

enssib
Ecole Nationale Supérieure
des Sciences de l'Information et des Bibliothèques

D E A
Sciences de l'Information et de la Communication

OPTION :
Systèmes d'Information Documentaire

MEMOIRE DE D.E.A

Traduction Assistée par Ordinateur
du français vers l'arabe :
Application à un corpus restreint

Auteur :
Sahbi SIDHOM

Sous la Direction :
Mohamed HASSOUN

1996



enssib
Ecole Nationale Supérieure
des Sciences de l'Information et des Bibliothèques

D E A
Sciences de l'Information et de la Communication

OPTION :
Systèmes d'Information Documentaire

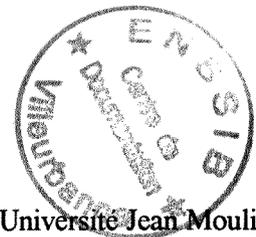
MEMOIRE DE D.E.A

Traduction Assistée par Ordinateur
du français vers l'arabe :
Application à un corpus restreint

Auteur :
Sahbi SIDHOM

Sous la Direction :
Mohamed HASSOUN

1996



Traduction Assistée par Ordinateur du français vers l'arabe : Application à un corpus restreint

par Sahbi SIDHOM ,
Sous la direction de Monsieur Mohamed HASSOUN
ENSSIB

Résumé :

L'informatique entretient des rapports complexes avec le langage naturel, car celui-ci convient le mieux à l'utilisateur dans sa communication avec l'ordinateur.

C'est dans cet esprit que l'on s'est fixé comme objectif la conception et l'implémentation d'un prototype de Traduction Assistée par Ordinateur (TAO) du français vers l'arabe, appliqué à un corpus restreint .

L'objectif dans la réalisation du prototype est, pour une phrase donnée en français, de faire son analyse syntaxique, son transfert structural français-arabe afin de produire en génération l'équivalent de la phrase en arabe.

Descripteurs :

Traduction Assistée par Ordinateur (TAO); Dictionnaire électronique bilingue, Analyse lexicale; Analyse syntaxique; Transfert structural français-arabe; Génération; Grammaire-DCG; Programmation Logique (PROLOG); français (langue); arabe (langue); langage naturel; compréhension; modélisation.

Abstract :

Computer science maintains complex relationships with Natural Language because this one is the most suited for communication between a user and a computer system.

It is in this spirit that we have fixed as objective , the design and implementation of a French-Arabic Computer Aided Translation (CAT) prototype applied to a restricted corpus.

The aim of the realization of such prototype, for a given sentence in French, is to make its syntactic analysis and its French-Arabic structural transfer in order to generate the equivalent in Arabic sentence.

Keywords :

Computer Aided Translation (CAT); automatic bilingual dictionary; lexical analysis; syntactic analysis; french-arabic structural transfer; generation; DCG grammar; Logic Programming (PROLOG); French (language); Arabic (language); natural language; understanding; patterning.

DEDICACES

Je dédie ce modeste travail à toute ma famille en témoignage de toute mon affection et de ma profonde gratitude .

A mon père et ma mère qui n'ont lésiné aucun moyen à mon égard.

A mes frères, ma soeur et ma belle soeur pour leur dévouement et leur sagesse et à ma nièce .

A mon oncle Fredj et sa famille .

REMERCIEMENTS

Mes remerciements s'adressent tout particulièrement à mon Directeur Monsieur Mohamed **HASSOUN** , de m'avoir fait bénéficier de ses pertinents conseils et de ses encouragements.

Je tiens aussi à remercier vivement Monsieur Joseph **DICHY**, Professeur à Lyon 2, pour sa collaboration , ses précieux conseils et ses encouragements .

Mes sincères remerciements vont

à

Monsieur Richard **BOUCHÉ**, Professeur et Directeur de l'option Systèmes d'Information Documentaire à l'ENSSIB , pour sa disponibilité, son aide et ses connaissances qu'il m'a transmises au cours de cette année de Recherche en DEA ,

à

Monsieur Michel **LE GUERN**, Professeur à Lyon 2 , pour m'avoir fait bénéficier d'une petite part de ses connaissances en linguistique .

et à

Monsieur Jean Marie **PINON** , Professeur à l'INSA de Lyon , pour l'aide et le soutien qu'il m'a accordé .

Mes remerciements s'adressent également au Président et aux membres du jury pour l'honneur qu'ils me font en acceptant de juger ce travail.

Je saisis cette opportunité pour remercier aussi :

Monsieur Salem **GHAZALI**
Madame Lamia **EL ABED**
Mademoiselle Nacera **GARBOUT**
Monsieur Adel **HANNACHI**

mes collègues et chercheurs au Département Communication Homme-Machine à l'IRSIT de Tunis pour leur extrême coopération et leur gentillesse.

Mes vifs remerciements vont à tous mes collègues de promotion et ami(e)s de l'ENSSIB et l'INSA de Lyon, à tous ceux qui m'ont aidée à mener à terme mon travail, en particulier Mohammed **SASSI** et les triathlètes Jérôme **POINAS** et Philippe **BELLYNCK**.

Que tout le personnel de l'ENSSIB, trouve ici l'expression de ma profonde gratitude .

PLAN

§ INTRODUCTION.....	1
---------------------	---

Partie 1 : Tour d'horizon

Chapitre I. La Traduction Automatique:	4
Chapitre II. Etat de l'art de la TA ou la TAO :	9
Chapitre III. Description de quelques systèmes de TA:.....	13

Partie 2 : Conception et réalisation d'une application

Chapitre IV. Modèle de traduction :.....	41
Chapitre V. Difficultés du traitement de la langue arabe:	48
Chapitre VI. Description de notre prototype de TAO :.....	59
Chapitre VII. Réalisation informatique du prototype de TAO :.....	80

§ CONCLUSION.....	99
-------------------	----

§ BIBLIOGRAPHIE	101
-----------------------	-----

SOMMAIRE

INTRODUCTION.....	1
-------------------	---

Partie 1 : Tour d'horizon

I. La Traduction Automatique:	4
I.1. Introduction:	4
I.2. Les connaissances nécessaires pour la TA:	5
I.2.1. Les dictionnaires:	5
I.2.2. Les connaissances syntaxiques :	5
I.2.3. Les connaissances sémantiques:	6
I.2.4. Les connaissances contextuelles:	7
II. Etat de l'art de la TA ou la TAO :	9
II.1. Première génération:(1950-1962)	9
II.2. Deuxième génération : (1960-1970)	9
II.2.1. Evolution de l'architecture des systèmes de deuxième génération:	11
II.2.1.1. Le modèle à transfert:(type PIVOT)	11
II.2.1.2. Le modèle à interlangues:(type SHALT2)	11
II.3. Troisième génération:	12
II.3.1. Architecture d'un système de troisième génération:	12
III. Description de quelques systèmes de TA:	13
III.1. Le système ALT-J/E:	13
III.1.1. Les dictionnaires:	13
III.1.2 Le schéma général du système:	14
III.1.3. Fondement théorique du système:	14
III.1.4 Mécanisme de traduction du système:	15
III.2 Le système ARIANE:	16
III.2.1 Le processus de traduction:	16
III.2.1.1 Analyse:	17
III.2.1.2 Transfert:	17
III.2.1.3 Génération:	17
III.2.2 Evolution d'ARIANE: ARIANE-G5, 1988	18
III.2.3 Utilisation de l'aspect multilingue d'ARIANE:	18
III.2.4 Application - traducteur de l'anglais vers l'arabe :	19
III.3. Le système AlethTrad:	21
III.3.1. L'analyse des textes:	21
III.3.2. Traduction , révision et validation:	21
III.3.3. Traduction multi-niveaux:	22
III.3.4 Architecture d'AlethTrad:	24
III.4 Le projet METAL:	25
III.4.1 Historique et évolution de METAL:	25
III.4.2 Le système METAL:	25
III.4.3 Les outils du développement lexical:	27
III.4.4 Evolutions:	28
III.5 Le système TOSHIBA :	29
III.5.1 Introduction:	29

III.5.2 Les caractéristiques du système:	29
III.5.3 Configuration du système:	29
III.5.4 La méthode de traduction :	30
III.5.4.1 Analyse morphologique - dictionnaires :	30
III.5.4.2 Analyse Syntaxique:	31
III.5.4.3 Analyse Sémantique:	32
III.5.4.4 Transfert structural:	33
III.5.4.5 La génération:	34
III.5.4.5.1 La génération syntaxique:	34
III.5.4.5.2 La génération morphologique:	34
III.6. Le système TORJOMAN:	36
III.6.1. Introduction:	36
III.6.2. Aspects fonctionnels :	36
III.6.3. Aspects techniques :	36
III.6.3.1. La compilation des dictionnaires :	37
III.6.3.2. L'analyse de l'anglais :	38
III.6.3.3. Le transfert et la génération :	38
III.6.4. Caractéristiques du système TORJOMAN :	38
III.6.5. Architecture du 'moteur' de traduction :	39
III.6.6. Remarques sur le système TORJOMAN :	40

Partie 2 : Conception et réalisation d'une application

IV. Modèle de traduction :	41
IV.1. Formalisme de description du langage:	41
IV.1.1. Définition:	41
IV.1.2. Choix du formalisme :	42
IV.2. Présentation de la grammaire DCG :	42
IV.3. Principe général de la conversion:	43
IV.4. Caractéristiques de notre modèle:	45
IV.5. Environnement de développement:	46
IV.5.1- Transcription des caractères arabes :	46
V. Difficultés du traitement de la langue arabe:	48
V.1. Généralités:	48
V.2. Quelques propriétés graphématiques de l'arabe:	48
V.2.1. Les voyelles dans la langue arabe:	49
V.2.2. Les signes diacritiques:	49
V.3. Propriétés morphologiques de l'arabe:	51
V.3.1. La racine:	51
V.3.2. Les morphèmes :	51
V.3.3. Les proclitiques:	51
V.3.4. Les enclitiques:	52
V.4. Propriétés syntaxiques de l'arabe:	52
VI. Description de notre prototype de TAO :	59
VI.1. Organisation du lexique:	59
VI.1.2. Le dictionnaire électronique réduit propre à cette application:	59
VI.2. Grammaire d'analyse:	61
VI.2.1. Rôle du formalisme grammatical:	61
VI.2.2. Grammaire du prototype de TAO :	62
VI.2.3. L'analyseur du prototype de TAO:	65

VI.3. Prototype de TAO basé sur le modèle à transfert de structures :	66
VI.3.1. Le modèle ML-TASC:	67
VI.3.2. Corpus du prototype :	69
VI.4. La génération :	73
VI.4.1. Les différents niveaux de la génération :	73
VI.4.2. La génération de niveau I :	74
VI.4.3. La génération de niveau II:	75
VI.4.4. La génération de niveau III:	78
VI.5. Architecture du système prototype de TAO :	79
VII. Réalisation informatique	80
VII.1. Introduction :	80
VII.2. Caractéristiques de l'application :	80
VII.2.1. Type de phrases analysées :	80
VII.2.2. Les catégories syntaxiques retenues:	80
VII.2.3. Les flexions retenues :	81
VII.3. Environnement du prototype :	82
VII.3.1. Le lexique :	82
VII.3.2. La grammaire implémentée :	82
VII.4. Présentation du système prototype :	83
VII.4.1. L'analyseur lexical :	83
VII.4.1.1. Principales règles de l'analyseur lexical :	84
VII.4.1.2. Quelques résultats du programme :	86
VII.4.2. L'analyseur syntaxique :	87
VII.4.2.1. Principales règles de l'analyseur syntaxique :	87
VII.4.2.2. Quelques résultats du programme :	90
VII.4.3. Le transfert structural :	91
VII.4.3.1 Principales règles du transfert structural :	91
VII.4.3.2 Quelques résultats du programme :	94
VII.4.4. La génération :	95
VII.4.4.1. Principales règles de la génération :	95
VII.4.4.2. Quelques résultats du programme :	97
CONCLUSION	99
BIBLIOGRAPHIE	101

INTRODUCTION

Depuis le lancement de la recherche vers les systèmes de Traduction Automatique (TA), des périodes de grand enthousiasme et des périodes de profonde désillusion se sont succédées vis-à-vis de la TA.

Le rapport « Automated Language Processing Advisory Committee » (ALPAC) se présentait comme un compte rendu d'échec des travaux antérieurs en TA et qui devait ralentir considérablement les recherches dans ce domaine.

Ceci n'a pas empêché des sociétés japonaises ou européennes, des universités anglaises ou américaines... de continuer à promouvoir les recherches pour élaborer de nouveaux systèmes de TA ou de Traduction Assistée par Ordinateur (TAO).

Les causes de l'échec des premiers systèmes de TA sont multiples. On en retiendra essentiellement deux :

- 1°) La prédominance: des informaticiens et/ou des linguistes dans les équipes de TA et la quasi-absence de l'expert du domaine (qui peut être un traducteur praticien). Il en résulte que la TA a été réduite à une simple confrontation entre langues et que l'équation a été faite trop rapidement entre automatisation et informatisation.
- 2°) La pression économique: privilégier les résultats rapides et au moindre coût, la TA se voyait ainsi réduite au rôle de la traduction utilitaire.

Parallèlement aux systèmes de TA se sont développés des systèmes de TAO. Ceux-ci ont l'avantage d'être plus légers, moins onéreux mais plus efficaces. Ils présentent l'inconvénient de contraindre le traducteur à « dialoguer » avec la machine pendant tout le long de la traduction.

Par cette technique, on vise à réduire le nombre et la complexité des erreurs produites par le système

Un grand nombre de travaux théoriques appliqués dans le domaine de la TA ou la TAO ont vu le jour à travers plusieurs périodes, caractérisés par des architectures de systèmes différentes. Ainsi, la vocation des projets de recherche en ingénierie linguistique & informatique est de perfectionner les modèles de dialogue homme-machine, les technologies et les méthodes de Traitement Automatique du Langage Naturel (TALN).

L'informatique peut être accablée de bien des défauts, mais l'une de ses principales vertus est d'être souvent plus performantes que les humains, lorsqu'un mécanisme peut être décrit par une algorithmique précise qui rend possible l'automatisation.

Cependant, malgré la diversité des approches et des théories appliquées à la TA ou la TAO, aucun système de traduction ne peut être qualifié de définitivement opérationnel. Si certains semblent être satisfaisants pour des langues à haut degré de ressemblance comme les langues européennes, la traduction de langues appartenant à des familles différentes reste cependant un

terrain d'étude et de recherche pour adapter les diverses algorithmes d'analyse ou de génération d'une langue à une autre (cas de la langue arabe).

Par ailleurs, la représentation du Langage Naturel (LN) ou de sous-ensemble de LN aux niveaux morpho-lexical, syntaxique et sémantique constitue un problème fondamental dans la conception des systèmes de TA ou TAO.

En effet, le choix de formalismes adéquats pour la représentation des langages en vue de leur traitement automatique est l'une des difficultés dans le domaine du TALN.

Notre objectif dans ce mémoire de D.E.A est de contribuer au traitement des aspects de la représentation des formalismes du LN - cas de sous-ensemble - en vue de la traduction .

Aussi, le présent travail s'inscrit dans le prolongement des recherches effectuées à Lyon1, Lyon2 et l'ENSSIB sur le traitement automatique de la langue arabe (particulièrement les travaux sur la morphologie de l'arabe, la thèse de Doctorat en informatique de M. Hassoun 1987 et la thèse de Doctorat en linguistique de J. Dichy , 1990).

Il prolonge par ailleurs la recherche donnant lieu à la conception d'un prototype de TAO du français vers l'arabe - application à un sous-ensemble du LN - .

Il s'agit de mettre au point un prototype de TAO (français-arabe) exploitant des travaux qui sont réalisés dans le projet SYDO (**SY**stèmes **DO**ocumentaires, créé en 1975 et rassemble le CRLS-Lyon2, le LID-Lyon1 et l'ENSSIB) sur la langue française et particulièrement sur la langue arabe au sein du programme SAMIA (**S**ynthèse et **A**nalyse **M**orphologiques **I**nformatisées de l'**A**rabe).

Le programme SAMIA se caractérise par l'élaboration de modèles de simulation informatique de certains aspects de l'activité langagière :

- ⇒ Base de Données lexicales (relative aux bases nominales et bases verbales),
- ⇒ Conjugueur des verbes arabes.
- etc.

Ces aspects sont analysés et représentés dans des modèles linguistiques opérant en synthèse et/ou en analyse.

Nous présentons dans ce mémoire, la description de prototype de TAO du français vers l'arabe par application à un corpus qui concerne des recommandations sur un carnet de cheques bancaires. Le principe de la conception est basé sur un modèle à transfert de structures dit aussi de deuxième génération.

Nous présentons, dans ce travail par chapitre, les aspects suivants:

Premier chapitre: les domaines de connaissances de la TA.

Deuxième chapitre: un état de l'art de la TA ou la TAO, en classant les systèmes par génération et en spécifiant leurs caractéristiques.

Troisième chapitre: la description de quelques systèmes de TA et TAO commercialisés ou réalisés.

Quatrième chapitre: les principes de base et la description globale d'un modèle pour un système de TA français-arabe.

Cinquième chapitre: la présentation des aspects et difficultés de traitement informatique de l'arabe en comparaison avec le français.

Sixième chapitre: la description de notre conception pour notre prototype de TA du français vers l'arabe - appliquée à un corpus / sous ensemble du LN - .

Septième chapitre: la réalisation informatique de notre prototype de TAO du français vers l'arabe. La présentation des principaux algorithmes de traitement sous forme de règles Prolog et l'édition de quelques résultats du prototype .

Les principales motivations qui ont guidé le présent travail sont :

- ⇒ le besoin en modèles de représentations du langage écrit,
- ⇒ le besoin en systèmes de TA ou de TAO de la langue arabe,
- ⇒ le besoin en TA ou TAO de documents particulièrement techniques et scientifiques.

Notre contribution dans le présent travail par rapport aux recherches effectuées à Lyon1, Lyon2 et l'ENSSIB sur le traitement automatique de l'arabe, se caractérise par une orientation vers le domaine de la TAO pour mettre en relation les différents travaux déjà réalisés sur l'informatisation de la langue arabe.

Partie 1 :

TOUR D'HORIZON

Chapitre 1:

La traduction automatique

I. La Traduction Automatique:

I.1. Introduction:

La tâche d'un système de traduction automatique peut être définie simplement comme suit:

L'ordinateur doit être capable, à partir d'un texte d'une Langue Source (LS) de produire son équivalent dans une Langue Cible (LC) [NIRENBURG 87b] .

La qualité du texte produit en génération LC est irréprochable et similaire au travail d'un traducteur professionnel [ARNOLD 87].

Les efforts pour la conception et la réalisation des systèmes de TA peuvent être résumés par des solutions aux problèmes posés par le texte en LS ou en LC (ie. les diverses facettes de compréhension).

Un certain nombre de questions importantes peuvent être envisagées sur ce point de vue :

- 1- Quelle est la signification du texte ?
- 2- A-t-il certaines composantes structurelles ?
- 3- Quel est le représentant de la compréhension d'un texte ?
- 4- Quel est l'ensemble des extractions de significations d'un texte?
- 5- Est-il absolument nécessaire d'extraire la signification (tous les termes signifiants ou de compréhension) dans un but de traduire ?

La **question 1-** soulève un problème de base en linguistique et en philosophie du langage. Notre tâche est dévouée à la théorie et la méthodologie dans le champ de la linguistique et de l'Intelligence Artificielle (IA).

La **question 2-** concerne les composantes structurelles qui nous intéressent sont d'ordre : logico-morphologique, lexicale, syntaxique, lexical-sémantique et contextuelle (sémantique inférentielle et pragmatique).

La **question 3-** se rapporte aux problèmes de la représentation des connaissances , aussi bien dans le style et le contexte fait en IA.

La **question 4-** concerne les reflets des problèmes computationnels, qui sont l'entreprise de la traduction automatique .

La **question 5-** implique les réponses négatives qui reflètent les souhaits des chercheurs en TA et les limites pratiques de l'état de l'art en TA ou TAO.

Ces questions soulèvent des problèmes difficiles à résoudre et pour le moment aucune solution définitive n'a été trouvée dans le développement en linguistique computationnelle et en IA .

I.2. Les connaissances nécessaires pour la TA:

I.2.1. Les dictionnaires: [HASSOUN 87],[FARWELL 93]

Les traducteurs humains utilisent les dictionnaires comme source d'information pour les langues LS et LC.

Le type de dictionnaire souvent utilisé par l'homme est le dictionnaire bilingue , qui connecte les unités (mots) de la LS et la LC. Un point important à remarquer est que ces dictionnaires bilingues sont conçus pour l'usage humain.

La situation est peu différente quand le dictionnaire est utilisé par le programme d'un ordinateur.

On peut illustrer ces types de dictionnaire et les modules d'exploitation, comme suit: on suppose que le système reçoit en entrée un texte allemand:

"Das Buch liegt auf dem Tisch." (1) .

La traduction en anglais de (1) est : "The book is on the table." (2). Le dictionnaire nécessaire pour la performance de cette traduction est le suivant :

<u>Allemand</u>	<u>Anglais</u>
auf	on
Buch	book
das	the
dem	the
liegt	is
Tisch	table

Le programme de traduction a fourni une substitution des mots allemands en anglais un par un.

L'analyseur à ce niveau est **lexical** ou **orthographique**, ie: fondé sur l'aspect des mots ou aspect **morphologique**.

Peut on utiliser le même système pour passer de l'anglais vers l'allemand ?

Non , parce que dans ce cas plusieurs relations vont du mot "the" à "das" et "dem".

Les connaissances inscrites dans ce dictionnaire sont insuffisantes pour résoudre cette ambiguïté et il faut ajouter d'autres connaissances pour faire le choix adéquat.

I.2.2. Les connaissances syntaxiques : [LYTINEN 87], [SABAH 90a]

L'un des problèmes de l'analyse automatique que résout la syntaxe est le processus de désambiguïsation relevant des différentes catégories d'un mot.

La syntaxe décrit la structure des mots et des connexions associées dans une phrase. L'analyseur à ce niveau est **grammatical**, fondé sur les règles d'agencement des groupes de mots et des mots entre eux.

Comme par exemple : "The coach lost a set." , sans connaissance de la structure de la phrase anglaise , il est impossible de savoir si :

- 1- le mot "coach" est un nom ou un verbe ,
- 2- le mot "lost" est un verbe ou un adjectif ,
- 3- le mot "set" est un nom ou un verbe ou un adjectif .

Ainsi les connaissances syntaxiques de la langue anglaise , nous permettent, dans ce cas, d'éliminer douze (12) cas d'ambiguïtés et de faire le choix correct pour la compréhension.

Pour ce faire , la reconnaissance de la structure syntaxique de la phrase par des règles de réécriture permet d'identifier les différents mots, unités ou groupes de la phrase (syntagmes nominaux , syntagmes verbaux...).

Ce type de connaissance est enregistré dans la grammaire anglaise ou dans toute autre grammaire de la LS d'un système de TA.

Les règles de réécritures pour notre exemple :

S → NP + VP
NP → D + N
VP → V + NP

avec: S : (Sentence) phrase,
NP : (Noun Phrase) Syntagme nominal,
VP : (Verbal Phrase) Syntagme verbal,
D : (Determiner) déterminant,
N : (Noun) nom et V: (Verb) verbe.

Le résultat de l'analyse est :

➤ S[NP[D(the),N(coach)], VP[V(lost),NP[D(a),N(set)]]].

I.2.3. Les connaissances sémantiques:

[PITRAT 85],[RICHARD 95], [LYTINEN 87]

La sémantique s'occupe du signifié des mots et des connexions associées dans une phrase. L'analyseur est fondé sur **des liens de signification entre les mots**, comme les anaphores, et traite les problèmes de polysémies (ie. les mots ayant plusieurs significations).

L'une des premières tentatives dans le domaine de l'analyse sémantique est d'utiliser les représentations des connaissances d'un langage naturel pour un système de traduction.

Ceci a été réalisé à l'université de YALE aux USA par les travaux de Shank en 1975: les Dépendances Conceptuelles (DC) dans un langage de représentation des connaissances . Les DC sont utilisées pour représenter le sens de la phrase; les dictionnaires bilingues utilisés dans ces systèmes prennent en considération des choix de marques ou de mailles , pour représenter les connaissances sémantiques.

Ces mailles sont utilisées lors de la phase de génération pour que le système connecte la signification au mot adéquat dans la LC.

On illustre la représentation en DC d'une phrase anglaise :

phrase: "Mary hit John."

DC-phrase : (event EV001
 (action PROPEL)
 (agent Mary)
 (object John)
 (instrument *?* unknown)
 (force * above-average *)
 (intentionality * positive *)
).

I.2.4. Les connaissances contextuelles: [CARBONELL 87],[SABAH 90b]

La pragmatique s'occupe de la cohérence générale des éléments sémantiques entre eux. L'analyseur est fondé sur le bon sens.

Exemple: « Alors que Marie parlait, Jean buvait ses paroles. »

Morphologie: tous les mots sont corrects du point de vue orthographique.

Syntaxe: les règles de grammaires sont correctement appliquées.

Sémantique: l'anaphore « ses » se rapporte à Marie, ie: Alors que Marie parlait, Jean buvait les paroles de Marie.

Pragmatique: le verbe « boire » implique un complément d'objet direct « liquide » or « les paroles » ne sont pas un « liquide » donc il y a une erreur pragmatique ! .

Par contre, si l'on sait que le contexte est « poétique », alors il s'agit d'une métaphore et la phrase est correcte.

Les connaissances contextuelles permettent d'éliminer des ambiguïtés provenant de causes très diverses [PITRAT 85] :

a) L'ambiguïté peut venir de la grammaire :

L'ambiguïté peut venir de la grammaire qui n'a pas pu choisir entre deux analyses. On illustre ce cas, par l'exemple suivant:

"Je viens d'écrire un article sur mon bureau."

Le syntagme prépositionnel "sur mon bureau", peut se rattacher à :

- 1- " article " ou
- 2- " écrire ".

b) L'ambiguïté peut venir du sens des mots :

Dans la phrase : "J'achète un terrain à mon voisin.", le verbe "acheter" exprime que le sujet reçoit la possession du complément direct qui était au paravant la possession du groupe prépositionnel.

c) La pragmatique ne peut pas tout résoudre:

Dans la phrase : " Pierre a acheté un bijou à ma voisine.", nous avons ici non seulement deux événements peu probables mais aussi des connaissances générales.

Chapitre 2 :

Etat de l'art de la TA ou la TAO

II. Etat de l'art de la TA ou la TAO :

Nous allons décrire l'état de l'art de la TA ou de la TAO (Traduction automatique Assistée par Ordinateur) en classant les systèmes par génération [RADWAN 94],[BOUALEM 93],[MELBY 87].

II.1. Première génération:(1950-1962)

Un système de ce type est bilingue et fonctionne grâce à plusieurs dictionnaires et à un ensemble de programmes .

Il n'est fondé ni sur des théories linguistiques ni sur des théories de compilation des langages formels.

Le processus de traduction consiste essentiellement en :

- 1- le remplacement des mots de la LS par ceux de la LC selon les dictionnaires ,
- 2- quelques arrangements locaux par des programmes ad-hoc,

L'aspect purement bilingue de ces systèmes exige :

- le recensement des phénomènes linguistiques les plus intéressants sur un couple de langue,
- la codification de ces phénomènes sous forme de dictionnaires et de programmes.

Inconvénients:

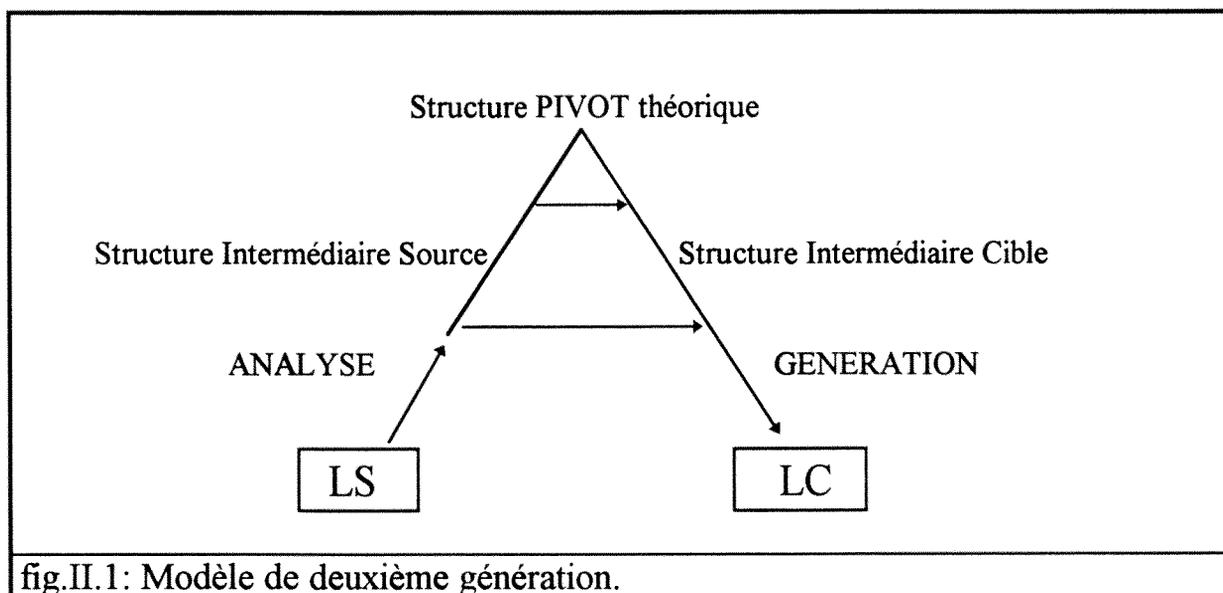
- L'accumulation des phénomènes linguistiques devient si importante, que l'insertion de nouveaux phénomènes dans les programmes, de plus en plus complexes, devient pratiquement impossible [CULLINFORD 87]. Le niveau de qualité atteint assez vite sa limite.
- Les programmes linguistiques sont directement écrits dans les langages de programmation de bas niveau .

II.2. Deuxième génération :(1960-1970)

Une autre approche pour améliorer la qualité de la TAO , consiste en la division du processus de traduction en trois phases logiques :

- 1- Analyse,
- 2- Transfert,
- 3- Génération.

Illustration par la fig.II.1 .



La sortie de la phase **analyse** est un descripteur structural du texte en LS.

Ce dernier étant transformé en descripteur structural équivalent en LC par la phase de **transfert** [KABBAJ 91].

Ce descripteur cible est alors transformé en texte de LC par la phase de **génération**.

Seule la deuxième phase logique fait intervenir le couple de langues ; plus l'analyse est profonde, plus la distance entre les deux descripteurs structuraux est courte [NAGAO 87]. Les systèmes de la deuxième génération permettent la séparation des données linguistiques (les dictionnaires et les grammaires) et les programmes (compilateurs et moteurs des transducteurs).

Cette séparation permet d'utiliser le même outil logiciel pour toutes les langues et la division du processus en trois phases est bien adaptée à la traduction multilingues [DORR 93], [NIRENBURG 87a].

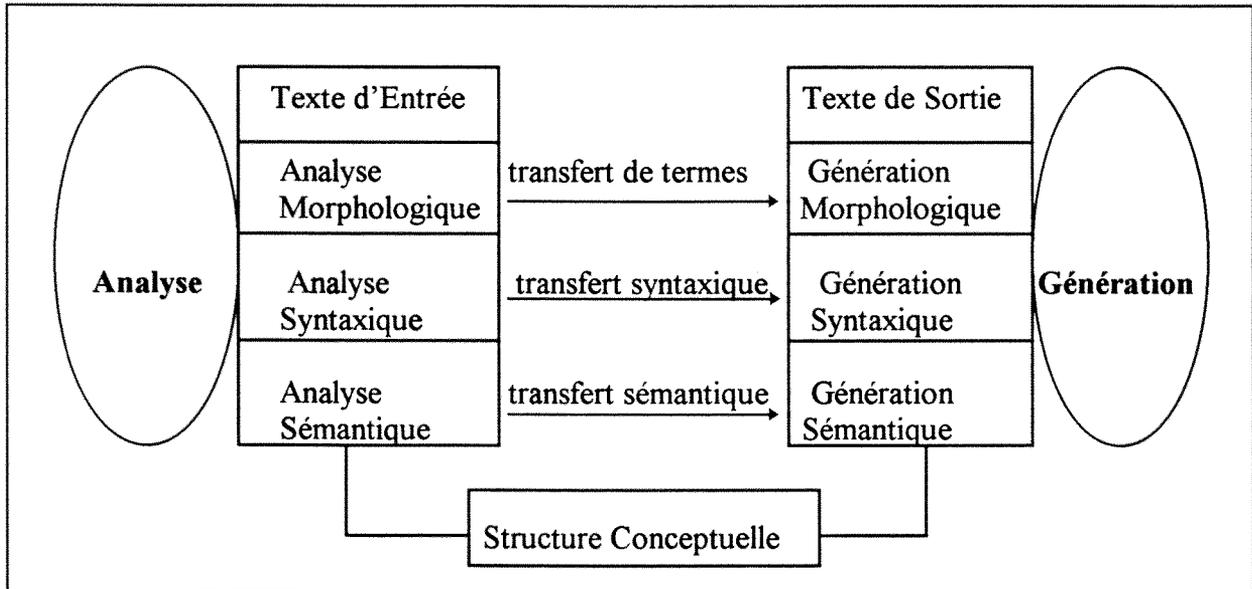
Inconvénients:

- Dans un système de deuxième génération la sémantique n'est exprimée que par des traits sémantiques qui sont analogues à des traits grammaticaux .
- Le problème des ambiguïtés est souvent difficile à résoudre, la cause est le manque de critères linguistiques ou sémantiques pertinents.

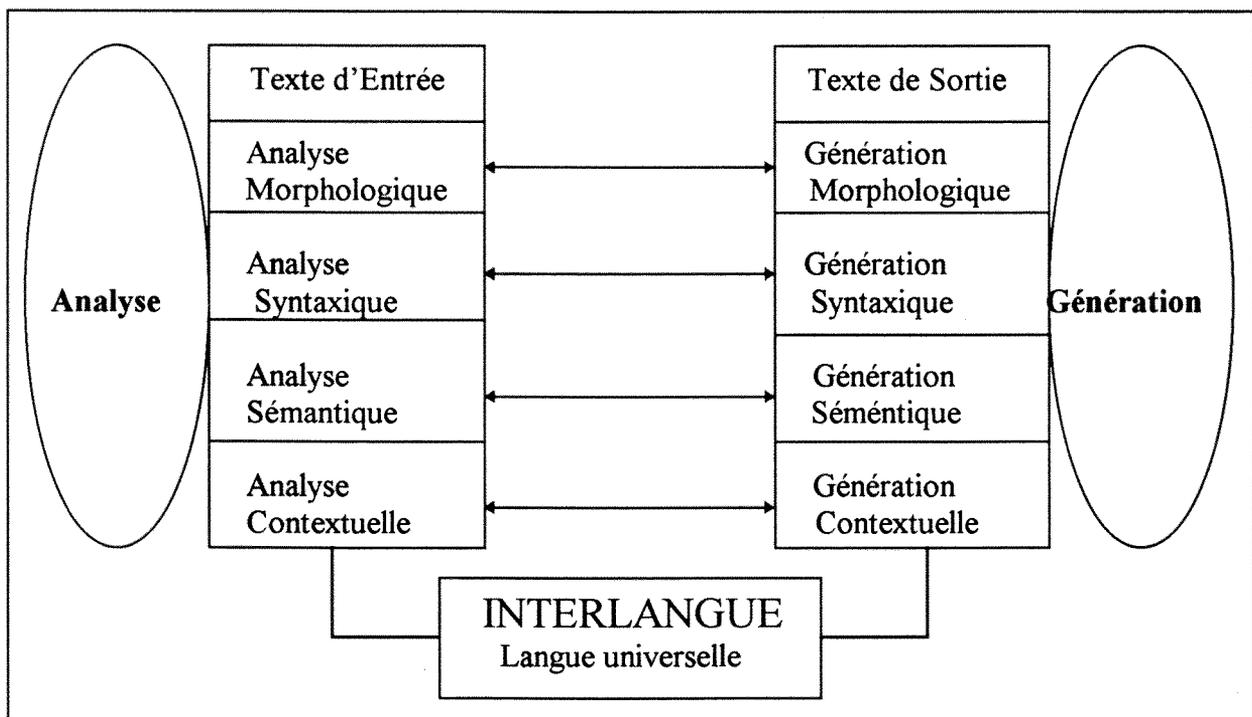
II.2.1. Evolution de l'architecture des systèmes de deuxième génération: [ABBOU 95a]

L'architecture des systèmes est comme suit:

II.2.1.1. Le modèle à transfert:(type PIVOT)



II.2.1.2. Le modèle à interlangues:(type SHALT2)



II.3. Troisième génération:

Le principe des systèmes de troisième génération est de viser une compréhension explicite du texte source:

- 1- Le texte source est analysé et transformé en une représentation conceptuelle indépendante de la langue [MCDONALD 87], il s'agit d'une interprétation (au sens logique) dans un modèle formalisé.
- 2- Divers mécanismes d'inférence utilisant des connaissances extra-linguistiques et celle du contexte ont permis d'enrichir la représentation conceptuelle [GROSS 95].
- 3- A partir de la structure obtenue, on produit le texte cible.

Les études et les réalisations actuelles ne sont que des approches partielles de certains aspects du problème, et cela pour des textes concernant des domaines très restreints.

II.3.1. Architecture d'un système de troisième génération:

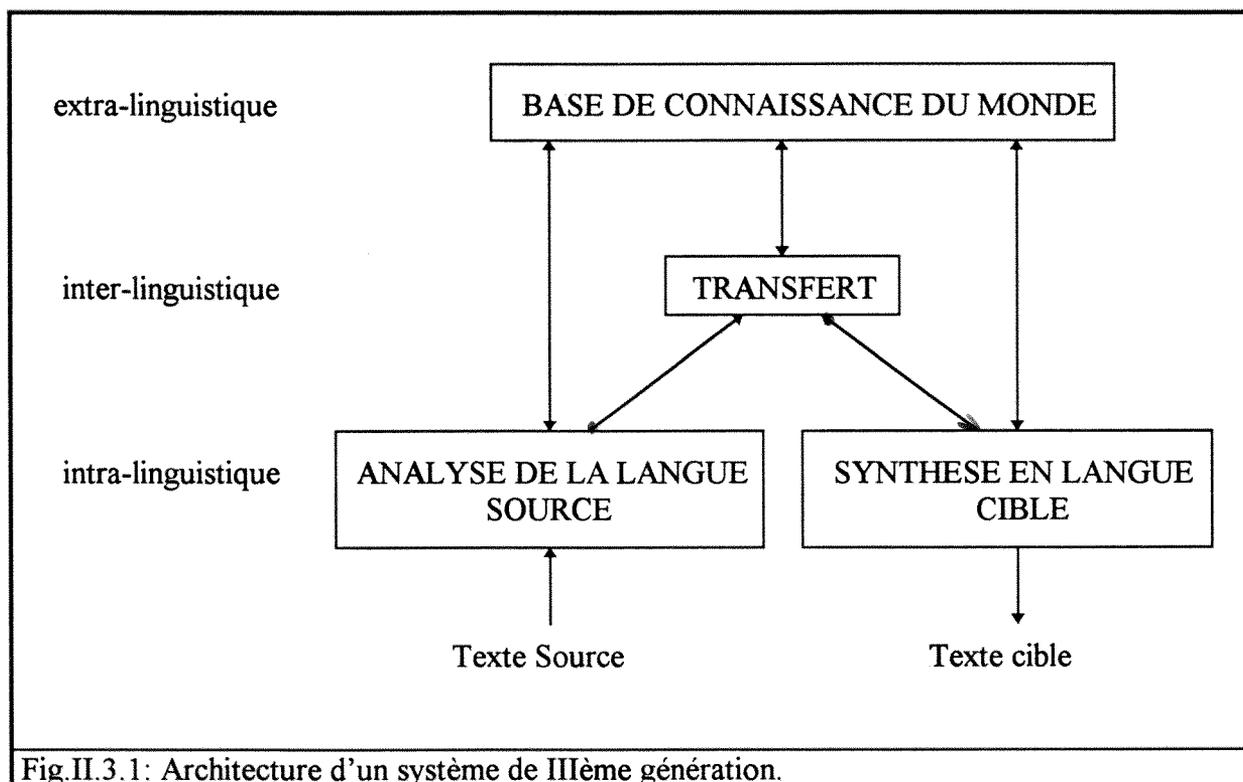


Fig.II.3.1: Architecture d'un système de IIIème génération.

Chapitre 3 :

Description de quelques systèmes
de TA

III. Description de quelques systèmes de TA:

Note : *Tous les schémas dans ce chapitre sont empruntés aux auteurs des travaux présentés.*

III.1. Le système ALT-J/E: [ABBOU 95b]

Développé par le NTT: Laboratoire des Sciences de la Communication au Japon. Le projet a débuté en 1985.

III.1.1. Les dictionnaires:

Le système ALT-J/E est doté d'un grand dictionnaire sémantique:

	mots communs	noms propres	termes techniques	autres
Nombre d'entrées	100.000	200.000	70.000	30.000

Sachant qu'il y a 3000 catégories sémantiques qui sont ventilées sur 80 domaines .

- Un dictionnaire de transfert japonais/anglais de 16.000 structures verbales.
- Un dictionnaire de règles syntaxiques et sémantiques adapté au domaine d'application requis.

Celui-ci permet de prendre en compte le contexte au niveau d'un paragraphe, de traduire des expressions idiomatiques, de réécrire automatiquement des phrases japonaises en phrases standards, de traduire spécifiquement les enclitiques japonais, de traiter les oppositions, de générer des nombres et de comptabiliser les noms anglais, de générer des pronoms possessifs en anglais et enfin d'ordonner correctement les mots pour l'écriture des adresses.

III.1.2 Le schéma général du système: (Voir Fig.III.1)

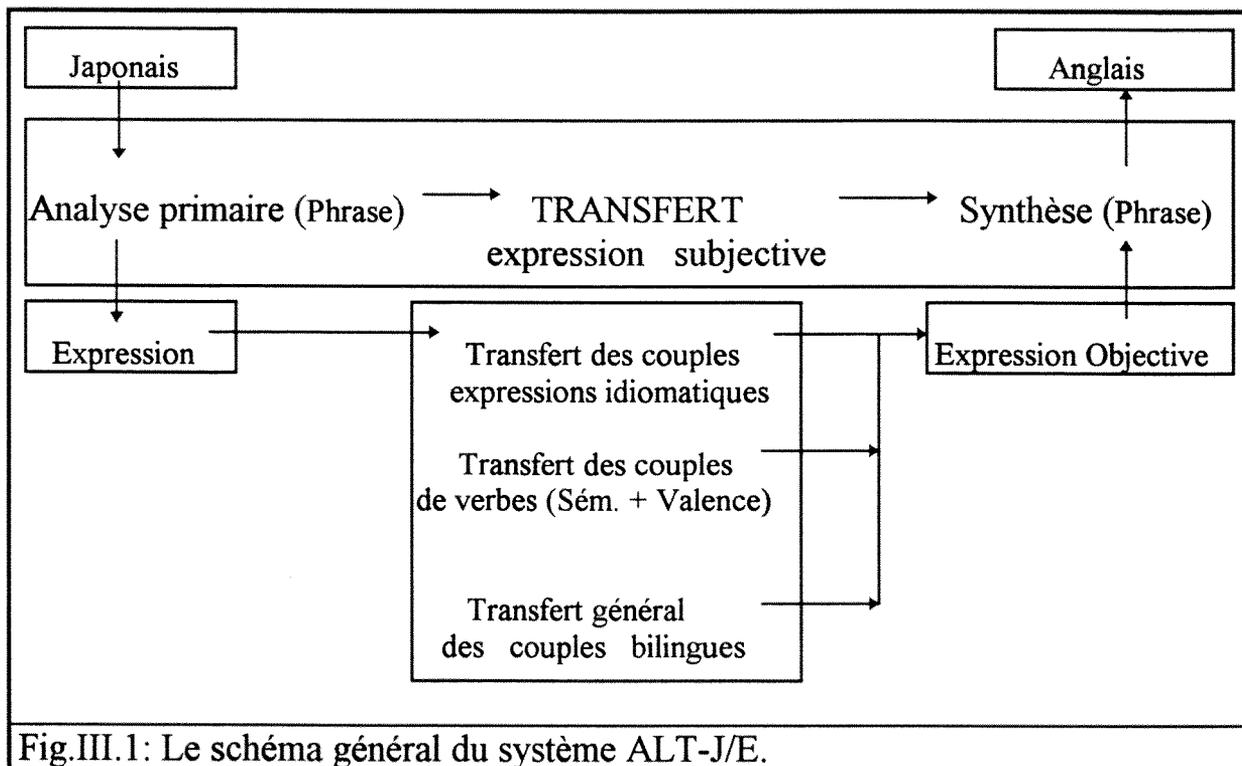


Fig.III.1: Le schéma général du système ALT-J/E.

III.1.3. Fondement théorique du système: [ABBOU 95b]

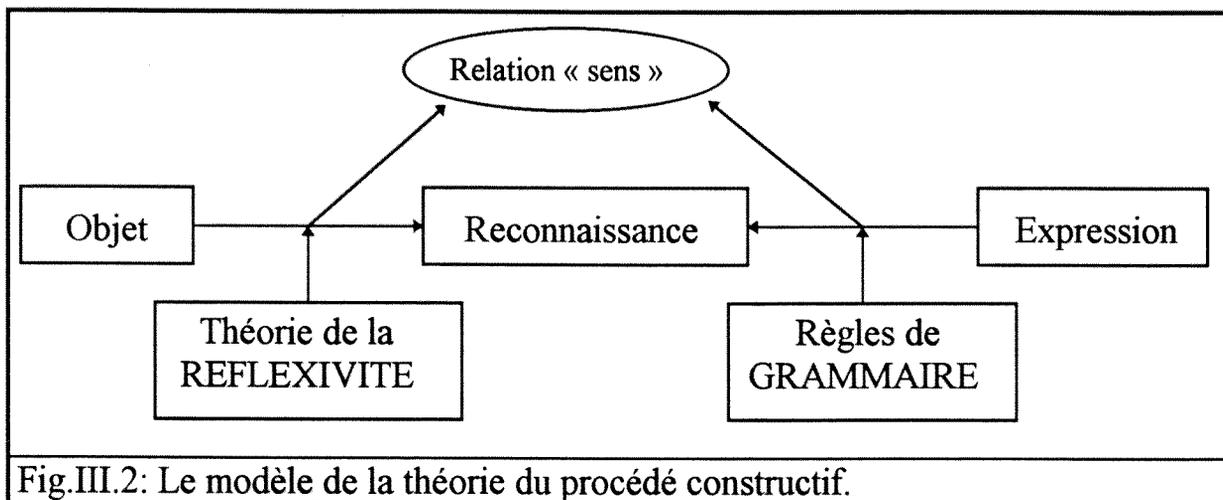
La *Théorie du Procédé Constructif* (TCP) est issue des travaux d'un grande école linguistique japonaise (cf. Pr M. Miyazaki, Japon IA N° 28).

Elle s'appuie sur la grammaire générale de Port-Royal (1660) et plus récemment, sur la grammaire générative de N. Chomsky (1950). Un des fondements essentiels de cette théorie réside dans la théorie du sens.

Selon cette théorie, trois processus interagissent:

- 1- Le monde des objets du discours,
- 2- la reconnaissance de ce monde par son interlocuteur,
- 3- leur expression (des interlocuteurs).

Une relation dynamique s'établit entre ces trois pôles pour construire un sens intelligible par des êtres en communication. (Voir [fig.III.2](#))



III.1.4 Mécanisme de traduction du système:

La TCP repose elle-même sur une méthode de traductions multiniveaux.

Les connaissances du traducteur en matière de catégories subjectives sont mises en relation avec le caractère subjectif de la phrase à traduire.

Ensuite, le caractère objectif de cette phrase est rendu objectivement en LC (exp. l'anglais), en respectant les niveaux idiomatiques, sémantiques de valence verbale et standard.

Enfin, les catégories subjectives servent à former une phrase anglaise correcte à partir de la structure objective de la phrase.

Remarques:

- 1- Pour la langue japonaise, langue agglutinante, l'expression finale englobe une partie objective représentée par les noms, les verbes et les adjectifs, et une partie subjective représentée par des "joshis", ie: des suffixes.
- 2- Le japonais accumule les mots et n'indique l'inflection subjective qu'à l'aide de ces suffixes ajoutés au verbe; alors que les langues indo-européennes (anglais, français,...) sont construites d'une manière diamétralement différente et affectent aux signes linguistiques des catégories.
- 3- Diverses théories et méthodologies sont mis en oeuvre pour le traitement de la langue japonaise.
- 4- La rapidité de traduction est observée avec les ordinateurs à structure parallèle ou massivement parallèle pour la résolution des ambiguïtés syntaxiques et la génération.

III.2 Le système ARIANE: [MONEIMNE 89],[VEILLON 70],[VAUQOIS 75]

Le système ARIANE est développé par le groupe (ex-CETA) GETA à Grenoble en France en 1978.

Le système ARIANE 78.4 est destiné à développer des programmes de traduction de documents écrits dans diverses langues naturelles.

Il appartient à la famille des systèmes de deuxième génération utilisant la technique de transfert, avec beaucoup d'améliorations.

Le système ARIANE est un environnement complet de programmation pour la TAO, il comporte:

- 1- Quatre langages spécialisés pour la programmation linguistique: TRANSF, ROBRA, ATEF, SYGMOR.
- 2- Un moniteur interactif est l'interface entre les utilisateurs et tout le système. Il assure le dialogue avec les utilisateurs en demandant des paramètres et enchaîne les programmes du système en réalisant des tâches complexes en fonction des paramètres obtenus.
- 3- Un ensemble de programmes utilitaires qui facilitent la mise au point et l'exploitation des programmes linguistiques.

III.2.1 Le processus de traduction:

Le processus de traduction d'une LS vers une LC est divisé en 3 étapes logiques:

- 1- Analyse
- 2- Transfert
- 3- Génération.

Chacune étant décomposée en au moins deux phases successives, comme l'illustre le diagramme ci-dessous: fig.III.2.1.

Phase logique	phase physique	langage (LSPL)
ANALYSE :	Analyse Morphologique : AM	ATEF
	Complement Lexical : AX	EXPANS
	Complement Lexical : AY	EXPANS
	Analyse Structural : AS	ROBRA
TRANSFERT :	Transfert Lexical : TL	EXPANS
	Complement Lexical : TX	EXPANS
	Transfert Lexical : TS	ROBRA
	Complement Lexical : TY	EXPANS

GENERATION :	Complement Lexical : GX	EXPANS
	Génération Structurale : GS	ROBRA
	Complement Lexical : GY	EXPANS
	Génération Morphologique : GM	SYGMOR
Fig.III.2.1: Les phases du système.		

III.2.1.1 Analyse:

Le texte d'entrée en LS est soumis à l'Analyseur Morphologique (AM), écrit en ATEF, qui produit comme résultat un arbre décoré.

Ce résultat est ensuite transformé en un autre arbre (descripteur multiniveau) par l'Analyseur Structural (AS), écrit en ROBRA.

III.2.1.2 Transfert:

a) Transfert Lexical: TL

Le TL est écrit en TRANSF. Il transforme l'arbre au moyen d'une consultation des dictionnaires bilingues multichoix. Le traitement des polysémies et des tournures complexes est possible.

b) Transfert Structural: TS

Le TS traite les différences structurales entre 2 langues, en rendant la structure linguistique intermédiaire conforme à la nature de la langue cible.

Cette phase finit également une partie du TL (choix d'équivalents en fonction d'un contexte étendu, traitement des tournures par le TL), et traite les phénomènes qui n'ont pas pu être calculés assez profondément par l'AS.

III.2.1.3 Génération:

a) Le Générateur Syntaxique: GS

A partir du descripteur produit par le transfert, le GS calcule :

- Une structure de surface qui est une paraphrase possible du texte (contraintes par les impératifs de la traduction). Concrètement, elle recalcule les fonctions et les classes syntaxiques à partir des relations logiques et sémantiques.
- L'ordre des mots.

- La génération des mots « outils » absents et de toutes les questions liées à l'actualisation.

b) Le Générateur Morphologique: GM

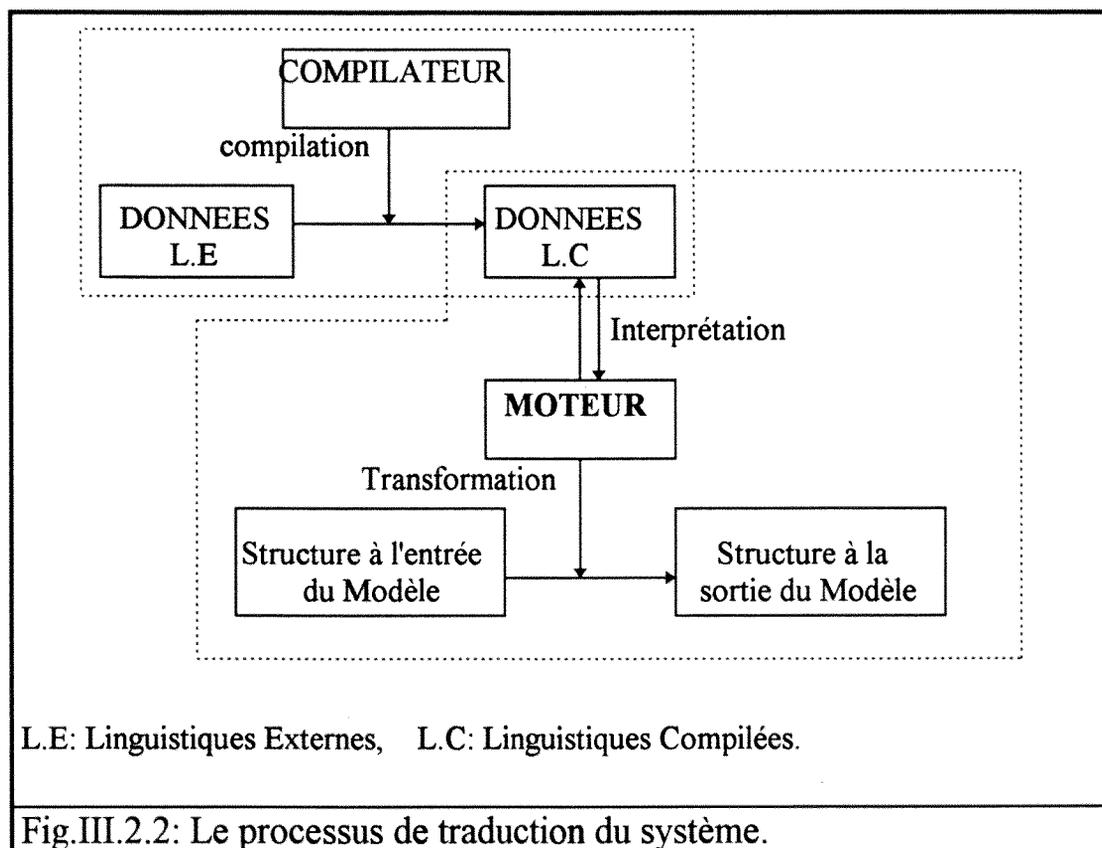
Il travaille sur les feuilles de l'arbre (ou terminaux), produit par le GS, et produit des formes morphologiques correctes constituant un texte en LC .

Chaque étape est écrite dans un LSPL réalisé à l'aide d'un compilateur et d'un moteur interprétant les tables produites par le compilateur.

III.2.2 Evolution d'ARIANE: ARIANE-G5, 1988

Depuis 1988, une nouvelle version d'ARIANE a été mise en service ARIANE-G5.

Le processus de traduction est toujours subdivisé en 3 étapes logiques; chacune des étapes est décomposée en au moins deux phases successives comme l'illustre le tableau fig.III.2.2 .



III.2.3 Utilisation de l'aspect multilingue d'ARIANE:

Une des particularités importantes du système ARIANE réside dans son utilisation multilingue. Ainsi pour développer un système complet multilingue avec N langues, il faut réaliser:

- ↳ N analyses ,
- ↳ $N*(N-1)$ transferts,
- ↳ N générations .

L'architecture du système avec ses différentes phases : (fig.III.2.3).

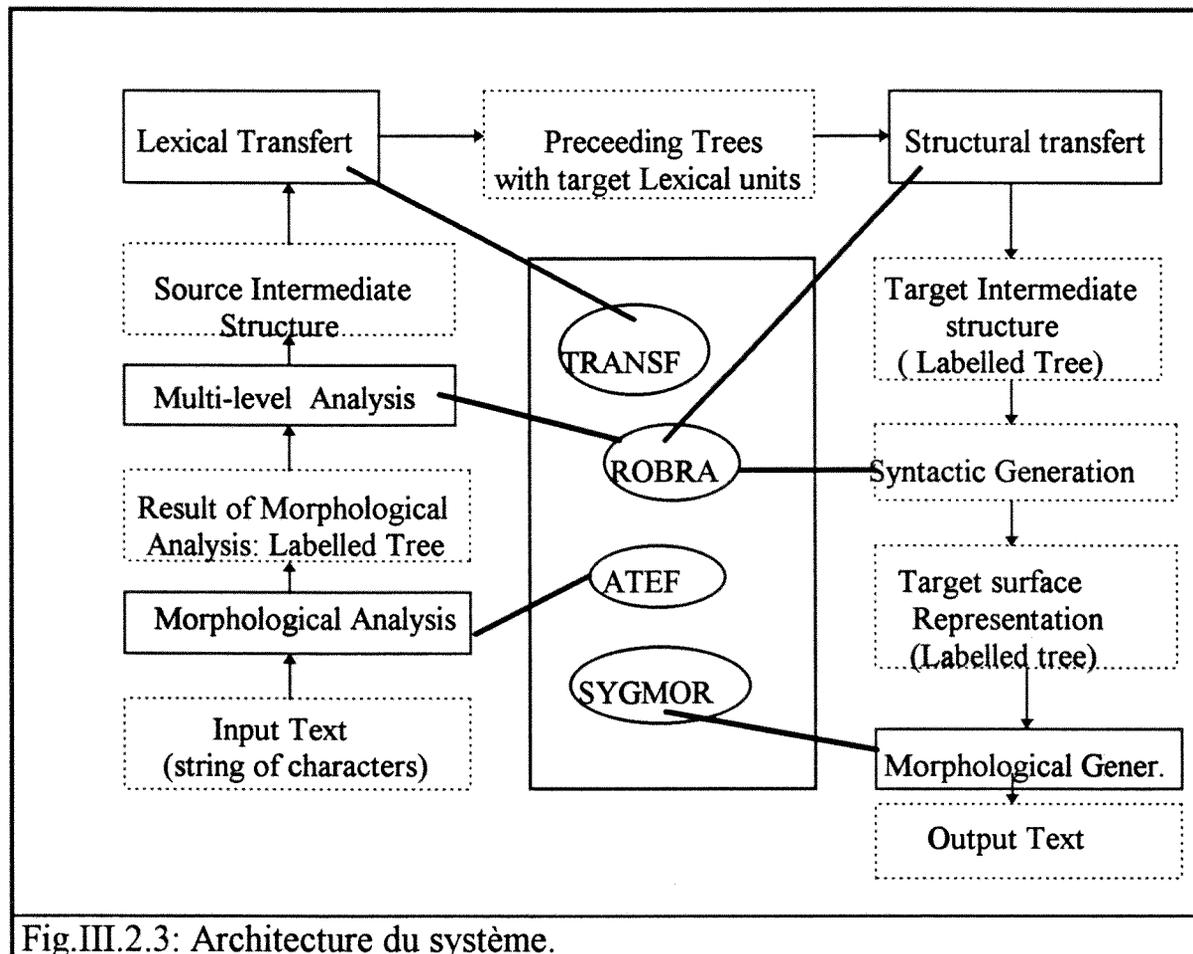


Fig.III.2.3: Architecture du système.

III.2.4 Application - traducteur de l'anglais vers l'arabe :

[MONEIMNE 89]

Grâce à l'aspect multilingue du système, Moneimne a réutilisé l'analyse de la LS: anglais pour la génération vers l'arabe (fig.III.2.3).

a) Développement du transfert vers l'arabe:

Dans l'approche du GETA, la phase doit être la plus courte possible. Elle se limite aux dictionnaires bilingues: TL et à un TS minimal. Cette particularité permet de gagner du temps dans le développement de nouveaux couples de langues.

b) Développement de la génération de l'arabe:

La génération de l'arabe est une des composantes les plus importantes du travail mené par Moneimne en 1989. L'utilisation de la langue arabe dans le système ARIANE représentait une dure épreuve pour:

- les tests sur le système lui-même,
- la théorie linguistique du GETA .

Toutes les possibilités du système sont utilisées pour développer la génération syntaxique (GS) .

Pour la GM, il a fallu faire un traitement particulier des verbes compte tenu du système dérivationnel de la langue arabe. Cette partie a été traité par le langage ROBRA.

Enfin, une spécification de la grammaire arabe, sous forme de grammaire statique, a été élaborée .

Remarques:

- 1- L'abandon par le GETA du concept de langage pivot contenu dans le projet du CETA.
En fait, cette langue pivot est la chimère des projets de TA/TAO: une représentation machine unique et non ambiguë qu'on donnerait à tout texte pour le traduire en n'importe quelle langue. Une telle représentation nécessiterait une représentation totale du signifié, valeurs pragmatiques comprises. On en est bien loin !
- 2- La caractéristique essentielle du langage informatique de haut niveau ROBRA, utilisé en analyse, en transfert structural et en génération syntaxique, est sa capacité d'analyse récursive.
Cela permet au système, lorsqu'il s'aperçoit qu'il suit "*un mauvais cheminement*" à travers le graphe de contrôle des sous-grammaires de revenir en arrière et de défaire une partie du travail pour prendre un autre cheminement possible.
Cette faculté risque naturellement d'engendrer un certain bouclage si aucune voie ne semble aboutir.
- 3- La maîtrise des procédures permettant d'interrompre certains traitements et de produire des traductions "*sous-optimales*" à différents niveaux, semble particulièrement délicate.
- 4- Sur le plan calcul, le système ARIANE réalise 1,5 million d'opérations pour traduire chaque mot dans le traitement.

Le résultat de cette traduction est un fichier qui présente les phrases du texte source suivies de leur traduction.

Après, la phase du traducteur/réviseur et pour apporter d'éventuelles corrections, le texte est livré au client, et au stockage en mémoire de traduction. Voir fig.III.3.1.

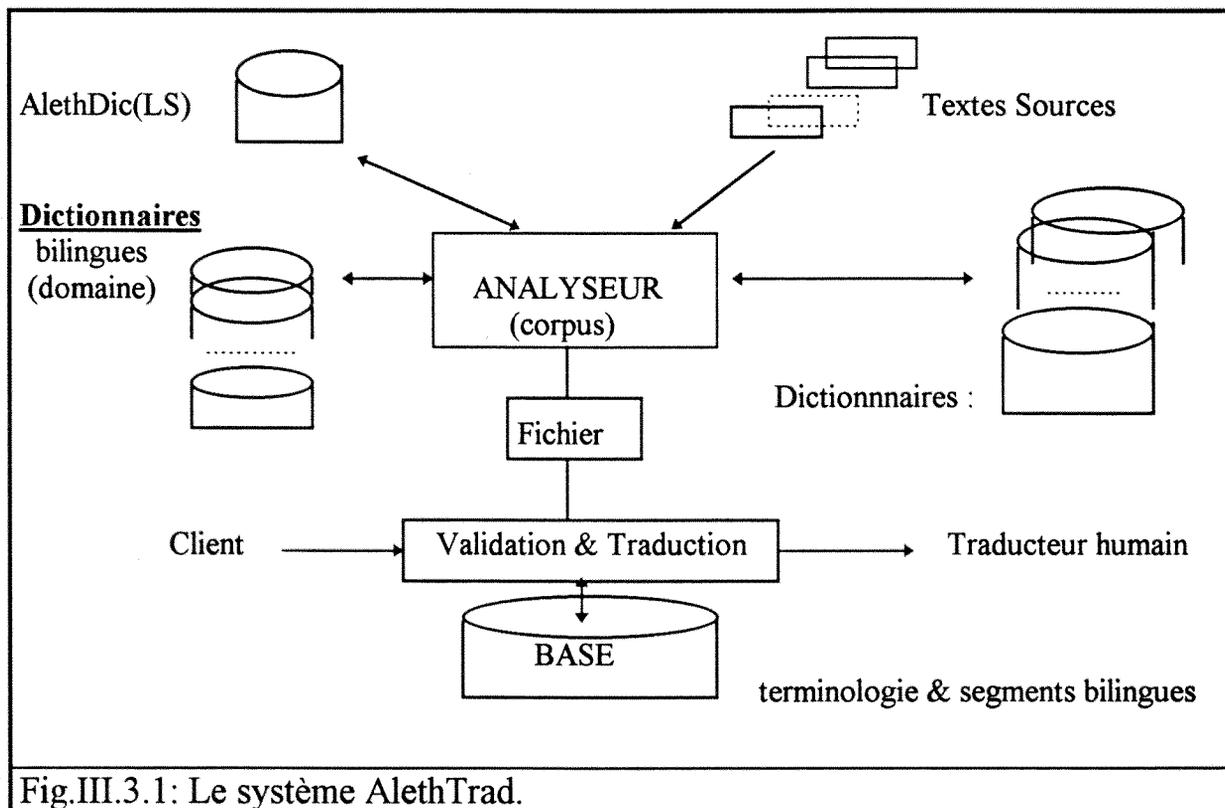


Fig.III.3.1: Le système AlethTrad.

III.3.3. Traduction multi-niveaux:

Au niveau de développement actuel, AlethTrad travaille sur les trois premiers niveaux:

a) Niveau 1: Le traitement

- Chercher dans une mémoire de traduction une phrase correspondant à celle en cours d'analyse.
- Dans le cas où aucun résultat n'est obtenu, le système déclenche un algorithme d'appariement approximatif (fuzzy matching) qui va lui permettre de faire une recherche sur des chaînes peu différentes (exp. les numériques, mots de types grammaticaux, variantes flexionnelles...).

Ainsi, des séquences de phrases à traduire déjà présentes dans la mémoire de traduction seront traitées automatiquement.

b) Niveau 2: analyse morphologique

Si le premier niveau échoue, le système passe au traitement du deuxième niveau qui effectue une analyse morphologique sur des éléments de la phrase .

Il procède à une lemmatisation :

- les noms : leur forme au singulier,
- les adjectifs : masculin | féminin ,
- les verbes : l'infinitif, etc.

Puis le système construit des groupes de mots pour tenter un traitement du niveau-1.

↳ Le résultat de l'analyse prend la forme d'une succession d'éléments phraséologiques ou bien d'éléments pharséologiques suivis d'éléments simples.

Dans le premier cas, le système déclenche une traduction directe à partir de la juxtaposition d'éléments phraséologiques. Dans le deuxième cas, le système déclenche un nouvel examen .

c) Niveau 3: Le moteur de traduction

Ce niveau intervient sur le résidu du document non-traduit par l'approche « *mémoire de traduction* » .

Le moteur déclenche des algorithmes impliquant l'utilisation des grammaires d'analyse, de transfert et de génération. La traduction se déclenche en 4 étapes :

1- *L'analyse morphologique* : niveau 2 .

2- *L'analyse syntaxique* : Le travail, à ce niveau, consiste à identifier certains groupes ou syntagmes (SN,SP,SV) sur lesquels interviendront des transformations structurelles.

3- *Les transformations structurelles* : Une fois les structures identifiées, celles-ci sont transformées en structures de la langue cible .

Pour ce faire, le système s'appuie sur des règles de transformation guidées par des informations contenues dans les dictionnaires multilingues.

Les règles de traduction implémentées traitent des cas génériques. Cependant, elles cherchent à couvrir les phénomènes linguistiques identifiés dans les documents à traduire .

4- *La génération* : Elle est prise en charge par un automate qui exploite les informations de ces structures .

Le système opère des substitutions lexicales, des insertions d'éléments lexicaux, générer les flexions des mots en langue cible , etc.

III.3.4 Architecture d'AlethTrad: (Voir fig.III.3.2)

AlethTrad a été développé en langage C et il tourne sur des machines Unix (stations de travail). Il travaille en mode serveur: des postes de traduction en réseau (Unix, Mac, PC).

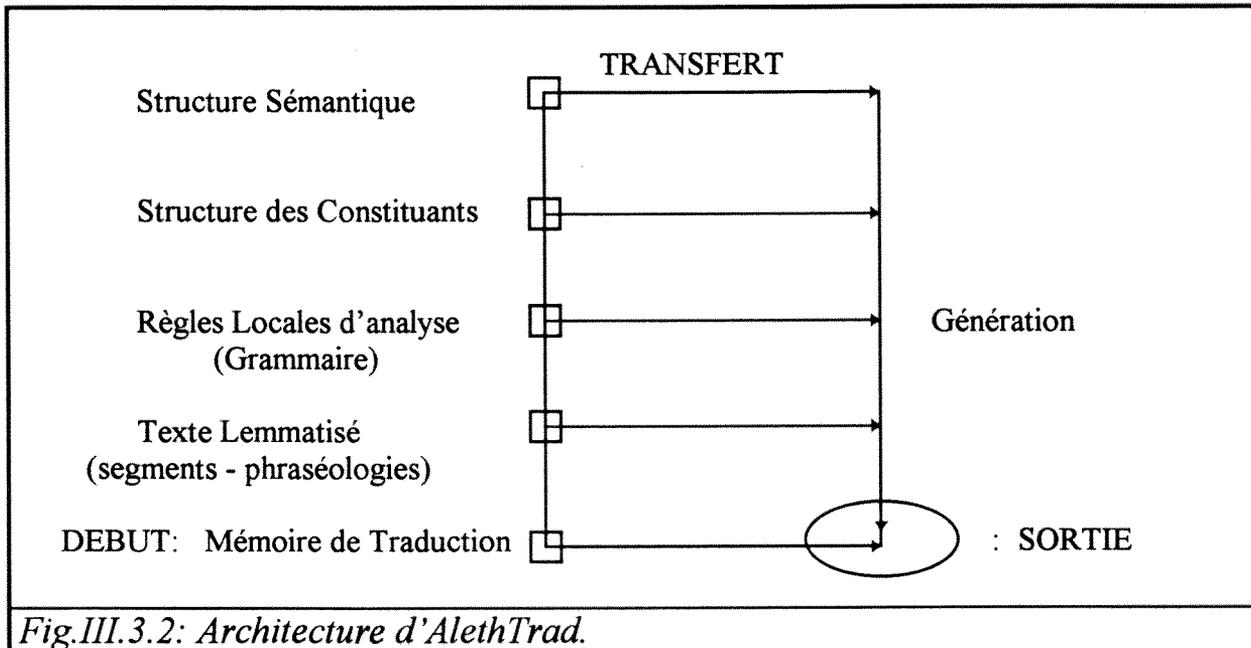


Fig.III.3.2: Architecture d'AlethTrad.

Le logiciel d'édition de textes utilisé en interface par les traducteurs est FRAMBuilder s'offre des fonctionnalités supplémentaires, comme la gestion des documents structurés en format SGML ou la communication de ces documents à d'autres logiciels par intermédiaire d'un API.

Une interface Word est en cours de développement et le premier bilan d'AlethTrad est de plus 40.000 pages traduites.

Remarques:

- 1- AlethTrad n'est pas un système de TA/TAO, mais un Outil d'Aide à la Traduction.
- 2- Le système est adapté aux besoins des utilisateurs.
- 3- La phase de post-traitement consiste à adapter le texte traduit vis-à-vis du texte source en recherchant, par exemple, des calques de structures et d'attributs. Ce point constitue une préoccupation.
- 4- La génération en langue cible est prise en charge par un automate, en opérant par des substitutions (lexicales, insertions d'éléments lexicaux, génération des flexions,...). Chacune de ces opérations est en cours d'innovations.

III.4 Le projet METAL: [WHITE 87]

Origine: Université de Texas - USA et SIEMENS en Allemagne .

Période: 1961-1990 .

III.4.1 Historique et évolution de METAL:

Année	Langage de programmation & Machine
1961 - 1970	Fortran, CDC 6600
mi- 1970	Lisp UCI-Lisp InterLisp, DEC-10 InterLisp, DEC System-20
fin- 1980	InterLisp, Symbolics Computers LM-2

III.4.2 Le système METAL:

Le système METAL traduit un texte d'une LS vers une LC par analyse automatique :

- 1- Morphologique,
- 2- Syntaxique,
- 3- Sémantico-syntaxique : relations et liens exprimés en LS.

METAL est qualifié de système intermédiaire de 2-ème et 3-ème génération.

a) Les outils du développement de la grammaire :

La composante linguistique qu'utilise METAL est une grammaire à structure de phrase augmentée, classiquement c'est la forme de description selon Winograd, 1983.

Dans cette grammaire, les règles s'appliquent lorsque la structure décrite est rencontrée et que certaines conditions sur les constituants de la structure sont vérifiées par l'occurrence des fonctions des constituants.

S'il y a succès, la règle construit un noeud gauche de l'arbre, qui a certaines caractéristiques spécifiques, généralement des Traits de Valeur (TdV).

Une des actions de base dans la construction de l'arbre est le passage relevant des TdV du noeud fils au noeud père et le transfert père-fils.

b) Traduction de phrases :

Dans ce cas, la phrase identifiée par le système permet la production de plusieurs interprétations et donc leurs traductions .

Le système procède par étapes successives :

- ① : L'appel de la fonction TRANSLATE (appel à l'environ. Lisp).
- ② : La mise sous forme d'une liste de la phrase.
- ③ : Le nombre d'interprétations de l'analyseur.
- ④ : Le transfert et la génération .
- ⑤ : La traduction de l'interprétation préférée.

Fig.III.4.1: Les étapes du système METAL.

On illustre par un exemple la traduction de l'allemand vers l'anglais:

- ① (translate)
- ② Sentence: (Ein alter, verheirater Mann hat dieses Programm niemals geschrieben.)
- ③ 1 interpretation in 25314 ms. 286 phrases: 250 rejected.
- ④ Transfert plus generation time: 14538 ms.
- ⑤ (|an| |old| |married| |man| |has| |never| |written| |this| |program| .)

Fig.III.4.2: Traduction de l'allemand vers l'anglais par METAL.

c) Le « P-TREE » :

Le P-tree ou P-arbre ou arbre décoré est la forme d'interprétation étalée communément utilisée par les développeurs du système. Il apporte du sens en visualisant la phrase analysée.

Le P-tree prend obligatoirement les arguments de base et ceux optionnels. Le premier argument (de base) est le nombre des interprétations, le second est le « tree-type » .

Le tree-type étale la forme de la structure de la phrase après la construction de tous les noeuds et l'application des règles sur la structure .

Le P-tree peut être visualisé soit en phase d'analyse soit en phase de transfert. Voir Fig.III.4.3 et Fig.III.4.4 .

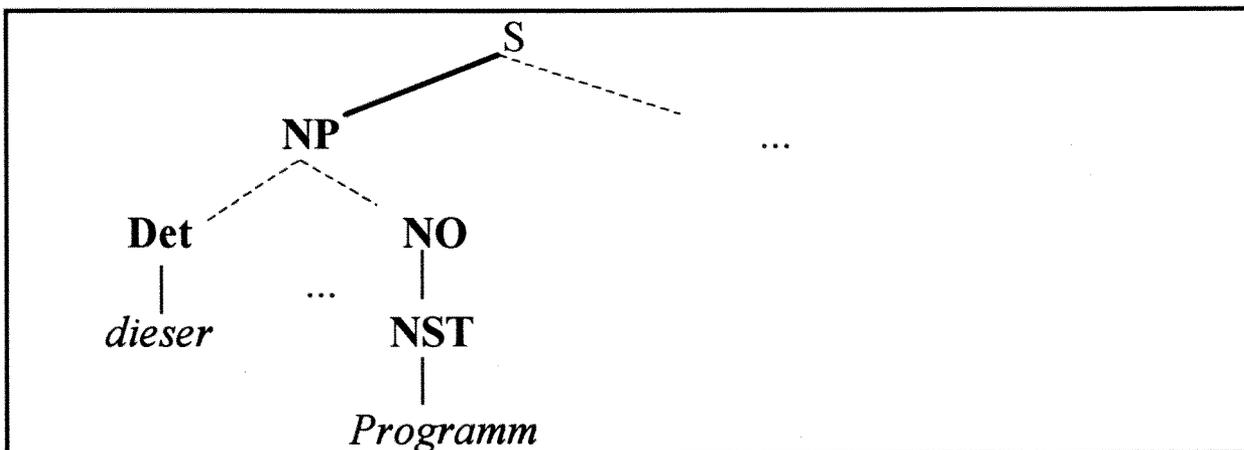


Fig.III.4.3: Arbre d'analyse.

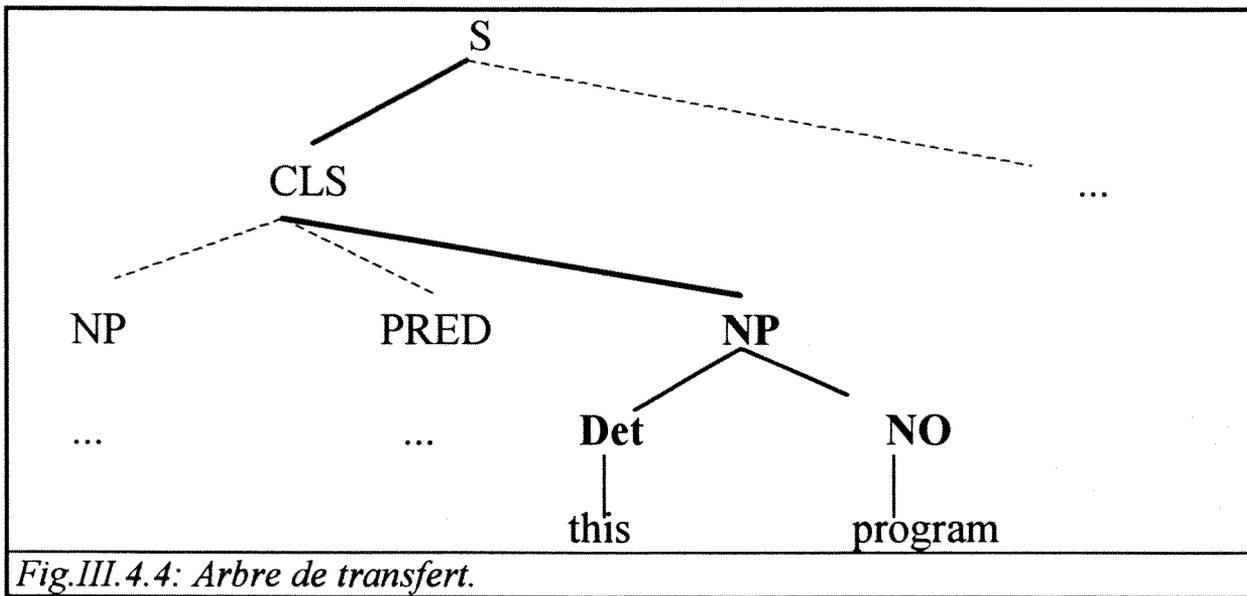


Fig.III.4.4: Arbre de transfert.

III.4.3 Les outils du développement lexical:

a) L'inter-codeur :

L'interface entre le système METAL et l'utilisateur est le moyen de coder et de modifier le lexique.

Quand l'inter-codeur est appelé pour consulter un mot ou un terme en LS et si ce mot n'existait pas dans le lexique du système, celui-ci crée une entrée dans la base, en présentant un menu complet pour remplir les différents champs correspondant et c'est ainsi que l'utilisateur peut intervenir .

b) Les fonctions lexicales :

Le système METAL permet l'accès à la BD lexicale dans un mode différent de celui d'un utilisateur averti . Cet accès demande des moyens d'aides et en quantité de lexique.

Ainsi, la création et l'installation d'un ensemble terminologique ou le recodage de la base existent quand il y a changement ou ajout dans la grammaire du système. Ceci demande une forme d'accès qui importe des mesures de sûreté et une exécution en batch.

Les fonctions lexicales sont converties par le détecteur d'erreur dans la base « Defaulter », le pré-analyseur « Preanalyser » et le valideur « Validator » pour le lexique en batch.

III.4.4 Evolutions:

Les recherches associées au projet METAL ont pris en considération le fait que le système s'oriente à des fins de traductions industrielles.

Pour ce faire, la concentration des travaux menés par les linguistes et les informaticiens assure l'efficacité des résultats .

Les outils de travail au sein du projet sont implémentés dans un langage structuré et ayant un format familier pour les linguistes.

Ainsi, l'ajout, la modification et la suppression de certaines structures sont facilement manipulables à l'expérimentation et les structures deviennent de plus en plus efficaces .

Ces outils restent utiles pour la création d'un nouveau langage et d'un nouveau système de traitement automatique de la langue (NLP-system) possible et efficient.

Remarques:

- 1- Métal est un système de traduction assistée par ordinateur, le processus de traduction comporte une phase de post-édition.
- 2- Le même résultat de l'analyse, pour une langue donnée, est utilisé pour la génération dans plusieurs autres langues: le concept d'analyse permet d'effectuer des traductions multilingues.
- 3- Enrichissement permanent des dictionnaires.
- 4- Métal est initialement conçu pour traduire de l'allemand vers l'anglais, puis étendu vers d'autres langues européennes.
Les résultats diffusés ne montrent pas la capacité du système pour la traduction dans des langues de classes différentes, comme l'arabe.

III.5 Le système TOSHIBA : [AMANO 89]

III.5.1 Introduction:

Le système TOSHIBA de TA fonctionne sur un mini-ordinateur 32-bits (UX-700) , et traduit des textes scientifiques et techniques de l'anglais vers le japonais.

Il est basé sur la technique de transfert sémantique, pour ceci un modèle de grammaire appelé LTNG:« Lexical Transition Network Grammar » a été développé.

Le système comporte 3 types de dictionnaires:

- 1- Un dictionnaire général ,
- 2- Un dictionnaire pour les termes techniques ,
- 3- Un dictionnaire utilisateur .

Le système est équipé d'un éditeur bilingue anglais/japonais pour le pré/post-édition et des utilitaires (logiciels) .

III.5.2 Les caractéristiques du système:

Le système comporte :

- Un puissant analyseur sémantique,
- un moteur de traduction (Parser): le parseur,
- un éditeur bilingue.

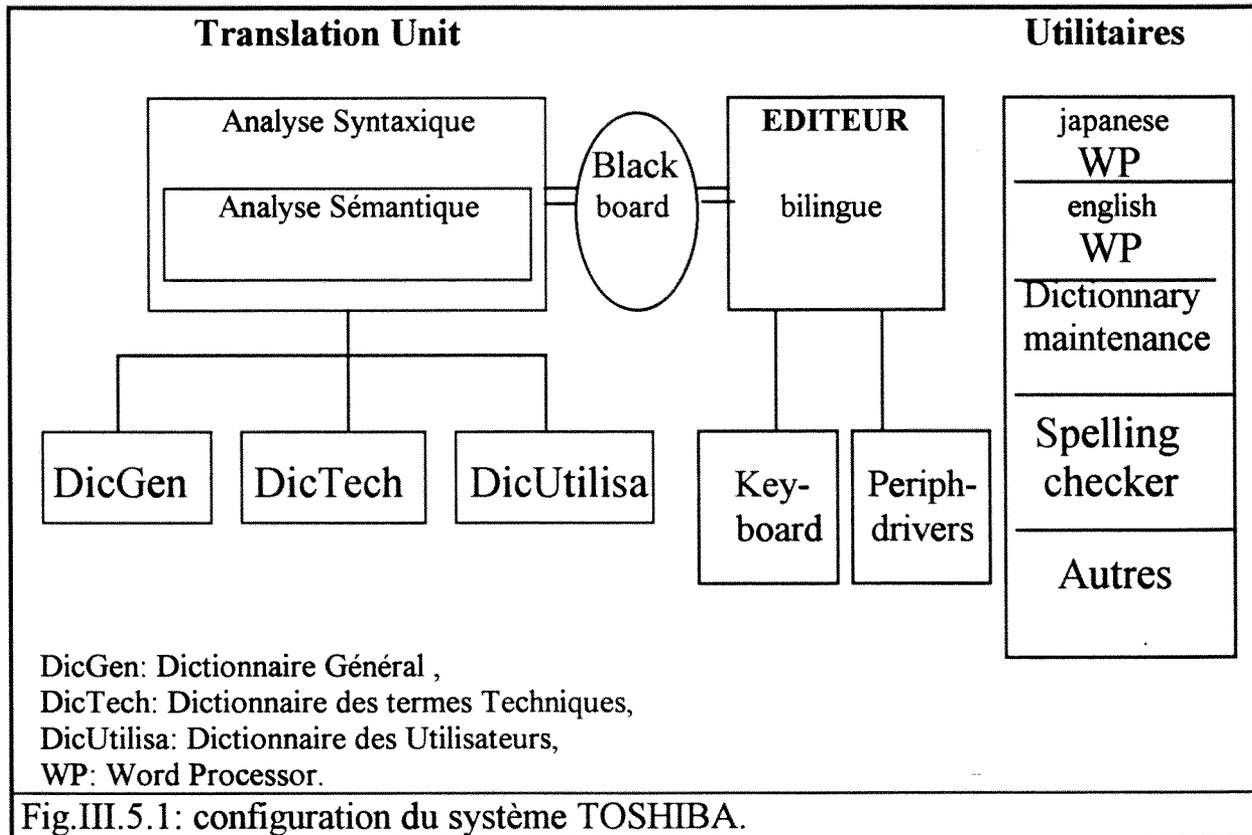
L'analyseur sémantique représente la signification de la phrase comme une structure conceptuelle de celle de la structure syntaxique et de la signification des mots qui apparaissent dans la phrase.

Le travail du système est en conjonction avec l'analyseur sémantique et le parseur avec une stricte séparation entre analyseur syntaxique et sémantique .

L'éditeur bilingue est une interface homme-machine avancée, entre l'opérateur humain et le système TOSHIBA .

III.5.3 Configuration du système:

Le système de TOSHIBA est écrit en langage C et sous Unix (UX-700) et a comme configuration celle du schéma de la Fig.III.5.1 .



III.5.4 La méthode de traduction :

III.5.4.1 Analyse morphologique - dictionnaires :

Il y a 3 types essentiels de dictionnaires utilisés par le système:

- 1- Le dictionnaire général: il comporte environ 50.000 mots d'usage général et d'idiomes.
- 2- Le dictionnaire des termes techniques: il comporte au maximum 50.000 termes techniques partitionnés selon des thèmes : électronique, mécanique...
- 3- Le dictionnaire utilisateur: il comporte des mots spécifiques des utilisateurs qui seront sauvegardés et éventuellement leur traduction.

Les mots dans une phrase anglaise apparaissent avec une morphologie complexe. L'analyseur morphologique divise le mot en morphèmes et construit une structure de mots . (fig.III.5.2)

- ① Reconnaissance des inflexions morphemiques dans un mot :
exp. comme le pluriel '(e)s' et le passé '(e)d' en anglais .
- ② Construction de la structure de mots.

Fig.III.5.2: L'analyseur morphologique .

III.5.4.2 Analyse Syntaxique: ASy

Les analyses syntaxique et sémantique sont séparées et non complètement indépendantes : elles fonctionnent séquentiellement et procèdent dans un mode interactif. (Fig.III.5.3)

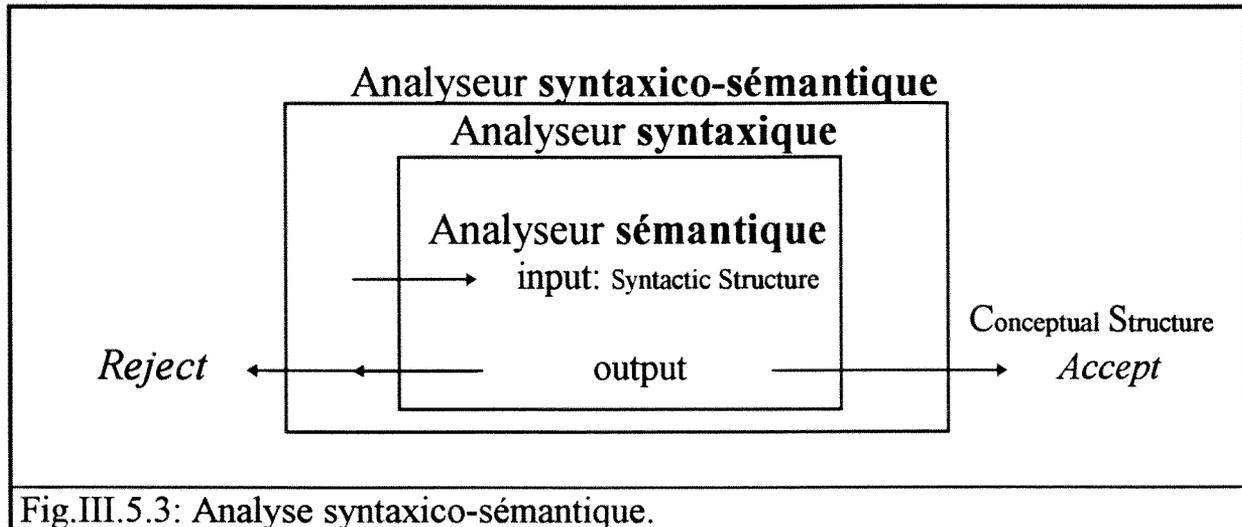


Fig.III.5.3: Analyse syntaxico-sémantique.

Les traits de l'analyse syntaxique du système sont :

1- Une seule structure syntaxique est dérivée de l'ASy. L'ambiguïté est résolue en éliminant les incohérences des catégories des mots et leurs combinaisons.

Si l'ambiguïté persiste, l'analyse sémantique construira une structure conceptuelle plausible et résoudra implicitement les ambiguïtés.

2- L'analyse syntaxique est purement syntaxique et les règles syntaxiques ont des conditions sémantiques .

3- L'analyse syntaxique est composée de deux parties :

- l'Interprétation Syntaxique : IS ,
- la Construction de la Structure Syntaxique .

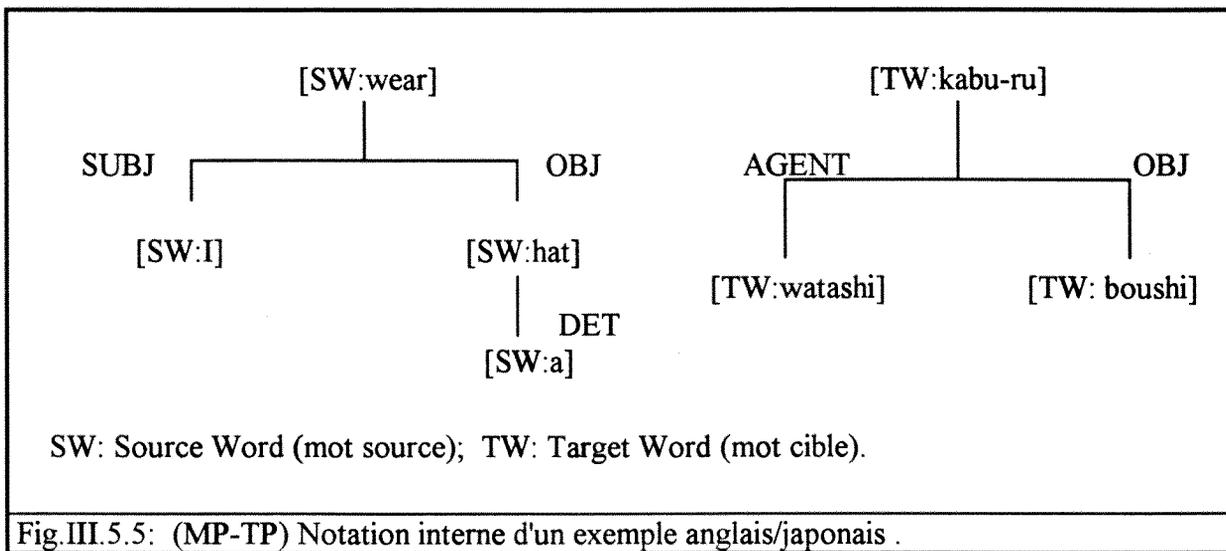
L'interprétation syntaxique des phrases en entrée est séparée du processus de construction de la structure .

L'IS consiste à rechercher les règles syntaxiques à employer et de cette façon les ambiguïtés lexicales sont résolues.

La construction de la structure applique ces règles à la manière d'un parseur pour construire la représentation syntaxique .

Les règles syntaxiques sont décrites par une grammaire ATN (Augmented Transition Network). La fig.III.5.4 , montre la forme générale d'une règle.

MP et TP représentent un arbre, utilisant la même notion pour un arbre syntaxique , i.e. les noeuds sont connectés par des arcs nommés et le nom de l'arc dénote une relation. Voir [fig.III.5.5](#)



III.5.4.4 Transfert structural:

Le transfert lexical est évoqué par les mots ou termes .

Le transfert structural est un autre type de transfert qui est évoqué par les traits des structures syntaxiques .

Le transfert lexical est une sorte de conversion de la syntaxe à la sémantique alors que le transfert structural est une conversion de la langue source à la cible et par conséquent une réalisation de grammaire contrastive .

Un exemple de transfert structural est le traitement de la différence dans le mode "temps" entre le japonais et l'anglais .

L'assignement correct du temps est traité dans le transfert structural qui est syntaxique, non sémantique et dépend d'une grammaire contrastive.

III.5.4.5 La génération:

III.5.4.5.1 La génération syntaxique:

Le rôle de la génération syntaxique est de :

- 1- Déterminer l'ordre des mots.
- 2- Attacher des postpositions (Joshi en japonais).

Les règles de génération décrivent ces deux fonctions. La procédure de génération commence par la règle attachée à la phrase et le noeud tête de la structure conceptuelle.

III.5.4.5.2 La génération morphologique:

En général, la génération morphologique est inscrite dans les dictionnaires .

La façon simple d'enregistrer la traduction des mots est de les mettre dans le champ TW: Target Word de la structure des mots. Cette méthode est employée quand la traduction équivalente est non-inflexionnelle, comme les noms.

Dans le cas de la traduction inflexionnelle, comme les verbes ,les adjectifs... , l'information inflexionnelle doit être présente.

On illustre par l'exemple suivant :

la traduction du mot « read » est « yo-mu », et les informations utilisées pour cette génération sont:

- ① stem : racine « yo »
- ② conjugation type : «5-dan»
- ③ kind of conjugational part : « ma »
- ④ other information (irregular form) : « onbin-kei » .

Le système emploie six catégories morphologiques pour le japonais: (tense) temps, aspect, (modality) modalité, (factitive) factif, passive et négation .

L'exemple, fig.III.5.6, illustre une génération morphologique.

INPUT : « I could not run . »

Morphological analysis: [SW: I] [SW: can, TENSE: past] [SW: not]
[SW: run, TENSE: present]

Syntactic analysis : [SW: I] [SW: run, MODALITY: (can (TENSE:past))
TYPE : negation]

Fig.III.5.6: Informations pour la génération morphologique.

Remarques:

- 1- La langue japonaise est très différentes des langues indo-européennes, notamment par sa morphologie externe.
- 2- Le besoin en systèmes de TA/TAO au japon est important, ceci pour les échanges technologiques et industriels avec les autres pays du monde.
- 3- Le domaine de recherche en linguistique et TALN sont largement considérés, particulièrement au développement de la morphologie (les dictionnaires électroniques et les bases de données morphologiques) et la sémantique (les grammaires appropriées au traitement de la sémantique).

III.6. Le système TORJOMAN: [LABED 90],[LABED 92],[SIDHOM 94]

TORJOMAN est un système de traduction assistée par ordinateur Anglais/Arabe.

Origine: IRSIT, Institut de Recherche des Sciences Informatiques et des Télécommunications en Tunisie.

Période: 1989-1995.

III.6.1. Introduction:

Le système TORJOMAN de TAO fonctionne sur machine compatible IBM-PC sous DOS/Windows .

Il a pour but de traduire des structures de base de phrases anglaises et sans restriction à un domaine particulier.

Le système a débuté en 1989 à l'IRSIT en Tunisie avec la collaboration d'une firme américaine PC-Linguistics au Texas (USA).

III.6.2. Aspects fonctionnels :

TORJOMAN est une première version d'un moteur de traduction automatique. C'est un système complet d'aide à la traduction dans le sens où il est capable de répondre à une demande de traduction , soit:

- 1- en donnant directement un équivalent arabe de la phrase anglaise,
- 2- en proposant à l'utilisateur des choix dans le vocabulaire de base,
- 3- en fournissant en dernier recours, un accès direct au dictionnaire des mots, pour une traduction mots à mots.

Comme le système TORJOMAN est un système de traduction assistée par ordinateur, ce dernier fait appel à une traduction extérieure :

- si le moteur de traduction se trouve confronté à plusieurs traductions possibles d'un terme anglais, il propose une liste de synonymes permettant à l'utilisateur un choix adéquat ,
- convivialité du système en donnant moyen à l'interprète humain d'affiner la traduction fournie par le système.

III.6.3. Aspects techniques :

Le processus de traduction est articulé essentiellement autour de trois phases:

- 1- compilation des dictionnaires ,
- 2- analyse de l'anglais ,
- 3- transfert et génération .

III.6.3.1. La compilation des dictionnaires :

Le moteur de traduction qui est à la base de TORJOMAN fonctionne grâce à trois types de dictionnaires :

- le dictionnaire général des mots ,
- le dictionnaire des expressions ,
- le dictionnaire de l'utilisateur.

➤ Dans le cas où la phrase à traduire correspond à une expression idiomatique, TORJOMAN opère automatiquement sa traduction à partir du **dictionnaire des expressions**.

Exemple:

« like father like son » : ?iNNa HaDaA ?aL-SSiBL MiN DaAKa ?aL-?aSaD .

Le système, dans ce cas, sait qu'il ne s'agit pas d'analyser la phrase car une expression idiomatique porte un sens global indépendant des termes qui la composent.

➤ Dans le cas où il ne s'agit pas d'une expression idiomatique entière, c'est le cas de traitement des expressions idiomatiques à usage verbal. En effet, le verbe en anglais peut être suivi d'une séquence spécifique de mots donnant un sens global au verbe différent de son sens en tant qu'unité autonome.

Torjoman opère par la détection de cette séquence puis sa traduction en l'intégrant au sens global de la phrase.

Exemple 1:

« the man breaks a journey » , si le système ne tient pas compte de l'usage idiomatique du verbe : KaSSaRa ?aL-RaJuLu ?aL-RiHLa& , alors qu'en réalité sa traduction est plutôt : TaWaQQaFa ?aL-RaJuLu ?aTNaA?a ?aL-RiHLa& .

Ainsi, Torjoman reconnaît ce genre de tournure verbale.

Exemple 2:

« pull out » : ?iQTaLa^Ca .

« call off » : ?aL^CGaỴ .

➤ Dans le cas où il ne s'agit pas d'une expression idiomatique ou idiomatique verbale, le système soumet la phrase à l'analyseur syntaxique pour ensuite effectuer les phases de transfert Anglais/Arabe et la génération en arabe (consultation du dico. général des mots).

➤ Quand la phrase contient un nom propre ou un acronyme, le système cherche sa traduction dans un dictionnaire spécial réservé à l'utilisateur. Ce dernier y introduit son vocabulaire propre et spécifique (**dico. de l'utilisateur**).

Exemple:

« IRSIT » : Ma^CHaD ?aL-BuHuT̄ Li^CuLuM ?aL-?i^CLaAMiYa& Wa ?aL-?iTTiSaALaAT^CaN Bu^CD .

➤ En dernier recours, quand la phrase n'est pas acceptée par l'analyseur syntaxique, une traduction mot à mot est fournie.

III.6.3.2. L'analyse de l'anglais :

Des algorithmes d'analyse et de génération forment le noyau du moteur de traduction et permettent au système de réaliser des tâches 'intelligentes' .

Le principal problème en analyse est celui de l'ambiguïté du langage naturel. Ces ambiguïtés peuvent prendre de formes variées:

- 1- lexicales : homographie, homonymie, polysémie.
- 2- structurelles : anaphore, coordination, rattachement.
- 3- sémantiques : sens , texte.

L'analyseur syntaxique de l'anglais est un automate ATN (W.Wood,1970 et T.Winograd,1983) sans le retour-arrière (backtracking), mais complété par des règles ou heuristiques d'élimination des ambiguïtés lexicales (par filtrage et sélection au voisinage des mots).

Ainsi, la grammaire d'analyse est une grammaire transformationnelle qui construit l'arbre de dérivation en donnant la structure profonde de la phrase .

III.6.3.3. Le transfert et la génération :

La stratégie de traduction utilisée par le système est la méthode de transfert, qui est fondée sur une description structurale des langues concernées (anglais/arabe) .

Cette phase de processus de traduction consiste à exploiter les résultats de l'analyse du langage source et à la définition des différentes structures grammaticales du langage cible puis la génération en langue cible.

III.6.4. Caractéristiques du système TORJOMAN :

Torjoman dans des versions antérieures traduisait des phrases simples au mode déclaratif et la voix affirmative. Plusieurs améliorations ont permis l'évolution du système :

- Traitement des coordinations (and, or, but) ainsi la structure de la phrase traitée devient complexe et comporte plus qu'un verbe, adjectif, adverbe ... et plus qu'une phrase simple. Ce qui rapproche progressivement à la structure de texte.

L'introduction des coordinations au niveau des syntagmes a permis au système de tenir compte des cas de duel, tout en considérant les accords des adjectifs et des verbes y afférents.

- Traitement des noms composés complexes qui constituent une particularité de la langue anglaise, des subordinations et des expressions idiomatiques à usage non-verbal.

III.6.5. Architecture du 'moteur' de traduction :

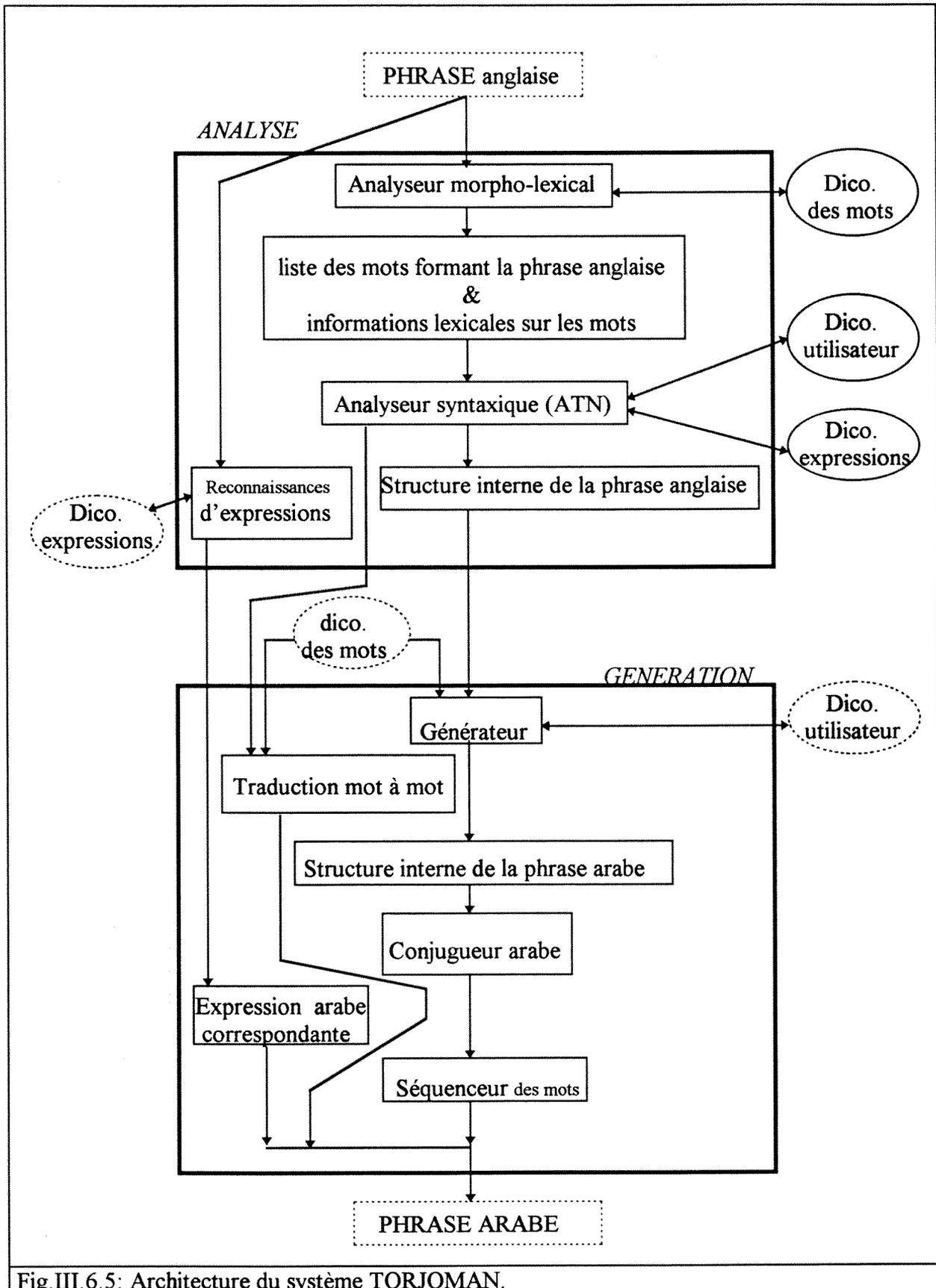


Fig.III.6.5: Architecture du système TORJOMAN.

III.6.6. Remarques sur le système TORJOMAN :

- 1- Le modèle théorique des grammaires transformationnelles, en l'implémentant sous forme des ATN, se heurtait à des problèmes d'implémentation :
Le résultat des recherches de S. Peters et R. Ritchie, 1973 ([ABEILLE 94]-p13) montrait que dans le cas général, on se heurtait à des problèmes d'indécidabilité si l'on tentait d'analyser une phrase incorrecte.
- 2- Les ATN eux-mêmes se révèlent mal adaptés :
Ils étaient lourds à programmer et à modifier, ils appartenaient à un style de programmation dit « procédural » où l'on décrit les objets par les procédures à utiliser pour les construire, mêlant connaissances linguistiques et instructions informatiques proprement dites.
- 3- Dans l'optique d'une analyse plus complète, le lien était difficile à faire avec le calcul d'une représentation sémantique.
- 4- Le dictionnaire général des mots ne permet pas l'accès direct à certaines dérivations d'un mot, comme dans le cas de traduction des noms composés complexes,...

Partie 2 :

**CONCEPTION ET RÉALISATION
D'UNE APPLICATION**

Chapitre 4 :

Modèle de traduction

IV. Modèle de traduction :

Introduction:

Nous présentons dans ce chapitre, les principes de base et la description globale d'un modèle pour un système de TA/TAO (cas du français vers l'arabe).

La description détaillée du système que nous avons cherché à modéliser, à savoir:

- la représentation des données ;
- les algorithmes d'analyse, de transfert et de génération ;
- l'architecture, le fonctionnement ...

sera présentée au fur et à mesure que l'idée sera clairement décrite et que le choix sera effectué dans chaque étape du modèle.

IV.1. Formalisme de description du langage:

[KOUNIALI 93], [BILANGE 92],[ARCHIBALD 91],[METZGER 88].

IV.1.1. Définition:

Le formalisme de description du langage naturel est défini comme étant un langage spécialement conçu pour coder la description des objets constituant le langage à formaliser.

Du point de vue linguistique, un formalisme est construit pour exprimer la description des énoncés dans une langue donnée.

Exemple:

Si on prend l'énoncé: « Le texte est traduit. », il est construit d'un enchaînement: « article+substantif+verbe », qui constitue un énoncé autorisé dans la langue française.

Cet énoncé est décrit aussi à l'aide de règles syntaxiques autorisant l'ordre:

Phrase → Syntagme_nominal + Syntagme_verbal
Syntagme_nominal → Article + Nom
Syntagme_verbal → Verbe .

Ainsi, la traduction de cet énoncé en arabe: *TuRjiMa ?aL-NaSSu* , qui constitue aussi un énoncé autorisé et compréhensible sans ambiguïté dans la langue arabe.

Il est aussi décrit par des règles grammaticales « de génération », comme suit:

Phrase' → Verbe' + Syntagme_Nominal'
Syntagme_Nominal' → Article' + Nom' .

A l'aide de ces règles on n'autorise pas l'inverse ou autre transformation (des prémisses) susceptibles de changer la bonne structuration, *comme par exemple* :

Syntaxme_Nominal'' → Nom' + Article' ou Nom + Article .

IV.1.2. Choix du formalisme :

Le style de programmation dit « déclaratif » ou « programmation logique » est conçu à l'origine pour le traitement automatique des langues, mais rapidement utilisé pour toutes sortes d'applications .

Ce type d'approche à été surtout représenté par la création du langage PROLOG (PROgrammation LOGique).

Ainsi, on décrit dans ce type d'approche les propriétés que doivent prendre les différents types d'objets , indépendamment des procédures utilisées pour construire ces objets ou vérifier ces propriétés .

Sur le plan pratique, on peut souligner l'intérêt du langage de programmation PROLOG pour la mise en oeuvre des analyses par les grammaires (exp. DCG), puisque les mécanismes d'unification et de retour arrière, inhérents à ce langage, permettent de réduire considérablement l'effort nécessaire à l'écriture de l'algorithme d'analyse ou de génération.

IV.2. Présentation de la grammaire DCG : [PEREIRA 80]

Les grammaires DCG, définies par F. Pereira et D. Warren en 1980, appartiennent à la famille des grammaires logiques, qui trouvent leur origine dans des recherches en logique (les démonstrateurs automatiques des théorèmes) et en programmation, ayant aboutit à la mise au point du langage PROLOG, et dont les premiers représentants ont été les grammaires de métamorphoses d'A. Colmerauer en 1975.

Dans ce type d'approche, les règles de réécriture de la grammaire sont conçues comme des axiomes (ou Clauses) et l'analyse syntaxique n'est qu'une mise en oeuvre des principes de déduction utilisés en logique.

La question de savoir si une séquence de mots donnée est une phrase du langage, revient à savoir si elle est le théorème déductible des axiomes de départ, et l'analyse syntaxique d'une phrase est le développement de la preuve de ce théorème.

Dans une grammaire logique, les symboles (P: pour une phrase, ou SN: pour Syntagme nominal,...) sont des termes constitués de prédicats et d'arguments, pouvant contenir des variables, par exemple: pour SN{variables: genre, nombre}.

On doit donc, lorsqu'on remplace une catégorie par une autre au cours d'une dérivation, recourir à des opérations supplémentaires, dont l'unification, et de remplacer toute variable apparaissant plusieurs fois dans une même règle par une même valeur.

Par exemple, à la règle de réécriture du syntagme nominal (SN), on peut associer des contraintes d'accord (écrites entre accolades selon la syntaxe DCG) portant sur les variables représentant le genre et le nombre des catégories correspondantes: Fig.IV.1.

```
SN([mot1,mot2], genre, nombre) → Determinant([mot1],x,y),
                                Nom([mot2],z,w)
                                {
                                    Accord_en_nombre(y,w),
                                    Accord_en_nombre(y,nombre),
                                    Accord_en_genre(x,z),
                                    Accord_en_genre(x,genre)
                                }.
```

Fig.IV.1: Analyse syntaxique d'un syntagme nominal en syntaxe DCG.

Ce type de grammaire est facile à implémenter, surtout en PROLOG, en raison de leur lien naturel avec les grammaires hors contextes, même si l'ajout de conditions peut rendre le formalisme aussi puissant .

IV.3. Principe général de la conversion:

Le modèle de conversion du système est adapté au modèle à transfert, composé de trois modules : (Fig.IV.3a) [ABBOU 95c]

- 1- un module d'analyse, qui génère une représentation structurée du texte source,
- 2- un module de transfert, qui transforme celle-ci en une représentation du texte cible,
- 3- un module de génération qui traduit cette représentation en un texte cible.

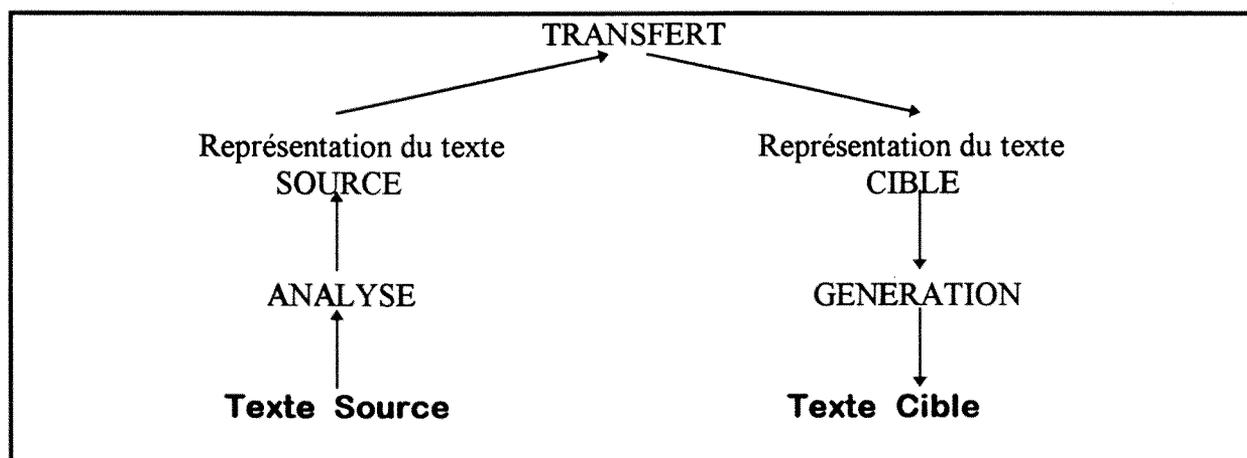


Fig.IV.3a: Le modèle de traduction à transfert.

La phase d'analyse transforme le texte de la langue source (cas du français) en un descripteur structural source, la phase de transfert transforme ce dernier en un descripteur structural cible et qui est transformé en texte de la langue cible (cas de l'arabe) par la phase de génération.

Notre intérêt qui s'est porté au modèle à transfert (dit système de la deuxième génération) pour les raisons suivantes :

- 1- isolation des informations linguistiques, qui sont principalement les grammaires et les lexiques des programmes de traitement (pour un futur traitement multilingue),
- 2- réutilisation des mêmes outils logiciels indépendamment des langues traitées,
- 3- les programmes de traitement peuvent alors fonctionner sur des types de grammaires diverses permettant ainsi des traductions multilingues.

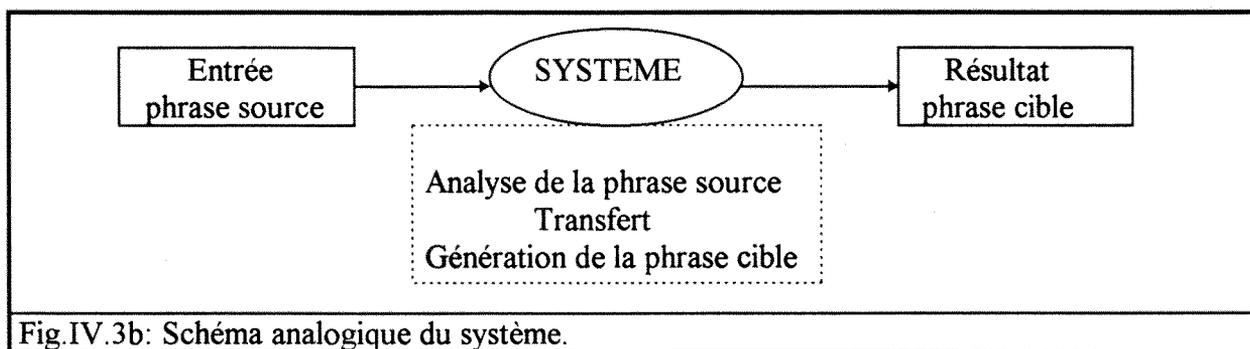


Fig.IV.3b: Schéma analogique du système.

Le système utilise dans son fonctionnement une base de lexiques et de grammaires, ainsi qu'un moteur de traitement (comme exemple, un compilateur PROLOG).

La syntaxe des structures de phrases en français autorisées sera contrôlée par l'ensemble des règles définies.

La génération des phrases arabes sera contrôlée par l'ensemble des règles de transfert (associer à chaque groupe syntaxique français le groupe syntaxique arabe équivalent).

Cette idée de *transfert structural* entre groupes syntaxiques a été adoptée dans le projet MALIN de traduction multilingue. Des relations bijectives sont donc définies entre les groupes syntaxiques sources et cibles.

En fait, cette correspondance [BOUALEM 93] associe les transitions de l'automate de la grammaire du français aux transitions de l'automate de la grammaire arabe. Le transfert syntaxique (français → arabe) fonctionne sur la base de la table correspondante suivante :

Structures en français	Structures en arabe
① Det+Nom composé	Nom composé
② Prep+ V	Prep + Det + Nom
③ Conj + GN	Conj + GN
④ Det + Nom + Adj	Det + Nom + Det + Adj
⑤ Nom composé + Adj	Nom composé + Det + Adj
⑥ Prep + Nom	Prep + Det + N
⑦ Det + Nom + V	V + Det + Nom
⑧ V + V-infinitif	V + Det + N
⑨ N + de + N	N + Det + N

Ainsi, dans le système ML-TASC de conversion de formalismes de langages techniques et scientifiques dans un environnement à syntaxe contrôlée et à contexte limité [BOUALEM 93], la phase de transfert convertit donc la structure intermédiaire du texte source (ensemble de structures syntaxiques) en une structure cible. Cette structure cible non ambiguë permet de générer le texte cible de manière convergente lors du processus de génération.

Remarque importante :

Il va de soi que le fonctionnement de ces tables de correspondances est extrêmement rigide et limité.

IV.4. Caractéristiques de notre modèle:

Le modèle à développer débute avec les principes suivants:

- ① Le système à développer dans sa première version est un prototype de traduction automatique assistée par ordinateur (TAO) du français vers l'arabe.
- ② Le système traite des phrases simples de base appartenant à un contexte général et/ou des phrases appartenant à un corpus étudié.
- ③ Les phrases sont exprimées selon une syntaxe pré-définie (syntaxe contrôlée). Les règles de grammaire décrivent un certain nombre de structures autorisées où tous les constituants (nom, article, verbe, ...) apparaissent dans les structures des phrases traitées.
- ④ Le contexte des phrases traitées ne comporte pas des phénomènes linguistiques complexes (ellipses, anaphores, structures comparatives, négation ...).
- ⑤ Les phrases simples peuvent être étendues à d'autres contextes (techniques et scientifiques) par amélioration des règles d'analyse et de transfert.
- ⑥ Le système se limite à la structure externe du langage. Cette structure permet de ne considérer qu'une seule représentation ou interprétation de la phrase analysée. Les seules formes d'ambiguïté sont d'ordre lexical et l'analyseur syntaxique permet de les éliminer.

- ⑦ Le lexique à utiliser dans le système correspond à un prototype d'une base lexicale qui pourra être conçu comme un dictionnaire électronique restreint français-arabe.
- ⑧ Une entrée de la base lexicale correspond à plusieurs champs séquentiels et comportant des informations morphologiques (le mot et ses variations), morpho-syntaxiques (les catégories: classe,genre,nombre...) et sémantiques (humain, objet, concret,...) pour le mot français et ses équivalents en arabe.
- ⑨ L'unité de traduction du système est une phrase qui est identifiée par une chaîne de caractères. Le programme de traitement identifie le caractère d'espacement comme séparateur entre les mots et le point comme fin de la phrase.

IV.5. Environnement de développement:

Le système ainsi que les modules utilitaires seront développés sur un ordinateur IBM-PC (micro-processeur famille 486 INTEL, avec 8 M de RAM).

L'environnement logiciel choisi pour le traitement des caractères arabes Arabic-MSDOS et/ou Windows arabisé et la programmation est réalisé en langage PROLOG II (Turbo PROLOG 2.0 de Borland) pour IBM-PC.

IV.5.1- Transcription des caractères arabes : [DICHY 95]

La transcription adoptée ici pour des raisons de « portabilité » informatique, est librement empruntée à A. ROMAN 1990 [ROMAN 90].

Les emphatiques et la constrictive vélaire *ha:?* sont en caractères gras; le soulignement distingue, lorsqu'il y a lieu, la constrictive de l'occlusive correspondante, ou un phonème d'un phonème 'voisin' (ainsi, s notera le son correspondant au 'ch' français: il ne s'agit pas d'une transcription phonémique au sens strict !).

Le présent travail portant sur le traitement automatique de l'écrit, les exemples sont en représentation graphématique (Dichy & Hassoun ,1990) [DICHY 89], c'est-à-dire en translittération en caractères latins des caractères arabes .

Cette dernière note au moyen de lettres minuscules les graphèmes diacritiques (correspondant à des signes diacritiques habituellement omis dans l'écriture courante), et pour des lettres majuscules les graphèmes de base (=lettres présentes dans le corps du mot), sauf pour les 'signes spéciaux': {^c?, &}. Pour les morphogrammes (D. Cohen, 1961/70) [COHEN 70].

Voir tableau (ci-après) de translittération en caractères latins des caractères arabes .

	caractère arabe	translittération en caractères latins	représentation graphémique
voyelles brèves:		a	a
		u	u
		i	i
voyelles longues:	?alif ا	a:	aA
	?alif-medda آ	?a:	Ä
	wa:w و	u:	uW
	ya:? ي	i:	iY
consonnes:	hamza ء	? ou ´	? ou ´
	ba:? ب	b	B
	ta:? ت	t	T
	ta:? ط	t	<u>T</u>
	ji:m ج	j	J
	ha:? ه	h	H
	xa:? خ	x	X
	da:l د	d	D
	da:l ذ	d	<u>D</u>
	ra:? ر	r	R
	za:y ز	z	Z
	si:n س	s	S
	si:n ش	s	<u>S</u>
	sa:d ص	s	S
	da:d ض	d	<u>D</u>
	ta:? ظ	t	T
	da:? ظ	d	<u>D</u>
	ayn ع	°	°
	gayn غ	g	G
	fa:? ف	f	F
	qa:f ق	q	Q
	ka:f ك	k	K
	la:m ل	l	L
	mi:m م	m	M
	nu:n ن	n	N
	ha:? ه	h	H
	wa:w و	w	W
	ya:? ي	y	Y
morphogrammes:	?ali:f maqsura ي	ÿ	ÿ
	ta:? marbu:ta ة	&	&

Chapitre 5 :

Difficultés du traitement de la langue arabe

V. Difficultés du traitement de la langue arabe:

Introduction:

Afin d'illustrer les difficultés et les problèmes du traitement automatique du langage naturel (TALN), nous décrivons quelques propriétés de la langue arabe, ([DICHY 90]), comme étape préliminaire avant la description de la conception et l'implémentation de notre prototype de TAO aux chapitres VI et VII.

Le but de ce chapitre n'est pas de traiter tous les aspects de cette langue, mais de permettre au lecteur, à travers des comparaisons de la langue source de notre prototype de TAO (le français), de voir les problèmes qui se posent dans le domaine du traitement informatique de l'arabe .

V.1. Généralités:

Les pays arabes utilisent souvent des langues étrangères latines, comme le français ou l'anglais, dans les domaines du traitement automatique de l'information. Des difficultés d'apprentissage s'imposent aux personnes qui se servent d'un système qui ne fonctionne pas dans sa langue natale.

Chaque pays arabe possède sa langue parlée, alors que l'enseignement dans les institutions d'éducation est le même arabe littéraire écrit.

D'ailleurs, le même arabe écrit, qui est utilisé dans l'information administrative et médiatique, les journaux et les livres.

Un grand intérêt se dévoile, comme l'arabe écrit est compris par tous, le problème de l'informatisation de cette langue ne diffère donc pas d'un pays à un autre.

V.2. Quelques propriétés graphématiques de l'arabe:

La langue arabe fait partie de la classe des langues sémitiques, c'est donc une langue alphabétique qui s'écrit de la droite vers la gauche.

L'alphabet arabe comprend 28 consonnes, correspondent à des graphèmes (y compris le hamza), auxquelles s'ajoutent des « segments d'allongement: *huruf al-madd* (*alif* = allongement du |a|), (*waw* = allongement du |u|) et (*ya'* = allongement du |i|) ».

Le signe "hamza" a longtemps été considéré dans les anciens documents comme signe complémentaire. Dans les nouvelles écritures, il vient s'ajouter à l'alphabet comme lettre à part entière.

En général, cette lettre est suscrite ou souscrite à des voyelles longues : le "aleph", le " ya" et le "waw", et peut aussi apparaître seule en fin de mot.

V.2.1. Les voyelles dans la langue arabe:

a) Les voyelles brèves:

Ce sont des signes pouvant se placer au-dessus ou bien en-dessous des consonnes.
On distingue :

- ① la "fetha" est une voyelle ouverte notée "--̄--", elle se place au-dessus de la consonne et correspond au son "a".
- ② la "kasra" est une voyelle fermée antérieure notée "--_--", elle se place en-dessous de la consonne et correspond au son "i".
- ③ la "dhamma" est une voyelle fermée postérieure notée "--'--", elle se place au dessus de la consonne et correspond au son "u".

b) Les voyelles longues:

On distingue les voyelles longues suivantes :

- ① "aleph" et "aleph-maqsurā",
- ② "ya",
- ③ "waw".

V.2.2. Les signes diacritiques:

On distingue plusieurs **signes diacritiques secondaires** en arabe:

- ① Le "sukun" noté "--°--" se place au-dessus de la consonne notant l'absence de voyelle, il marque une consonne post-vocalique suivie d'une autre consonne ou d'une frontière de mot.
- ② La "chedda" notée "--_'" note l'allongement de la consonne au-dessus de laquelle elle se positionne. Ce signe peut accompagner une voyelle brève pour donner une voyellation à deux niveaux.
- ③ La "medda" suscrite au aleph, elle est notée " |̄ ", marque l'allongement phonétique d'une voyelle |a| précédée d'une *hamza* " | ".
- ④ La "wasla" est un signe syllabique toujours suscrit à un aleph pour lui ôter sa valeur phonétique.
- ⑤ Le "tanwiin" est représenté par la séquence λn , avec $\lambda = a | i | u$, et est transcrit par le redoublement de la voyelle. Le redoublement des fetha, kasra et dhamma est respectivement noté: "--̄_--", "--_'" et "--'_" .

Par opposition aux **signes diacritiques primaires**, qui sont réalisés en général par des points placés au-dessus ou au-dessous de la forme de base de la lettre .

Exemple: ba:ʔ = b , ta:ʔ = t , ta:ʔ = T , etc.

Remarques importantes:

- ① La plupart des documents arabes qui sont diffusés au grand public, comme les journaux et les livres, sont écrits en arabe non-vocalisé où les mots sont représentés sous forme de successions de consonnes (squelettes).

Cet aspect pose des problèmes de compréhension des textes écrits, c'est comme essayer de comprendre que la phrase:

" L prcsss d tradctn cmprt 3 phss lgqs.", doit vouloir dire : "Le processus de traduction comporte 3 phases logiques."

Une telle comparaison est cependant très superficielle par rapport à la langue arabe car, compte tenu des différences structurelles de ces deux langues (français et arabe), l'arabe non-vocalisé reste tout de même plus compréhensible que le français sans voyelles.

- ② Le réel problème de la compréhension des mots non-vocalisés en arabe est le fait qu'un même mot peut avoir plusieurs significations, parfois n'ayant aucun rapport les unes avec les autres, selon la manière dont ce mot peut être vocalisé.

Exemple:

On illustre cet aspect par le mot "SLM", qui peut avoir les significations de :

- 2.a) *SuLLaMun* ou « échelle »,
- 2.b) *SiLMun* ou « paix »,
- 2.c) *SaLiMa* ou « est guéri »,
- 2.d) *SaLLaMa* ou « a transmis » , etc.

Ce genre d'ambiguïté peut engendrer des difficultés dans la lecture de textes non-voyellés. Cependant, le contexte suffit en général à résoudre ces difficultés.

- ③ On retrouve dans certains manuels, livres, dictionnaires et textes officiels que seulement la voyelle la "chedda" n'est pas omise.

Ainsi, le nombre d'ambiguïtés diminue dans le texte et la compréhension devient plus aisée.

Exemple:

On illustre par le même exemple que précédemment, ou le mot "SLM" se subdivise en deux catégories de significations et non quatre:

- 3.a) *SuLLaMun* et *SaLLaMa* ,
- 3.b) *SiLMun* et *SaLiMa* , etc.

V.3. Propriétés morphologiques de l'arabe:

Une projection simplifiée d'un mot en arabe peut comprendre la racine, les morphèmes, les antéfixes et les postfixes.

V.3.1. La racine:

La racine se présente sous la forme d'une succession de deux, trois ou quatre consonnes et qui représente une notion .

Exemple:

Le mot *KTB* , représente la notion "écrire" .

Ainsi, la racine associée à des voyelles (des signes diacritiques ou des voyelles longues) constitue le **formant-noyau du mot** .

Exemple : (de formant-noyau du mot)

Le mot *?aKTuBu* « j'écris » a pour formant-noyau du mot: *KTuBu* , le préfixe "*?a*" désigne la première personne du singulier et cache le pronom *?aNaa* « je ».

La racine ainsi que le formant-noyau du mot ne peuvent être rencontrés seuls dans un texte. Ils sont toujours accompagnés de **morphèmes** et/ou de **proclitiques** et/ou d'**enclitiques**.

V.3.2. Les morphèmes :

Les morphèmes sont composés de voyelles et de certaines consonnes accompagnant (en préfixe ou en suffixe) le radical pour former le mot minimal .

Un mot minimal est un mot possible qui peut figurer dans un texte .

Exemple:

Le mot *TaKTuBuWNa* « vous écrivez » , ayant pour:

- formant-noyau du mot: *KTuBu* ,
- morphème préfixe: *Ta* qui désigne la deuxième personne du singulier, duel ou pluriel ,
- morphème suffixe: *uWNa* qui désigne le pluriel.

V.3.3. Les proclitiques:

Les proclitiques sont des consonnes et des voyelles en nombre défini et en ordre fixe, et qui servent à déterminer certains éléments comme l'article, l'interrogatif, le coordinatif, l'affixe temporel, etc.

Exemple:

- l'article défini : ?aL ,
 - l'interrogatif : ?a , ...
 - le coordinatif : Wa , ?aW,...
 - l'affixe temporel : Ya , Ta ,...
- etc.

V.3.4. Les enclitiques:

Les enclitiques sont des consonnes et des voyelle en nombre défini et qui désignent le suffixe personnel.

Exemple:

- la première personne du singulier: ?a ,
 - la deuxième personne du singulier: Ta ,
- etc.

Remarque :

La racine est formée uniquement de consonnes. Les voyelles n'ont pas le rôle de définir la notion, mais ils servent pour la dérivation : distinguer les différents mots constituables à partir de la racine .

Pour cette raison, l'arabe est une langue à forte dérivation.

Dans les dictionnaires arabes, les mots sont classés par rapport à l'ordre alphabétique des racines et non pas celui des dérivations, d'où la difficulté de recherche de mots dérivés dans ces dictionnaires.

Des nouvelles versions de dictionnaires qui sont commercialisés sur le marché adoptent l'ordre alphabétique pour les racines et les dérivations.

V.4. Propriétés syntaxiques de l'arabe:

Dans cette paragraphe, on citera quelques propriétés morphologico-syntaxiques de la grammaire arabe:

- ① Le verbe de la phrase arabe se construit d'une manière particulière et que dans sa morphologie on retrouve:

- 1.a°) le sujet,
- 1.b°) le noyau verbal, et
- 1.c°) une désinence verbale .

Exemple:

En français : "le professeur lit le texte", est une construction SVO (Sujet-Verbe-Objet), où le professeur (Sujet), lit (Verbe) et le texte (Objet).

En arabe : *YaQRa?u ?aL-Mu^cLLiMu ?aL-NaSSa* est une construction dans laquelle:

- *YaQRa?u* (le Verbe) se décompose en: *Ya + QRa? + u*, avec
Ya : pronom personnel 'implicite' 3PMS qui remplace le pronom « il »,
QRa? : noyau verbal, et
u : désinence verbale.

D'après l'analyse syntaxique traditionnelle, il s'agit d'une **phrase verbale**, qui comporte un sujet et un verbe.

- *?aL-Mu^cLLiMu ?aL-NaSSa* : est le reste de la phrase qui contient un « sujet explicite » *?aL-Mu^cLLiMu* qui instancie le pronom "il".

- ② Le sujet peut parfois se placer avant le verbe dans le cas, par exemple, où l'on veut désigner d'une manière plus précise l'acteur (le sujet de l'action). C'est ce que la syntaxe arabe appelle **une phrase nominale**.

La phrase arabe nominale comprend: un Sujet ou *MuBTaDa?* et un Attribut ou *XaBaR* qui peut être soit un nom, soit un SN, un SV, un SP ou une phrase.

Exemple 1:

En arabe : *?aL-Mu^c aLLiMu YaQRa?u ?aL-NaSSa* « Le professeur lit le texte », dans cette phrase :

- *?aL-Mu^cLLiMu* (Sujet): est un Syntagme nominal qui comprend un sujet 'explicite' composé de (Article défini + Nom commun),
- *YaQRa?u ?aL-NaSSa* : une analyse identique à l'exemple précédent, que c'est une construction SVO, le verbe comporte un pronom personnel "il" instancié par le sujet 'explicite'.

Exemple 2:

Une autre construction de phrases ne faisant apparaître aucun verbe, et dans ce cas on parle aussi de **phrase nominale**, en voici un :

En arabe: *?aL-NaSSu MuFiYDun* « le texte est intéressant », cette phrase est constituée d'un sujet 'explicite': *?aL-NaSSu* « le texte » et d'un attribut: *MuFiYDun* « intéressant ».

- ③ Le complément d'objet, lorsqu'il se présente sous la forme d'un pronom personnel, est souvent positionné après le verbe (contrairement à la langue française où il se place avant le verbe).

Exemple:

En arabe: *YaQRa?uHu* = *Ya* + *Qra?u* + *Hu* « il le lit », (en désignant le texte), le complément d'objet désigné par *Hu* ou le pronom « le » qui est un suffixe personnel et est placé après le noyau verbal *Qra?u* « lit » .

- ④ La construction de phrases arabes utilise souvent des 'formes implicites' pour désigner le sujet et ceci à l'aide des 'pronoms implicites' : *DaMiYR MuSTaTiR* (selon la grammaire traditionnelle).

Des pronoms sont utilisés aussi pour désigner l'Objet, mais dans ce cas ils sont 'explicites'.

Exemples:

En arabe: *?aL-Mu^CLLiMu YaQRa?u ?aL-NaSSa* « le professeur lit le texte », le sujet est *?aL-Mu^CLLiMu* « le professeur ».

Dans le cas: *HuWa YaQRa?u ?aL-NaSSa* « il lit le texte », le sujet est *HuWa* « il », comme pronom personnel 'explicite'.

Mais, dans le cas: *YaQRa?u ?aL-NaSSa* « il lit le texte », le sujet est un pronom 'implicite' 3PMS « il », qui désigne le professeur.

- ⑤ Dans la langue arabe, il n'existe qu'un article défini: *?aL* ou le / la / les , qui vient se positionner dans la partie proclitique des noms. Il n'existe pas d'article indéfini, il est exprimé par l'absence de l'article défini = \emptyset , mais son apparition est 'explicite' sur le nom par le signe diacritique le *TaNWiYN* : λn .

Exemple:

Pour le défini : *?aL-^CiLmu* « la science »,

pour l'indéfini: *^CiLmun* « Science / une science », le $\lambda n = un$.

En pratique, dans l'activité de traduction, parfois on trouve des ajouts comme par exemple :

en français: "un livre de Paul", traduite

en arabe : *KiTā?Bun MiN KuTuBi Bu:L* , qui se traduit littéralement encore

en français par : " un livre parmi d'autres (livres) de Paul ".

⑥ L'arabe comporte deux genres: le féminin et le masculin. Un troisième genre s'ajoute dit neutre pour désigner les "choses" qui ne possèdent pas de genre particulier. Il est possible de ramener tout mot en arabe de genre "chose neutre" à l'un des deux genres classiques.

Très souvent, le féminin se distingue du masculin par l'intermédiaire du signe & « T-rond », *ta:ʔ marbuta* dans la partie des morphèmes suffixes du mot féminin.

Pendant, il existe des exceptions pour cette règle et il est à noter que le seul article défini *ʔaL* ne peut servir pour distinguer les deux genres (et ceci par opposition au rôle de l'article dans la détermination du genre dans la langue française).

Exemple:

indéfinis: *MaLiK +un* , *MaLiKa& +un* ou « Roi , Reine », etc.

définis : *ʔaL-MaLiK +u* , *ʔaL-MaLiKa& +u* « Le Roi , La Reine », etc.

exceptions : *RaJuL* , *ÄMRaʔa&* «Homme , Femme », etc.

⑦ En plus du singulier, la langue arabe comporte les nombres duels.

Le duel présente une désinence par rapport au singulier et qui est symbolisé par les morphèmes suffixes: *aANi* ou *aYNi* pour le masculin; et *TaANi* ou *TaiYNi* pour le féminin, selon le comportement du mot à dualiser sujet ou complément ...

Exemple:

➤ *ʔaL-KiTaaBu HuNaA* « Le livre est ici », le mot *KiTaaBu* « livre » est au singulier.

➤ *ʔaL-KiTaaBaANi HouNaA* « les deux livres sont ici », le mot *KiTaaBaANi* est duel et se comporte comme sujet de la phrase nominale.

➤ *ʔaNaA QaRaʔTu ʔaL-KiTaaBaYNi* « J'ai lu les deux livres », le mot *KiTaaBaYNi* «les deux livres » a le rôle d'un COD dans la phrase verbale et c'est pourquoi le morphème suffixe *aYNi* est employé.

⑧ Le pluriel en arabe présente aussi, par rapport au singulier, une désinence qui est symbolisée par un certain nombre de morphèmes suffixes distincts selon le genre et le comportement du mot à pluraliser.

Le pluriel obtenu par l'adjonction de l'une de ces désinences est dit 'pluriel sain', on trouve donc un 'féminin pluriel sain' et un 'masculin pluriel sain'.

Mais, il existe un autre pluriel dit 'pluriel brisé' et qui est issu d'une rupture des consonnes de la racine. Ce dernier, ne correspond à aucune règle et nécessiterait sans doute un traitement spécial.

Exemple:

- masculin 'pluriel sain': (*FaLLaAHun*, mas, sing) et (*FaLLaAHuWNa*, mas, plur)
« agriculteur et agriculteurs » .
- féminin 'pluriel sain': (*TuFFaAHa&un*, fem, sing) et (*TuFFaAHaATun*, fem, plur)
« pomme et pommes ».
- pluriel 'brisé': (*TiFLun*, mas, sing) et (*?aTFaALun*, mas, plur) « enfant et enfants » .

⑨ La conjugaison des verbes en arabe présente plusieurs valeurs temporelles :

A) L'accompli: il énonce une action achevée.

A.a) Employé seul, il peut exprimer un passé simple ou composé en français .

Exemple: *?aNaa QaRa?Tu* « J'ai lu ».

A.b) Employé dans des verbes exprimant un souhait, une volonté, une décision..., il exprime un résultat présent d'une action ayant eu lieu dans le passé.

Exemple: *?aNaa ?aRaDTu ...* « J'ai voulu ... ».

A.c) Employé dans des faits constatés ou acquis définitivement, il exprime un présent continu .

Exemple: *?iNna ?aL-LaHa KaANa ^caLiYmun* « Allah est omniscient ».

A.d) Employé dans des phrases contenant un engagement solennel négatif (subjonctif présent en français), il exprime un présent (ou même un futur) sous la forme d'un passé déjà accompli.

Exemple: *LaA ?aKaLTu HaTTaY* ... « je ne mangerai que si... ».

A.e) Employé dans le verbe *KaANa* « être » , précédant un autre verbe accompli, il donne à celui-ci un sens du passé dans le passé (passé antérieur en français).

Exemple: *LaW KaANa QaD DaHaBa...* « s'il était parti... ».

etc.

B) L'inaccompli: il énonce une action en cours d'être effectuée ou qui se répète en un temps vague (en français cela représente le présent, l'imparfait ou le futur).

B.a) L'inaccompli indicatif exprime, en dehors de toute indication de temps fournie, le présent réel.

Exemple: *?aNaa ?aKTuBu ?aL-ANa* « j'écris en ce moment ».

B.b) L'inaccompli subjonctif indique une intention annoncée et qui devrait avoir lieu dans le futur .

Exemple: *?aMaRaHu ?aN YaDHaBa* « il lui ordonna de partir ».

B.c) L'inaccompli apocopé exprime une action incertaine ou conditionnelle. Il énonce une injonction ou une interdiction.

etc.

C) L'impératif: il ne s'utilise qu'avec les deuxièmes personnes au singulier, au duel ou au pluriel (masculin ou féminin). Il exprime en général une injonction ou un ordre.

Exemple: *?uKTuB !* « écris ! » .

D) Le passif: il correspond à une action accomplie du type *^CuMiLa* « il a été fait », ou une action inaccomplie du type *Yu^CMaLu* « il se fait ».

Conclusion:

Ainsi, d'après ces cas illustrés, la langue arabe présente un nombre important de difficultés liées à ses caractéristiques morphologiques et syntaxiques.

Ces caractéristiques sont différentes des langues indo-européennes (en particulier le français) communément utilisées dans le traitement automatique de l'information.

La construction des phrases en arabe est nominale ou verbale. Donc, il faudra distinguer les différentes constructions de phrases avant de les traiter.

Aussi, le problème de présence obligatoire du verbe arabe ou de son absence dans la génération de la phrase.

Les analyseurs lexico-syntaxiques doivent aussi distinguer les formes de conjugaison, les valeurs temporelles, les différences de genre et de nombre ainsi que les nombreuses formes de dérivation à partir d'une racine...

Dans la langue arabe, un seul mot peut parfois constituer une phrase complète (plusieurs éléments syntaxiques). On illustre ce cas par le mot: *?aSaTaKtuBuHu ?* « Est-ce-que tu vas l'écrire ? ». Ce mot constitue une phrase interrogative complète contenant un sujet, un verbe, un complément, une valeur temporelle, un genre et un nombre.

Un autre phénomène dans la construction de phrases arabes qui concerne la notion de distribution avec une 'case vide' ou 'valeur par défaut': certains éléments syntaxiques, tout en continuant à jouer leur rôle, n'apparaissent guère dans la phrase. Ils sont repérables par le phénomène de déduction:

Exemple:

QaRa?a ?aL-Mu^caLLiMu ?aL-NaSSa « le maître a lu le texte », on trouve dans cette phrase que le sujet est 'explicite'.

Par contre, *QaRa?a ?aL-NaSSa* « il a lu le texte », le sujet est 'implicite' comme pronom personnel « il » et est déduit dans le contexte par *?aL-Mu^caLLiMu* .

Chapitre 6 :

Description de notre prototype
de TAO

VI. Description de notre prototype de TAO :

VI.1. Organisation du lexique:

Introduction:

Il est clair que dans n'importe quel système de traduction, la puissance de l'analyse des textes sources et de la génération en textes cibles dépend, en grande partie, de l'organisation des dictionnaires électroniques.

Dans notre système expérimental (ou prototype), nous avons adopté pour le besoin de la TAO des phrases du français vers l'arabe, une structure modulaire dans l'organisation du lexique.

Cette structure se manifeste par la conception d'un dictionnaire électronique bilingue français-arabe. L'intérêt de ce genre de conception a deux portées:

- 1- Analyse des phrases françaises
- 2- Synthèse des phrases arabes (ou génération)

Deux questions se posent à se propos:

- Quelles sont les informations nécessaires pour la phase d'analyse ?
- Quelles sont les informations nécessaires pour la phase de synthèse ?

VI.1.2. Le dictionnaire électronique réduit propre à cette application:

Notre vision personnelle de la solution du problème est que la conception du lexique devrait être modulaire et symétrique.

On sous-entend par:

- **Modulaire:** le lexique comporte trois modules (ou dictionnaires) indépendants.
Le premier module est un dictionnaire bilingue des mots simples (article, nom, adjectif, verbe,...). Le second est un dictionnaire bilingue des mots composés et le troisième dictionnaire pour les expressions figées (ou idiomatiques).

Dans le cas de conception de notre prototype de TAO on s'intéressera qu'au premier type de dictionnaire.

- **Symétrie:** chaque ligne du dictionnaire comporte deux parties (ou segments) .
La première partie est réservée uniquement pour la reconnaissance et le traitement du français et la seconde pour la génération et le traitement de l'arabe.

A) Description des segments:

- **Premier segment:** les informations du mot français Mot_fr(i).

Le premier segment présente une entrée unique (ou clé d'entrée) qui est le mot simple en français puis des champs pour les informations morphologiques, syntaxiques et sémantiques concernant ce mot.

- **Deuxième segment:** les informations du mot arabe Mot_ar(i).

Le deuxième segment présente une occurrence des mots arabe équivalents au mot Mot_fr(i).

Cette occurrence est le mot arabe, Mot_ar(i), le plus probable de la liste des équivalents au Mot_fr(i), puis des informations morphologiques, syntaxiques, sémantiques concernant Mot_ar(i).

On rajoutera à ce segment deux autres champs: le premier est la liste des mots équivalents au mot Mot_fr(i), le second des informations supplémentaires sur Mot_ar(i) pour le besoin des traitements (comme les flexions...).

Nous schématisons cette conception comme suit:

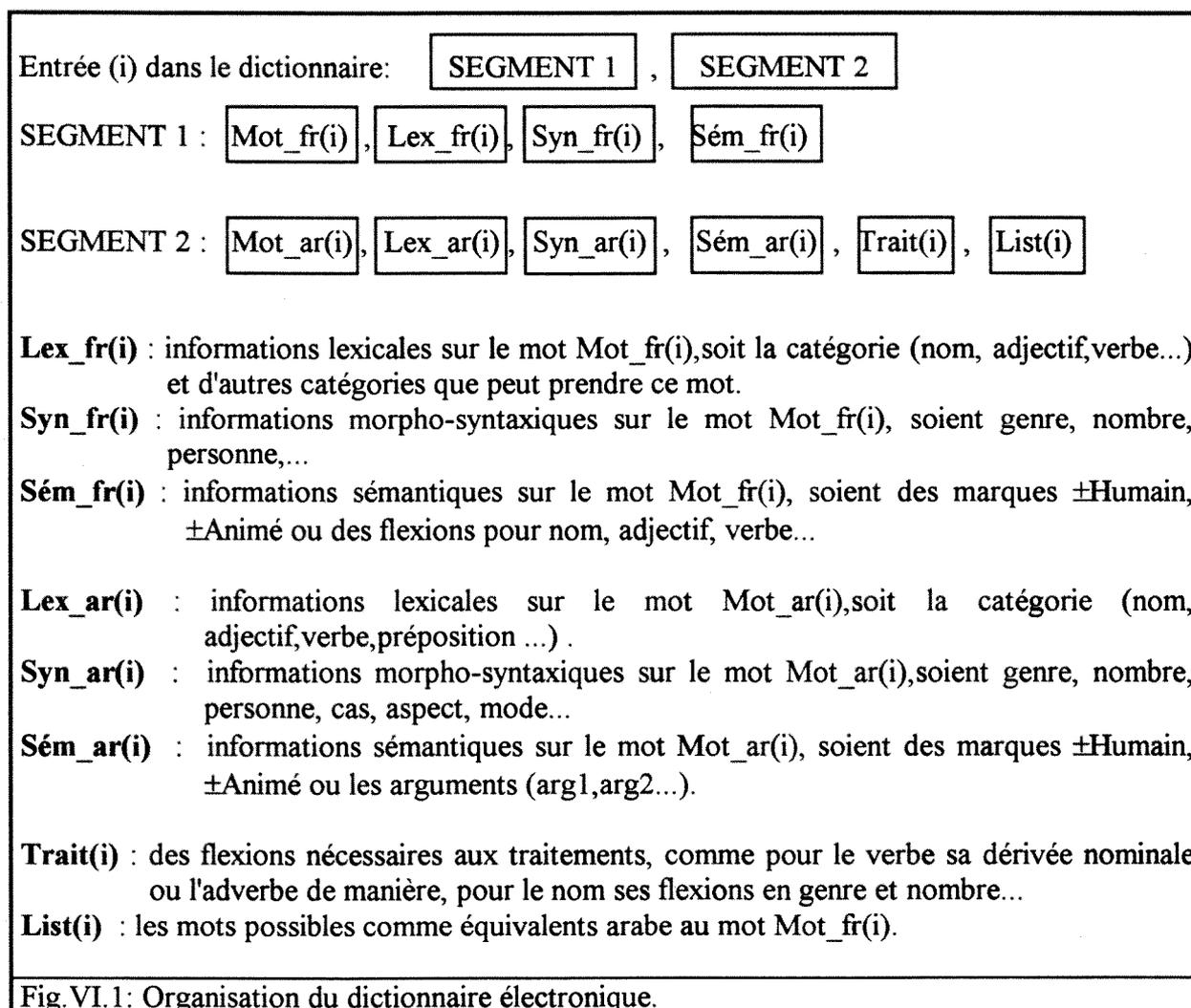


Fig. VI.1: Organisation du dictionnaire électronique.

B) Intérêt de la segmentation: [BOURBEAU 85]

L'intérêt de la segmentation relève du principe de symétrie, et grâce à cette composition on peut obtenir:

- 1- un nouveau dictionnaire électronique bilingue arabe-français, en inversant les deux segments (SEGMENT 2 , SEGMENT 1).
- 2- deux dictionnaires électroniques monolingues français et arabe, en séparant chacun des deux segments. Chaque nouveau dictionnaire obtenu (SEGMENT 1 et SEGMENT 2) est complet pour l'analyse de sa langue (respectivement français et arabe).
- 3- la possibilité de coupler l'un des deux segments avec un autre segment différent (autre que le français et l'arabe) pour obtenir un nouveau dictionnaire bilingue pour un traitement automatique multilingue:
soit (SEGMENT ½ , SEGMENT x) ou (SEGMENT x , SEGMENT ½) .

VI.2. Grammaire d'analyse:

Introduction:

La puissance d'un analyseur syntaxique dans un système de TA ou TAO dépend de l'organisation des dictionnaires électroniques, mais aussi du bon choix du type de grammaire d'analyse syntaxique.

Ainsi, pour la phase d'analyse des phrases françaises dans notre prototype de TAO, on a besoin d'une grammaire.

VI.2.1. Rôle du formalisme grammatical :

[BOLC 87],[CHURCH 93], [GAZDAR 89].

Les formalismes grammaticaux sont des langages destinés à la description des langages elles-mêmes: Soit, à décrire l'ensemble des phrases possibles d'une langue, les propriétés structurales de ses phrases (leurs syntaxes) et leur signification (la sémantique).

Ainsi, chaque grammaire particulière est écrite dans un formalisme syntaxique ou le **métalangage**, qui encode une analyse d'une langue objet.

Le métalangage a pour buts suivants:

- 1- fournir un outil formel de description des langues (naturelles),
- 2- donner des caractérisations des langues (naturelles) qui soient interprétables par l'ordinateur,
- 3- délimiter la classe des langues naturelles possibles.

Le choix du métalangage dans lequel les analyses sont encodées est déterminé en fonction des trois paramètres suivants: [SHIEBER 86]

➤ Félicité linguistique:

La description des phénomènes linguistiques est exprimée directement ou indirectement comme les linguistes y expriment.

➤ Expressivité:

Le type de classe d'analyse exprimée.

➤ Efficacité:

Pour le traitement automatique, les outils de traitement informatique appropriés à l'interprétation des grammaires exprimées dans le formalisme considéré et les limites propres du point de vue informatique .

VI.2.2. Grammaire du prototype de TAO :

Pour le besoin de notre prototype de TAO, on propose la reconnaissance des structures de phrases de notre corpus (Voir Fig.VI.3.2a), comme suit :

A) Les règles de reconnaissance:

La présentation de la grammaire pour l'analyse des phrases de notre corpus est inspirée des travaux de recherches du groupe SYDO (SYstèmes DOcumentaires, créé en 1975 et rassemble le CRLS-Lyon 2, le LID-Lyon 1 et l'ENSSIB), qui a créé un modèle linguistique adapté au traitement automatique de l'information (identification des syntagmes nominaux) [METZGER 88],[DE BRITO 91],[LAROUK 94].

Les aspects linguistiques qui sont attribués au modèle proviennent de la contribution de plusieurs professeurs universitaires: A. BERRONDONNER, M. Le GUERN, R. BOUCHÉ, J. ROUAULT, J-P METZGER et tant d'autres dans les équipes de recherches.

Nous retenons dans notre travail sur le corpus, certaines règles de réécriture du modèle SYDO et l'introduction d'autres nouvelles règles nécessaires pour la reconnaissance des phrases.

Le vocabulaire terminal (V_T) et le vocabulaire non terminal (V_N) sont :

$$V_T = \{F, W, D, P, V, C, T\} ,$$

$$V_N = \{PH, PH_c, SN, N', SV, V', N, A', A, D', SP, SP^n, Sp^n_c, SV, EP, EP_c \}.$$

La liste des catégories selon la description de [DE BRITO 91]:

CATEGORIES	SYMBLES
Nominales:	F: noms-adjectifs {F-NOM, F-ADJ, F-NAN} D: prédéterminants {D-DEF, D-NUM, D-IND} P: prépositions
Intermédiaires:	C: conjonctions W: adverbess

	{W-AAJ, W-QUA, W-PRO, W-TAM} T: signes de ponctuation
Verbales ou Propositionnelles:	V: verbes Y: pronoms préverbaux Q: subordinants

N° Règle	Règle de réécriture	Illustration
1	PH → SN + SV	PH= « L'attention de notre clientèle » + « est attirée sur les points suivants: »
2	PH _c → PH + C + PH	PH _c = « Il convient d'éviter... » + « et » + «de veiller ... existe entre les mots... »
3	SN → D' + N'	SN= « les » + « points suivants »
4	SN → F-PRO	SN = « il »
5	SN → D' + F-PRP	SN = « la Banque Nationale... »
6	N' → N + SP ⁿ	N' = « responsabilité » + « en cas de perte, de soustraction... »
7	N' → N	N' = « usage»
8	N' → N + A'	N' = « rectifications » + «éventuelles »
9	SP ⁿ → SP	SP ⁿ = « dans le présent chéquier» ; n=1
10	SP ⁿ → SP + (T) + SP ⁿ	SP ⁿ = « de toute responsabilité en cas de perte, de soustraction ...» ; n>1
11	SP → P + SN	SP = « dans » + «le présent travail »
12	SP → P-en + ('ne') V-pp ('pas')	SP = « en » + « (n') omettant (pas) » selon [Le GOFFIC 93], appelé groupe prépositionnel (GP)-participial.
13	SP → P-de + V-inf + (A')	SP=« de »+«reproduire »+(« integralement ») selon [Le GOFFIC 93], GP-infinitival.
14	SP → P-pour + V-inf + (SN)	SP =« pour » + « refuser »+ (« le paiement ») selon [Le GOFFIC 93], GP-infinitival.
15	SP → P-à + W + (SP)	Groupe adverbial = [expansion gauche] + adverbe + [expansion droite] , selon [Le GOFFIC 93], GP-adverbial. [expansion gauche] =trés, beaucoup plus,... [expansion droite] = expansion P ou SP .
16	D' → D	D' = « la »
17	D' → F-ADJ (possessif)	D' = « notre »
18	D' → W-QUA + P-de	D' = « beaucoup » + « de »
19	D' → « ce que » + D	D' = « ce que » + « un »

20	$A' \rightarrow F\text{-ADJ}$	$A' = \text{« suivants »}$
21	$SV \rightarrow V' + SP^n + (SN)$	$SV = \text{« convient »} + \text{« d'éviter de laisser »} + \text{« un blanc devant les sommes... »}$
22	$SV \rightarrow V' + Sp_c^n + (SN)$	$SV = \text{« est déchargée »} + \text{« de toute responsabilité en cas de perte, ... »} + \text{« des formules contenues dans le présent chéquier »}$
23	$V' \rightarrow V$	$V' = \text{« convient »}$
24	$V' \rightarrow AUX + V$	$V' = \text{« est »} + \text{« attirée »}$
25	$V' \rightarrow V + AUX + V$	$V' = \text{« doivent »} + \text{« être »} + \text{« approuvées »}$
26	$N \rightarrow N + EP$	$N = \text{« sommes »} + \text{« en chiffres »}$
27	$N \rightarrow N + EP_c$	$N = \text{« sommes »} + \text{« en chiffres ou en lettres »}$
28	$EP \rightarrow P + N'$	$EP = \text{« en »} + \text{« lettres »}$
29	$EP_c \rightarrow EP + C + EP$	$EP_c = \text{« en chiffres »} + \text{« ou »} + \text{« en lettres »}$
30	$Sp_c^n \rightarrow Sp^n + C + Sp^n$	$Sp_c^n = \text{« d'éviter de laisser ... »} + \text{« et »} + \text{« de veiller à ce qu'un minimum... »}$

B) Formulation de la grammaire en PROLOG:

[GAZDAR 89],[GAL 89],[KABBAJ 91]

Les unités non-terminales de cette grammaire correspondent à des buts Prolog.
Si la phrase est représentée par une liste de mots L, les buts Prolog auront essentiellement deux arguments L1 et L2 :

L qui représente la phrase en entrée pour l'analyse et L2 représente ce qui reste de la phrase après reconnaissance d'une partie par le but en question (Syntagme_Nominal).

Ainsi, la partie consommée est la différence des deux : $L1=L-L2$.

On formule la règle d'analyse d'une phrase P en Prolog: (voir Fig.VI.2.2B)

où :

$$\begin{aligned}
 P &\rightarrow SN_{(\text{genre,nombre})} + SV_{(\text{genre,nombre})} \\
 SN_{(\text{GEN,NBR})} &\rightarrow Det_{(\text{GEN,NBR})} + N_{(\text{GEN,NBR})} \\
 SV_{(\text{G,N})} &\rightarrow V_{(\text{G,N})} + Adj_{(\text{G,N})} .
 \end{aligned}$$

```
Analyser_Phase (L) :- Analyser_Syntagme_nominal (L,L2,GEN,NBR),  
                      Analyser_Syntagme_Verbal (L2,GEN,NBR) .
```

```
Analyser_Syntagme_nominal (L1,L2,G_,N_) :-  
    extraire_mot(L1,Rest,M1),  
    est_determinant(M1,G_,N_),  
    extraire_mot(Rest,L2,M2),  
    est_nom(M2,G_,N_).
```

```
Analyser_Syntagme_Verbal (L,G_,N_) :-  
    extraire_mot(L,Rest,M1),  
    est_verbe(M1,G_,N_),  
    extraire_mot(Rest,[],M2),  
    est_adjectif(M2,G_,N_).
```

```
extraire_mot ([Tête|Q], Q, Tête).
```

Fig. VI.2.2B: Règle d'analyse d'une phrase en Prolog.

La règle « Analyser_Phase » permet de faire l'analyse syntaxique d'une phrase sans donner l'arborescence de la phrase (ie. arbre syntagmatique). D'autres règles ou arguments dans celle-ci pourront exprimer les traces de l'analyse :

P(SN(est_determinant(M1),est_nom(M2)),SV(est_verbe(M3),est_adjectif(M4))).

VI.2.3. L'analyseur du prototype de TAO:

Un programme est qualifié de reconnaiseur (recognizer) lorsque sa tâche consiste à vérifier si oui ou non la phrase est conforme à la grammaire. La sortie d'un reconnaiseur est donc oui/non selon que la phrase est acceptée ou rejetée.

Un analyseur est un reconnaiseur qui produit en outre la structure syntaxique de la phrase reconnue. La production d'un analyseur (la sortie) n'est pas strictement limitée à l'*arbre syntaxique* de la phrase. On peut avoir comme sortie la *représentation sémantique* de cette phrase, ou une *forme transformée* ([KITTREDGE 87],[MILLER 90])de celle-ci proche de notre but qui est la traduction.

Aussi, l'analyseur peut produire plusieurs sorties: l'arbre syntaxique, la forme corrigée selon les règles d'accord, la représentation sémantique,...

VI.3. Prototype de TAO basé sur le modèle à transfert de structures :

Introduction:

Notre choix qui s'est porté au modèle à transfert de structures pour la conception et l'implémentation du prototype de TAO, dit aussi de deuxième génération, pour les buts suivants:

- 1- séparation des données aux traitements (dictionnaires et grammaires aux programmes de traitements).
- 2- évolution rapide et amélioration souple du prototype vers un système complet de TAO.
- 3- le prototype de TAO opère en trois phases appelées analyse, transfert et génération . Cette dernière distinction correspond aux trois étapes fondamentales de l'opération de traduction: la compréhension de la phrase source, la transposition de l'idée et la formulation de cette idée résultante en langue cible [HATON 91].
- 4- la possibilité de faire évoluer un système de deuxième génération vers un modèle de troisième génération [RASKIN 87],[TUCKER 87].

Un système de troisième génération associe au système de deuxième génération une base de connaissances dans laquelle le système peut puiser pour trancher les ambiguïtés de sens.

Dans notre démarche de travail et particulièrement dans l'élaboration des structures de transfert du français vers l'arabe, le travail de thèse de Doctorat de A-M. Boualem 93 nous a inspiré dans le contexte de modélisation.

Son modèle est structuré comme suit:

- Choix d'un contexte de travail: qui est basé sur un corpus technique très restrictif.
- Analyse du corpus et élaboration des structures de transfert: cette structuration est bijective, ie: Structure_française ↔ Structure_arabe .

A ce modèle, on complète par d'autres étapes et sans tenir compte de l'aspect bijectif :

- ↳ Recherche des régularités après l'analyse du corpus puis élaboration des structures de transfert.
- ↳ Test et vérification de ces structures dans la possibilité de rajouter d'autres règles qui complètent le bon fonctionnement et la fiabilité de ces structures. Comme la détermination (en arabe), l'accord de l'adjectif avec le nom (en arabe, l'adjectif ne s'accorde pas dans tous les cas en genre et nombre), possibilité de changement de catégorie de certains mots (le verbe en français peut être traduit en un nom-verbal dans le cas du gérondif) ...

VI.3.1. Le modèle ML-TASC: [BOUALEM 93]

Dans ce modèle de conversion de formalismes de langages techniques et scientifiques dans un environnement à syntaxe contrôlée et raisonnement limité, nous présentons le corpus en question, puis les structures qui ont été élaborées et les améliorations qu'on a apporté.

Voir ci-après: (Fig. VI.3.1a: Corpus de [BOUALEM 93]); et
(Fig. VI.3.1b: Tableau descriptif des structures de transfert).

Mode d'emploi:	<i>TaRiYQa& ?aL-?iSTi^CMaAL :</i>
Suivre le mode d'emploi pour obtenir un dossier ou un livre relié correctement.	<i>?iTba^C TaRiYQa& ?aL-?iSTi^CMaAL MiN ?aJLi ?aL-HuSuLi ^CLaY MiLaFF ?aW KiTaAB MuMSiK JaYYiD.</i>
Utiliser le courant électrique ou une pile électrique Type-A.	<i>?iSta^CMiL ?aL-TaYYaAR ?aL- KaHRuBaA?iY ?aW BaTTaARiYYa& KaHRuBaA?iYa& Type-A .</i>
Mettre l'interrupteur sur la position ON.	<i>Da^C ?aL-MiQFaL ^CaLaY ?aL-WaD^CiYYa& ON.</i>
La température normale de fonctionnement est atteinte après 4 minutes.	<i>DaRaJa& ?aL-HaRaARa& ?aL-^CaADiYYa& LiL-TaSGiYL WaASiLa& Ba^CDa 4 DaQaA?iQ .</i>
L'appareil chauffe mais peut rester sous tension.	<i>YuSaXXiN ?aL-JiHaAZ LaKiN YaSTaTiY^Ca ?aL-BaQaA?a MuSTaGiL .</i>
Un thermostat évite le surchauffage.	<i>MuNaDDiMu ?aL-HaRaARa& YaMNa^C ?iRTiFaA^C ?aL-HaRaARa& .</i>
Placer la pile de feuilles dans la zone de chauffe .	<i>Da^C MaJMu^Ca& ?aL-?aWRaAQ FiY MiNTaQa& ?aL-HaRaARa& .</i>
Retirer la pile reliée après le signal lumineux.	<i>?iHDaF ?aL-MaJMu^C a& MuMSiKa& Ba^CDa ?aL-?iSaARa& ?aL-DaW?iYYa&.</i>
Fig. VI.3.1a: Corpus de [BOUALEM 93] .	

Structure en français	Structure en arabe	Justifications et améliorations
Det + Nom-composé	Nom-composé	① 'le mode d'emploi': <i>TaRiYQa& ?aL-?iSTi^CMaAL</i> ② 'la zone de chauffe': <i>MiTaQa& ?aL-HaRaARa&</i> ⇒ si (Det=défini) alors (Det+Nom_composé) sinon (Nom_composé)

Prép + V	Prép + Det V	<p>① 'pour obtenir': <i>MiN ?aJLi ?aL-HuSuLi</i></p> <p>⇒ 'de laisser', 'de reproduire': (se traduisent en MaSDaR ou Nom-verbal) <i>TaRKi , ?i^caADa& DiKRi</i></p> <p>si (Prép+V_infinif) alors (Nom_verbal)</p>
Conj + GN	Conj + GN	<p>① 'ou un livre': <i>?aW KiTaABun</i></p>
Det + Nom + Adj	Det Nom + Det Adj	<p>① 'le courant électrique'</p> <p>⇒ en arabe l'adjectif est toujours postposé, donc:</p> <p>si (Non+Adj)ou(Adj+Nom) alors { (Nom+adj), si (Det=défini) alors (Det Nom +Det Adj), Appliquer_accord_gen&nbr(Nom,Adj) }</p>
Prép + Nom	Prép + Det Nom	<p>⇒ 'en cas de perte': il n'y a pas détermination, donc si (Prép+Nom) alors (Prép+Nom)</p>
Det + Nom + V	V + Det Nom	<p>⇒ c'est le cas typique de construction de phrases en arabe:</p> <p>si (structure_fr=SVO) alors { structure_ar := VSO, si (S=Pronom) alors structure_ar := VO, }</p> <p>avec: S(sujet), V(verbe) et O(compl. objet).</p>
V + V	V + Det V	<p>① 'peut rester': <i>YaSTaTiY^ca ?aL-BaQaA?a</i></p> <p>⇒ c'est le cas de (V + V_infinif):</p> <p>si (structure_fr=V+V_infinif) alors { V_infinif → <i>MaSDaR</i> (ou déverbal), structure_ar := soit: V + Det*Nom_verbal, soit: V + Prép(?aN) + V_conjugué, }</p> <p>Illustration: 'Je veux partir ...' <i>?uRiYDu + ?aL-DDaHaAB / ?aN ?aDHaBa</i></p>

Nom + 'de' + Nom	Nom + Det Nom	<p>① '(la) pile de feuilles': <i>MaJMu^Ca& (?aL-)?aWRaAQ</i></p> <p>⇒ on propose comme traduction convenante pour cet exemple: <i>(?aL-)MaJMu^Ca& MiN(a) (?aL-)?aWRaAQ</i></p> <p>Donc:</p> <p>si (structure_fr=[Det]+Nom1+'de'+Nom2) alors { structure_ar := soit: ([Det] Nom1 + [Det] Nom2), soit:(Nom1 + Prép(<i>MiN/Li</i>) +[Det]+Nom2 } Illustration: 'Ecole de filles': <i>MaDRaSa& LiL-BaNAT</i></p>
------------------	---------------	--

Fig. VI.3.1b: Tableau descriptif des structures de transfert.

VI.3.2. Corpus du prototype :

Notre choix de corpus, pour notre système expérimental, s'est orienté vers un texte bilingue (français-arabe) à portée de main: il s'agit de recommandations qu'on trouve sur un carnet de chèques (d'un pays maghrébin).

Ce choix se justifie pour les raisons suivantes:

- fréquence de lecture du texte;
- diversité des structures syntaxiques: bien que ce texte soit court, il est porteur de plusieurs constructions syntaxiques et la diversité de styles de phrases;
- l'étude de la traduction du texte, nous a permis d'approfondir certaines connaissances utiles à la bonne démarche pour la conception du système prototype de TAO :
 - ↳ modèle de connaissances pour le dictionnaire électronique;
 - ↳ modèle d'analyse des structures syntaxiques;
 - ↳ modèle pour les structures de transfert français-arabe .

Voir Fig. VI.3.2a: Le corpus de notre travail .

Le texte en français	Le texte en arabe
L'attention de notre clientèle est attirée sur les points suivants:	<i>YuLFaT ?iNTiBaAH ZaBaA?iNuNaA ?iLaY ?aL-NiQaAT ?aL-TTaALiYa&:</i>
1- Les chèques doivent être remplis avec beaucoup de soin.	<i>1- YaJiBu ?aN TuMLa?a ?aL-SSuKuK Bi-?HTiMaAMin KaBiYR .</i>
2- Les rectifications éventuelles doivent être approuvées en n'omettant pas de remplir intégralement, le cas échéant, le montant en toutes lettres.	<i>2- YaJiBu ?aN YuSaADiQa ^caLaY ?aL-TTaSHiYHaAT ?aA-MuHTaMaLa BiDuWNI ?iGFaAL ?i^caADa& DiKri ?aL-MaBLaGi Bi-KaAMiLiHi ^ciNDa ?aL-?iQTiDaA? Wa Bi-?aL-?aHRouFi ?aL-KaAMiLa&.</i>
3- Il convient d'éviter de laisser un blanc devant les sommes en chiffres ou en lettres et de veiller à ce qu'un minimum d'espace existe entre les mots ou entre les lettres, afin d'empêcher toute modification ultérieure.	<i>3- YaNBaGiY TaHaASiY TaRKi BaYaAD ?aMaAMa ?aL-MaBaALiG Bi-?aL-?aRQaAM ?aW Bi-?aL-?aHRuFi Wa ?aL-SSaHaRi ^caLaY ?aLLa YaBQaY BaYNa ?aL-KaLiMaAT ?aW BaYNA ?aL-?aHRuF SiWaY FaDaA?in ?aNna Wa DaLiKa LiL-HaYLuLa& DuWNa KuLLi TaGYiRin.</i>
4- La Banque Nationale est déchargée de toute responsabilité en cas de perte, de soustraction ou d'usage irrégulier ou frauduleux des formules contenues dans le présent chéquier, à moins qu'elle n'ait été prévenue à temps, pour refuser le paiement.	<i>4- ?aL-BaNK ?aL-WaTaNiY LaA YaTaHaMMaL ?aYYa MaS?uLiYYa& FiY HaALa& DaYaA^c , ?iXTiLaAS ?aW ?iSTi^cMaAL GaYR QaANuNiY ?aW TaDLiYSiY LiL-SSiYaG ?aL-LaTiY YaHTaWiY ^caLaYHaA DaFTaR ?aL-SSuKuK HaDaA , ?iLLa ?iDaA ?uXTiRa Bi-DaLiKa FiY ?aL-WaQTu ?aL-MuNaASiB , Li-YuRFaDa ?aL-DDaF^ca .</i>
Fig. VI.3.2a: Corpus de notre travail.	

Notre démarche de travail pour ce corpus consiste à :

- 1- analyser le corpus phrase par phrase;
- 2- retrouver les régularités entre les phrases en analyse et en synthèse;
- 3- généraliser ces régularités en structures de transfert français-arabe;
- 4- compléter si possible chaque structure de transfert par des règles d'accord, de détermination ou de transformation de catégories;
- 5- dresser la liste finale des structures retenues et les règles complémentaires à celles-ci.

Nous résumons ces différentes étapes par le tableau ci-dessous (Fig. VI.3.2b).

Structure en français	Structure en arabe	Commentaires																					
SN: Det(=Adj_poss) + Nom_com	SN: Nom_com * Det	'notre clientèle': <i>ZaBaA?iNu * NaA</i> ⇒ le(*)est un symbole binaire qui dénote la concaténation des deux mots.																					
SN: Det(=défini) + Nom_com	SN: Det * Nom_com	'les chèques': <i>?aL-SSuKuWK</i>																					
SN: Det + Nom_com + Adj	SN: Det*Nom_com + Det*Adj et Accord (Nom,Adj): <table border="1" style="margin-left: 20px;"> <thead> <tr> <th colspan="2">Nom(gen,nbr)</th> <th>Adj(gen,nbr)</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>fem,sin</td> <td>fem,sin</td> </tr> <tr> <td>2</td> <td>mas,sin</td> <td>mas,sin</td> </tr> <tr> <td>3</td> <td>fem,plur, H+</td> <td>fem,plur</td> </tr> <tr> <td>4</td> <td>fem,plur, H-</td> <td>fem,sin</td> </tr> <tr> <td>5</td> <td>mas,plur, H+</td> <td>mas,plur</td> </tr> <tr> <td>6</td> <td>mas,plur, H-</td> <td>fem,sin</td> </tr> </tbody> </table> H+: humain et H-: non-humain	Nom(gen,nbr)		Adj(gen,nbr)	1	fem,sin	fem,sin	2	mas,sin	mas,sin	3	fem,plur, H+	fem,plur	4	fem,plur, H-	fem,sin	5	mas,plur, H+	mas,plur	6	mas,plur, H-	fem,sin	'les points suivants': <i>?aL-NNiQaAT ?aL-TaALiYa&</i>
Nom(gen,nbr)		Adj(gen,nbr)																					
1	fem,sin	fem,sin																					
2	mas,sin	mas,sin																					
3	fem,plur, H+	fem,plur																					
4	fem,plur, H-	fem,sin																					
5	mas,plur, H+	mas,plur																					
6	mas,plur, H-	fem,sin																					
SN: Det + Adj + Nom	SN: Det*Nom + Det*Adj et Accord (Nom,Adj)																						
SP-infinitival: Prép(de) + V_infinitif	Déverbal: Nom_verbal	'de laisser': <i>TaRKi</i> 'd'éviter': <i>TaHaASi</i>																					
SP-complément: Prép(en) + Nom	SP: Prép(Bi)*Det_def*Nom	'en lettres': <i>Bi-?aL-?aRQaAM</i>																					
SP-gérondif: Prép(en) + V_pp	SV / SN: (?aL-HaAL) Adverbe_de_manière / adjectif. { SV: Pronom + V <u>ou</u> SN: ?aL-HaAL }	c'est un cas typique en arabe qui se traduit, soit par une phrase verbale, soit par une phrase sans verbe: '(β=Le garçon mange) en marchant': soit: β + <i>WaHWa YaMSiY</i> soit: β + <i>MaASin</i>																					

SP: Prép + SN	SP: Prép */+ SN et si (V+SP) alors { le V_arabe qui donne Prép, par: Propose(V,Prép) }	'sur (la table)': <i>aLaY</i> 'V(attirer) sur (les points suivants)': attirer= <i>YuLFiTu</i> +P(? <i>iLaY</i>), <i>?iLaY ?aL-NiQaT</i> ...
Det: Det(=indéfini) + X	<i>TaNWiYN</i> : X * λ_n ; $\lambda = a u i$	'une banque': <i>BaNKun</i>
Det: Det(=groupe_adv) + X	X + Det	'(avec) beaucoup de (soin)': <i>Bi-?iHTiMaAMin KaBiYRin</i>
Prép: Prép(afin de)	Conj+Pro_dems+ prép(Li)	'afin d'empêcher': <i>Wa DaLiKa Li*</i> ...
SN: SN + SP	SN: SN + SP	'le montant en toutes lettres': <i>?aL-MaBLaGi Bi-?aL- ?aHRuFi ?aL-KaAMiLa&</i>
P: (Phrase) SN + SV(= V + SN/SP)	P: V + SN + SN/SV	Toutes les phrases de notre corpus.
P: SN(= Pronom) +SV(=V + X)	P: V + X	'il convient d'éviter': <i>YaNBaGiY TaHa:SiY</i>
P: SN(=Nom_propre)+SV(=V+X)	P: V + Nom_propre + X	'Lamia mange une pomme': <i>Ta?KuLu LaMYa? TouFFa:Ha&</i>

Fig. VI.3.2b: Structures de transfert du corpus.

VI.4. La génération :

Introduction:

La génération en langue cible (en arabe) constitue la dernière phase dans la conception de notre prototype de TAO, après la conception du dictionnaire électronique, l'analyse de la phrase source (en français) et l'élaboration des structures de transfert (français-arabe).

VI.4.1. Les différents niveaux de la génération :

La génération dans notre étude se décompose en trois étapes:

1- Niveau I : génération des mots simples

Par la consultation directe du dictionnaire électronique et dans ce cas :
Mot_ar(i) = générer(Mot_fr(i)) ;

2- Niveau II : génération des mots complexes

Il s'agit pour notre prototype de TAO :

- génération des verbes conjugués (Conjuguer_verbe),
- génération des adjectifs (par la règle Accord(Nom,Adjectif)),
- génération des prépositions dans un SP (Propose (verbe,prép)),

```
selon ( catégorie (Mot_ar(i))= V ou Adj ou Prép )
  alors
  {
    V      : Mot_ar(i) ← Conjuguer_verbe (Mot_ar(i)) + {Pers.,Gen.,Nbr.};
    Adj    : Mot_ar(i) ← Accord (Nom,Mot_ar(i)) + {Gen., Nbr.};
    Prép   : Mot_ar(i) ← Propose (Verbe,Mot_ar(i));
  }
```

3- Niveau III : génération en phrase cible

D'après les structures de transfert on établit la génération finale en langue cible:

```
selon (Structure = (Phrase: P_fr) ou
                (Groupe_nominal :GN_fr) ou
                (Groupe_verbal : GV_fr) )
  alors
  {
    Phrase      : générer_ST( P_fr,P_ar);
    Groupe_nominal : générer_ST(GN_fr,GN_ar);
    Groupe_verbal  : générer_ST(GV_fr,GV_fr);
  }
```

VI.4.2. La génération de niveau I :

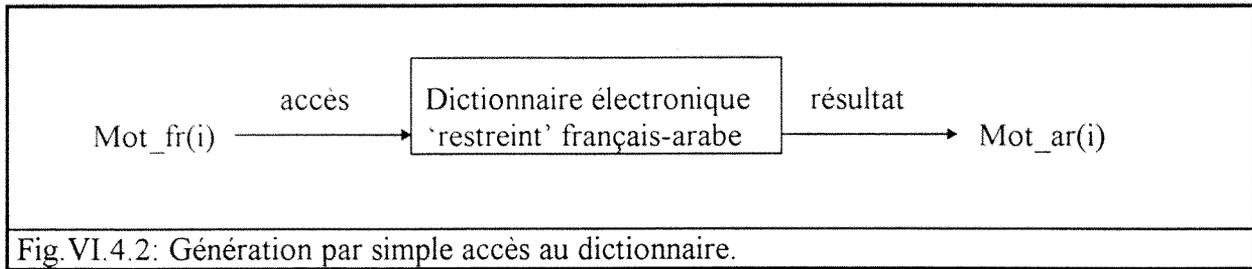


Fig.VI.4.2: Génération par simple accès au dictionnaire.

A la recherche de la traduction du mot Mot_fr(i) dans le dictionnaire bilingue, le résultat est le mot Mot_ar(i) qui est l'occurrence la plus probable .

L'occurrence convient-elle dans le traitement ?

Pour résoudre ce genre de problème, on peut puiser dans la liste des équivalences, à la recherche d'un autre mot arabe Mot_ar(i') convenant :

- ① La liste des mots est présentée à l'utilisateur pour faire son choix, dans ce cas on parle de système TAO. Ceci rentre dans le cadre du présent travail, mais peut être un prolongement ou un sujet d'étude.
- ② Un traitement automatique est prévu pour ce niveau de génération. Dans ce cas chaque élément de la liste y compris la première occurrence de traduction subissent un test de cohérence des traits.

➤ On illustre ce genre de traitement, par l'exemple suivant:

```
si (trait(Sujet)=H+) et ( trait(Verbe)=v_état) alors (trait(Objet)=H+);  
sinon  
si (trait(Sujet)=H-) et ( trait(Verbe)=v_état) alors (trait(Objet)=H-);  
etc.
```

➤ Un autre cas d'illustration, que le verbe peut être prévisible pour ce genre de traitement:

```
selon ( arguments( VERBE) ) alors  
{  
  arg1 : si (arg1=H+) alors (arg2=H-);  
  arg2 : si (arg2=H-) alors (arg3=H-);  
  etc.  
}
```

Ce type de traitement de prévisibilité, assurant le bon choix de traduction des mots, nécessite :

- d'une part, une étude sémantique des verbes pour dresser la liste des traits de chacun de ces arguments, et
- d'autre part, une étude des traits des constructions syntaxiques (SN, SP, SV, ...) dans des positions différentes (Sujet, COD, COI, CC, ...) pour prévoir les traits dans chaque type de construction. Par conséquent, le choix du mot parmi la liste devient plus aisé selon un trait prédéterminé.

Ceci ne rentre pas dans le cadre du présent travail, mais pourra être un prolongement de celui-ci et faire appel à un nouveau sujet d'étude en collaboration.

VI.4.3. La génération de niveau II:

A) Génération des verbes :

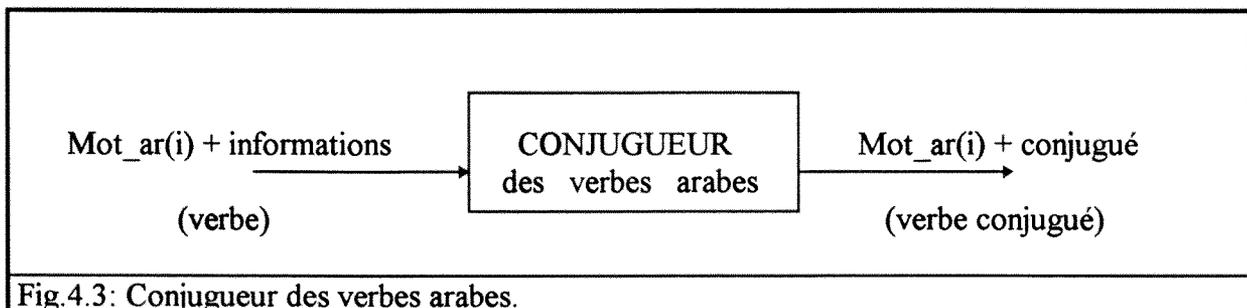


Fig.4.3: Conjugueur des verbes arabes.

Selon les travaux qui sont réalisés sur la conjugaison des verbes arabes [ABU AL-CHAY 88] et [HASSOUN 87], l'incorporation du système conjugueur des verbes à notre prototype de TAO, nous permet de générer le verbe au mode et aspect, la personne et le nombre désirés.

Les données de la conjugaison sont:

- 1- le verbe à la 3PMS de l'accompli actif,
- 2- le ou les modes de conjugaisons désirés.

Ainsi, il est possible de définir trois types de conjugaison:

- la conjugaison du verbe pour tous les modes (accompli, inaccompli, impératif...),
- la conjugaison d'un verbe pour un mode donné,
- la conjugaison d'un verbe à un mode et pour une personne, un genre et un nombre donnés.

⇒ C'est bien ce troisième type qui nous intéresse pour le prototype de TAO.

Selon le travail de [ABU AL-CHAY 88] le programme de traitement a été écrit en PROLOG, et la règle de conjugaison est écrite comme suit:

```
conjuguer(_verbe,_aspmode,_pers,_nomb,_gen,PRF,BASE,SUF) :-  
    verbe(_verbe,NCPS,NCB),  
    aspmode_lvps(_aspmode, (N,M) ),  
    compatible(_pers,_nomb,_gen), pronom(PNG,PRO),  
    classe(NCPS,L), dans(NVS,L), dans(NVP,L),  
    >(NVS,N), <(NVS,M), <(NVP,3),  
    base(NVS,PNG,BASE),  
    suffixe(NVS,PNG,SUF),  
    conc(BASE,SUF,BASESUF),  
    sortir(BASESUF,_aspmode,PNG,NVP,PRF,N).
```

avec:

_verbe: le verbe;
_aspmode: l'aspect et le mode;
_pers: la personne (1ère,2ème ou 3ème);
_nomb: le nombre (S:singulier, P:pluriel ou D:duel);
_gen: le genre (M:masculin, F:féminin ou N:commun);
PRF: le préfixe ;
BASE: la base ;
SUF: le suffixe ;

Fig. VI.4.3: règle de conjugaison en PROLOG-FOLL [ABU AL-CHAY 88].

La première étape de cette règle consiste à vérifier le verbe, l'aspect-mode, la personne, le nombre, le genre et leurs compatibilités. Ensuite, en fonction de ces informations, on trouve le suffixe, la base et éventuellement le préfixe relatif du verbe conjugué.

Il y'a trois types 'sortir', cette règle peut fournir:

- ① l'aspect accompli et le mode impératif,
- ② l'aspect inaccompli voix « active »,
- ③ l'aspect inaccompli voix « passive ».

Il reste le problème de conversion de « temps », « aspect » et « mode » du français vers l'arabe. Ce problème constitue un long travail de recherche à proposer, du fait que:

- La langue française est une langue très précise dans la description de l'état du verbe et son action dans l'espace et le temps. Cette caractéristique permet de situer le sujet, son état et la continuité du phénomène sans ambiguïté.
- Ces caractéristiques de description ne se retrouvent pas toutes dans la langue arabe. Parfois l'ambiguïté se prolonge jusqu'aux choix de mode ou aspect.

B) Génération des adjectifs:

On rappelle la règle de génération de l'adjectif arabe, qui s'accorde selon le genre, le nombre et le trait ±Humain du nom arabe:

<p>si ((Nom+Adj) ou (Adj+Nom)) alors { Accord (Nom,Adj):</p> <table border="1" style="margin-left: 20px;"> <thead> <tr> <th></th> <th>Nom(gen,nbr)</th> <th>Adj(gen,nbr)</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>fem,sin</td> <td>fem,sin</td> </tr> <tr> <td>2</td> <td>mas,sin</td> <td>mas,sin</td> </tr> <tr> <td>3</td> <td>fem,plur, H+</td> <td>fem,plur</td> </tr> <tr> <td>4</td> <td>fem,plur, H-</td> <td>fem,sin</td> </tr> <tr> <td>5</td> <td>mas,plur, H+</td> <td>mas,plur</td> </tr> <tr> <td>6</td> <td>mas,plur, H-</td> <td>fem,sin</td> </tr> </tbody> </table> <p>Adj ← générer(Adj,gen_N,nbr_N), } H+: humain et H- : non-humain .</p>		Nom(gen,nbr)	Adj(gen,nbr)	1	fem,sin	fem,sin	2	mas,sin	mas,sin	3	fem,plur, H+	fem,plur	4	fem,plur, H-	fem,sin	5	mas,plur, H+	mas,plur	6	mas,plur, H-	fem,sin	<p>Illustration:</p> <p>① <i>SaYYaARa& SaRiY^ca&</i> : 'une voiture rapide' ② <i>TaALiB DaKiYY</i> : 'un étudiant intelligent' ③ <i>FaTaYaAT JaMiYLaAT</i> : 'des belles filles' ④ <i>?aSJaAR MuTMiRa&</i> : 'des arbres fruitiers' ⑤ <i>TuLLaAB ?aDKiYaA?</i> : 'des étudiants intelligents' ⑥ <i>KuTuB QaYYiMa&</i> : 'des livres intéressants'</p>
	Nom(gen,nbr)	Adj(gen,nbr)																				
1	fem,sin	fem,sin																				
2	mas,sin	mas,sin																				
3	fem,plur, H+	fem,plur																				
4	fem,plur, H-	fem,sin																				
5	mas,plur, H+	mas,plur																				
6	mas,plur, H-	fem,sin																				

C) Génération des prépositions:

Dans le cadre d'étude de notre corpus, on distingue les cas de génération suivants:

Le cas (en français)	La génération	Exemples
CAS 1: SP-infinitival Prép(de) + V_infinitif	Nom_verbal ☞ Prép=∅	'de laisser': <i>TaRKi</i> 'd'éviter': <i>TaHaASi</i>
CAS 2: SP-complément Prép(en) + Nom	Prép(Bi)*Det_def*Nom ☞ Prép= Bi	'en lettres': <i>Bi-?aL-?aRQaAM</i>
CAS 3: SP-gérondif Prép(en) + V_pp	?aL-HaAL ☞ Prép=∅ ie: ≈ Adverbe_de_manière / adjectif (en français). soit: PV: Pronom + V ou SN: ?aL-HaAL	c'est un cas typique en arabe et se traduit soit par une phrase verbale, soit par une phrase sans verbe: 'le garçon mange ... en marchant': soit: β... <i>WaHuWa YaMSiY</i> soit: β... <i>MaASin</i>

<p>CAS 4: SP Prép + SN</p>	<p>Prép */+ SN \Rightarrow Prép $\neq \emptyset$ si (cas: V+SP) alors { le V qui peut donner Prép, par: Propose(V,Prép) }</p>	<p>'sur (la table)': $^c aLa\ddot{Y} (...)$ 'V(attirer)': ① sur (les points suivants)': attirer = $V(YuL\ddot{F}iTu)$ +Prép(?iLa\ddot{Y}) $\Rightarrow ?iLa\ddot{Y} (?aL-NiQaT...)$</p>
---------------------------------------	--	---

VI.4.4. La génération de niveau III:

C'est le niveau de génération en langue cible (ou synthèse de la phrase arabe).

On illustre à ce niveau de génération que le système a réussi toutes les étapes d'analyse et les deux premiers niveaux de génération :

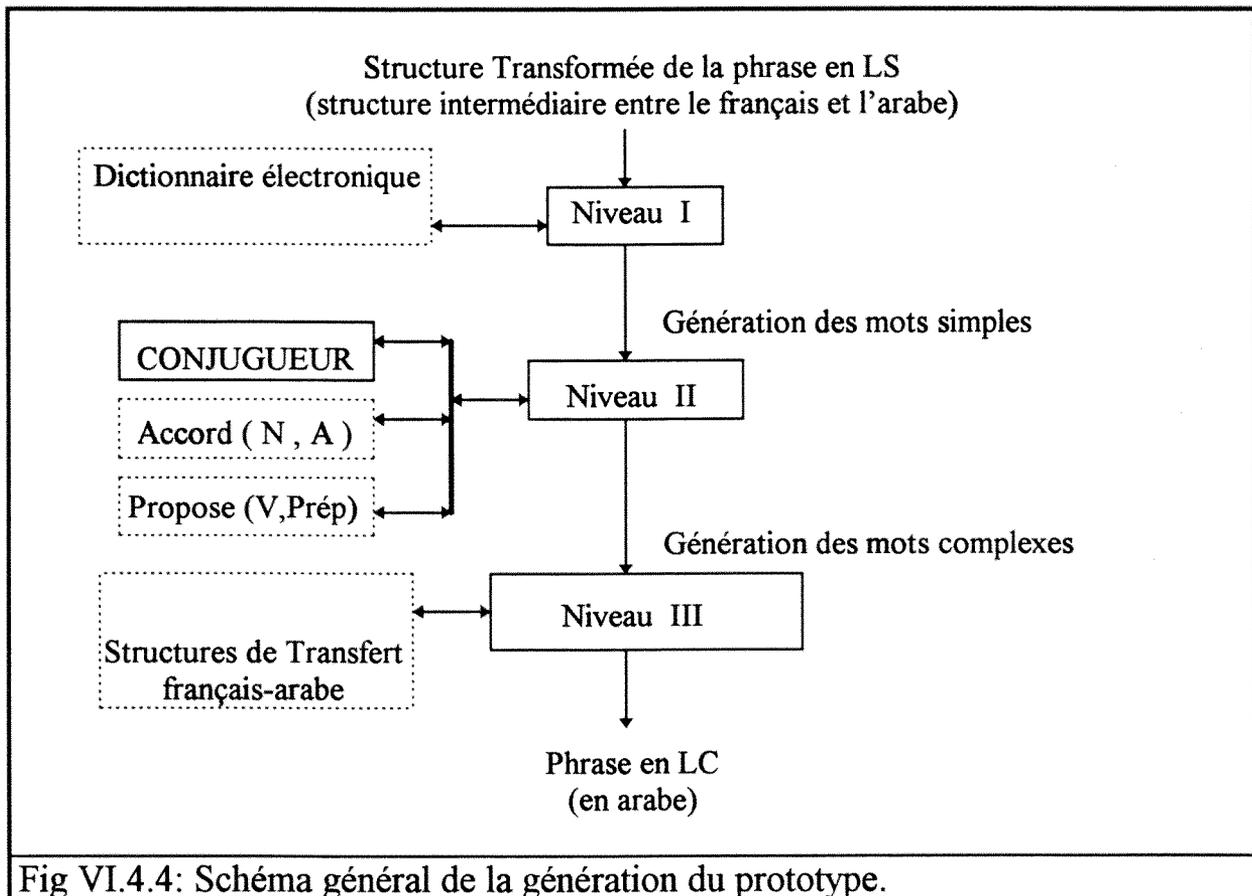


Fig VI.4.4: Schéma général de la génération du prototype.

VI.5. Architecture du système prototype de TAO :

Essayons de résumer en un schéma les composants nécessaires de notre système prototype .

Au coeur du système, on trouve bien entendu les programmes découpés classiquement en trois ensembles : **Analyse**, **Transfert** et **Synthèse**, eux-mêmes subdivisés en modules indépendants réalisant chacun une tâche majeure bien déterminée .

Ces programmes seront écrits en un langage adapté aux besoins de la linguistique (comme exemple, un langage déclaratif), aussi ces programmes seront mis en oeuvre pour diriger et d'en surveiller l'application dans chaque niveau de traitement .

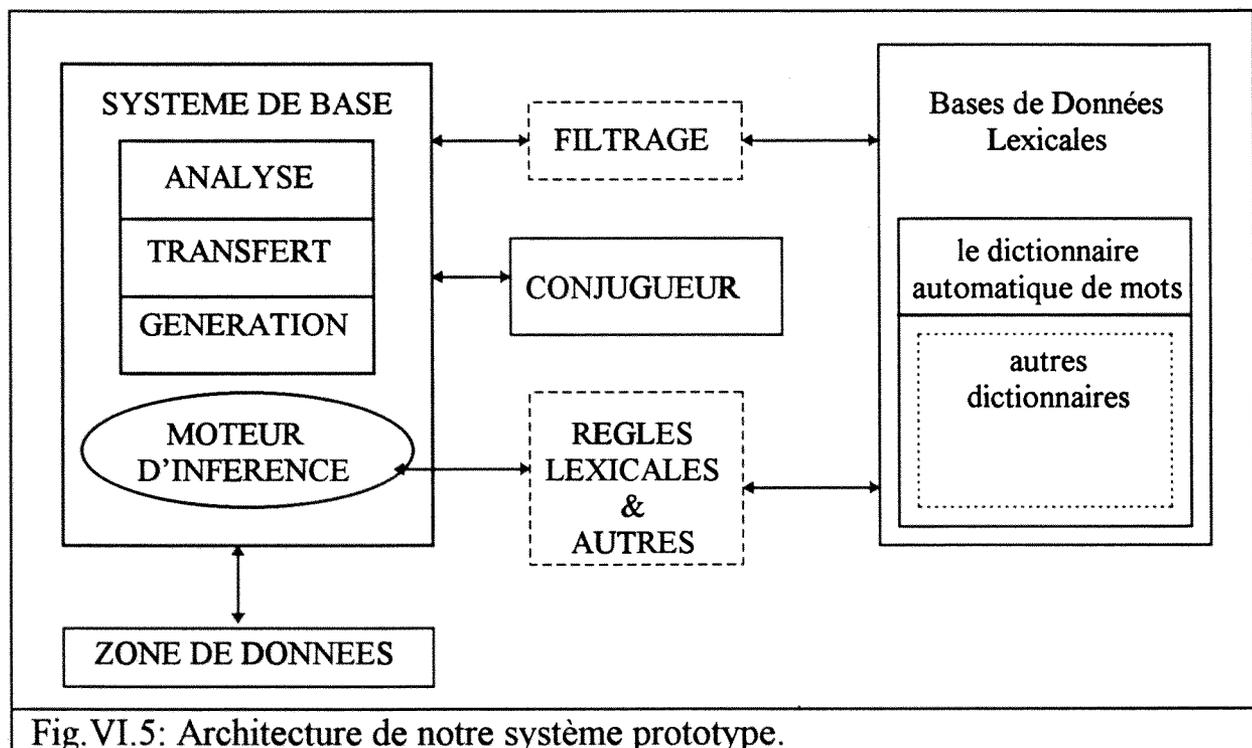


Fig. VI.5: Architecture de notre système prototype.

Chapitre 7 :

Réalisation informatique

- Prototype de TAO du français vers l'arabe -

VII. Réalisation informatique

- Prototype de TAO du français vers l'arabe - :

VII.1. Introduction :

L'objectif de notre réalisation informatique du prototype de TAO est pour une phrase donnée en français de faire son analyse syntaxique, d'effectuer le transfert structural français-arabe afin de produire en sortie la traduction de cette phrase en arabe.

Exemple:

(entrée) « Jean mange une pomme dans la rue. »

(sortie) « Ya?KuLu JuWN TuFFaAHa& FiY ?aL-TTaRiYQ. »

VII.2. Caractéristiques de l'application :

VII.2.1. Type de phrases analysées :

Les phrases acceptées de la langue française sont des phrases simples (Voir caractéristiques du modèle chap. IV.4.), analysables par la grammaire implémentée dans notre prototype.

VII.2.2. Les catégories syntaxiques retenues:

Catégories	Symboles
Nominales :	f1 : nom propre ; f2 : pronom personnel ; f3 : nom commun ; f4 : adjectif ; d : prédéterminant ; p : préposition
Intermédiaires :	c : conjonction ; w : adverbe ; t : ponctuation
Verbales :	v : verbe

VII.2.3. Les flexions retenues :

A°) Pour le lexique français :

Le choix des flexions est effectué selon la syntaxe du projet SYDO [DE BRITO 91] et [METZGER 88].

Flexions	Symboles
Genre :	mas : masculin ; fem : féminin ; _ : indéterminé
Nombre :	sin : singulier ; plur : pluriel ; _ : indéterminé
Personne :	1 : première personne ; 2 : deuxième personne ; 3 : troisième personne

B°) Pour le lexique arabe :

Le choix des flexions est effectué selon la syntaxe adoptée par [AL-CHAY 88] et [HASSOUN 87].

Flexions	Symboles
Genre :	M : masculin ; F : féminin ; _ : indéterminé
Nombre :	S : singulier ; D : duel ; P : pluriel ; _ : indéterminé ;
Personne :	1 : première personne ; 2 : deuxième personne ; 3 : troisième personne

VII.3. Environnement du prototype :

VII.3.1. Le lexique :

L'acquisition de données pour la traduction est basée sur les entrées lexicales ou terminologiques bilingues. Notre approche consiste à construire un dictionnaire électronique bilingue restreint pourvu d'un composant d'analyse morphologique.

Les informations morpho-syntaxiques (genre, nombre, personne...) sont conservées sous forme de traits qui accompagnent les mots. Nous aurons besoin de ces informations lors des traitements pour l'analyse syntaxique, le transfert structural et la synthèse en langue cible.

Une entrée dans le dictionnaire bilingue de notre application est décrit comme suit :

dic (Mot_fr, Cat_fr, [Gen_fr, Nbr_fr, (Pers_fr)],
Cat_ar, [GNP_ar¹, GNP_ar²,...], [Mot_ar¹, Mot_ar²,...]).

avec :

Mot_fr : un mot du lexique français ;

Cat_fr : catégorie de Mot_fr ;

[Gen_fr, Nbr_fr, (Pers_fr)] : la liste des flexions de Mot_fr en genre (Gen_fr), en nombre (Nbr_fr) et optionnellement en personne (Pers_fr) ;

Cat_ar : catégorie équivalente de Mot_fr en arabe ;

[GNP_ar¹, GNP_ar²,...] : la liste des flexions des mots arabes équivalents à Mot_fr en genre (G), en nombre (N) et en personne (P) ;

[Mot_ar¹, Mot_ar²,...] : liste des mots équivalents en arabe .

Exemples:

dic(le,d,[mas,sin], d, [], ["?aL-"]).

dic(socrate,f1,[mas,sin,p3], f1, ["_SM"], ["SuQRaAT"]).

dic(elle,f2,[fem,sin,p3], f2, ["3SF"],["HiYa"]).

dic(homme,f3,[mas,sin], f3, ["3SM","3SM"], ["RaJuL","?iNSaAN"]).

dic(mange,v,[_sin,p3], v, ["3SM","3SF"], ["Ya?KuLu(+Bi)","Ta?KuLu"]).

dic(intelligent,f4,[mas,sin], f4, ["_SM","_SM"], ["DaKiYY","FaTiN"]).

dic(peu,w,[_], w, [], ["QaLiYL"]).

dic(et,c,[_], c, [], ["Wa"]).

etc.

VII.3.2. La grammaire implémentée :

A notre Base de Données lexicales (BDD ou dictionnaire automatique bilingue), on ajoute un composant d'analyse syntaxique. Celui-ci, muni des informations lexicales, peut alors décider de choisir les composants d'une phrase et ses différentes structures .

La grammaire implémentée dans notre application pour l'analyse syntaxique d'une phrase française est la suivante :

<phrase> ::= <SN> <SV>

<SN> ::= <Nom propre> (<SP>) |
<Pronom> (<SP>) |
<Déterminant> (<Adj>) <Nom commun> (<Adj>) (<SP>)

<SV> ::= <Verbe> |
<Verbe> (<Adj>) |
<Verbe> (<Adv>) | <Verbe> <SN>

<SP> ::= <Prép> <SN>

<Nom propre> ::= nom propre
<Pronom> ::= pronom
<Nom commun> ::= nom commun | substantif
<Déterminant> ::= déterminant | article
<Adj> ::= adjectif
<Verbe> ::= verbe
<Adv> ::= adverbe
<Prép> ::= préposition | locution-prépositionnelle

où : <X> ::= y ; X est un non-terminal et y est un terminal ,
' | ' : est le symbol de disjonction 'ou' ,
(Z) : Z est un symbole optionnel .

VII.4. Présentation du système prototype :

VII.4.1. L'analyseur lexical :

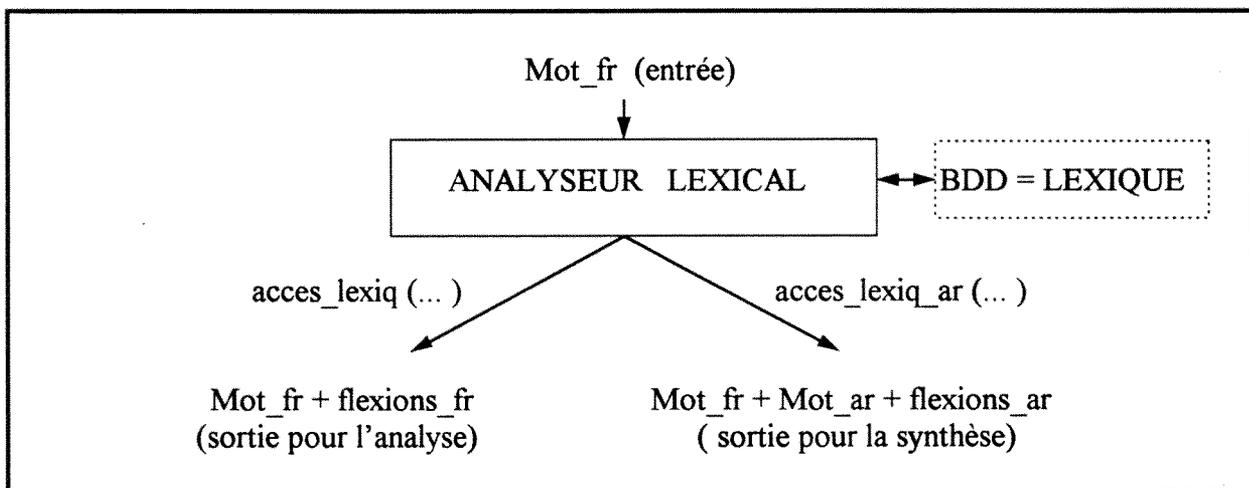


Fig. VII.4.1 : L'analyseur lexical .

VII.4.1.1. Principales règles de l'analyseur lexical :

La recherche d'un mot français dans le lexique permet aussi de retrouver les informations le concernant : sa catégorie et ses flexions en genre , nombre et personne, etc. .

Ces informations seront utiles pour faire l'analyse et le contrôle syntaxique des mots de la phrase .

La règle de recherche d'un mot français dans le lexique est identifiée par le prédicat `acces_lexiq`.



```
acces_lexiq(MOT,CAT,[GEN_mot,NB_mot,PERS_mot]) :-  
    dic(MOT,CAT,[GEN_mot,NB_mot,PERS_mot],_,_,_),  
    write("\n R1:"),  
    !.
```

```
acces_lexiq(MOT,CAT,[GEN_mot,NB_mot|_]) :-  
    dic(MOT,CAT,[GEN_mot,NB_mot|_],_,_,_),  
    write("\n R2:"),  
    !.
```

A la phase de transfert structural français-arabe et à celle de génération, la règle `acces_lexiq_ar` permet d'identifier le mot arabe équivalent au mot français qui a subi le traitement par l'analyseur syntaxique . Cette règle permet aussi d'identifier la catégorie et les flexions du mot arabe qui sont des informations nécessaires au transfert et à la génération en arabe .



```
acces_lexiq_ar(MOT_fr,[MOT_ar,CAT,GEN,NBR,PERS]):-  
    dic(MOT_fr,_,_,CAT,[PNG|_],[MOT_ar|_]),  
    personne(PNG,PERS),  
    nombre(PNG,NBR),  
    genre(PNG,GEN).
```

Les règles d'identification des flexions :



```
personne(PNG,_):- free(PNG),!.  
personne(PNG,PERS):- frontchar(PNG,CHAR,REST),  
    identifier_png(CHAR,PERS), !.
```



```
nombre(PNG,_):- free(PNG),!.  
nombre(PNG,NBR):- frontchar(PNG,_REST1),  
    frontchar(REST1,C,_),  
    identifier_png(C,NBR), !.
```



```
genre(PNG,_):- free(PNG),!.  
genre(PNG,GEN):- frontchar(PNG,_REST1),  
    frontchar(REST1,_REST2),  
    frontchar(REST2,C,""),
```

identifie_r_png(C, GEN), !.

Les Faits (ou clauses) pour les flexions :

```
identifie_r_png('M', "mas") .  
identifie_r_png('F', "fem") .  
identifie_r_png('S', "sin") .  
identifie_r_png('P', "plur") .  
identifie_r_png('1', "p1") .  
identifie_r_png('2', "p2") .  
identifie_r_png('3', "p3") .  
identifie_r_png('_', _) .
```

VII.4.1.2. Quelques résultats du programme :

```
Files      Edit      Run      Compile      Options      Setup
----- Dialog -----
? votre mot à chercher dans le dico. =il

R1:
MOT = il
CAT = f2
GEN = mas
NBR = sin
PERS = p3

MOT_ar = HuWa
CAT_ar = f2
GEN_ar = mas
NBR_ar = sin
PERS_ar = p3

Press the SPACE bar

F2-Save  F3-Load  F6-Switch  F9-Compile                               Alt-X-Exit
```

```
Files      Edit      Run      Compile      Options      Setup
----- Dialog -----
? votre mot à chercher dans le dico. =intelligente

R2:
MOT = intelligente
CAT = f4
GEN = fem
NBR = sin
PERS = _

MOT_ar = DaKiYa&
CAT_ar = f4
GEN_ar = fem
NBR_ar = sin
PERS_ar = _

Press the SPACE bar

F2-Save  F3-Load  F6-Switch  F9-Compile                               Alt-X-Exit
```

VII.4.2. L'analyseur syntaxique :

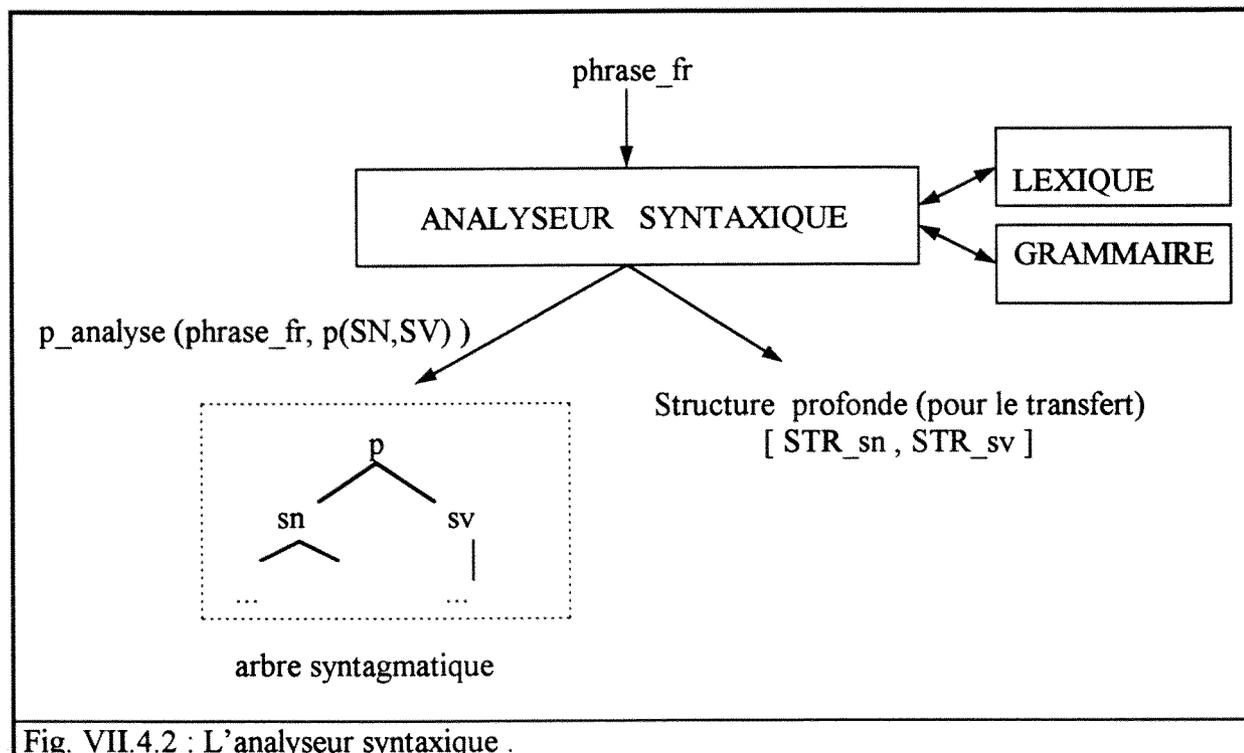


Fig. VII.4.2 : L'analyseur syntaxique .

VII.4.2.1. Principales règles de l'analyseur syntaxique :

L'analyse syntaxique de la phrase en français est déclenchée par la règle **p_analyse** . Ce prédicat a quatre arguments : le premier argument est **PHRASE_e** qui représente la chaîne de caractères de la phrase française (entrée), le deuxième argument est de structure complexe **p(ARBRE_sn,ARBRE_sv)** et représente le résultat de l'analyse syntaxique , c'est à dire l'arborescence de la phrase (ou arbre syntagmatique). Les derniers arguments représentent les structures à transmettre comme entrées pour la règle de transfert structural.

La règle **p_analyse** fait appel à deux autres règles : **sn_analyse**, la règle d'analyse des syntagmes nominaux et **sv_analyse**, la règle d'analyse des syntagmes verbaux.

La règle d'analyse d'une phrase :

P ::= SN + SV :



```
p_analyse( PHRASE_e,p(ARBRE_sn,ARBRE_sv) ,STR_sn, STR_sv) :-
    sn_analyse( PHRASE_e,REST_s,[GENRE,NOMBRE],ARBRE_sn, STR_sn ),
    sv_analyse( REST_s,[GENRE,NOMBRE],ARBRE_sv , STR_sv),
    !.
```

Des règles d'analyse de syntagmes nominaux :

SN ::= Nom-propre + [PP] :



```
sn_analyse( PHRASE_e,REST_s,[GN,NB],sn_1(n_prp(NOM_prp),LISTE_pp) ,
           [ sn_1, GN, NB, PERS, CHAINE_sn,[MODIF]] ) :-
fronttoken( PHRASE_e,NOM_prp,REST_prp ),
acces_lexiq( NOM_prp,f1,[GN,NB] ),
liste_sp_analyse( REST_prp,REST_s,LISTE_pp,MODIF),
!.
```

SN3 ::= Det + [Adj] + Nom-com + [Adj] :

```
sn_analyse( PHRASE_e,REST_s,[GN,NB],
           sn_3(d(DETER),LISTE_adj1,n_com(NOM),LISTE_adj2,LISTE_pp) ,
           [ sn_3, GN, NB, PERS, CHAINE_sn, [MODIF]] ) :-
fronttoken( PHRASE_e,DETER,REST_det ),
acces_lexiq( DETER,d,[GN,NB] ),
liste_adj(REST_det,REST_adj1,LISTE_adj1,[GN,GB]),
fronttoken( REST_adj1,NOM,REST_nom ),
acces_lexiq( NOM,f3,[GN,NB] ),
liste_adj(REST_nom,REST_adj,LISTE_adj2,[GN,GB]),
liste_sp_analyse( REST_adj,REST_s,LISTE_pp,MODIF),
bound(LISTE_pp),
!.
```

La règle d'analyse d'un syntagme prépositionnel :

SP ::= prép + SN :



```
sp_analyse( PHRASE,REST,sp(p(PREP),SN) ,[sp,G_p,NB_p,PERS,CHAINE_sp]) :-
fronttoken(PHRASE,PREP,REST_prep),
acces_lexiq(PREP,p,_),
sn_analyse(REST_prep,REST,[G_p,NB_p],SN,STR_sn),
!.
```

Des règles d'analyse de syntagmes verbaux :

SV1 ::= verbe :



```
sv_analyse(REST_e,[GEN_sn,NB_sn],sv_1(v(VERB)),
           [ sv_1, GEN_sn, NB_sn, PERS, VERB,[] ] ):-
fronttoken(REST_e,VERB,""),
acces_lexiq(VERB,v,[GEN_sn,NB_sn]),
!.
```

SV2 ::= verbe + [ADJ] :

```
sv_analyse(REST_e,[GEN_sn,NB_sn],sv_2(v(VERB),ADJ), [ sv_1, GEN_sn,
    NB_sn, PERS, VERB, [f4,GEN_sn,NB_sn,PERS,CHAINE_adj]]):-
    fronttoken(REST_e,VERB,REST_v),
    acces_lexiq(VERB,v,[GEN_sn,NB_sn]),
    liste_adj(REST_v,"",ADJ,[GEN_sn,NB_sn]),
    !.
```

SV4 ::= verbe + GN :

```
sv_analyse(REST_e,[GEN_sn,NB_sn],sv_4(v(VERB),SN),
    [ sv_4, GEN_sn, NB_sn, PERS, VERB,[MODIF]]) :-
    fronttoken(REST_e,VERB,REST_v),
    acces_lexiq(VERB,v,[GEN_sn,NB_sn]),
    sn_analyse(REST_v,REST,[Gsn,Nsn],SN,MODIF),
    !.
```

VII.4.2.2. Quelques résultats du programme :

```
Files      Edit      Run      Compile      Options      Setup
===== Dialog =====
Faites entrez votre phrase en français :
      >> un homme mange une pomme dans la rue
P:
  SN:
  sn_3(d("un"), [], n_com("homme"), [], [])
  SV:
  sv_4(v("mange"), sn_3(d("une"), [], n_com("pomme"), [], [sp(p("dans"), sn_3(d("la
"), [], n_com("rue"), [], [])]))))
STR_sn:
  [{"sn_3", "mas", "sin", "_", "un homme"}, []]
STR_sv:
  [{"sv_4", "mas", "sin", "mange"}, [{"sn_3", "fem", "sin", "_", " une pomme"}, {"sp", "fe
m", "sin", "_", " dans la rue"}]

Press the SPACE bar

F2-Save  F3-Load  F6-Switch  F9-Compile  Alt-X-Exit
```

```
Files      Edit      Run      Compile      Options      Setup
===== Dialog =====
Faites entrez votre phrase en français :
      >> jean est un homme intelligent
P:
  SN:
  sn_1(n_prp("jean"), [])
  SV:
  sv_4(v("est"), sn_3(d("un"), [], n_com("homme"), [a("intelligent")], []))
STR_sn:
  [{"sn_1", "mas", "sin", "jean"}, []]
STR_sv:
  [{"sv_4", "mas", "sin", "est"}, [{"sn_3", "mas", "sin", "_", " un homme intelligent"},
[]]

Press the SPACE bar

F2-Save  F3-Load  F6-Switch  F9-Compile  Alt-X-Exit
```

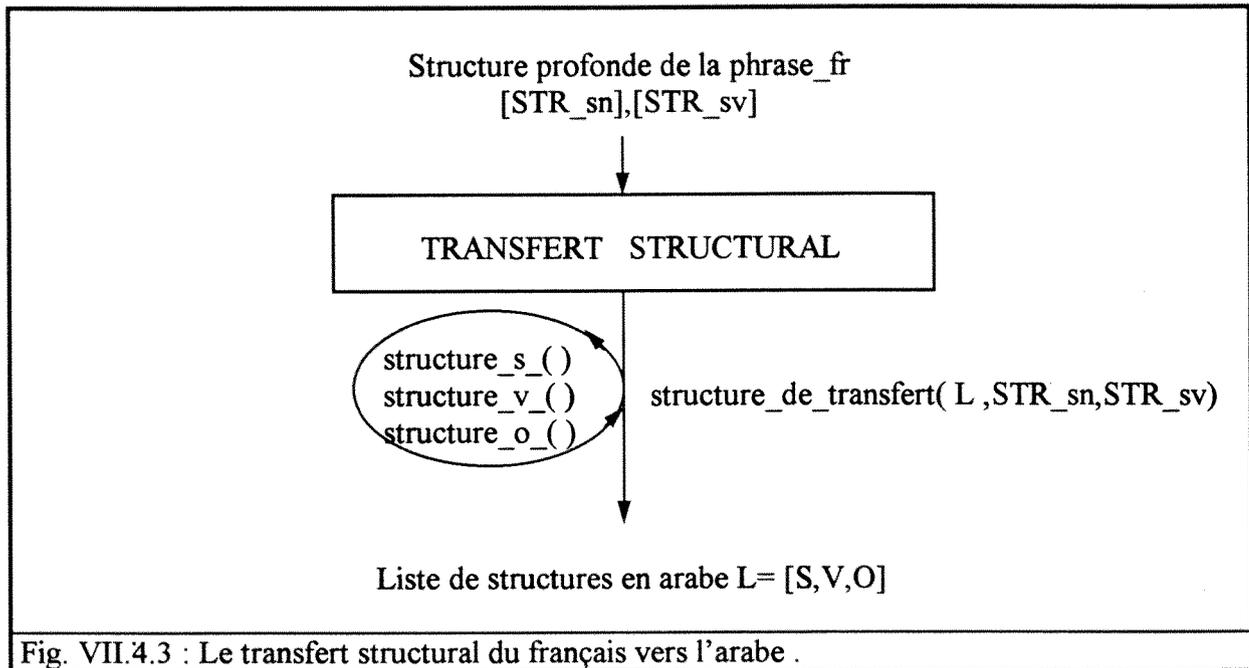
```
Files      Edit      Run      Compile      Options      Setup
===== Dialog =====
Faites entrez votre phrase en français :
      >> la belle femme intelligente est nadia
P:
  SN:
  sn_3(d("la"),[a("belle")],n_com("femme"),[a("intelligente")],[])
  SV:
  sv_4(v("est"),sn_1(n_prp("nadia"),[]))
STR_sn:
  [{"sn_3","fem","sin",_,"la belle femme intelligente"},[]]
STR_sv:
  [{"sv_4","fem","sin","est"},["sn_1","fem","sin","nadia"],[]]
Press the SPACE bar
```

F2-Save F3-Load F6-Switch F9-Compile Alt-X-Exit

```
Files      Edit      Run      Compile      Options      Setup
===== Dialog =====
Faites entrez votre phrase en français :
      >> un enfant dans la rue avec un chien mange une pomme
P:
  SN:
  sn_3(d("un"),[],n_com("enfant"),[],[sp(p("dans"),sn_3(d("la"),[],n_com("rue
"),[],[sp(p("avec"),sn_3(d("un"),[],n_com("chien"),[],[]))]))))
  SV:
  sv_4(v("mange"),sn_3(d("une"),[],n_com("pomme"),[],[]))
STR_sn:
  [{"sn_3","mas","sin",_,"un enfant"},["sp","fem","sin",_," dans la rue avec
un chien"]]
STR_sv:
  [{"sv_4","mas","sin","mange"},["sn_3","fem","sin",_," une pomme"],[]]
Press the SPACE bar
```

F2-Save F3-Load F6-Switch F9-Compile Alt-X-Exit

VII.4.3. Le transfert structural :



VII.4.3.1 Principales règles du transfert structural :

La règle d'analyse syntaxique **p_analyse** permet de transmettre deux arguments : STR_sn et STR_sv , nécessaires à la règle **structure_de_transfert** . Ces arguments ont la structure de liste et comprennent les informations suivantes :

STRUCTURE ::= STR_sn | STR_sv .

STR_sn ::= [
 [Nom_règle_sn, Flexion_Genre, Flexion_Nbr, (Flexion_Perso), CHAÎNE_sn],
 MODIFICATEUR
].

MODIFICATEUR ::= [Nom_modif, Flexion_Genre, Flexion_Nombr, (Flexion_Personne),
 CHAÎNE_modif]
| [] .

STR_sv ::= [
 [Nom_règle_sv, Flexion_Genre, Flexion_Nbr, (Flexion_Perso), CHAÎNE_sv],
 MODIFICATEUR
].

Nom_règle_sn ∈ { sn_1, sn_2, sn_3 } .

Nom_règle_sv ∈ { sv_1, sv_2, sv_3, sv_4 } .

Nom_modif ∈ { sp, f4, w } .

Transfert de données par la règle **p_analyse** à la règle **structure_de_transfert** :

```
p_analyse( PHRASE_fr, ARBRE_p, STR_sn, STR_sv ) - (i,_o,o)
```

où : i (input) , _ (variable libre) et o (output) .

La règle **structure_de_transfert** permet de transformer chacune des structures françaises contenues dans STR_sn et STR_sv en structures syntaxiques arabes et de contrôler la cohérence entre les structures engendrées (les flexions en genre et en nombre ...).

La règle de transfert structural français-arabe :



```
structure_de_transfert( [SUJET_ar,VERBE_ar,OBJET_ar],STR_sn,STR_sv ) :-  
  structure_s_ ( [S,GEN_sn,NBR_sn,SUJET_ar] ,STR_sn),  
  structure_v_ ( [V,GEN_sn,NBR_sn,VERBE_ar] ,STR_sv),  
  structure_o_ ( [O,GEN_sp,NBR_sp,OBJET_ar] ,STR_sv).
```

Identification de la structure « structure_s_ » :

Le cas SUJET ::= Nom-propre :



```
structure_s_ ( ["sn1",GEN,NBR, Npropre_ar] ,[Type,G,N,P,SN_phrase,MODIF] ) :-  
  fronttoken(SN_phrase_fr,NPropre_fr,REST_fr),  
  structure_mot_ ( NPropre_fr,[ Npropre_ar,"f1",GEN,NBR] ),  
  structure_o_ (["sp",GEN_sp,NBR_sp,PERS_sp,SP_str],MODIF).
```

Le cas SUJET ::= DET+[a]+N+[a]+[SP]:

```
structure_s_ ( ["sn3",GEN,NBR, SN_phrase_ar],[Type,G,N,P,SN_phrase,MODIF] ) :-  
  fronttoken(SN_phrase,DET_fr,REST_d),  
  structure_mot_ ( DET_fr, [ DET_ar,"d",GEN_d,NBR_d] ),  
  
  option_adj( ["f4",GEN_a,NBR_a,ADJ_ar1],REST_d,REST_a ),  
  
  fronttoken(REST_a,NOM_com,REST_n),  
  structure_mot_ ( NOM_com, [ NOM_ar,"f3",GEN,NBR] ),  
  concat(DET_ar,NOM_ar, SN1_dn),  
  
  option_adj( ["f4",GEN_a2,NBR_a2,ADJ_ar2],REST_n,REST_a2 ),  
  
  concat(SN1_dn," ", SN2_dn),  
  concat(ADJ_ar1,ADJ_ar2,ADJ_ar),  
  concat(SN2_dn,ADJ_ar, SN_dn),  
  
  structure_o_ ( ["sp",GEN_sp,NBR_sp,Prep_ar] ,MODIF),  
  concat(SN_dn," ", SN1_ar),  
  concat(SN1_ar,Prep_ar, SN_phrase_ar), !.
```

Identification de la structure « structure_v_ » :

VERBE ::= verbe :



```
structure_v_ ( ["v",GEN,NBR,VERBE], [Type,G,N,P,SV_phrase,[]]) :-  
    structure_mot_ ( SV_phrase, [ VERBE,"v",GEN,NBR] ).
```

Identification de la structure « structure_o_ » :

Le cas OBJET ::= SP(preposition+SN) :



```
structure_o_ ( ["sp",GEN,NBR,SP_phrase_ar] , , [Type,G,N,P,SP_phrase,MODIF]) :-  
    fronttoken(SP_phrase_fr,PREP_fr,REST_p),  
    structure_mot_ ( PREP_fr, [ Prep_ar,"p",GEN_p,NBR_p] ),  
  
    structure_s_ ( [ SN ,GEN,NBR, SN_phrase_ar] ,MODIF),  
    concat(Prep_ar," ",SP1_ar),  
    concat(SP1_ar,SN_phrase_ar,SP_phrase_ar),  
    !.
```

Identification de la structure « structure_mot_ » :



```
structure_mot_ ( Mot_fr, [Mot_ar,Cat,Gen,Nbr] ):-  
    acces_lexiq_ar(Mot_fr, [Mot_ar,Cat,Gen,Nbr,PERS]).
```

VII.4.3.1 Quelques résultats du programme :

```
Files      Edit      Run      Compile      Options      Setup
----- Dialog -----
Rappel de votre phrase en français :
    >> un homme mange une pomme

Structure VSO:

Structure de transfert:
S = RaJuL
V = Ya'KuLu(+Bi)
O = TuFFaAHa&

Press the SPACE bar

F2-Save  F3-Load  F6-Switch  F9-Compile                               Alt-X-Exit
```

```
Files      Edit      Run      Compile      Options      Setup
----- Dialog -----
Rappel de votre phrase en français :
    >> un enfant dans la rue mange une pomme

Structure VSO:

Structure de transfert:
S = TiFL    FiY 'aL-TaRiYQ
V = Ya'KuLu(+Bi)
O = TuFFaAHa&

Press the SPACE bar

F2-Save  F3-Load  F6-Switch  F9-Compile                               Alt-X-Exit
```

Files Edit Run Compile Options Setup

Dialog

Rappel de votre phrase en français :
>> une belle femme marche dans la rue avec un homme

Structure VSO:

Structure de transfert:

S = AMRa'a& JaMiYLa&
V = TaMSiY
O = FiY 'aL-TaRiYQ Maca RaJuL

Press the SPACE bar

F2-Save F3-Load F6-Switch F9-Compile

Alt-X-Exit

Files Edit Run Compile Options Setup

Dialog

Rappel de votre phrase en français :
>> nadia aime les enfants

Structure VSO:

Structure de transfert:

S = NaADiYa&
V = TuHiBBu
O = 'aL-'aTFaAL

Press the SPACE bar

F2-Save F3-Load F6-Switch F9-Compile

Alt-X-Exit

VII.4.4. La génération :

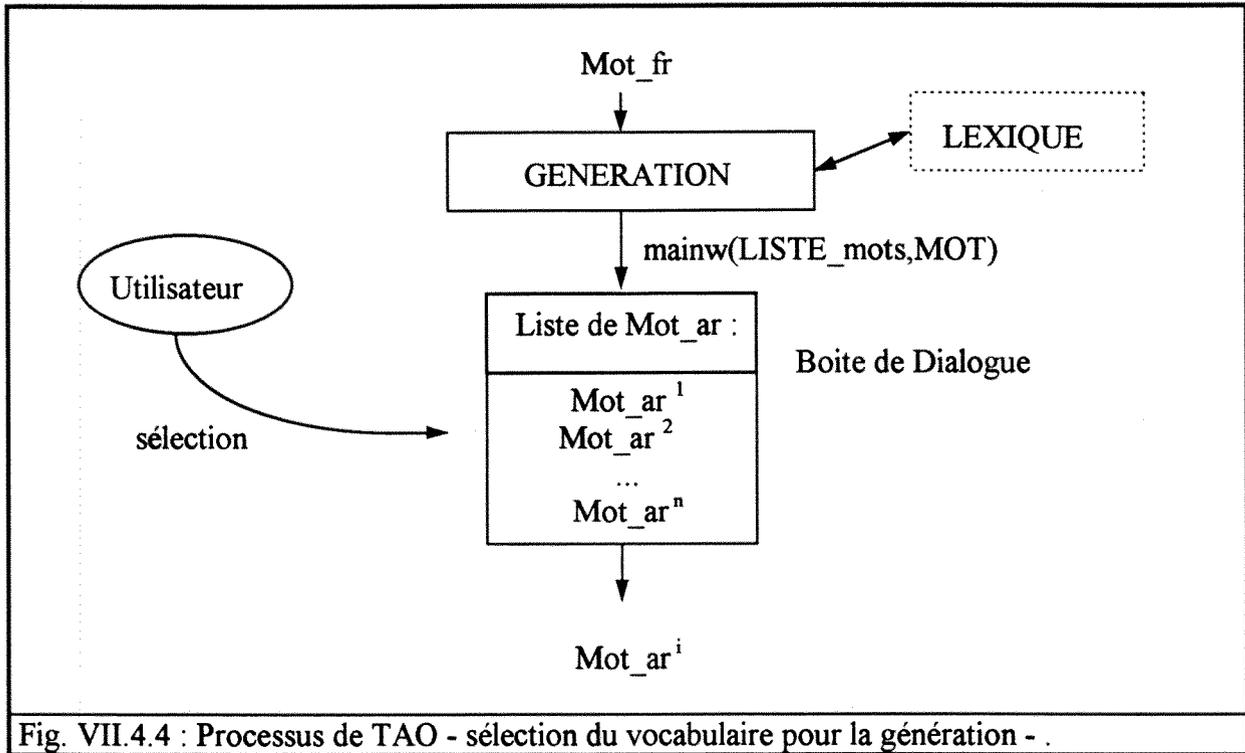


Fig. VII.4.4 : Processus de TAO - sélection du vocabulaire pour la génération - .

VII.4.4.1. Principales règles de la génération :

La génération de la phrase arabe s'effectue après le traitement de la règle `structure_de_transfert`. Celle-ci donne, en sortie, une structure de liste composée par trois éléments: SUJET, VERBE et OBJET. Ces éléments constituent des structures syntaxiques pour la phrase en arabe (le groupe sujet, le verbe principal et le groupe complément). Ceux-ci par arrangement permettent de générer la phrase nominale ou verbale en arabe (*JuMLa& ?iSMiYYa& ?aW Fi^cLiYYa&*) selon le choix de l'utilisateur.

Génération de la phrase en arabe :

PHRASE ::= phrase_verbale_ar



```

generation_VSO_ar([SUJET, VERBE, OBJET], PHRASE_ar) :-
    write("\nStructure VSO:\n"),
    structure_de_transfert([SUJET,VERBE,OBJET],STR_sn,STR_sv),
    concat(VERBE," ",P1),
    concat(P1,SUJET,P2),
    concat(P2," ",P3),
    concat(P3,OBJET,PHRASE_ar)
  
```

PHRASE ::= phrase_verbale_ar



```
generation_SVO_ar([SUJET, VERBE, OBJET], PHRASE_ar) :-  
    write("\nStructure SVO:\n"),  
    structure_de_transfert([SUJET, VERBE, OBJET], STR_sn, STR_sv),  
    concat(SUJET, " ", P1),  
    concat(P1, VERBE, P2),  
    concat(P2, " ", P3),  
    concat(P3, OBJET, PHRASE_ar) .
```

Le choix du lexique de traduction s'effectue en fonction de la règle **mainw** qui ouvre une boîte de dialogue, affiche à l'utilisateur la liste des traductions en arabe du mot français traité et attend la validation d'une *ligne* (= mot). La sélection d'un mot dans la liste permet sa prise en compte pour le reste du traitement et le passage au mot suivant à traiter .

Selection dans le lexique arabe :

```
mainw (LISTE_X , MOT_user) :-  
    menu (  
        5,27,7,7,  
        LISTE_X,  
        " MENU-Liste des mots arabes-",  
        1,  
        MOT_user  
    )
```

VII.4.4.2. Quelques résultats du programme :

PRINCIPAL MENU

Faites entrer votre MOT en français :
>> vous

CHOICE MENU

- ?aNTa
- ?aNTi
- ?aNTuM
- ?aNTuNNa
- ?aNTuMaA

<1>

GENERATION Morphologique : <1> <2> <3> <4>

PRINCIPAL MENU

Faites entrer votre MOT en français :
>> vous

CHOICE MENU

- ?aNTa
- ?aNTi
- ?aNTuM
- ?aNTuNNa
- ?aNTuMaA

MENU RESULTAT

Mot sélectionné :?aNTi
Press the SPACE bar

<2>

PRINCIPAL MENU

Faites entrer votre MOT en français :
>> homme

CHOICE MENU

RaJuL
?iNSa?N

MENU RESULTAT

Mot sélectionné :RaJuL
Press the SPACE bar

<3>

PRINCIPAL MENU

Faites entrer votre MOT en français :
>> intelligente

CHOICE MENU

DaKiYa&
FaTiNa&

MENU RESULTAT

Mot sélectionné :FaTiNa&
Press the SPACE bar

<4>

GENERATION_MENU

Votre phrase en français :
>> un homme mange une pomme

CHOICE MENU
STRCURE SVO
STRUCTURE VSO

GENERATION : { phrase nominale (SVO)
 { phrase verbale (VSO)

GENERATION_MENU

Votre phrase en français :
>> un homme mange une pomme

Structure VSO:

PHRASE_VSO : Ya?KuLu(+Bi) RaJuL TuFFaAHa&

Structure SVO:

PHRASE_SVO : RaJuL Ya?KuLu(+Bi) TuFFaAHa&

Press the SPACE bar

GENERATION_MENU

Votre phrase en français :

>> un homme mange une pomme

Structure VSO:

PHRASE_VSO : Ya?KuLu(+Bi) RaJuL TuFFaAHa&

Press the SPACE bar

GENERATION_MENU

Votre phrase en français :

>> la belle femme mange un fruit

Structure VSO:

PHRASE_VSO : Ta?KuLu 'aL-AMRa'a& 'aL-JaMiYLa& FaAKiHa&

Press the SPACE bar

Remarques sur notre prototype :

- 1- Un système de TAO et particulièrement le prototype réalisé nécessite une grande quantité d'informations . Ces informations sont essentiellement composées :
 - de *données linguistiques* : des grammaires et des dictionnaires automatiques.
 - de *programmes* .
- 2- La réalisation de tels systèmes informatiques nécessite des compétences en divers domaines : linguistique, lexicographique, informatique et logico-mathématiques. Il est difficile sinon impossible d'exiger de quiconque une compétence dans tous ces domaines . Des méthodes et des outils logiciels sont indispensables pour une organisation modulaire où chacun contribue par ses connaissances propres à la réalisation de ce système.
- 3- L'objectif de notre réalisation informatique est de développer une maquette pédagogique (prototype) permettant l'évolution vers un système de TAO opérationnel.
- 4- L'outil logiciel (langage de programmation) est de grande importance pour l'implémentation des concepts théoriques du langage naturel et pour l'adaptation du modèle théorique au système réalisé .
Le langage PROLOG est un outil adéquat au traitement du langage naturel et particulièrement à la TAO . Turbo PROLOG 2.0 (INC. Borland) qui a servi comme outil logiciel à notre réalisation informatique est un outil pédagogique dont il est nécessaire d'acquérir une nouvelle version adaptée au développement logiciel des applications du langage naturel et la TAO .

CONCLUSION

Le domaine de la TA ou la TAO est un domaine difficile, l'élaboration de tel système met systématiquement en jeu les difficultés liées au TALN.

Le travail que nous avons présenté s'inscrit dans le cadre de recherche sur l'informatisation de la langue arabe, l'un des axes de recherches au CERSI (Centre d'Etudes et de Recherches en Sciences de l'Information) à l'ENSSIB en accord avec le Centre de recherche en Terminologie et Traduction (CRTT) à Lyon 2 et la collaboration de l'Institut de Recherche en Science Informatique et des télécommunications (IRSIT) en Tunisie.

c'est dans ce contexte que nous avons effectué le présent travail pour la représentation des langages:

- ⇒ conception d'un modèle pour un dictionnaire électronique bilingue,
- ⇒ choix d'un formalisme grammatical pour l'analyse de la langue source (français),
- ⇒ élaborations des structures de transfert vers la langue cible (arabe),
- ⇒ possibilité d'intégration d'outils développés sur la base de données lexicales en arabe,
- ⇒ modélisation de l'architecture de notre prototype de TAO ,
- ⇒ rélisation informatique d'un prototype de TAO du français vers l'arabe .

Les systèmes de traitement informatique des langues naturelles fonctionnent grâce à un certain nombre d'informations linguistiques mises à leur disposition. Parmi ces informations, certaines sont attachées à un mot et décrivent la façon dont il intervient dans les phénomènes linguistiques: ce sont les données lexicales.

Notre premier soucis dans ce travail était la conception d'une base de données lexicales bilingue français-arabe pour notre prototype de TAO. Notre conception peut être considérée comme un dictionnaire électronique restreint et la démarche peut être confiée à un opérateur humain, spécialisé en terminologie pour mettre au point une véritable base de données lexicales.

Il ne fait pas de doute que l'analyse syntaxique est nécessaire à l'interprétation des énoncés, et l'intérêt des formalismes syntaxiques pour le TALN.

Les représentations syntaxiques nécessaires à l'interprétation n'ont pas forcément besoin d'être très sophistiquées: le découpage peut se limiter à l'identification des principaux syntagmes susceptibles de recevoir un rôle sémantique [ABEILLÉ 95] (les groupes nominaux qui correspondent à des arguments, les phrases ou les subordinées et certains groupes prépositionnels).

Le formalisme DCG nous a permis des analyses linguistiques originales dans notre travail avec les facilités d'implémentation de la grammaire et la réécriture des règles de transfert syntaxiques du français vers l'arabe.

Nous avons montré, à travers un modèle comparatif celui de ML-TASC (Système de conversion de formalismes vers la traduction multilingue [BOUALEM 93]) dans le cadre du projet Malin à l'INRIA (Sophia Antipolis - Nice), des critères d'amélioration des règles de transfert syntaxique du français vers l'arabe .

Aussi, l'intégration des outils développés, sur la base de données lexicales (relatives aux bases nominales et verbales) en arabe et le conjugueur des verbes arabes, permettent l'évolution de notre modèle prototype de TAO vers un système de TAO français-arabe appliqué à un domaine particulier (contexte scientifique ou technique).

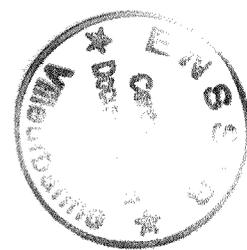
Le domaine du TALN de manière générale et le domaine de la TAO en particulier sont complexes, cela n'est qu'une conséquence de la complexité même du LN.

Notre modèle pour la conception du prototype de TAO du français vers l'arabe, appliqué à un corpus restreint, présente les caractéristiques souhaitées des modèles de formalisation et de conversion des langages naturels.

Il nous a permis d'étudier les différentes structures syntaxiques du français en adaptant les aspects du modèle linguistique développé par le groupe SYDO pour l'identification des syntagmes nominaux et l'introduction de nouvelles règles pour l'analyse de notre corpus.

Nous pensons que les traitements mettant en jeu le LN ou des sous-ensembles de celui-ci, doivent être d'abord positionnés dans leur contexte d'application. L'histoire de la TA / TAO et les résultats de certains systèmes dans ce domaine ont montré que, dès qu'il s'agit de couvrir des domaines d'application bien définis, restreints et contrôlés les traitements deviennent satisfaisants aux attentes des professionnels de la traduction .

Une étroite collaboration dans ce genre de projet, entre linguistes, informaticiens et experts du domaine permet l'écriture d'algorithmes de traitements plus performants en matière de connaissances linguistiques et les particularités du domaine, rendant ainsi possible l'automatisation du LN dans le processus de la traduction (TA ou TAO).



BIBLIOGRAPHIE

[ABBOU 95a]:

ABBOU André, *Le Japon: Traduction et autoroutes d'information*, La tribune des industries de la langue et de l'information électronique N° 17-18-19, OFIL, 1995. p.76-83.

[ABBOU 95b]:

ABBOU André, *La théorie du procédé constructif*, La tribune des industries de la langue et de l'information électronique N° 17-18-19, OFIL, 1995. p.35-36.

[ABBOU 95c]:

ABBOU André, *Traduction Assistée par Ordinateur - Actes du séminaire international (Paris, Mars 1988)*, Ed. DAICADIF, 1989. 234 p.

[AL-ABED 92]:

AL-ABED Lamia, *Traduction Assistée par Ordinateur de l'anglais vers l'arabe*, Rapport de synthèse sur le laboratoire TAO de 1990-1992 - Département Arabisation & Communication Homme-Machine, IRSIT Tunisie Déc.1992, p.10 .

[AL-ABED 90]:

AL-ABED Lamia, *Elaboration d'un système de traduction de phrases de l'anglais vers l'arabe*, Mémoire d'ingénieur concepteur en informatique, FST, 1990, p.205 .

[AL-CHAY 88]:

ABU AL-CHAY Najim, *Un système expert pour l'analyse et la synthèse des verbes arabes dans un cadre d'enseignement assistée par ordinateur*, Thèse de doctorat d'Etat, Université Claude Bernard , Lyon 1, 1988. 139 p.

[ALLEN 95]:

ALLEN James , *Natural language understanding* ,2nd Ed. , The Benjamin / Cummings publishing Company INC, 1995 . 654 p.

[AMANO 89]:

AMANO S. (& HIRAKAWA H. , NOGAMI H. , KUMANO A.), *The TOSHIBA machine translation*, Future Computing System vol.2 N°3, Oxford University Press and Maruzen Company Limited, 1989. p. 227-246.

[ARCHIBALD 91]:

ARCHIBALD Michiels, *Traitement du langage naturel et Prolog*, HERMES Ed. , 1991.

[ARNOLD 87]:

ARNOLD Dong (& DES TOMBE Louis) , *Basic theory and methodology in EUROTRA*, Machine translation theoretical and methodological issues, University Press of Combridge, 1987. p. 114-134.

[BALLON 94]:

BALLON S. (& SARRON C.), *Industries de la langue - Centre Français du commerce extérieur - Direction des industries et services*, Ed. du CFCE, Mai 1994. 76 p.

[BIERIN 90]:

BIERIN E. (& MOULIN A. , PICHault F.), *Les industries de la langue: un marché en devenir*, OWIL, 1990. 98 p.

[BILANGE 92]:

BILANGE Eric, *Dialogue personne-machine - Modélisation et réalisation informatique*, Edition HERMES, 1992. 192 p.

[BOLC 87]:

BOLC Léonard, *Natural language parsing systems*, Springer-Verlag: *Symbolic Computation A.I, Germany*, 1987. p. 60-110.

[BOUALEM 93]:

BOUALEM Abdel-Malek, *ML-TASC Système de conversion de formalismes de langages techniques et scientifiques dans un environnement à syntaxe contrôlée et à raisonnement limité- Traduction Automatique Multilingue*, Thèse de Docteur en Science, Université Sophia Antipolis, 25 Nov. 1993. 262 p.

[BOULICAUT 92]:

BOULICAUT J-F , *Vers une programmation grammaticale à large spectre - Application à la reconstruction de programmes avec le métacompilateur STARLET/GL*, Thèse de Doctorat en Science, INSA Lyon, 1992. 319 p.

[BOURBEAU 85]:

BOURBEAU Laurent (& ISABELLE Pierre), *TAUM-AVIATION: Its Technical Features and some experimental results*, Computational Linguistics Vol: 11 N° 1, 01-03 1985. p. 18-27.

[CARBONELL 87]:

CARBONELL Jaime G. (& TOMITA Masaru), *Knowledge-based MT the CMU approach, Machine translation theoretical and methodological issues*, University Press of Cambridge, 1987. p. 68-89.

[CHURCH 93]:

CHURCH K. W. (& HOVY E. H.), *Good applications for CRUMMY machine translation*, *Machine Translation 8*, Kluwer Academic Publishers Netherlands, 1993. p. 239-258.

[CULLINFORD 87]:

CULLINFORD Richard E. (& ONYSHKEVYCH Boyan A.), *An experiment in lexicon-driven MT , Machine translation theoretical and methodological issues*: University Press of Cambridge, 1987. p. 278-301.

[COHEN 70]:

COHEN D., *Essai d'une analyse automatique de l'arabe (1961-70)*, in *Etudes de linguistiques sémitique et arabe*, Paris: Mouton, 1970.

[DICHY 95]:

DICHY Joseph, Pour une lexicomatique de l'arabe: l'unité lexicale simple et l'inventaire fini des spécificateurs du domaine du mot, A paraître in *Lexicomaitique et dictionnaire* - Actes des journées de Lyon -, sept. 95.

[DICHY 90]:

DICHY Joseph, *L'écriture dans la représentation de la langue: la lettre et le mot en arabe*, Thèse de Doctorat d'Etat, Lyon 2 - Université Lumière, 1990, 660 p.

[DICHY 89]:

DICHY Joseph et **HASSOUN M.O.**, *Simulation des modèles linguistiques et Enseignement Assistée par Ordinateur de l'arabe - Travaux SAMIA I*, Paris: Conseil International de la Langue Française, 1989.

[DORR 93]:

DORR Bonnie J., Interlingual machine translation a parameterized approach, *A.I N°63*, Elsevier Science Publishers, 1993. p.429-492.

[ELLUL 87]:

ELLUL COMMARC Aline, *Interface coopérative en L.N*, Thèse de Doctorat d'Etat: I.A, ENST Paris, 1987. 162 p.

[FALZON 89]:

FALZON Pierre, *Ergonomie Cognitive du Dialogue*, Presse universitaire de Grenoble, 1989. 175 p.

[FARWELL 93]:

FARWELL D. (& GUTHRIE L. , WILKS Y.), Automatically creating lexical entries for ULTRA a multilingual MT system, *Machine Translation 8*, Kluwer Academic Publishers Netherlands, 1993. p.127-145.

[FORD 91]:

FORD Nigel, *Expert Systems and A.I*, Library Association Publishing London, 1991. 277 p.

[GAL 89]:

GAL A. (& LAPALME G. , DIZIER P-St), *Analyse et synthèse d'un langage - PROLOG*, Ed. EYROLLES, 1989. 135 p.

[GAZDAR 89]:

GAZDAR G. (& MELLISH C.), *Natural language processing in PROLOG*, Ed. Addison-Wesley Pub Company, 1989. 386 p.

[GROSS 95]:

GROSS Gaston, Une sémantique nouvelle de la TA- les classes d'objets, *La tribune des industries de la langue et de l'information électronique N° 17-18-19*, OFIL, 1995. p. 16-19.

[HASSOUN 87]:

HASSOUN Mohamed, *Système d'analyse morphologique automatique de la langue arabe*, Thèse de Doctorat d'Etat, Université Claude Bernard, Lyon 1, 1987. 142 p.

[HATON 91]:

HATON J-P (& al.), *Le raisonnement en I.A.*, InterEditions, 1991. 480 p.

[JOHNSON 87]:

JOHNSON Roderick L. (& **WHITELOCK Peter**), MT as an Expert task, *Machine translation theoretical and methodological issues*, University Press of Combridge, 1987. p. 136-144.

[KABBAJ 91]:

KABBAJ A., *Intelligence artificielle en LISP et PROLOG*, Ed. MASSON, 1991. p. 257-322.

[KITTREDGE 87]:

KITTREDGE Richard I., The significance of sublanguage for automatic translation, *Machine translation theoretical and methodological issues*, University Press of Combridge, 1987. p. 59-67.

[KOUNIALI 93]:

KOUNIALI Samy Habib, *Désambiguïsation de groupes nominaux complexes en conformité avec les connaissances du domaine (Application à la traduction Automatique)*, Thèse de Doctorat d'Etat, INP de Lorraine, 23 septembre 1993. 185 p.

[LANGE 95]:

LANGE J-M, La TAO des dernières aux prochaines années, *La tribune des industries de la langue et de l'information électronique N° 17-18-19*, OFIL, 1995. p. 20-24.

[LAROUK 94]:

LAROUK OMAR, *Extraction des connaissances à partir de documents textuels: traitement automatique de la coordination (connecteurs et ponctuation)*, Thèse de Doctorat d'Etat, UCB Lyon1, Fev. 1994.

[LUZZATI 95]:

LUZZATI Daniel, *Dialogue Verbal Homme-machine*, Ed. MASSON, 1995. 197 p.

[LYTINEN 87]:

LYTINEN Steven L., Integrating syntax and semantics, *Machine translation theoretical and methodological issues*, University Press of Combridge, 1987. p. 302-316.

[MCDONALD 87]:

MCDONALD David D., Natural language generation: Complexities and techniques, *Machine translation theoretical and methodological issues*, University Press of Combridge, 1987. p. 192-224.

[MELBY 87]:

MELBY Alan, On human-machine interaction in translation, *Machine translation theoretical and methodological issues*, University Press of Combridge, 1987. p. 145-154.

[METZGER 88]:

METZGER J-Paul, *Syntagmes nominaux et information textuelle- Reconnaissance automatique et représentation*, Thèse de Doctorat d'Etat, Université Claude Bernard, Lyon 1, 1988.324 p.

[MILLER 90]:

MILLER Philip (& **TORRIS Thérèse**), *Formalismes syntaxiques pour le traitement automatique du langage naturel*, Edition HERMES, 1990. 359 p.

[MONEIMNE 89]:

MONEIMNE Walid, *TAO vers l'arabe - Spécification d'une génération standard de l'arabe - Réalisation d'un prototype anglais-arabe à partir d'un analyseur existant*, Thèse de Doctorat d'Etat, Université Joseph Fourier, Grenoble 1, 14 juin 1989. 244 p.

[NAGAO 87]:

NAGAO Makoto, Role of structural transformation in a MT system, *Machine translation theoretical and methodological issues*, University Press of Combridge, 1987. p. 262-277.

[NIRENBURG 87a]:

NIRENBURG Sergei (& **RASKIN V.**, **TUCKER Allen B.**), The structure of interlingua in Translator, *Machine translation theoretical and methodological issues*, University Press of Combridge, 1987. p. 90-113.

[NIRENBURG 87b]:

NIRENBURG Sergei, Knowledge and choices in MT, *Machine translation theoretical and methodological issues*, University Press of Combridge, 1987. p. 1-21.

[PEREIRA 80]:

PEREIRA Fernando C.N. & **WARREN David H.D.**, Definite Clause Grammars for language analysis- A survey of the formalism and comparison with Augmented Transition Networks, *A.I n° 13*, 1980, p.213-278.

[PITRAT 85]:

PITRAT Jacques, *Textes Ordinateurs et Compréhension*, Ed. EYROLLES, 1985. 201 p.

[RADWAN 94]:

RADWAN Khaled, *Vers l'accès multilingue en langage naturel aux bases de données textuelles*, Thèse de Doctorat d'Etat, Université Paris-Sud, Centre ORSAY, 11/02/1994.243 p.

[RASKIN 87]:

RASKIN Victor, Linguistics and natural language processing, *Machine translation theoretical and methodological issues*, University Press of Combridge, 1987. p. 42-58.

[RICHARD 95]:

RICHARD Jean-Luc, *STELO LINGVA: un système de TA centré sur l'Espéranto - Phase0: la traduction Espéranto-Français*, Mémoire de DEA, INSA de Lyon, Sep. 1995. 58 p.

[ROMAN 90]:

ROMAN André, *Grammaire de l'Arabe*, Paris: P.U.F. (coll. 'Que sais-je?'), 1990.

[SABAH 90a]:

SABAH Gérard, *L'intelligence artificielle et le langage - Volume 1: représentation des connaissances*, Ed. HERMES, 1990. 352 p.

[SABAH 90b]:

SABAH Gérard, *L'intelligence Artificielle et le langage Volume 2: Processus de compréhension*, Edition HERMES, 1990. 411 p.

[SHIEBERT 88]:

SHIEBERT Y. , *Lexicalized Grammars: on application to tree adjoining grammars*, acte de colloque: 12th colling Budapest, 1988.

[SIDHOM 94]:

SIDHOM Sahbi, *Le système TORJOMAN: traducteur assistée par ordinateur de l'anglais vers l'arabe*, Nov.1993-94, IRSIT Tunisie, p. 1-12 , Rapport d'activités sur le système TORJOMAN.

[SOMERS 93]:

SOMERS Harold L. , Current research in MT, *M.T* N°7, Kluwer Academic Publishers Netherlands, 1993. p. 231-246.

[TUCKER 87]:

TUCKER Allen B., Current strategies in MT research and development, *Machine translation theoretical and methodological issues*, University Press of Combridge, 1987. p. 22-41.

[VAUQUOIS 75]:

VAUQUOIS Bernard, *La traduction automatique à Grenoble*, DUNOD, 1975. 179 p.

[VEGA 95]:

VEGA J., AlethTrad outil et service, *La tribune des industries de la langue et de l'information électronique* N° 17-18-19, OFIL, 1995. p. 24-27.

[VEILLON 70]:

VEILLON Gerard, *Modèles et algorithmiques pour la traduction automatique*, Thèse de Doctorat, Faculté des sciences de l'université de Grenoble, 1970. 156 p.

[WHITE 87]:

WHITE John S., The research environnement in the METAL project, *Machine translation theoretical and methodological issues*, University Press of Combridge, 1987. p. 225-246.

[WILKS 94]:

WILKS Y. , Developpments in MT research in the US, *Aslib Proceedings vol.46 N°4*, April 1994. p.111-116.

[WILLS 93]:

WILLS wolfram , Basic concepts of MT, *Meta XXXVII.3*,1993. p. 403-413.

[ZAHARIN 88]:

ZAHARIN Yossoff , Towards an analyser (parser) in a machine translation system based on ideas from expert systems, *Comput. Intelligence 4*, 1988. p. 180-191.

