupée par l'ensemble des cas.

Nous allons présenter les différentes stratégies que nous avons testées, en indiquant à chaque fois les avantages et les inconvénients liés :

5.4.1. Sélection de phrases dont le score est supérieur à un seuil

Toutes les phrases dont le score est supérieur à un seuil limite sont sélectionnées.

Le principal inconvénient de cette solution vient du fait que le seuil est défini de manière statique, donc

--> en cas de mauvaise reconnaissance des groupes, on peut arriver à un échec

--> en cas de bonne reconnaissance le nombre de phrases sélectionnées devient trop important et entraîne une explosion combinatoire lors de l'étape d'interprétation.

5.4.2. Sélection des N meilleures phrases

On sélectionne arbitrairement les N meilleures phrases trouvées.

L'inconvénient majeur vient du fait que plusieurs phrases peuvent avoir un score de reconnaissance identique si elles sont instanciées par les mêmes groupes mais avec des imbrications différentes. En choisissant N de manière arbitraire, une phrase représentant la bonne interprétation de l'énoncé peut être rejetée car elle se trouvait au rang N+1 alors qu'elle avait le même score que la phrase de rang N. Selon le même principe, on peut sélectionner des phrases dont le score est vraiment mauvais par rapport aux autres.

L'avantage de cette solution est qu'en fixant le nombre de phrases sélectionnées on maîtrise totalement les risques d'explosion combinatoire et de plus dans ce cas on est sûr d'obtenir des phrases à interpréter même si leur score ne dépasse pas un seuil déterminé.

5.4.3. Sélection de phrases adaptée aux résultats

L'idée est d'adapter la sélection aux résultats obtenus à l'étape de reconnaissance. Soit MSCORE le score de la meilleure phrase, on sélectionnera les phrases jusqu'à ce que :

--> l'écart entre MSCORE et celui de la phrase courante soit supérieur à un seuil donné,

--> ou l'écart entre les scores de deux phrases successives dépasse la limite autorisée

inconvénient :

- la sélection peut être arrêtée entre deux groupes de scores très proches (par le premier critère d'arrêt).

avantages :

- si les écarts autorisés sont correctement définis, on évite l'explosion combinatoire,

- on ne sélectionne que les " meilleures " phrases en se basant sur le score de la phrase la mieux reconnue, ce qui évite un échec de reconnaissance dans le cas

où les groupes composant la phrase sont mal reconnus où lorsque la phrase contient des mots parasites qui n'ont pu être instanciés correctement.-

5.4.4. Conclusion

La sélection de phrases adaptée aux résultats nous paraît être la solution la plus astucieuse, cependant, pour laisser à l'utilisateur le choix du mode de sélection, nous lui permettons de paramétrer la sélection de phrases complètes, et de définir le seuil d'admission dans le premier cas et le nombre de phrases dans le second cas. Pour la sélection adaptée aux résultats, les paramètres sont fixés par le système. Quelque soit la stratégie utilisée pour la sélection de phrases complètes, la sélection des meilleurs squelettes se fera toujours en fonction des résultats.

Chapitre 6 Interprétation et gestion du dialogue

Dans les chapitres précédents, nous avons présenté les différentes étapes permettant d'obtenir à partir du signal une phrase sous forme de structure casuelle. La structure obtenue peut être incomplète, ambiguë voire incompréhensible. Dans ce chapitre, nous allons présenter les moyens mis en œuvre pour :

- compléter les énoncés elliptiques ou anaphoriques de manière à les rendre interprétables ;
- résoudre les cas d'échecs et les ambiguïtés d'interprétation en utilisant une stratégie de dialogue permettant de limiter les échanges entre l'utilisateur et la machine ;

6.1. Construction d'énoncés complets

La commande fournie par le module de reconnaissance peut s'avérer incomplète. Nous avons vu que certaines ellipses ont été résolues au moment de l'étape de coordination et de qualification des groupes. Cependant, la commande obtenue peut encore contenir certaines ellipses et références dont la résolution nécessite la mise en œuvre de traitements syntaxiques, sémantiques et pragmatiques. Nous présentons dans un premier temps la résolution des ellipses de cas, puis dans un second temps la résolution des références.

6.1.1. Résolution des ellipses

Dans ce paragraphe, nous ne traiterons que les ellipses de cas complet, les ellipses de mots à l'intérieur d'un cas seront traitées dans le paragraphe présentant la résolution des références.

Si l'on se reporte à la terminologie utilisée dans le chapitre 4, il y a ellipse dans un énoncé lorsqu'un cas obligatoire ou semi-obligatoire est omis. Rappelons

brèvement qu'un cas est obligatoire lorsque sa présence est nécessaire (de manière explicite) pour la compréhension de l'énoncé, et qu'un cas est semi-obligatoire lorsqu'il est nécessaire de connaître sa valeur pour interpréter l'énoncé, mais que celle-ci peut se déduire du contexte syntaxico-sémantique ou pragmatique.

Si l'ellipse porte sur un cas semi-obligatoire, les trois possibilités à envisager sont les suivantes :

- une mauvaise reconnaissance de l'énoncé,
- l'utilisation d'une valeur par défaut,
- l'omission d'un cas énoncé dans une commande précédente.

Si l'ellipse porte sur un cas obligatoire, il faut envisager :

- une mauvaise reconnaissance de l'énoncé,
- un énoncé volontairement incomplet car non ambigu (ex : La porte !).

Nous allons présenter dans un premier temps le cas particulier de l'ellipse du prédicat, puis nous présenterons la résolution d'ellipses de cas semi-obligatoires par l'utilisation de valeurs par défaut et par recherche dans l'historique, et enfin nous présenterons la résolution des ellipses des cas obligatoires par l'intermédiaire de connaissances pragmatiques.

6.1.1.1. Cas particulier de l'ellipse du prédicat

Dans ce paragraphe, nous ne présentons que l'ellipse du prédicat due à une factorisation du prédicat précédent, l'ellipse de type conventionnel (ex : la porte!) étant traitée dans le paragraphe 6.1.1.3.

La reconnaissance de la commande est centrée sur le prédicat, bien que celui-ci puisse être éliminé. Pour que l'analyse de la phrase n'aboutisse pas à un échec dans ce cas de figure, le prédicat de la commande précédente est systématiquement rajouté à l'ensemble des prédicats reconnus dans l'énoncé. Lorsque l'énoncé contient effectivement une ellipse du verbe, il convient de faire hériter la nouvelle commande de tous les cas de la commande précédente qui n'ont pas été reformulés.

Ex : - *Pose le livre rouge sur la table près de la fenêtre ...*
- ... *et le cahier bleu*
--> *pose le cahier bleu sur la table près de la fenêtre.*

Cependant, si l'héritage de cas paraît évidente, l'ellipse du prédicat peut aussi entraîner une factorisation des groupes qualifiants.

Ex : - *Pose le livre sur la table près de la fenêtre*
- ... *et le classeur sur le bureau*
--> *Pose le classeur sur le bureau [près de la fenêtre]*

Cette factorisation n'est pas forcément automatique, c'est pour cela qu'à cette étape, nous nous contentons de définir pour chaque nouveau cas, l'ensemble des groupes qualifiants des cas correspondants dans l'énoncé précédent. Ces groupes ne seront utilisés que lors de l'interprétation si la phrase est ambiguë (cf § 6.2.3.1.).

Ex : - *Pose le livre qui est sur le bureau ,sur la table à côté de la fenêtre*
- ... *et le cahier aussi*
--> *Pose le cahier sur la table à côté de la fenêtre*
avec : Qualif-objet = sur le bureau.
si plusieurs cahiers sont candidats mais qu'un seul se trouve sur le bureau, c'est celui-ci qui est sélectionné

6.1.1.2. Valeur par défaut et recherche dans l'historique

Une ellipse de cas semi-obligatoire peut être due soit à l'utilisation d'une valeur par défaut, soit à la factorisation d'un cas défini dans une commande précédente.

Ex : - *Pose la chaise*
--> *valeur par défaut = par terre*

- *Pose le livre sur le bureau*
- *Pose le cahier*
--> *factorisation = sur le bureau.*

Les cas semi-obligatoires utilisés dans notre système sont les cas AGENT, BÉNÉFICIAIRE et DESTINATION. Nous présentons cas par cas le mode de résolution de l'ellipse.

- **CAS AGENT** : Dans un langage de type commande, le cas agent est celui qui est le plus souvent éli­di­é, en effet, les commandes étant à la forme impérative, l'agent est implicitement le robot.

Ex : - *Ouvre la porte.*

Le traitement de l'ellipse de l'agent se décompose en trois phases successives, la solution étant donnée par la première phase fournissant une valeur correspondant aux contraintes qui lui sont associées.

Les trois étapes sont :

- l'utilisation de la valeur par défaut (robot) si le prédicat se trouve en tête d'énoncé
- la recherche dans l'historique,
- une question de type QUI.

Les exemples suivants illustrent la résolution d'ellipse du cas agent par chacune des trois étapes :

Ex1 : - *Prends le livre sur le bureau*
--> *prédicat en tête de commande, utilisation de la valeur par défaut.*

Ex2 : - *Il pose le livre sur le bureau*
- ... *et le reprend*
--> *le prédicat n'est pas en tête, échec de la valeur par défaut*
--> *recherche dans l'historique : agent = IL*

Ex3 : - ... *prend le livre et le classeur*
--> *le prédicat n'est pas en tête, échec de la valeur par défaut*
--> *l'historique est vide, échec de la recherche dans l'historique*
--> *question : Qui prend le livre et le classeur ?*

- **CAS BÉNÉFICIAIRE** : comme pour le cas agent, le cas bénéficiaire a une valeur implicite dans notre type d'application, il s'agit ici du locuteur ce cas se résout en deux phases :

- recherche dans l'historique
- utilisation d'une valeur par défaut qui sera le locuteur si l'agent est le robot et le robot si l'agent est le locuteur.

Ex1 : - *Donne le livre à Pierre*
- *Donne aussi le cahier*

--> *recherche dans l'historique : bénéficiaire = Pierre*

Ex2 : - *Donne le livre*

--> *valeur par défaut : bénéficiaire = locuteur*

Ex3 : - *Je donne le livre*

--> *valeur par défaut : bénéficiaire = robot*

- **CAS DESTINATION** : l'é­li­si­on de ce cas est de loin la plus complexe à résoudre, voici plusieurs exemples permettant de mettre en évidence les différentes possibilités de résolution.

Ex1 : - *Pose le livre sur le bureau*
- *Pose le cahier aussi* --> *destination = sur le bureau*

Ex2 : - *Prends la chaise*
- ... *et pose la* --> *destination = par terre*

Ex3 : - *Prends le livre*
- ... *pose le* --> *destination = Où ?*

Ex4 : - *Tourne* --> *destination = Vers où ?*

Ex5 : - *Avance* --> *destination = Tout droit*

Nous pouvons remarquer que les valeurs par défaut utilisées (par terre, tout droit ...) dépendent d'une part du type du prédicat, et d'autre part de l'objet sur lequel celui-ci porte (lorsqu'il y a un cas objet associé au prédicat).

Nous avons vu dans le chapitre 4, que l'ensemble des prédicats est défini selon une hiérarchie correspondant à des classes et des sous-classes bien précises. Pour chaque classe ou sous-classe contenant un cas destination, nous définissons une méthode¹ qui sera invoquée lors de la résolution de l'ellipse.

¹ Une méthode est une fonction associée à une classe d'objets. Les sous-classes héritent de cette fonction à condition qu'elle ne soit pas redéfinie à leur niveau. Une méthode est activée par l'envoi de messages.

La figure 6.1. nous décrit les méthodes de résolution d'ellipses associées aux classes de prédicats ACTS (comme "poser", "mettre"...) et DEPL (comme "avancer", "tourner", "reculer" ...).

- | |
|--|
| <ul style="list-style-type: none">- Pour ACTS
rechercher la position normale de l'objet (ex : par terre pour une chaise).- Pour DEPL
rechercher la valeur par défaut de destination associée au prédicat. |
|--|

Figure 6.1. Description de deux méthodes de résolution d'ellipse du cas destination

Si l'appel d'une méthode ne résout pas l'ellipse, il convient alors de faire une recherche dans l'historique, si celle-ci échoue à son tour, une question de type Où est posée à l'utilisateur.

La recherche dans l'historique, quant à elle, se fait en partant de l'énoncé le plus récent dans la limite d'une certaine profondeur, en recherchant le premier cas correspondant au cas cherché.

6.1.1.3. Résolution pragmatique

Pour un cas obligatoire, ou pour un cas semi-obligatoire n'ayant pu être résolu par les techniques de recherche vues au paragraphe 6.1.1.2, il est nécessaire d'utiliser une stratégie pragmatique. Nous pouvons être en présence de deux cas différents vus au paragraphe 6.1.1, c'est-à-dire une partie de phrase non reconnue ou une ellipse conventionnelle. Dans les deux cas, les connaissances pragmatiques peuvent aider à résoudre l'ellipse ou tout au moins à faire des suppositions sur les éléments manquants.

Si le prédicat est éliidé :

- à partir du nom de l'objet, rechercher tous les prédicats applicables,
- pour chaque occurrence d'objet et pour chaque prédicat, rechercher si l'objet satisfait aux préconditions d'application du prédicat.

Si l'objet est éliidé :

- à partir du prédicat, rechercher tous les objets sur lequel il est applicable,
- pour chaque occurrence de chaque type d'objet, vérifier les préconditions d'application du prédicat.

Si la solution ainsi obtenue n'est pas unique, il faudra décider si le système choisit au hasard une des solutions possibles ou s'il demande au locuteur de répéter sa commande. Si les hypothèses émises par le système permettent de restreindre les suppositions à deux ou trois éléments, il peut être envisagé d'en choisir une au hasard en attendant la confirmation, mais au delà de ce nombre, il est préférable de poser une question au locuteur.

6.1.2. Résolution des références

Les références peuvent être de deux types : les références à un concept et les références à une occurrence. La résolution de ces deux types de références est développée dans les paragraphes suivants.

6.1.2.1. Référence à un concept

Une référence à un concept se fait par l'éliision d'un nom dans un groupe nominal (ex : le rouge, sur la petite...). Ce type de référence pourra être fait dans les cas OBJET, SOURCE et DESTINATION, en effet, dans notre application, il n'existe pas de référence conceptuelle dans un cas agent ou bénéficiaire comme dans la figure 6.2.

- | |
|---|
| <ul style="list-style-type: none">- Le grand garçon a pris le livre- ... le petit a pris le cahier
- le fils de Paul lit un livre- ... celui de Pierre joue dans le jardin |
|---|

Figure 6.2. Exemples de références conceptuelles du cas agent

Dans le chapitre 4, nous avons défini un ensemble de règles syntaxico-sémantiques permettant de résoudre les éliisions lorsque le groupe référent et le groupe elliptique sont coordonnés. Ces règles restent valables lorsque les groupes

ne sont pas des voisins directs. La figure 6.3. nous donne un exemple de règle de résolution de références que nous avons utilisée lors de l'étape de coordination de groupes.

GN + GNR :	
conditions --->	le GN et le GNR contiennent un qualificatif caractéristique d'une même propriété ou le GN contient un groupe imbriqué les deux groupes ont le même genre
transformations--->	transformer le GNR en groupe nominal en lui ajoutant le nom du GN
Ex :	le livre rouge et le bleu le livre qui est sur la table et le rouge

Figure 6.3. Exemple de règle de résolution d'ellipse

Les règles permettant de résoudre les ellipses étant connues, il reste à définir le mode de recherche du groupe référent. Les règles syntaxiques utilisées s'appuient sur un fort parallélisme de structure entre le groupe référent et le groupe elliptique ; cependant, on peut trouver le nom référent dans un groupe structurellement différent du groupe elliptique (Ex : le livre rouge (GN) sur le bleu (GPR)).

Le groupe référent est le groupe le plus proche possible structurellement et se trouvant le plus près possible du groupe elliptique. Il faut noter que si un groupe offre un fort parallélisme de structure avec le groupe elliptique mais que les règles de résolution ne peuvent s'appliquer, la recherche du groupe référent ne pourra s'effectuer qu'entre ce groupe et le groupe elliptique et a fortiori dans des groupes dont la structure s'éloigne de celle du groupe elliptique.

Ex : *Le livre sur le bureau et le cahier rouge sur le bleu*

Le groupe prépositionnel (sur le bureau) présente un parallélisme structurel avec le groupe elliptique (sur le bleu), cependant la règle de résolution (qui exigerait ici que le GP contienne un adjectif indiquant une couleur) n'est pas vérifiée donc le référent se trouve obligatoirement entre ces deux groupes, ici ce sera le GN (le cahier rouge)

Pour trouver le groupe référent, on donne la priorité au parallélisme de structure et ce n'est qu'en cas d'échec que la recherche s'effectue dans les autres groupes. Nous allons définir l'ordre dans lequel les différents cas de la commande courante et de la commande précédente sont étudiés pour trouver le référent, en illustrant chaque possibilité par des exemples. Dans l'ordre nous étudions :

- Le cas du groupe elliptique s'il y a une coordination ou si l'ellipse est dans une imbrication d'un GP.

Ex 1 : - *Prends (le livre sur la table rouge et le cahier sur la noire)*
(objet - (GN (GP)) et (GN (GPR)))

--> *sur la table noire*

Ex 2 : - *Pose le livre sur la table noire à côté de la grise*
(dest - (GP (GPR)))

--> *a côté de la table grise*

Ex 3 : - *Prends le livre sur le bureau et le cahier bleu sur le rouge*
(objet - (GN (GP)) et (GN (GPR)))

--> *sur le cahier rouge*

- Les cas de la commande dont les groupes ont une structure proche de celle du groupe elliptique.

Ex 4 : - *Déplace l'armoire de la grande pièce vers la petite*
(source - GP) (dest - GPR)

--> *vers la petite pièce*

- Les cas de la commande précédente dont les groupes ont une structure proche de celle du groupe elliptique.

Ex 5 : - *Pose le livre sur la grande table* (dest - GP)
- *Pose le cahier sur la petite* (dest - GPR)

--> *sur la petite table*

- Les cas de la commande courante dont les groupes ont une structure différente de celle du groupe elliptique ainsi que le groupe elliptique lui-même.

Ex 6 : - *Ouvre la fenêtre*
- *Pose le livre rouge sur le bleu* (objet - GN) (dest - GPR)

--> *sur le livre bleu*

Ex 7 : - *Ferme la porte*
- *Prends le livre rouge sur le bleu*

(objet - (GN (GPR)))

--> sur le livre bleu

- Les cas de la commande précédente dont les groupes ont une structure différente de celle du groupe elliptique.

Ex 8 : - Prends le livre sur la table noire (objet - (GN (GP)))

- Pose le cahier sur la grise (dest - GPR)

--> sur la table grise

Ex 9 : - Prends le livre rouge (objet - GN)

- Pose le cahier sur le bleu (dest - GPR)

--> sur le livre bleu

Dans le cas où plusieurs groupes sont candidats, nous appliquons les heuristiques suivantes :

- si le groupe elliptique et les référents possibles sont des groupes prépositionnels, on choisira comme groupe référent celui qui a la même préposition que le groupe elliptique.

Ex 10 : - Pose le livre sur la table rouge près de la fenêtre verte

- Pose le cahier sur la noire

--> sur la table noire

- si le test sur la préposition ne peut être fait (cas des GN) ou n'est pas vérifié (prépositions différentes), le groupe choisi sera celui qui est le plus proche du groupe elliptique.

Ex 11 : - Pose le livre sur le bureau noir

à côté du fauteuil rouge

- Pose le sac contre le bleu

--> contre le fauteuil bleu

Nous n'avons pas jugé nécessaire d'approfondir ces heuristiques car nous supposons que l'utilisateur est coopératif et donc que les informations qu'il donne sont pertinentes et évitent au maximum les ambiguïtés, ainsi l'exemple 11 devient pour un locuteur coopératif :

Ex 11' : - Pose le livre sur le bureau noir

à côté du fauteuil rouge

- Pose le sac contre le fauteuil bleu

Lorsque le nom référent est trouvé, celui-ci est rajouté au groupe elliptique afin de former un groupe complet et permettre ainsi son interprétation. C'est pour cette raison que nous ne remontons pas au delà de la commande précédente pour résoudre les références conceptuelles. En effet, la séquence de commandes suivante :

- Prends le livre noir

- Prends le bleu

- Pose le rouge

est conservée dans l'historique de la façon suivante :

- Prends le livre noir

- Prends le livre bleu

- Pose le livre rouge

Lors de l'étape de coordination et de qualification de groupes, nous avons choisi de supprimer les factorisations de groupes qualificatifs, ainsi l'énoncé :

- Prends le livre et le cahier sur la table

est "développé" de la manière suivante :

- Prends le livre sur la table et le cahier sur la table

Si le groupe qualifiant est un groupe elliptique, il faut rajouter le nom résolvant l'ellipse dans chacun des groupes coordonnés de manière à éviter de recommencer plusieurs fois la recherche du référent :

Ex 12 : - Prends le livre sur le bureau noir

- Prends le cahier et le classeur sur le gris

1^{ère} étape --> Prends le cahier sur le gris et le classeur sur le gris

2^{ème} étape --> Prends le cahier sur le bureau gris
et le classeur sur le bureau gris

Nous avons supposé que l'éllision se faisait uniquement sur le nom du groupe nominal, elle peut cependant se faire aussi sur le complément du nom (groupes qualificatifs) comme dans l'exemple ci-dessous :

Ex 13 : - Prends le livre rouge sur le bureau

- Ouvre le bleu

--> le livre bleu sur le bureau

Pour tenir compte de ces éllisions éventuelles, nous procédons de la même manière

que pour l'éllision du prédicat, c'est-à-dire que lorsqu'un groupe elliptique est complété par un groupe de même niveau appartenant au même cas mais dans l'énoncé précédent, nous conservons les éventuels compléments du nom pour résoudre les ambiguïtés qui peuvent apparaître lors de l'interprétation de la commande.

6.1.2.2. Référence à une occurrence

La résolution d'une référence à une occurrence consiste à retrouver un ou plusieurs objets ou personnes dont on a déjà parlé ou qui sont définis implicitement. Les cas pouvant faire l'objet de référence à une occurrence sont les cas AGENT, BÉNÉFICIAIRE, OBJET, SOURCE et DESTINATION.

- Cas AGENT et BÉNÉFICIAIRE

Une référence à un agent va être de la forme : "je", "tu" ... dans ce cas, il ne faut pas rechercher dans l'historique le référent, mais simplement dans le sens lié aux mots. Pour cela il faut rechercher la correspondance dans le modèle de l'univers. Pour tous les mots susceptibles de jouer un rôle d'agent dans une forme de type "ref" (référence à une occurrence), il faut associer dans le lexique la valeur d'instanciation.

ex : je, moi, me... --> locuteur
tu, toi, te ... --> robot

Pour les références de la forme : "il", "elle"... il est nécessaire de faire une recherche dans l'historique, cependant dans notre application qui ne concerne qu'un robot et un utilisateur, ce genre de référence (pour un agent) ne doit pas exister.

Il en est de même pour le cas BÉNÉFICIAIRE où les références seront du type : "moi", "toi", "à moi" ...

- Cas OBJET

Rappelons la syntaxe du groupe nominal faisant référence à une occurrence :

GNREF --> pronp⁺ / art adjn⁺
Ex : *le, les deux, une*

La recherche du référent d'un cas objet se fait en tenant compte des

contraintes pragmatiques associées au prédicat, par exemple pour le prédicat ALLUMER, les objets recherchés seront du type lampe, télévision... Le référent est donc le premier nom figurant dans un cas objet et satisfaisant les contraintes pragmatiques associées au prédicat. Cependant ceci ne s'applique que si le référent est unique, s'il y a plusieurs référents possibles, il est nécessaire de les déterminer précisément.

Ex 1 : - *Prends le livre rouge*
- *Prends le classeur et le cahier sur la table*
- *Pose les sur le bureau*
les --> le livre, le classeur et le cahier

Ex 2 : - *Prends le livre sur le bureau*
- *Pose le sur la table*
- *Prends le classeur et le cahier sur la table*
- *Mets les dans le tiroir*
les --> le classeur et le cahier

Lorsque le groupe est au pluriel, il faut rechercher tous les objets concernés, pour cela on recherche le premier objet satisfaisant les contraintes pragmatiques du prédicat, si le cas objet est instancié par une conjonction d'objets, ils sont tous sélectionnés et la recherche continue en remontant dans l'historique tant que le prédicat de la commande est égal au prédicat de celle ayant fourni les premiers objets de référence (dans l'exemple 1, on étudie toutes les commandes dont le prédicat est PRENDRE, dans l'exemple 2, on fait le même traitement mais on s'arrête dès que l'on rencontre un autre prédicat, ici POSE).

L'utilisation d'un adjectif numérique permet de faire une vérification du nombre de référents. Si N est la valeur de l'adjectif numérique et que l'article est défini, on sélectionne les N objets satisfaisant les contraintes pragmatiques.

- Cas SOURCE et DESTINATION

ex : - *Ouvre le tiroir.*
- *Mets le livre dedans*

La référence à une occurrence dans un cas source ou destination se rapporte toujours à une occurrence d'un cas objet d'un énoncé précédent. Les autres possibilités relevant d'une mauvaise construction

syntaxique ne sont pas prises en compte ici. Le groupe peut prendre deux formes :

GPREF --> adv +s / prep+s GNREF

Dans le cas d'un adverbe, il faut effectuer la recherche du référent de la même manière que pour le cas général de la résolution du cas OBJET mais sans tenir compte des contraintes pragmatiques et en sélectionnant toutes les occurrences concernées, lorsque l'ensemble des occurrences esr défini, il suffit de leur rajouter la préposition associée à l'adverbe (ex : dessus --> sur bureau14). Dans le cas de la deuxième forme possible (prep + GNREF), le traitement est identique au traitement du cas OBJET (sans les contraintes pragmatiques) et la préposition est rajoutée aux occurrences trouvées.

En plus de toutes les vérifications sémantiques faites au cours de la recherche du référent, des vérifications d'ordre syntaxique (accord en genre et en nombre) sont effectuées, lorsque cela est possible, pour renforcer la validité du référent.

Contrairement à la résolution des références à un concept, la résolution de références à une occurrence ne modifie pas la forme de l'énoncé, les occurrences servent d'interprétation du groupe, si bien que lorsqu'une même référence apparaît plusieurs fois dans une séquence de commandes, le travail de recherche est simplifié, il suffit de reprendre la dernière interprétation possible.

Ex : - Prends le livre rouge sur le bureau
- Pose le sur la table
--> le = le livre rouge
- Ouvre le
--> le = le livre rouge

Lorsqu'une commande admet plusieurs références à des occurrences, il faut adopter la règle suivante : une occurrence ne peut être utilisée en tant que référent qu'une seule fois dans la même commande.

Ex : - Ouvre le tiroir
- Prends le cahier
- Pose le dedans
--> le = le livre, dedans = dans le tiroir

6.2. Interprétation des commandes

6.2.1. Confrontation au modèle de la tâche

La confrontation au modèle de la tâche permet de faire une vérification pragmatique qui consiste à vérifier que l'objet sur lequel porte l'action demandée est bien du type autorisé pour cette action. Cette vérification se fait en parallèle avec l'étape de résolution des références. Elle doit se faire avant la recherche de l'occurrence concernée car si la commande ne s'applique pas au type d'objet cité, il est inutile de perdre du temps à rechercher une occurrence qui ne permettra pas de valider le cas.

Ex : - *Allume la petite table rouge près de la fenêtre*

Il est inutile de rechercher la table concernée avec tous les risques d'ambiguïté ou d'échec que cela peut entraîner alors que la commande est invalide. Le système doit fournir la réponse suivante :
- *Allumer ne s'applique pas aux objets de type table.*

6.2.2. Résolution des groupes complexes

La résolution d'un groupe consiste à trouver la ou les occurrences concernées (figure 6.4.).

- sur le bureau	---> sur bureau15	sur bureau18
- la petite table rouge près de la fenêtre	---> table27	près de fenêtre13

Figure 6.4. Exemples de résolution de groupes.

Pour un groupe simple (sans imbrication), il suffit de rechercher dans la liste des occurrences de l'objet concerné celui ou ceux dont les propriétés correspondent à la description (figure 6.5.).

- la petite table rouge
 ---> sélection de la liste des occurrences des tables
 ---> recherche des occurrences dont les propriétés
 taille et couleur valent respectivement petit et rouge.

Figure 6.5. Résolution d'un groupe simple.

Pour des groupes complexes (avec imbrication), il faut commencer par rechercher les objets correspondant au niveau le plus profond de la description et remonter jusqu'au niveau supérieur en sélectionnant les objets correspondant aux descriptions faites (figure 6.6).

- le livre rouge sur la table près de la fenêtre
 ---> recherche de toutes les fenêtres
 ---> recherche et sélection des tables se trouvant près d'une
 des occurrences de fenêtre
 ---> recherche et sélection des livres rouges se trouvant sur
 une des occurrences de table sélectionnée à l'étape
 précédente.

Figure 6.6. Résolution d'un groupe complexe

Lorsqu'un cas est instancié par plusieurs groupes coordonnés, si la conjonction est un "et", chacun des groupes sera instancié, si la conjonction est un "ou" l'instanciation s'arrête lorsqu'une solution satisfaisante est trouvée.

Nous avons vu dans le chapitre 4 que certains phénomènes de factorisation avaient été résolus au niveau syntaxique.

Ex : - le livre et le cahier sur le bureau
 ---> le livre **sur le bureau** et le cahier sur le bureau.

Mais dans ce cas, une contrainte sémantique importante disparaît celle qui consiste à instancier le groupe mis en facteur une seule fois de la même manière pour chacune de ses formes développées (figure 6.7). Cette contrainte doit être réintroduite pour éviter d'engendrer des phénomènes d'ambiguïté artificielle. La règle qui s'applique dans ce cas est la suivante :

- si deux groupes coordonnés sont qualifiés par le(s) même(s)
 groupe(s), l'instanciation de la qualification doit être identique pour
 les deux groupes.

- Le livre sur le bureau et le cahier sur le bureau
 ---> livre1 sur bureau1 cahier1 sur bureau2
 livre2 sur bureau2
 ===> livre2 sur bureau2 et cahier1 sur bureau2

Figure 6.7. Application de la règle de développement

Pour chaque cas instancié nous calculons un score d'interprétation permettant de rendre compte des cas d'échecs et d'ambiguïtés. A partir de ces différents scores, nous calculons le score global d'interprétation de la phrase, ce qui permet de classer les unes par rapport aux autres les phrases obtenues par les combinaisons différentes des mêmes groupes. Le meilleur score obtenu correspondra à une phrase dont chaque cas instanciable est réalisé de manière optimale. pour chaque ambiguïté ou échec, le score d'interprétation est pénalisé.

Il y a ambiguïté lorsque le groupe décrit un objet et un seul et que plusieurs occurrences satisfont la description, il faut rendre la pénalité affligée proportionnelle au nombre d'objets en conflit. Lorsqu'un groupe est de type indéfini (ex : un livre) ou au pluriel (ex : les livres) le nombre de réalisations n'entre pas en compte, il n'y a pas d'ambiguïté, dans le premier cas, il suffit de choisir une occurrence parmi celles sélectionnées et dans le second cas il faut les sélectionner toutes.

Il y a échec lorsqu'une tentative d'instanciation n'a reconnu aucun objet de l'univers (ex : le livre ---> pas de livre dans l'univers).

Nous avons pris comme convention d'affecter un score nul à une phrase interprétée de manière satisfaisante et un score positif pour à une phrase incorrecte. Le score d'un cas est calculé de la manière suivante :

- pour un cas d'ambiguïté : $sc = coefa * (nbocc-1)$
 - pour un cas d'échec : $sc = coefe$
- avec coefa : coefficient d'ambiguïté
coefe : coefficient d'échec
nbocc : nombre d'occurrences

Le score de la phrase est obtenu en cumulant les scores d'interprétation de chaque cas.

6.2.3. Résolution des ambiguïtés

Le score d'interprétation d'une phrase nous permet de faire un classement parmi toutes les phrases ayant été sélectionnées par l'analyseur syntaxico-sémantique. Si l'on se réfère aux maximes de qualité et de quantité de Grice, les informations fournies par le locuteur sont suffisantes tant en qualité qu'en quantité pour interpréter correctement la phrase. Il paraît donc naturel de considérer que la phrase la mieux interprétée correspond à ce que voulait dire le locuteur. Cependant, il se peut que certaines phrases restent ambiguës, mais si l'on considère que les maximes de Grice sont respectées, il faut se donner les moyens d'affiner l'interprétation pour résoudre (ou réduire au maximum) les cas d'ambiguïtés.

Nous avons défini quatre étapes permettant de réduire les ambiguïtés. Ces étapes ne sont pas concurrentes mais complémentaires. Le système étant amené à faire des hypothèses, l'utilisateur a la possibilité de contester ce choix. Dans les quatre paragraphes suivants, nous présentons les différentes étapes de résolution des ambiguïtés et dans le cinquième, la gestion des contestations.

6.2.3.1. Recherche des compléments

La première étape consiste à vérifier s'il n'existe pas de groupe qualificatif en attente pour ce cas. En effet, nous avons vu que lors d'une ellipse du prédicat, les groupes qualificatifs de certains cas sont stockés de manière à pouvoir aider à lever les ambiguïtés :

Ex : - *Prends le livre sur le bureau*

- ... *et le cahier*

Qualif-objet = sur le bureau

Il en est de même lors de la résolution de références à un concept.

Si les qualifications existent elles sont rajoutées au groupe ambigu. Si le nombre d'occurrences en est diminué les qualificatifs sont conservés, par contre si l'interprétation aboutit à un échec (aucune occurrence) ils sont rejetés.

6.2.3.2. Recherche dans l'historique

Si la qualification n'existe pas ou ne permet pas de résoudre l'ambiguïté, la deuxième étape consiste à rechercher dans l'historique un des objets sur lesquels porte l'ambiguïté. En effet, si l'utilisateur vient de parler d'un objet précis celui est considéré comme un focus possible de la conversation, ce seul argument suffit à le caractériser vis-à-vis des autres objets.

Ex : - *Ferme le livre rouge qui est sur le bureau*

- *Prends le cahier sur la table*

- *Range le livre dans le tiroir*

--> *il s'agit du livre que l'on vient de fermer*

6.2.3.3. Confrontation entre le modèle de la tâche et l'univers

Nous sommes donc en présence d'une commande s'appliquant à un objet sur lequel porte le conflit. Nous ne nous intéressons ici qu'aux ambiguïtés portant sur le cas objets, celles-ci étant les seules solubles par cette méthode. La recherche des occurrences satisfaisant la description fournie par le cas objet a plusieurs solutions possibles. La confrontation va consister à rechercher dans le modèle de la tâche les préconditions que doit vérifier l'objet concerné pour que la prédicat soit applicable, puis à rechercher dans l'univers, parmi les occurrences sélectionnées, celles qui vérifient ces préconditions. Un exemple est donné à la figure 6.8.

```

- Ferme la porte
  occurrences possibles : porte1 porte2 porte3
  précondition          : état(objet) = ouvert
  univers               : état(porte1) = fermé
                       : état(porte2) = ouvert
                       : état(porte3) = fermé
  --> l'ambiguïté est levée, l'objet sélectionné est : porte2

```

Figure 6.8. Résolution d'ambiguïté par confrontation avec le modèle de la tâche et l'univers.

Dans le meilleur des cas la confrontation permet de lever totalement l'ambiguïté, dans d'autres cas elle permet d'améliorer la situation en diminuant le nombre d'objets en conflit et parfois elle ne résout pas le problème du tout. Lorsque la confrontation n'a pas permis de lever complètement l'ambiguïté, il faut utiliser une autre stratégie que nous développons dans le paragraphe suivant.

6.2.3.4. Mise en évidence de la pensée du locuteur

La solution la plus naturelle consiste à choisir une occurrence parmi celles qui sont en conflit et de la soumettre au locuteur en attente d'une confirmation ou d'une infirmation.

Ex : - Prends le livre sur le bureau
 - le livre rouge?
 - non, le noir

Cependant tous les cas d'ambiguïté ne sont pas aussi simples. En effet, les objets en conflit n'ont pas forcément une caractéristique qui les différencie les uns des autres (comme la couleur dans l'exemple, il pourrait y avoir plusieurs livres rouges). De plus lorsque la description est complexe (groupes imbriqués et coordonnés), il est nécessaire d'adopter une stratégie pour poser une question qui soit la plus pertinente possible.

D'après les maximes de Grice [Grice-75], les informations fournies par le locuteur sont, par hypothèse, suffisantes pour déterminer l'objet, tant en qualité qu'en quantité ; donc s'il y a un conflit, c'est suite à une erreur du locuteur. Si l'on reprend l'exemple : **le livre sur le bureau près de la fenêtre**, et que l'on constate

qu'il existe deux bureaux, dont l'un supporte deux livres et l'autre un seul, il paraît évident que le locuteur parle de celui qui est seul sur le bureau. En effet, si le locuteur porte son attention sur le bureau ne supportant qu'un seul livre, il n'éprouve pas le besoin d'apporter des informations supplémentaires. Le but que nous cherchons à atteindre est de sélectionner l'ensemble d'objets sur lequel le locuteur focalise son attention, de manière à ce que la question qui va lui être posée ait un maximum de chances de recevoir une confirmation (ce qui est plus facile à traiter qu'une contestation)

Lorsqu'un objet est décrit de manière complexe, il paraît évident que les objets du niveau le plus profond sont les plus facilement caractérisables, les autres ayant été décrits à partir d'eux!

Nous allons essayer de définir une stratégie permettant de sélectionner l'objet sur lequel le système doit poser une question en cas d'ambiguïté. Nous allons représenter les instanciations des objets grâce à un arbre. La figure 6.9, nous montre un exemple d'arbre d'instanciation du groupe : le livre sur le bureau près de la fenêtre. Une occurrence est désignée par l'initiale de l'objet suivie d'un numéro (par exemple F1 pour la fenêtre 1).

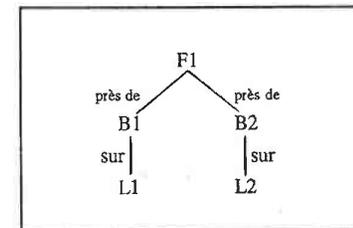


Figure 6.9. Arbre d'instanciation

La technique de sélection consiste dans un premier temps à sélectionner dans l'arbre d'instanciation, la (ou les) branche(s) optimisant la description, c'est-à-dire, celle(s) qui n'engendrent (nt) qu'une seule feuille et qui soi(en)t la plus longue possible. Si les maximes de quantité sont respectées, l'objet désigné par le locuteur doit être une feuille figurant au bout d'une de ces branches. C'est

pourquoi la question posée portera sur un objet d'une branche sélectionnée, ceci dans le but d'augmenter les chances de confirmation.

Lorsque les branches sont sélectionnées, il faut déterminer sur quel type d'objet va porter la question (i.e. à quel niveau de profondeur de l'arbre). Pour cela, on recherche à partir des feuilles, s'il existe une caractéristique permettant de faire la discrimination entre les différents objets du même niveau. A chaque niveau testé, nous ne nous intéressons qu'aux occurrences n'engendrant qu'une seule feuille ; donc, la discrimination doit porter sur chacune de ces occurrences d'une part, et sur l'ensemble des autres occurrences, d'autre part. Les occurrences n'engendrant qu'une seule feuille sont regroupées dans ce que nous appelons l'espace de discrimination stricte. La figure 6.10 illustre, en utilisant l'espace de discrimination { L1, L2 }, les contraintes que doivent vérifier les valeurs d'une caractéristique pour que celle-ci soit considérée comme discriminante.

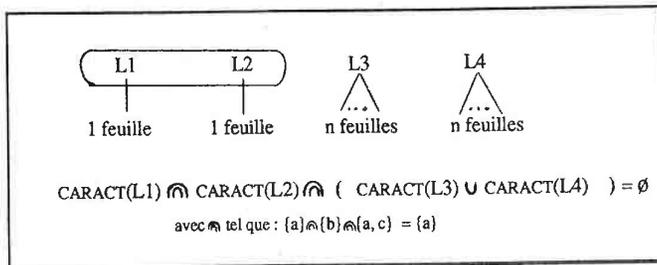


Figure 6.10. Contraintes de discrimination

La question posée portera sur une occurrence se trouvant à l'intersection d'une branche sélectionnée et d'un espace de discrimination stricte. La figure 6.11 illustre différents cas d'instanciation menant à une ambiguïté (à partir du groupe donné dans l'exemple), les branches sélectionnées sont entourées verticalement et les espaces de discrimination stricte, horizontalement ; à chaque niveau, on indique l'ensemble des objets sur lesquels on peut poser une question et les contraintes que doit vérifier la caractéristique retenue.

arbre d'instanciation	objets de la question	contraintes sur CARACT
	F2 B3 L4	$\text{CARACT}(F2) \cap \text{CARACT}(F1) = \emptyset$ $\text{CARACT}(B3) \cap \text{CARACT}(B2) \cap \text{CARACT}(B1) = \emptyset$ $\text{CARACT}(L1) \cap \text{CARACT}(L2) \cap \text{CARACT}(L3) \cap \text{CARACT}(L4) = \emptyset$
	F1, F2 B1, B2 L1, L2	$\text{CARACT}(F1) \cap \text{CARACT}(F2) = \emptyset$ $\text{CARACT}(B1) \cap \text{CARACT}(B2) = \emptyset$ $\text{CARACT}(L1) \cap \text{CARACT}(L2) = \emptyset$
	\emptyset B1 L1	\emptyset $\text{CARACT}(B1) \cap (\text{CARACT}(B2) \cup \text{CARACT}(B3)) = \emptyset$ $\text{CARACT}(L1) \cap \text{CARACT}(L2) \cap \text{CARACT}(L3) \cap \text{CARACT}(L4) \cap \text{CARACT}(L5) = \emptyset$

Figure 6.11. Exemples de cas d'instanciation

6.2.4. Résolution des cas d'échecs

En cas d'échec d'instanciation, le locuteur doit être informé sur la cause de cet échec. Les exemples suivants illustrent les différents types d'explications qui peuvent lui être fournies :

- le livre rouge
--> il n'y a pas de livre rouge
- le livre rouge sur le bureau près de la fenêtre
--> il n'y a pas de fenêtre
ou --> il n'y a pas de bureau près de la fenêtre

ou --> il n'y a pas de livre rouge sur le bureau près de la fenêtre

Pour obtenir ces informations, l'analyseur construit l'arbre des occurrences d'objets en commençant par les niveaux les plus profonds. Pour l'exemple : *le livre rouge sur le bureau près-de la fenêtre*, il recherche dans un premier temps toutes les fenêtres, puis tous les bureaux se trouvant près de l'une de ces fenêtres et enfin tous les livres rouges se trouvant sur un des bureaux ainsi déterminés. Le diagnostic de l'échec se fait à partir du premier ensemble d'occurrences vide.

Dans les cas d'échec, on peut aussi apporter une aide à l'utilisateur dans la mesure du possible :

Ex : le livre sur le bureau près de la fenêtre

---> il n'y a pas de livre sur le bureau près de la fenêtre, mais il y en a un sur le bureau noir.

Le problème qui se pose est l'explosion combinatoire, en effet, il n'est pas question d'indiquer tous les livres qui se trouvent dans l'univers du robot, mais seulement ceux qui respectent le "mieux" la description, à condition qu'ils ne soient pas en trop grand nombre. Pour pouvoir faire ce genre de propositions au locuteur, il faut faire une recherche ascendante, contrairement à l'analyse descendante utilisée pour l'instanciation, c'est-à-dire partir des feuilles possibles de l'arbre d'instanciation et remonter jusqu'à un état d'échec. Si juste avant l'échec, le nombre de solutions possibles est restreint (une ou deux solutions par exemple), une proposition est faite à l'utilisateur.

6.2.5. Gestion des contestations

Nous venons de présenter les différentes méthodes utilisées pour compléter les énoncés incomplets ou elliptiques et pour lever les ambiguïtés. Ces méthodes utilisent des heuristiques qui peuvent entraîner une mauvaise déduction du système, c'est pourquoi l'utilisateur peut contester l'interprétation qui a été faite. Une contestation devra toujours commencer par le mot réservé NON de manière à pouvoir la différencier des autres commandes. Lorsque le NON est détecté, la nouvelle commande est analysée. Si le robot a fait une hypothèse sur une caractéristique d'un objet pour lever une ambiguïté, l'analyseur se met en attente d'une contestation portant sur cette caractéristique de manière à accélérer le

traitement. Si la contestation ne porte pas sur une hypothèse, le système analyse la commande de la même manière qu'une commande classique, mais dans un contexte restreint à la commande précédente. Le nouvel énoncé, une fois complété, remplace entièrement le précédent.

Si la nouvelle commande n'est pas une contestation, le robot considère que son interprétation est correcte et la valide en l'ajoutant dans l'historique.

Chapitre 7

Résultats et implémentation

Dans ce chapitre, nous allons présenter les résultats obtenus par notre système. Dans un premier temps, nous allons fournir quelques exemples de génération de phonèmes à partir du texte écrit. Nous présenterons ensuite un exemple de résultat détaillé de la phase de reconnaissance, puis des résultats de plusieurs types de phrases (simples, complexes, correctes, incomplètes ...).

Nous présenterons ensuite les résultats obtenus lors de l'interprétation des phrases sélectionnées, et plus particulièrement la résolution de groupes simples et complexes, et la résolution de certaines formes d'ellipses.

Dans la dernière partie nous décrirons l'implémentation du système ainsi que les modules en cours de développement.

7.1. Présentation générale

7.1.1. Génération de phonèmes

Comme nous l'avons signalé au chapitre 5, le système n'est pas connecté à un module de décodage acoustico-phonétique. A partir d'un énoncé écrit, nous générons la suite des phonèmes le représentant. Pour simuler les résultats du décodeur, chaque phonème est fourni avec deux indices correspondant aux bornes du phonème sur le signal. Ces bornes étant utilisées pour déterminer la succession possible de deux phonèmes ou pour mesurer leur écart, nous avons choisi de prendre la même valeur pour les bornes inférieure et supérieure de chaque phonème et d'incrémenter cette valeur de 1 à chaque nouveau phonème.

Voici quelques exemples de phrases et la suite de phonèmes associés. (La codification est celle définie en annexe.)

Ex 1 : Le soleil brille et les oiseaux chantent

(1, 1, l) (2, 2, &
(3, 3, s) (4, 4, o) (5, 5, l) (6, 6, e) (7, 7, j)
(8, 8, b) (9, 9, r) (10, 10, i) (11, 11, j) (13, 13, e)
(14, 14, l) (15, 15, e)
(16, 16, w) (17, 17, a) (18, 18, z) (19, 19, o)
(20, 20, ch) (21, 21, an)(22, 22, t)

Dans les exemples suivants, nous nous contenterons de mettre la suite de phonèmes, sans les bornes inférieures et supérieures.

Ex 2 : La fouille du secteur

l / a / f / u / j / d / y / s / e / k / t / e / u / r /

Ex 3 : On ira tous au paradis

o n / i / r / a / t / u / o / p / a / r / a / i /

On peut remarquer que le "s" de tous n'est pas généré, en effet, dans le cas général, un "s" en fin de mot ne se prononce pas, pour obtenir la forme souhaitée, il faudra écrire : "tousse"

Ex 4 : Le jeune cobaye vert est mouillé

l / & / g h / e u / n / k / o / b / a / j / v / a i / r / e / m / u / j / e /

Ex 5 : Ouvre la porte

u / v / r / l / a / p / o / r / t /

7.1.2. Résultats détaillés de la reconnaissance

Le module de reconnaissance de phrases utilise et fournit de nombreuses informations. Sur un exemple simple (nous choisirons la phrase "ouvre la porte"

dont la représentation phonétique est donnée ci-dessus), nous allons détailler les différentes étapes de reconnaissance ainsi que la structure finale fournie au module d'interprétation.

7.1.2.1. Structure du mot

Les mots reconnus sont caractérisés par :

- l'indice de début du premier phonème,
- l'indice de fin du dernier phonème,
- le score de reconnaissance du mot,
- le mot.

Voici quelques exemples de mots reconnus dans la suite de phonèmes représentant la phrase "ouvre la porte".

(2	3	.895	nous)
(4	6	.9785125	lampe)
(6	9	1.	porte)

Nous pouvons remarquer que les scores sont conservés tels qu'ils sont fournis par le module de reconnaissance de mots, ils ne sont pas arrondis de manière à obtenir un score de phrase très précis.

7.1.2.2. Structure du groupe

A partir des mots, le module de reconnaissance construit des groupes, voici la structure du groupe nominal :

Ex 6 : la porte

((gn p) 1. f s (art u) (nom ps)
((4 5 1 la) (6 9 1 porte))

Les informations fournies indiquent que nous sommes en présence d'un groupe nominal dont le score de reconnaissance vaut 1, qu'il est féminin et singulier, qu'il est composé d'un article (élément dont la présence est facultative) et d'un nom (élément dont la présence est obligatoire et porteur de la sémantique du groupe) et que les mots reconnus sont "la" avec un score de 1 entre les phonèmes 4 et 5, et "porte" avec un score de 1 entre les phonèmes 6 et 9.

L'exemple suivant nous montre la structure d'un groupe prépositionnel.

Ex 7 : Sur le bureau noir

```
( (gp p) 1 n n (prep ps)
  ( (gn p) 1 m s (art u) (nom ps) (adjp m) )
  ( (10 12 1 sur)
    ((13 14 1 le) (15 18 1 bureau) (19 22 1 noir)))
```

Dans l'exemple suivant, nous présentons un groupe nominal qualifié par un groupe prépositionnel.

Ex 8 : Le livre sur la table

```
( (gn p) 1 m s (art u) (nom ps)
  ( (gp p) 1 n n (prep ps)
    ( (gn p) 1 f s (art u) (nom ps) ) )
  ( (5 6 1 le) (7 10 1 livre)
    ( (1 1 13 1 sur)
      ((14 15 1 la) (16 19 1 table)) ) )
```

7.1.2.3. Structure de la phrase

La phrase est décomposée en un prédicat et un ensemble de cas, chaque cas étant instancié par un groupe de mots. Dans la commande "ouvre la porte", il y a le prédicatif "ouvrir" et le cas objet instancié par le groupe nominal "la porte". La représentation interne de cette structure est la suivante :

Ex 9 : Ouvre la porte

```
( (score 1)
  (cas ( (verbe
          ( (gv p) 1 n n #:verbe:pred:act:acts:#[1 0 0] p)
          ( (1 3 1 ouvrir) )
          (objet
            ( (gn p) 1 f s (art u) (nom ps)
              ((4 5 1 la) (6 9 1 porte))) ) )
```

Les informations obtenues indiquent que la phrase a été reconnue avec un score de 1, et que la liste des cas l'instanciant est composée d'un verbe et d'un objet.

Dans les exemples suivants, nous ne ferons figurer que les éléments nécessaires à la mise en évidence de certains résultats ainsi que les éléments importants caractérisant la phrase. Pour la commande "ouvre la porte", nous fournirons par exemple la structure simplifiée suivante :

Ex 10 : Ouvre la porte

```
( 1
  ( (verbe (gv 1) (ouvrir))
    (objet (gn 1) (la porte) ) ) )
```

Ex 11 : Pose le livre sur la table

```
(1.
  ((verbe (gv 1.) (poser))
    (objet (gn 1.) (le livre))
    (dest (gp 1. (gn 1.)) (sur (la table))))))
```

7.2. Résultats de la reconnaissance

Nous allons présenter, dans les paragraphes suivants, les résultats obtenus par le module de reconnaissance, au travers d'exemples de plusieurs types de phrases. Nous traiterons dans un premier temps des phrases simples et dont les phonèmes sont correctement reconnus, ensuite, pour valider le module de reconnaissance de mots, nous présenterons des exemples de commandes contenant des insertions, des omissions ou des confusions de phonèmes. Pour montrer la souplesse de la grammaire, nous donnerons quelques exemples de phrases dont les constructions syntaxiques sont incorrectes. Enfin nous présenterons la reconnaissance de phrases contenant des expressions complexes puis de phrases contenant des expressions factorisées. Dans chaque exemple,

nous ne fournirons que la (les) meilleure(s) phrase(s) obtenue(s) par le module de reconnaissance.

7.2.1. Exemples de phrases simples et correctes

Voici trois exemples de phrases simples, c'est-à-dire que les cas sont instanciés par des groupes simples sans coordination ni qualification. De plus, les phrases sont entrées sans insertion de fautes, donc le score de reconnaissance est à chaque fois de 1.

Ex 12 : Prends le livre

(1. ((verbe (gv 1.) (prendre)) (objet (gn 1.) (le livre))))

Ex 13 : Pose le cahier sur le bureau

(1
((verbe (gv 1.) (poser))
(objet (gn 1.) (le cahier))
(dest (gp 1. (gn 1.)) (sur (le bureau))))

Ex 14 : Va près de la fenêtre

(1.
((verbe (gv 1.) (aller))
(dest (gp 1. (gn 1.)) (presde (la fenetre))))

7.2.2. Validation du module de reconnaissance de mots

Dans ce paragraphe, nous présentons des phrases dans lesquelles nous avons introduit quelques erreurs (confusion, insertion et éllision de phonèmes) pour simuler les résultats d'un module de D.A.P. et valider notre module de reconnaissance de mots.

Ex 15 : Ouvre la bedite borte rouche

(.9813515
((verbe (gv .9785125) (ouvrir))
(objet (gn .9819599) (la petite porte rouge))))

Ex 16 : Ponce le pahier euh sor le puleau

(.975053
((verbe (gv .9785125) (poser))
(objet (gn .9929914) (le cahier))
(dest (gp .961941 (gn .9693117)) (sur (le bureau))))

Ex 17 : Prend le ptit lifre

(.9760807
((verbe (gv 1.) (prendre))
(objet (gn .9654498) (le petit livre))))

7.2.3. Résistance aux erreurs syntaxiques

Pour montrer la souplesse de la grammaire utilisée, nous présentons quelques exemples de phrases dont la construction syntaxique est incorrecte suite à de mauvais accords grammaticaux ou à un mauvais ordonnancement des groupes.

Ex 18 : Prends livre rouge

(1.
((verbe (gv 1.) (prendre))
(objet (gn 1.) (livre rouge))))

Ex 19 : Grande fenêtre fermer

```
(.8235294
((objet (gn 1.) (grande fenetre))
 (verbe (gv 1.) (fermer))))
```

Ex 20 : Sur le bureau le livre pose

```
(1.
((dest (gp 1. (gn 1.)) (sur (le bureau)))
 (objet (gn 1.) (le livre))
 (verbe (gv 1.) (poser))))
```

Ex 21 : La petite chaise noire déplace

```
(1.
((objet (gn 1.) (la petite chaise noir))
 (verbe (gv 1.) (déplacer))))
```

7.2.4. Reconnaissance d'expressions complexes

Nous présentons ici la reconnaissance d'expressions complexes composées de coordinations de groupes et de qualification de groupes.

Ex 22 : Prends le cahier sur la table à côté de la lampe

```
(1.
((verbe (gv 1.) (prendre))
 (objet (gn 1. (gp 1. (gn 1. (gp 1. (gn 1.))))
 (le cahier (sur (la table
 (acotede (la lampe))))))))
```

Ex 23 : Prends le livre et le cahier

```
(1.
((verbe (gv 1.) (prendre) )
 (objet (et 1. (gn 1.) (gn 1. ) )
 ((le livre)(le cahier))))
```

Ex 24 : Prends le livre, le classeur et le cahier

```
(.9607998
 (verbe (gv 1.) (prendre))
 (objet (et (gn 1.) (et (gn 1.) (gn 1.))
 ((le livre)
 ((le classeur) (le cahier))) ) )
```

Nous pouvons remarquer dans cet exemple que la factorisation de la conjonction de coordination a été résolue lors de l'étape de reconnaissance.

Ex 25 : Prends le livre sur la table et le classeur sur le bureau

```
(1.
((verbe ((gv 1.)( prendre))
 (objet (et 1. (gn 1. (gp 1. (gn 1.))
 (gn 1. (gp 1. (gn 1.))) )
 (( le livre) ( sur (la table))
 (le classeur) ( sur ( le bureau))) ) )
```

Ex 26 : Pose le petit livre rouge sur le bureau près de la fenêtre

Pour des exemples plus complexes, le module de reconnaissance n'étant pas capable de choisir la meilleure solution, il propose les différentes constructions possibles de la phrase.

```
(1.
((verbe (gv 1.) (poser))
 (objet (gn 1.) (le petit livre rouge))
```

```

(dest      (gp 1. (gn 1. (gp 1. (gn 1.))))
           (sur (le bureau (presde (la fenetre))))))

(1.
 ((verbe  (gv 1.)      (poser))
 (objet   (gn 1. (gp 1. (gn 1.)))
           (le petit livre rouge (sur (le bureau))))
 (dest    (gp 1. (gn 1.)) (presde (la fenetre))))

(1.
 ((verbe  (gv 1.)      (poser))
 (objet   (gn 1. (gp 1. (gn 1. (gp 1. (gn 1.))))
           (le petit livre rouge (sur (le bureau
           (presde (la fenetre))))))

```

7.2.5. "Développement" d'expressions

Ex 27 : Prends le cahier et le classeur sur le bureau

A priori, l'analyseur ne peut pas savoir si le groupe prépositionnel "sur le bureau" qualifie les deux groupes nominaux ou simplement le deuxième, c'est pour cela qu'il génère les deux possibilités avec le même score de reconnaissance, le module d'interprétation se chargeant de choisir la bonne. Les deux phrases générées sont les suivantes :

```

(1.
 ( (verbe  (gv 1.)      ( prendre))
   (objet  (et 1.   (gn 1. (gp 1.(gn 1)))
           (gn 1.) (gp 1. (gn 1.)) )
     (( le cahier)  (sur (le bureau)))
     ((le classeur) ( sur (le bureau))))))

```

```

(1.
 ( (verbe  (gv 1.)      ( prendre))
   (objet  (et 1.   (gn 1. )
           (gn 1.) (gp 1. (gn 1.)) )
     (( le cahier)
     ((le classeur) ( sur (le bureau))))))

```

Ex 28 : Pose le livre près de la lampe et du classeur

Nous avons ici un exemple de coordination de deux groupes prépositionnels, la préposition du deuxième groupe étant une contraction de celle du premier groupe, l'analyseur en déduit donc que la deuxième préposition est identique à la première :

```

( 1
 (verbe  (gv 1.)      ( poser))
 (objet  (gn 1.)      ( le livre))
 (dest  (et 1.   (gp 1. (gn 1.)) (gp 1.(gn 1.))
        ((presde ( la lampe))
        (presde ( classeur))) )

```

Ex 29 : Prends le livre rouge et le noir

Dans cet exemple, deux groupes sont coordonnés. le premier est un groupe nominal complet alors que le second est un groupe elliptique. Les deux groupes respectent les règles nécessaires à leur coordination, donc l'ellipse du deuxième groupe est levée immédiatement, le GNR est remplacé par un GN :

```

( 1.
 (verbe  (gv 1.)      ( prendre))
 (objet  (et 1.   (gn 1.) (gn 1.))
        (( le livre rouge)(le livre noir)) )

```

Ex 30 : Déplace la petite et la grande chaise

Nous sommes ici en présence d'un cas similaire excepté que le groupe elliptique est placé avant le groupe référant :

```
(1.
  (verbe      (gv 1. )      ( déplacer))
  (objet      (et 1. (gn 1. ) (gn 1. ))
              ((la chaise petite) (la grande chaise))) )
```

7.3. Résultats de l'interprétation

7.3.1. Groupes simples

L'étape d'interprétation de la phrase apporte de nouvelles informations dans la structure : un score d'interprétation globale de la phrase, un score d'interprétation de chaque cas ainsi que les occurrences instanciant les cas.

Ex 31 : Prends le livre rouge

```
((scoreinterp 0.)
  (lcas (verbe 0. (gv 1.) (prendre) ()))
  (objet 0. (gn 1. ) (le livre rouge) (livre110) )
  (agent 0. () () (robot))
  (score 1.))
```

Dans cet exemple, le score global d'interprétation de la phrase vaut 0 car il n'y a ni ambiguïté ni échec d'interprétation. Le verbe n'étant pas un cas interprétable, son interprétation vaut (), le livre rouge a été interprété comme l'occurrence dre livre portant le nom livre110 et le cas agent qui n'était pas spécifié dans l'énoncé par comme valeur, par défaut, le "robot".

7.3.2. Groupes complexes

Ex 32 : Prends le livre sur le bureau près de la fenêtre

```
((scoreinterp 0.)
  (lcas (verbe 0. (gv 1.) (prendre) ()))
  (objet 0. (gn 1. (gp 1. (gn 1. (gp 1. (gn 1.))))))
  (le livre (sur (le bureau
                  (presde (la fenetre))))))
  (livre118 (sur bureau114
             (presde fenetre102))))
  (agent 0. () () (robot))))
```

7.3.3. Groupes ambigus

Ex 33 : Prends le livre et le classeur sur le bureau

Les deux interprétations possibles de cette phrase sont :

- prends le livre et le classeur sur le bureau
- prends le livre sur le bureau et le classeur sur le bureau

C'est la confrontation avec l'univers qui va nous permettre de lever l'ambiguïté, car il n'existe qu'un seul bureau supportant à la fois un classeur et un livre, la règle de développement utilisée impliquant que le bureau où se trouve le livre doit être identique à celui où se trouve le classeur.

```
(scoreinterp 0.
  (lcas ( verbe 0. (gv 1) (prendre) ()))
  (objet 0.
    (et 1. (gn 1. (gp 1. (gn 1.)))
           (gn 1. (gp 1. (gn 1.))))
    ((le livre (sur (le bureau)))
     (le classeur (sur (le bureau))))
    (et ((livre118 (sur bureau114)))
         ( classeur116 (sur bureau114))))
  (agent 0. () () (robot))))
```

```

(scoreinterp 4.
  (lcas ( verbe 0. (gv 1) (prendre) () )
    (objet 4.
      (et 1. (gn 1.)
        (gn 1. (gp 1. (gn 1.))))
      ((le livre )
        (le classeur (sur (le bureau))))
      (et ((livre110 )(livre111 )(livre118 ))
        ( classeur116 (sur bureau114))))
    (agent 0. () () (robot))))

```

7.3.4. Groupes elliptiques

Nous présentons uniquement la résolution d'ellipses de cas semi-obligatoires à l'aide de connaissances pragmatiques, ou par une question.

Ex 34 : Pose la chaise

```

(scoreinterp 0.
  (lcas ((verbe 0. (gv 1.) (poser) () )
    (objet 0. (gn 1.) (la chaise) ((chaise113)))
    (dest 0. () () ((par terre) )
    (agent 0. () () ((robot) ) )

```

Ex 35 : Pose le livre rouge

```

(scoreinterp 3.
  (lcas ((verbe 0. (gv 1.) (poser) () )
    (objet 0. (gn 1.) (le livre rouge) ((livre111))
    (dest 3. () () (( ou ? )))
    (agent 0. () () ((robot) ) )

```

7.4. Implémentations et perspectives

Le système que nous avons développé est écrit en LELISP version 15.2 sur SUN 4. Nous avons utilisé la couche objet pour définir les classes des prédicats et les méthodes associées à la résolution des ellipses. Les stratégies de gestion du dialogue, si elles sont définies, ne sont pas encore totalement implémentées (cas d'ambiguïtés, résolution d'ellipses structurales, contestations...). Elles sont actuellement en cours de développement et demandent à être validées par un ensemble de tests significatifs.

En ce qui concerne la partie développée, quelques modifications sont envisageables à très court terme pour améliorer la stratégie de choix des phrases. Nous avons vu que l'analyseur reconnaît des phrases asyntaxiques, cependant, nous pensons qu'il serait intéressant de faire entrer le degré de correction syntaxique dans le calcul du score de reconnaissance. En effet, si plusieurs phrases sont candidates avec des scores de reconnaissance proches, il convient de favoriser celle dont la structure syntaxique est la plus correcte. De la même manière, il faut privilégier celle instanciant les cas de manière optimale. Si l'on se réfère à l'exemple 26, les deux premières phrases candidates doivent être préférées à la troisième pour laquelle le cas DEST n'est pas instancié.

Conclusion

Nous avons présenté un ensemble d'outils et de concepts permettant d'interpréter des commandes orales dans un langage proche du langage naturel. Le système n'étant pas encore connecté à un décodeur acoustico-phonétique, nous avons simulé la génération de phonèmes à partir du texte écrit de la commande. Pour rendre compte des imperfections du décodage, nous introduisons des erreurs dans l'énoncé (confusion de phonèmes, élision ...) ce qui nous permet d'obtenir un système pouvant fonctionner à partir de parole ou de texte écrit comportant des fautes d'orthographe.

La grammaire de cas que nous avons utilisée, permet de donner une grande liberté d'expression à l'utilisateur dans la mesure où elle indique quels sont les éléments nécessaires à la compréhension, mais n'impose pas de contraintes rigides.

Les grammaires locales décrivent la syntaxe des groupes instanciant les cas, elles permettent de définir l'ensemble minimal utile à l'identification du groupe, ce qui autorise, là aussi une grande liberté d'expression.

La structure casuelle obtenue après l'analyse d'une commande étant très proche d'une structure prédicative (prédicats + arguments), l'interprétation de la commande en est grandement simplifiée.

L'organisation des prédicats en classes, et la structure des lexiques permettent une augmentation ou un changement de vocabulaire souples qui ne remettent pas en cause la phase de reconnaissance des énoncés. La résolution des ellipses et des anaphores permet des échanges naturels.

Pour obtenir un système totalement opérationnel, il conviendrait de définir les grammaires locales de manière plus complète afin de pouvoir accepter une plus grande variation de langage. Pour augmenter la qualité des échanges, il faudrait

intégrer les verbes autres que des prédicats de façon à pouvoir nuancer les demandes, faire des requêtes concernant le dialogue ...

L'interfaçage du système de dialogue avec un véritable système de gestion de plans d'action permettrait d'enrichir les vérifications pragmatiques et aider davantage à la résolution d'ambiguïtés et à la gestion d'hypothèses.

Tout ceci correspond à des perspectives à moyen terme. Une perspective plus proche serait d'utiliser le système de dialogue dans le but de valider la reconnaissance des objets faite par le module de vision. Dans notre représentation de l'univers, nous avons supposé que les objets étaient parfaitement reconnus, ce qui est loin d'être le cas actuellement. On pourrait envisager de coupler les deux systèmes au moyen d'un processus d'hypothèse-validation. Lorsque le système de vision reconnaît un objet avec un certain score, le fait de nommer ou de décrire cet objet peut faire évoluer le score vers une plus grande certitude de reconnaissance ou au contraire, rejeter l'hypothèse. Les deux systèmes travaillant en parallèle, seraient l'un pour l'autre une source de connaissance de l'environnement.

Conclusion

Bibliographie

- [Alinat-87] : P. Alinat, E. Gallais, J.P. Haton, J.M. Pierrel, P. Richard.
A continuous speech dialog system for oral control of sonar console.
Proceeding of IEEE ICASSP-87, 1987.
- [Allen-80] : J.F. Allen, C.R. Perrault
Analysing intention in utterances.
Artificial Intelligence, N° 15, p.143-178. 1980.
- [Amalberti-86] : R. Amalberti, N. Carbonell, P. Falzon
Stratégie de contrôle du dialogue en situation d'interrogation téléphonique.
Actes du séminaire GRECO sur le dialogue oral homme-machine. Nancy 1984.
- [Bellman-57] : J. Bellman
Dynamic Programming.
Princeton University Press. USA 1957.
- [Bobrow-77] : D.G. Bobrow, R.M. Kaplan, D.A. Norman,
H. Thomson, T. Winograd.
Gus: A Frame-Driven Dialog System.
Artificial Intelligence, vol 11, n°3. 1977.
- [Bogurov-83] : B. Bogurov, K. Spark-Jones.
How to drive a database front-end using general semantic information.
Conference on applied natural language processing. Santa Monica. p. 81-61. 1983.
- [Borillo-77] : A. Borillo.
Représentation des connaissances: de l'analyse syntaxique de la phrase à son interprétation.
Journées de travail sur la compréhension, IRIA Satori. 1977.

Bibliographie

- [Caelen-88] : J. Caelen.
Méta-stratégie en reconnaissance dans le projet "DIRA".
Actes des XVII^{ème} J.E.P., p. 173-179. Nancy, 1988.
- [Calliope-89] : Calliope.
La parole et son traitement automatique.
Ed. Masson. Collection technique et scientifique des télécommunications. Paris, Milan, Barcelone, Mexico 1989.
- [Carbonell-86] : N. Carbonell, J.P. Haton, D. Fohr, F. Lonchamp, J.M. Pierrel.
APHODEX, Design and Implementation of an Acoustic-Phonetic Decoding Expert System.
Proc IEEE ICASSP, Tokyo, Japon 1986.
- [Causse-76] : B. Causse, D. Dours, R. Facca, G. Perennou.
Evaluation d'une méthode ascendante d'analyse lexicale dans le discours continu.
Actes des VII^{ème} journées d'étude sur la parole, p.37-53. Nancy 1976.
- [Charpillat-85] : F. Charpillat.
Un système de reconnaissance de la parole continue pour la saisie de textes lus.
Thèse de doctorat de l'université de Nancy I, 1985.
- [Chomsky-71] : N. Chomsky.
Aspects de la théorie syntaxique.
Le Seuil. Paris. 1971.
- [Chomsky-71bis] : N. Chomsky.
Deep structure, surface structure and semantic interpretation.
Steinberg & Jakobovits, p. 183-216. 1971
- [Chomsky-75] : N. Chomsky.
Reflexions on language.
Pantheon, New-Jersey. 1975.

- [Clark-79] : H. Clark.
Responding to indirect speech acts.
Cognitive psychology, vol 11, p. 430-477. 1979.
- [Cohen-79] : P.R. Cohen, C.R. Perrault.
Elements of Plan-Based Theory of Speech Acts.
Cognitive Science, vol 1, p. 177-212. 1979.
- [Colmerauer-78] : A. Colmerauer.
Metamorphosis grammars.
Natural language communication with computer.
Bolc, Springer-Verlag, p.139-189. Berlin 1978.
- [Colnet-86] : D. Colnet, G. Masini, A. Napoli, Y. Noiret, K. Tombre.
Les langages orientés objets.
Rapport interne n° 86-R-077. CRIN. Nancy 1986.
- [Corner-83] : D.F. Corner, A.O. Anbler, R.J. Popplestone.
Reasoning about the spatial relationships derived from a RAPT program for describing assembly by robot.
Actes du congrès IJCAI 1983, p 842-844.
- [Coulon-86] : D. Coulon, D. Kayser.
Informatique et langage naturel: présentation générale des méthodes d'interprétation de textes écrits.
TSI 1986, Vol 5-2, p 103-128.
- [Dell-85] : F. Dell.
La phonologie générative: les règles et les sons.
Herman, Paris. 1985.
- [Deville-86] : G. Deville, H. Paulussen.
A case grammar as an original linguistic model for semantic representation of utterances in a man-machine dialog system.
Thesis of computational linguistics, University of Antwerpen, Belgium, 1986.

- [Deville-87] : G. Deville, H. Paulussen, J.M. Pierrel.
Une grammaire de cas comme modèle de représentation sémantique d'énoncés de dialogues oraux homme-machine finalisés.
Actes du 6^{ème} congrès AFCET Reconnaissance des Formes et Intelligence Artificielle, p.159-173. Antibes 1987.
- [Deville-87bis] : G. Deville.
Corpus de dialogues oraux finalisés en situation réelle : Demande de renseignements auprès du ministère de l'emploi et du travail (cellule Action-Travail)
Ed. G. Deville, FUNDP. Namur, 1987.
- [Dinard-87] : M. Dinard.
Résolution d'ellipses extragrammaticales dans un dialogue en langage naturel.
Actes du 6^{ème} congrès AFCET Reconnaissance des Formes et Intelligence Artificielle, p.461-466. Antibes 1987.
- [Dubois-85] : D. Dubois, H. Prade.
Théorie des possibilités. Applications à la représentation des connaissances en informatique.
Ed. Masson 1985.
- [Falzon-84] : P. Falzon.
Les langages opératifs.
Actes du séminaire GRECO sur le dialogue oral homme-machine. Nancy 1984.
- [Fillmore-68] : Ch. Fillmore.
The case for case.
Universals in linguistics theory, E. Bach and R.T. Harn ed. Rinehart and Winston. p.1-90. 1968.
- [Fohr-86] : D. Fohr.
APHODEX : un système expert en décodage acoustico-phonétique de la parole continue.
Thèse de doctorat de l'université de Nancy I, 1986.

- [Ferber-87] : J. Ferber.
Des objets aux agents: une architecture stratifiée.
Actes du 6^{ème} congrès AFCET Reconnaissance des Formes et Intelligence Artificielle, p.275-286. Antibes 1987.
- [Frey-83] : W. Frey, U. Reyle, Ch. Roher.
Automatic construction of knowledge base by analysing texts in natural language.
Actes du congrès IJCAI 1983, p.727-729.
- [Gazdar-79] : G. Gazdar.
Phrases structure grammar.
The nature of syntactic representation, p.131-184.
Jacobson & Pullum. Dordrecht : Reidel (82). 1979.
- [Gazdar-79bis] : G. Gazdar.
Pragmatics implicature, presupposition and logical form.
Academic press. New-York. 1979.
- [GRECO-85] : GRECO
Dialogue oral homme-machine en situation assistée par l'action.
Actes du 5^{ème} congrès AFCET Reconnaissance des Formes et Intelligence Artificielle, p.281-296. Grenoble 1985.
- [Grice-75] : H.P. Grice.
Logic and Conversation in Syntax and semantics 3: speech acts,
Coles & Morgan , Academic press, New-York, p.41-58. 1975.
- [Grimes-72] : J. Grimes.
The thread of discourse.
Rapport technique NSF, Cornell university. 1972.
- [Gross-86] : M. Gross.
Grammaire transformationnelle du français. I- Syntaxe du verbe.
Ed. Cantilène. 1986.

- [Gross-86bis] : M. Gross.
Grammaire transformationnelle du français.
2- *Syntaxe du nom.*
Ed. Cantilène. 1986.
- [Haton-74] : J.P. Haton .
Une méthode dynamique de comparaison de chaînes de symboles de longueurs différentes : application à la recherche lexicale.
C.R.Ac. des sciences, A-278,p. 1527-1530. 1974.
- [Haton-85] : M.Ch. Haton.
Contribution à l'éducation vocale assistée par ordinateur : étude des voix et réalisation du système SIRENE.
Thèse d'état, Université de Nancy I, 1985.
- [Haton-90] : J.P. Haton, A. Bonneau, D. Fohr, Y. Gong, Y. Laprie, J.M. Pierrel.
Décodage acoustico phonétique : problèmes et éléments de solution.
Soumis à Traitement du signal. 1990.
- [Hayes-83] : Ph. J. Hayes and J. G. Carbonell.
A Framework for Processing Corrections in task-Oriented Dialogs.
Actes du congrès IJCAI 1983, p.668-670.
- [Hendler-90] : J. Hendler, A. Tate, M. Drummond.
A.I. Planning : systems and techniques.
AAAI-90, p. 61-77.
- [JdM-84] : J. di Martino.
Contribution à la reconnaissance globale de parole : mots isolés et mots enchaînés.
Thèse de docteur ingénieur, Nancy I, 1984.

Bibliographie

- [Kayser-86] : D. Kayser.
Représentation des connaissances en intelligence artificielle. Contribution à l'étude des représentations dans le cadre des sciences cognitives.
Ecole d'été de l'ARC. "Approches des sciences cognitives"
Bonas, 1986.
- [Kittredge-79] : R. Kittredge.
Textual Cohesion within Sublanguages : Implications for Automatic Analysis and Synthesis.
IRIA-LISH "représentation des connaissances et raisonnement pour les sciences de l'homme", p 42-72,
Saint-Maximin. 1979.
- [Klatt-75] : D.H. Klatt.
Word Identification in a Speech Understanding System.
Speech Recognition, D.R. Reddy (ed.), Academic Press.
1975.
- [Lehnert-83] : W. G. Lehnert, M. G. Dyer, P. N. Johnson, C.J. Yang, S. Harley.
Boris: An experiment in In-Depth Understanding of Narratives.
Artificial intelligence 20, p 15-62.1983.
- [Luzzati-85] : D. Luzzati.
Comportement langagier induit par la machine, étude des demandes initiales.
Analyse linguistique d'un corpus de règles finalisées,
rapport GRECO, M.A. Morel, 1985.
- [Mangeol-88] : B. Mangeol.
La composante lexicale dans le système de dialogue oral homme-machine du CRIN.
Thèse de doctorat de l'université de Nancy I, 1988.
- [Mariani-89] : J. Mariani.
Recent Advances in Speech Processing.
Actes du congrès ICASSP. 1989.

Bibliographie

- [Masini-89] : G. Masini, A. Napoli, D. Colnet, D. Léonard, K. Tombre.
Les langages à objets, langages de classes, langages de frames, langages d'acteurs.
Inter Editions. 1989.
- [Mercier-77] : G. Mercier, P. Quinton, R. Vives.
Dialogue homme-machine avec KEAL.
Recherches acoustiques 4, 1977.
- [Miclet-88] : L. Miclet.
Décodage acoustico-phonétique.
Actes des premières journées nationales du Greco-PRC
Communication Homme-Machine. Parole, Langage naturel
et Vision.
Paris, 24-25 Novembre 1988.
- [Minsky-75] : M. Minsky.
A framework for representing knowledge.
The psychology of computer vision.
Ed. P. Winston, MacGraw-Hill, New-York, 1975.
- [Mohr-88] : R. Mohr.
Le projet ORASIS.
Actes des premières journées nationales du Greco-PRC
Communication Homme-Machine. Parole, Langage naturel
et Vision.
Paris, 24-25 Novembre 1988.
- [Morin-87] : P. Morin, J.M. Pierrel.
PARTNER : un système de dialogue oral homme-machine.
Actes du congrès COGNITIVA, p. 354-361, 1987.
- [Neumann-84] : B. Neumann.
Natural Language Description of Time-Varying Scenes.
Fachbereich Informatik, Universität de Hambourg, 1984.
- [Neumann-87] : B. Neumann.
Integrating image and natural language understanding.
Artificial intelligence in facturing, p 141-148.
Ed. T. Bernol. GDI, 1987.

- [Norihiro-81] : A. Norihiro, S. Itsuya, T. Saburo.
A plot understanding system on reference to both image and language.
Actes du congrès IJCAI 1981, p. 77-84.
- [Nouhen-81] : A. Nouhen-Bellec, J. Siroux.
Cadi: constructeur automatique de dialogue intelligent intégré à un système de reconnaissance et de compréhension de la parole.
Thèse de 3^{ème} cycle. université de Rennes I. 1981.
- [Pereira-80] : F. Pereira, D. Waren.
Definite clause grammar for language analysis - a survey of formalism and a comparison with augmented transition networks.
Artificial Intelligence, vol 13, 3. P. 231-278. 1980.
- [Pierrel-75] : J.M. Pierrel.
Contribution à la compréhension automatique du discours continu.
Thèse de 3^{ème} cycle de l'université de Nancy I, 1975.
- [Pierrel-79] : J.M. Pierrel, J.F. Mari, J.P. Haton.
Le niveau lexical dans le système MYRTILLE II: représentation du lexique et traitements associés.
10^{èmes} journées d'étude sur la parole. Grenoble 30 Mai-1^{er} Juin 1979.
- [Pierrel-81] : J.M. Pierrel.
Etude et mise en œuvre de contraintes linguistiques en compréhension automatique du discours continu.
Thèse d'état de l'université de Nancy I. 1981.
- [Pierrel-87] : J.M. Pierrel.
Dialogue oral homme-machine : connaissances linguistiques, stratégies et architecture de systèmes.
Hermès, Paris. 1987.

- [Pister-84] : C. Pister.
Adaptation au locuteur par apprentissage automatique, application à un système de reconnaissance automatique de la parole.
Thèse de 3^{ème} cycle, Nancy I, 1984.
- [Pitrat-85] : J. Pitrat.
Textes, ordinateurs et compréhension.
Eyrolles 1985.
- [Roussanaly-88] : A. Roussanaly.
Dial, la composante dialogue d'un système de communication orale homme-machine finalisée en langage naturel.
Thèse de doctorat de l'université de Nancy I. 1988.
- [Sabah-88] : G. Sabah.
L'intelligence artificielle et le langage. Vol 1. Représentation des connaissances.
Hermès. Paris. 1988.
- [Sadek-87] : M.D. Sadek, M. Guyomard, J. Siroux.
Vers un système de dialogue basé sur les logiques intentionnelles et la planification des actes de langage.
Actes du 6^{ème} congrès AFCET Reconnaissance des Formes et Intelligence Artificielle, p.913-923. Antibes 1987.
- [Sakoe-78] : H. Sakoe, S. Chiba.
Dynamic Programming Optimization for Spoken Word Recognition.
IEEE Transactions of Acoustics Speech and Signal Processing. Vol 26, p43-49. 1978.
- [Segui-88] : J. Segui.
L'accès au lexique, données expérimentales et modèles.
Ed. Calliope 1988.

- [Schank-72] : R. Schank.
Conceptual dependency : a theory of natural language understanding.
Cognitive psychology, vol 3, p.552-631. 1972.
- [Schank-77] : R. Schank, R. Abelson.
Scripts, Plans, Goals and Understanding. An Inquiry into Human Knowledge Structures.
Lawrence Erlbaum associates, publishers 1977.
- [Shortliffe-76] : E.H. Shortliffe.
Computer based medical consultations, MYCIN.
American elsevier. 1976.
- [Simmons-74] : R. Simmons.
On managing sentence meanings.
Report NL 20, dept of computer science, university of Texas, Austin. 1974.
- [Siroux-85] : J. Siroux, D. Gillet.
A system for man-machine communication using speech.
North-Holland publishing company, Amsterdam. p. 289-315. 1985.
- [Thirion-89] : E. Thirion.
Interprétation et apprentissage géométrique en vision par ordinateur.
Thèse de doctorat de l'institut national polytechnique de Lorraine. 1989.
- [Véronis-87] : J. Véronis.
Vérification de cohérence dans le dialogue homme-machine en langage naturel.
Actes du 6^{ème} congrès AFCET Reconnaissance des Formes et Intelligence Artificielle, p.143-157. Antibes 1987.
- [Veronis-88] : J. Véronis.
Contribution à l'étude de l'erreur dans le dialogue homme-machine en langage naturel.
Thèse de l'université d'Aix-Marseille III. 1988.

- [Vilnat-85] : A. Vilnat.
Le dialogue dans une communication homme-machine.
Intellectica. Les interactions homme-ordinateur. A. Borillo,
J.M. Hoc, M. Quéré. Vol 1, N° 1. 1985.
- [Vives-85] : R. Vives.
*Mise en correspondance de descriptions phonologiques et
prosodiques de mots dans le système de reconnaissance de
parole KEAL.*
Xèmes JEP, GALF, Paris 1985, p. 253-256.
- [Walker-78] : D.E. Walker.
Understanding spoken language.
North-Holland publishing company. Amsterdam 1978.
- [Wei-66] : J. Weizenbraum.
*ELiza : a Computer Program for the study of Natural
Language Communication Between Man and Machine.*
Communications de l'A.C.M. vol 9, N° 1, p. 36-45. 1966.
- [White-78] : G.M. White.
*Dynamic Programming, the Viterbi Algorithm and Low
Cost Speech Recognition.*
IEEE ICASSP p. 413-417. Tulsa, Oklahoma, USA 1978.
- [Winograd-71] : T. Winograd.
*Procedure as a representation for data in computer program
for understanding natural language.*
MIT. Ph D. 1971.
- [Yin-83] : B. Yin.
*A framework for handling vision data in an object level
robot language. RAPT.*
Actes du congrès IJCAI 1983, p.814-820.

VOYELLES			CONSONNES		
phon	exemples	codification	phon	exemples	codification
a	patte, bas, pâte	a	p	pas	p
i	il	i	t	tas	t
y	nu	y	k	cas	k
o	bol	o	b	bon	b
ø	eau	ø	d	dans	d
œ	le	&	g	gars	g
e	peu, peur	eu	v	vie	v
ɛ	blé	e	z	zéro	z
ɛ̃	merci	ai	ʒ	je	gh
u	ou	u	f	feu	f
ã	an	an	s	sous	s
õ	on	on	ʃ	chat	ch
ẽ	lin	in	n	nous	n
œ̃	un	in	m	ma	m
SEMI-VOYELLES			ʝ	agneau	nj
			l	la	l
			r	rue	r
ɥ	huit	w			
w	oui	w			
j	yeux, baille	j			

Figure A.1. Codification des phonèmes

Groupe nominal et ses dérivées	
GN	--> art adjn adja nom ^{+s} adjp*
GNR	--> art ⁺ adjn adj ⁺ adj* / prond ⁺ rel GP ⁺ / prond ⁺ rel ⁺ adj ⁺ adj*
GNREF	--> pronp ^{+s} / art adjn ⁺
Groupe prépositionnel et ses dérivées	
GP	--> prep ^{+s} GN ⁺
GPR	--> prep ^{+s} GNR ⁺
GPREF	--> prep ^{+s} pronp ^{+s} / adv ^{+s}
Groupes interrogatifs	
GIN	--> adjj ⁺ adja nom ^{+s} adjp* / art ⁺ adjj ⁺ prepcn ⁺ adjn adja nom ^{+s} adjp
GINR	--> art adjj ⁺ / art ⁺ adjj ⁺ prepcn ⁺ adjn adj*
GINREF	--> proni ^{+s} / advi ^{+s}
GIP	--> prep ^{+s} GIN ⁺
GIPR	--> prep ^{+s} GINR ⁺ / reli ^{+s}
GIPREF	--> prep ^{+s} GINREF ⁺
Groupes relatifs	
GREL	--> rel ⁺ GP ⁺ / rel ⁺ GPR ⁺ / rel ⁺ adj ⁺ adj* / rel ⁺ GPREF ⁺

Figure A.2. Grammaires locales

GN + GN	conditions	-->	même rôle sémantique
	transformations	-->	-
GN + GNR	conditions	-->	- GN et GNR contiennent un adjectif ayant le même rôle ou GN contient une imbrication
	transformations	-->	- GN et GNR ont le même genre transformer le GNR en GN (en fonction de sa structure)
GNR + GN	conditions	-->	- GN et GNR contiennent un adjectif ayant le même rôle et GNR est de la 1 ^{ère} forme
	transformations	-->	- GN et GNR ont le même genre transformer le GNR en GN en lui ajoutant tout ce qui suit le nom du GN
GNR + GNR	conditions	-->	accord en genre
	transformations	-->	-
GP + GP	conditions	-->	même rôle sémantique
	transformations	-->	si la préposition du 2 ^{ème} GP est une forme contractée de celle du 1 ^{er} , la transformer
GP + GPR	conditions	-->	idem que GN + GNR et même rôle sémantique
	transformations	-->	idem que GN + GNR et idem que GP + GP
GP + GN	conditions	-->	les deux GN ont le même rôle sémantique
	transformations	-->	transformer le GN en GP en y insérant la préposition
GP + GNR	conditions	-->	idem que GN + GNR
	transformations	-->	idem que GN + GNR et idem que GP + GN

GPR + GN	conditions	-->	idem que GNR + GN
	transformations	-->	transformation du GNR en GN transformation du GN en GP
GPR + GP	conditions	-->	même rôle sémantique
	transformations	-->	-
GPR + GPR	conditions	-->	- même rôle sémantique
	transformations	-->	- accord en genre des GNR idem que GP + GP
GPR + GNR	conditions	-->	idem que GNR + GNR
	transformations	-->	transformation du GNR en GPR
GPREF + GP	conditions	-->	même rôle sémantique
	transformations	-->	s'il n'y a pas de conjonction, par défaut => ET
GP + GREL	conditions	-->	GREL est de la 1 ^{ère} forme
	transformations	-->	même rôle sémantique des GP transformation du

Figure A.3. Règles de coordinations et transformations associées

NOM DE L'ETUDIANT : KLEIN JOELLE

NATURE DE LA THESE : DOCTORAT DE L'UNIVERSITE DE NANCY I EN INFORMATIQUE



VU, APPROUVE ET PERMIS D'IMPRIMER

NANCY, le 16 OCT. 1990 n° 2093

LE PRESIDENT DE L'UNIVERSITE DE NANCY I



Résumé

La commande d'un robot doté d'un système de vision évolué implique la reconnaissance d'expressions naturelles complexes nécessaires à la désignation des objets, c'est pourquoi nous avons tenté de réaliser un système permettant à une personne de donner des ordres de la manière la plus naturelle possible. Dans le but de donner une grande souplesse à notre système, nous autorisons des constructions elliptiques, anaphoriques et même asyntaxiques (dans la mesure où elles sont sémantiquement correctes).

Les moyens que nous avons mis en œuvre pour atteindre nos objectifs reposent sur une analyse syntaxico-sémantique de l'énoncé. La sémantique est définie à l'aide d'une grammaire de cas dont la structure (verbe + cas) s'apparente fortement à la structure logique d'une commande (prédicat + arguments). La structure syntaxique est définie à l'aide de grammaires locales décrivant des groupes simples. Les grammaires locales sont partagées en trois classes mettant en évidence les variations syntaxiques dues aux différents types d'énoncés possibles (énoncés complets, elliptiques et anaphoriques). La construction de groupes complexes se fait à l'aide de coordinations et de qualifications de groupes en fonction de règles linguistiques qui permettent de résoudre localement certaines formes d'ellipses (ex : le livre rouge et le noir -> le livre rouge et le **livre** noir).

L'interprétation de la commande ainsi reconnue se déroule en deux étapes : la première consiste à compléter les énoncés incomplets à l'aide de valeurs par défaut, par recherche dans l'historique, raisonnement pragmatique..., la seconde est la recherche des objets dans l'univers. Nous nous sommes particulièrement intéressés au problème de l'ambiguïté et nous proposons quelques stratégies permettant de faire des hypothèses sur les objets en conflit en partant du principe que le locuteur est coopératif et que les informations qu'il fournit sont pertinentes.

Mots-clés : langage naturel, grammaire de cas, dialogue oral homme-machine finalisé, commande orale de robot, reconnaissance d'expressions complexes.