République Algérienne Démocratique et Populaire
Ministère de l'enseignement Supérieur et de la Recherche Scientifique
Université Blida 1

Faculté des sciences
Département d'Informatique

Domaine Mathématiques et Informatique
Spécialité : Traitement Automatique du Langage - TAL

Mémoire de Master

Thème

Conception et réalisation d'un outil d'aide à l'analyse grammaticale des textes arabes (الإعراب) (Niveau des textes : CEM)

Sujet proposé par :
Mr Amine Cherif Zahar

Présenté par :
Kharchi Rochdi
Fassih Anes

Soutenu le : 24 / 12 / 2020
Devant le Jury composé de :

Melle	Ykhlef H.	Présidente
Mme	Mezzi M.	Examinatrice
Mr	Amine Cherif Zahar	Promoteur

Résumé :

Notre travail consiste en la proposition d'un outil d'aide à l'identification du rôle grammatical des différentes composantes d'une phrase en arabe standard, autrement dit, associer aux différents mots qui composent un texte, la description grammaticale qui leurs convient tout en marquant leur nature et les rapports qui les unissent.

La solution que nous proposons se résume à la mise en place :

- D'une analyse morphologique qui permet de détecter les possibilités de catégories grammaticales d'un mot en se basant sur son schème et sa voyellation, ou sur des ressources s'il ne s'agit pas d'un mot flexionnel.

- D'une analyse grammaticale qui se base sur des règles grammaticales qui indiquent l'agencement des entités d'une phrase et que le programme va apprendre au fur et à mesure de son utilisation.

Mots Clés ***:*** *Analyse grammaticale, rôle des constituants, Étiqueteurs symboliques, Étiqueteurs avec apprentissage, Règle grammaticales.*

Abstract

Our work consists of proposing a tool to help identify the grammatical role of the different components of a sentence in standard Arabic, in other words, to associate with the different words that make up a text, the grammatical description that suits them all by marking their nature and the relationships that unite them.

The solution we offer comes down to the implementation:

- A morphological analysis which makes it possible to detect the possibilities of grammatical categories of a word based on its schema and its vowel, or on resources if it is not an inflectional word.

- A grammatical analysis which is based on rules which indicate the arrangement of the entities of a sentence and which the program will learn as it is used.

Keywords : *Grammatical analysis, roles of constituents, Symbolic tagger, Tagger with learning, grammatical rules.*

ملخص

يتكون عملنا من اقتراح أداة للمساعدة في تحديد الدور النحوي للمكونات المختلفة للجملة في اللغة العربية الفصحى، بعبارة أخرى، ربط الكلمات المختلفة التي يتكون منها النص بالوصف النحوي الذي يناسبها من خلال تحديد طبيعتهم والعلاقات التي توحدهم.

الحل الذي نقدمه يجمع بين:

- تحليل صرفي يجعل من الممكن الكشف عن الفئات النحوية للكلمة بناءًا على وزنها وتشكيلها، أو على أساس الموارد إذا لم تكن كلمة تصريفية.

- تحليل نحوي يعتمد على القواعد النحوية والتي سيتعلمها البرنامج عند استخدامه.

الكلمات المفتاحية : *الوصف النحوي ، أدوار الكلمات ، الأوسم الرمزية ، الوسم بالتعلم ، القواعد.*

Dédicaces

C’est avec un immense plaisir Que je dédie ce travail

A mes très chers parents qui sont tout ce que j’ai de plus cher au monde, en témoignage de ma reconnaissance infinie pour leurs nombreux sacrifices.

Qu’ils trouvent en ce travail la preuve de mon éternel amour et ma reconnaissance envers eux.

Que dieu les gardes et leur procure la santé et le bonheur incha’allah.

Aussi

À ma famille, mes frères mes sœurs sans oublier mes amis.

Rochdi Kharchi

Dédicaces

Je dédie ce travail à :

A ma chère mère

« Je ne saurais te rendre l'amour que tu me donne, la bienveillance que tu as envers moi et les sacrifices que tu fais. J'espère seulement le mériter. »

A mon cher père

« Nous avons appris à parler sans faire appel aux mots, mais laisse-moi te dire Je t'aime parce qu'il le faut bien. »

Qu'Allah vous garde.

A mes frères

« Même si la vie nous éloigne, nous resterons unis. Je vous aime. »

A ma chère famille qui me manque.

A mes amis car sans eux je ne serais pas ce que je suis.

A mes professeurs car ils m'ont transmis ce qu'ils avaient de plus précieux, et c'est ainsi que je les honore.

A tous ceux qui ont contribués de loin ou de près à l'élaboration de ce modeste travail.

Anes Fassih

Remerciement

En premier lieu, nous remercions Dieu, notre créateur, qui nous a donné la force, la volonté et le courage pour réaliser ce travail.

Aussi, nous tenons à remercier infiniment :

Nos chers parents pour leurs soutiens tout au long de notre parcours

Mr. Cherif Zahar d'avoir proposé et encadré ce sujet. Nous lui exprimons notre profonde gratitude pour nous avoir fait profiter de ses connaissances, mais aussi de ses méthodes de travail, et surtout de sa rigueur scientifique. Veuillez trouver ici l'expression de notre estime et considération.

Nos remerciements aux membres du jury pour avoir accepté d'examiner notre travail, pour leur lecture attentive de notre mémoire ainsi que pour les remarques qu'ils nous adresseront lors de cette soutenance.

Notre reconnaissance va aussi à tous ceux qui ont collaboré à notre formation en particulier les enseignants du département d'Informatique, Université Blida 1, ex Saad Dahleb, aussi à nos camarades de la promotion 2019-2020.

On remercie également tous ceux qui ont participé de près ou de loin à élaborer ce travail.

Table des matières

Introduction générale 15
1. Contexte global : 1
2. Problématique : 2
3. Objectif : 2
4. Organisation du mémoire 3
CHAPITRE 1 : Généralités sur l'Arabe standard et son traitement automatique 4
1. Introduction 5
2. La langue arabe 6
3. Traitement automatique de la langue Arabe (TALA) 7
3.1 Les caractéristiques de la langue arabe 8
3.2 Les éléments essentiels de la morphologie arabe 10
3.2.1 Le schème «الوزن » 10
3.2.2 La racine «الجذر » 12
3.2.3 Les affixes 12
3.3 Structure d'un mot 15
3.3.1 Catégories des mots 17
3.3.1.1 Morphologie Verbale 18
3.3.1.2 La Morphologie Nominale 20
3.3.1.3 Les particules 25
4. Les particularités de la langue arabe 27
4.1 Voyellation 27
4.2 Agglutination 28
4.3 Irrégularité de l'ordre des mots dans la phrase 28
4.4 Absence de ponctuation régulière 29
4.5 Détection de racine 29
4.6 Le caractère '_' 30
4.7 Mots étrangers translittérés en arabe 30
5. Difficultés du traitement automatique de la langue arabe 31
5.1 La segmentation de textes 31
5.2 L'analyse morphologique 31
5.3 L'orthographe de la langue arabe 32
5.4 L'étiquetage grammatical 33
5.5 L'analyse syntaxique 34
6. Conclusion 34

CHAPITRE 2 : Travaux antérieurs.....35

1. Introduction 36

2. L'étiquetage grammatical 37

3. Méthodes d'étiquetage 37

3.1 Etiqueteurs symboliques 37

3.2 Etiqueteurs avec apprentissage 38

3.2.1. Etiquetage non supervisé 39

3.2.2. Etiquetage supervisé 40

4. Catégorisation des étiqueteurs avec apprentissage 41

4.1. Etiquetage à base de règles 41

4.2. Etiquetage statistique 41

4.3. Etiquetage à base de réseaux de neurones 43

5. Analyse sur les étiqueteurs existants 45

5.1. AraMorph 45

5.2. L'étiqueteur ASVM 45

5.3. L'analyseur AlKhalil 46

5.4. L'étiqueteur APT de Khoja 47

5.5. Etiqueteur de XEROX 47

5.6. L'analyseur de Sakhr 48

5.7. L'analyseur de Buckwalter 48

6. Comparaison théorique entre les étiqueteurs morphologiques : 49

7. Conclusion 50

CHAPITRE 3 : Conception et modélisation de la solution 51

1. Introduction : 52

2. Programme ARIBLI 52

2.1 Description fonctionnelle : 52

2.2 Description de l'analyse morphologique 56

2.2.1 Découpage 57

2.2.2 Détection des entités spéciales 57

2.2.3 Recherche des schèmes 58

2.2.4 Recherche de la racine 60

2.2.5 Muarraf 60

2.2.6 Halat al i'rab 60

Récapitulatif de l'analyse morphologique : 61

2.3 Description de l'analyse grammaticale et de l'enrichissement 62
2.3.1 Analyseur 64
2.3.2 Enrichissement des règles grammaticales: 66
3 Récapitulatif : 68
4 Conclusion : 69
CHAPITRE 4 : Application et résultats 70
1. Introduction : 71
2. Langage Python 71
2.1 Caractéristiques du langage Python : 72
3 PyCharm 73
4. Analyse grammaticale d'un texte en langue arabe : 74
4.1 Choix du langage de programmation : 74
4.2 Matériel utilisé : 74
4.3 Bibliothèques utilisés : 75
4.4. Description de l'interface graphique d'ARIBLI : 76
4.5. Déroulement 77
4.6. Tests et résultats 82
5. Conclusion 82
Conclusion générale et perspec*tives* 83
Références bibliographiques 86

Liste des figures

Figure 1-1 : Les pays du monde arabe……..6

Figure 1-2 : Montre une structure possible d'un mot……16

Figure 1-3 : La segmentation de mot « أَسَتَذْ كُرُونَهُ »……16

Figure 1-4 : Hiérarchisation du mot en Langue Arabe……..……...17

Figure 1-5 : Classification des unités lexicales en Langue Arabe…….26

Figure 1-6 : Exemple sur l'effet du mot non voyelle « العلم » sur les extraits……27

Figure 2-1 : Les différentes méthodes d'étiquetage avec apprentissage automatique……...39

Figure 2-2 : Explication de l'algorithme de Viterbi ……..……...……42

Figure 2-3 : Exemple de réseau de neurones récurent..……44

Figure 2-4 : Décomposition d'un mot avec AL Khalil…….……..46

Figure 3-1 : Diagramme de classes du programme ARIBLI……53

Figure 3-2 : Exemples pour les champs des tables de la première partie de la BDD……54

Figure 3-3 : Schéma générale de l'analyse morphologique……..……56

Figure 3-4 : Illustre la recherche de schème du mot…….……..……...59

Figure 3-5 : Exemples pour les champs des tables de la deuxième partie de la BDD……..62

Figure 3.6 : Schéma général de l'analyse grammaticale et de l'enrichissement……..63

Figure 3-7 : Représentation de l'automate avant l'analyse de la phrase……..…….67

Figure 3-8 : Représentation de l'automate après l'enrichissement 67

Figure 4-1 : Interface graphique du système « ARIBLI »…….76

Figure 4-2 : Saisie du texte……..…….77

Figure 4-3 : Interface Results……..….78

Figure 4-4 : Interface Feed……..…….79

Figure 4-5 : Interface d'analyse impossible……..……...80

Figure 4-6 : Interface d'administration……..……..81

Liste des tableaux

Tableau 1-1 : Présente les 28 lettres de la langue arabe…….8

Tableau 1-2 : Représente différentes écritures de la lettre « qaf »……8

Tableau 1-3 : Montre un exemple pour les mots « كتب » et «شعر »……..……..…10

Tableau 1-4 : La dérivation de mots كتب écrire et حمل porter…….11

Tableau 1-5 : Quelque dérivation du verbe " كتب "……..…12

Tableau 1-6 : Table des préfixes……..……....13

Tableau 1-7 : Un exemple des suffixes divisés selon leurs types …..…….....14

Tableau 1-8 : Les pronoms personnels en Langue Arabe.....23

Tableau 1-9 : Les pronoms démonstratifs en Langue Arabe…….24

Tableau 1.10 : Les pronoms relatifs en Langue Arabe.....24

Tableau 1-11 : Combinaisons possibles d'inversion de l'ordre des mots dans la phrase……...28

Tableau 1-12 : La liste de Préfixes et suffixes les plus fréquents……....29

Tableau 1-13 : Exemple d'étiquettes grammaticales attribuées selon la voyellation ….....33

Tableau 2.1 : Tableau comparatif des étiqueteurs morphologiques…….49

Tableau 4-1 : Matériel hardware utilisé lors du développement et des tests.....74

Tableau 4-2 : Description du bouton de traitement.....76

Liste d’acronymes

AFP : l'Agence France Press (AFP).

ASCII : American Standard Code for Information Interchange

APT : Arabic Part-of-speech Tagger

MADA : Morphological Analysis and Disambiguation of Arabic

SVM : Support Vector Machine

POS : Part Of Speech

TT : Tree tagger

TBA : Tree bank arabic

ALECSO : Organisation arabe pour l'éducation, la culture et les sciences

HMM : Hiden markov model

IDE : Environment de development integé.

LDC : Linguistic Data Consortium

CWI : Centre national de recherches

VCS : Systèmes de contrôle de version

DIINAR.1 : DIctionnaire INformatisé de l'Arabe version 1

ORM : Mapping objet-relationnel

BDD : Base De Données

Introduction générale

Introduction générale :

1. ***Contexte global :***

Le traitement automatique du langage naturel, abrégé en TALN, est une discipline s'appliquant au domaine de l'informatique et du langage. Il est utilisé par exemple pour les traductions, la reconnaissance vocale ou encore les réponses automatiques aux questions. [14]

Il représente un défi colossal dans le domaine de l'informatique car le langage peut en effet être à double sens et pour le comprendre, il est nécessaire de bien connaître le contexte dans lequel il s'insère.

Toutefois, les ordinateurs comprennent de mieux en mieux le langage humain. Pour leur faire apprendre le langage, différents domaines d'application de la linguistique sont pris en compte :[16]

- La **morphologie**, qui s'intéresse à la forme des mots et à l'organisation des catégories grammaticales.
- La **syntaxe**, qui définit comment les mots sont agencés dans une phrase.
- La **sémantique**, qui correspond à la signification des mots et des groupes de mots
- La **pragmatique**, grâce à laquelle le contexte est pris en compte.
- Enfin la **phonologie**, qui s'occupe des sonorités de la langue orale, ce qui est important pour la reconnaissance vocale.

2. ***Problématique :***

En pédagogie, on appelle analyse grammaticale l'exercice de décomposition qui a pour matière les phrases et les mots. Il consiste à distinguer les propositions et membres de phrases, en marquant leur nature et les rapports qui les unissent, ou encore à discerner la nature et le rôle de chaque mot dans la proposition.

L'analyse, dans le premier cas, s'appelle analyse logique[1], dans le second, analyse grammaticale[2]. En réalité, ces deux genres d'analyse sont inséparables et stériles l'une sans l'autre.

Les mots qui composent un texte voyellé ou non voyellé sont donc éminemment ambigus. Comment, en contexte, faire le bon choix ? Autrement dit, comment associer aux différents mots qui composent un texte l'étiquette qui leur convient, compte tenu du contexte où ils occurrent ? Tel est le but de l'étiquetage grammatical, problématique posée dès la fin des années 60.

3. ***Objectif :***

Dans notre travail, nous nous intéressons à la conception et au développement d'un outil d'aide à l'analyse grammaticale autrement dit * AL I'RAB * qui sert à définir la nature et la fonction des mots contenus dans une phrase. Il s'agit donc de faire une décomposition séparée de chaque mot, pour reconnaître à quelle partie du discours il appartient, s'il est simple ou composé, primitif ou dérivé, à quel genre, à quel nombre, à quel cas appartiennent les substantifs, les adjectifs, les pronoms, et pourquoi, à quelle personne, à quel nombre, à quel temps, à quelle conjugaison les verbes appartiennent etc…

---

1 Décomposer la phrase en propositions et indiquer leur fonction c'est-à-dire les rapports qu'elles ont les unes avec les autres.

2 Etudier isolement chaque mot d'une phrase et déterminer la nature, les variations etc…

4. *Organisation du mémoire*

Le présent mémoire est articulé comme suit :

La première partie concerne l'étude bibliographique qui comprend deux chapitres :
- Un premier chapitre intitulé « **Généralités sur l'arabe et son traitement automatique** » qui a été consacré à la présentation de certaines bases linguistiques et grammaticales fondamentales de la langue arabe.
- Un second chapitre nommé «**Travaux antérieurs**» dans lequel nous nous sommes intéressés à l'étiquetage grammatical où nous avons évoqué quelques travaux récents qui nous ont semblés les plus intéressants.

Une deuxième partie qui concerne les ressources que nous avons utilisées, la solution que nous avons proposée ainsi que les résultats obtenus. Elle comprend à son tour deux chapitres :
- Le troisième chapitre du mémoire intitulé « **Conception et modélisation de la solution** » où nous avons présenté l'architecture générale proposée en détaillant les différents modules du programme, ainsi que la base de données implémentée.
- Le quatrième chapitre « **Application et résultats** » où nous avons détaillé les différentes étapes du déroulement de notre application et ensuite l'interface graphique ainsi que les résultats obtenus.

Enfin, une conclusion générale et quelques perspectives futures.

CHAPITRE 1 : Généralités sur l'Arabe standard et son traitement automatique

1. Introduction

La langue arabe est d'une origine très différente des langues européennes. Elle fait partie du groupe des langues sémitiques. Ce groupe se divise en langues sémitiques orientales, sémitiques occidentales et sémitiques méridionales. À la différence d'autres nations telles que les anciens égyptiens, les babyloniens et les chinois dont les systèmes d'écriture remontent à des milliers d'années, l'écriture arabe, telle que nous la connaissons aujourd'hui, n'est apparue qu'au VIe siècle[1].

Certains historiens et chercheurs pensent que l'origine de l'écriture arabe est le syriaque en se basant sur :

- l'ordre primitif des lettres arabes.
- les formes de l'ancien alphabet arabe dit « coufique », qui sont comparables à celles de « l'estranghelo », une forme de l'écriture syriaque.

L'arrivée de l'Islam a profondément marqué l'histoire de la langue et de l'écriture arabe. Le Coran, livre sacré recueillant la parole de Dieu, mais aussi code juridique et moral, occupe d'emblée une place centrale dans la vie du croyant et de la communauté musulmane.

L'écriture arabe a par la suite connu plusieurs réformes : naissance des points distinguant les différentes lettres, du tachkil (voyelles courtes)…

Puis vient l'âge de la calligraphie où de très belles écritures à rôle ornemental sont apparues.

[1] Les Arabes ont de tout temps utilisé l'écriture. Toutefois, ce n'est pas celle en usage aujourd'hui qui l'a toujours été. Le khatt al-musnad (ressemblant à l'alphabet tifinagh) était très répandu au Yémen…

2. La langue arabe

L'**arabe** (العربية , *al ʿarabīya*) est une langue sémitique aujourd'hui parlée en première ou seconde langue par plus de 220 millions de personnes dans le monde. Elle est marquée par une importante diglossie entre l'arabe littéral, langue véhiculaire surtout écrite, et l'arabe dialectal, langue vernaculaire surtout orale. L'arabe littéral comprend l'arabe classique (pré-coranique, coranique, et post-coranique) et l'arabe standard moderne. L'arabe dialectal comprend de nombreuses variétés régionales. L'arabe est parlé à des degrés divers dans les pays arabes du Moyen-Orient, en Iran (province du Khouzistan), dans les pays d'Afrique du Nord, du Sahara, du Sahel et sur les côtes de la Corne de l'Afrique. Il est également pratiqué par la diaspora arabe.

Figure 1-1 : Les pays du monde arabe.

3. Traitement automatique de la langue Arabe (TALA)

Le traitement automatique de la langue arabe est une discipline en pleine expansion, dans laquelle on voit de plus en plus de recherches et de technologies se soucier des spécificités de cette langue [6] et proposer des outils nécessaires au développement de son traitement automatique. Par ses propriétés morphologiques et syntaxiques la langue arabe est considérée comme une langue difficile à maîtriser dans le domaine du traitement automatique de la langue [2], [3]. L'arabe doit sa formidable expansion à partir du 7eme siècle grâce à la propagation de l'islam et la diffusion du Coran [4]. Les recherches pour le traitement automatique de l'arabe ont débuté vers les années 1970. Les premiers travaux concernaient notamment les lexiques et la morphologie.

Avec la diffusion de la langue arabe sur le Web et la disponibilité des moyens de manipulation de textes arabes, les travaux de recherche ont abordé des problématiques plus variées comme la syntaxe, la traduction automatique, l'indexation automatique des documents, la recherche d'information, etc.

Par exemple le mot français ***ferme***, est hors contexte, un substantif, un adjectif ou un verbe. Alors que le mot arabe **RaLaKa** غَلَقَ est un verbe à la 3eme personne masculin singulier de l'accompli actif, par contre sa forme non voyellée غلق (dans l'exemple donné ne sont représentées que les consonnes RLK) admet quatre catégories grammaticales :

- Substantif masculin singulier (RaLKun : une fermeture),
- Verbe à la 3eme personne masculin singulier de l'accompli actif (RaLaKa : il a fermé ou RaLLaKa il a fait fermer),
- Verbe à la 3eme personne masculin singulier de l'accompli passif (RuLiKa : il a été fermé),
- Verbe à l'impératif 2eme personne masculin singulier (RaLLiK: fais fermer).

Les voyelles jouent un rôle proche des accents en français pour un mot comme
peche qui peut être interprété comme *pêche, pèche et péché*. Par contre, en arabe chaque lettre de chaque mot devrait posséder sa voyelle ce qui n'est en général pas le cas, on constate donc l'étendue du rôle que jouent les voyelles dans les mots arabes, non seulement parce qu'elles enlèvent l'ambiguïté, mais aussi parce qu'elles donnent l'étiquette grammaticale d'un mot indépendamment de sa position dans la phrase.

3.1 Les caractéristiques de la langue arabe

L'alphabet de la langue arabe compte 28 consonnes (tableau1-1). L'arabe s'écrit et se lit de droite à gauche. Les lettres changent de forme de présentation selon leur position (au début, au milieu ou à la fin du mot).

Arabe	Translittéré	Nom	Arabe	Translittéré	Nom
أ	ā	Alif	ض	ḍ	Ḍād
ب	b	Bā	ط	ṭ	Ṭā
ت	t	Ta	ظ	ḍ	Ḍha
ث	th	Tha	ع	'	'Ayn
ج	j	Jīm	غ	gh	Ghain
ح	ḥ	Hā	ف	f	Fā
خ	kh	Khā	ق	g	Gāf
د	d	Dāl	ك	k	Kāf
ذ	ḍ	Thāl	ل	l	Lām
ر	r	Rā	م	m	Mīm
ز	z	Zāy	ن	n	Nūn
س	s	Sīn	ه	h	Hā
ش	sh	Shīn	و	w (u)	Wāw
ص	ṣ	Ṣad	ي	y (i)	Yā

Tableau 1-1 : présente les 28 lettres de la langue arabe

La représentation morphologique de l'arabe est assez complexe en raison de la variation morphologique et du phénomène d'agglutinement, les lettres changent de formes selon leur position dans le mot (isolée, initiale, médiane et finale). Le tableau 1-2 montre un exemple des différentes formes de la lettre «qaf » dans différentes positions. Nous pouvons donc observer plusieurs caractéristiques générales de cette langue.

Isolée	Initiale	Médiane	Finale
ق	قـ	ـقـ	ـق
	قرآن	القرآن	غسق

Tableau 1-2 : représente différentes écritures de la lettre **« qaf ».**

Un mot arabe s'écrit avec des consonnes et des voyelles. Les voyelles sont ajoutées au-dessus ou en-dessous des lettres et elles sont nécessaires à la lecture et à la compréhension correcte d'un texte, elles permettent de différencier des mots ayant la même représentation. Pour une meilleure précision de la prononciation, des signes ont été inventés. Il s'agit de trois voyelles brèves et de sept signes orthographiques qui s'ajoutent aux consonnes. Ces trois voyelles brèves sont :

- **Fatha « ـَ »,** elle surmonte la consonne et se prononce comme un «a» français ;
- **Damma « ـُ »,** elle surmonte la consonne et se prononce comme un «ou» français ;
- **Kasra « ـِ »,** elle se note au-dessous de la consonne et se prononce comme un « i » français).

Les sept signes orthographiques sont :

❖ **Sukun « ° » :** ce signe indique qu'une consonne n'est pas suivie par une voyelle. Il est noté toujours au-dessus de la consonne.

❖ **Les trois signes de tanwin :** lorsque (la Fatha, la Kasra et la Damma) sont doublées, elles prennent un son nasal, comme si elles étaient suivies de «n» et on les prononce respectivement :

✓ **an** pour les Fathatan.

✓ **in** pour les Kasratan.

✓ **un** pour les Dammatan.

❖ **Chadda «** ّ **»** comme dans le français, l'arabe peut renforcer une consonne quelconque

❖ **Wasla «** ٱ **» :** quand la voyelle d'un Alif au commencement d'un mot doit être aspirée par la dernière voyelle du mot qui précède.

❖ **Madda «** آ **» :** la madda (prolongation) se place sur l'Alif pour indiquer que cette lettre tient lieu de deux alifs consécutifs ou qu'elle ne doit pas porter le Hamza.

	Interprétation I		Interprétation II		Interprétation III	
كتب	كَتَبَ	Il a écrit	كُتِبَ	Il a été écrit	كُتُبٌ	Des livres
شعر	شَعَرَ	Il a senti	شِعْرٌ	Poème	شَعْرٌ	Chevelure

Tableau 1-3 : montre un exemple pour les mots « كتب » et « شعر »

Les lettres lunaires initiales d'un nom n'assimilent pas l'article qui les précède et par conséquent ne reçoivent pas le chadda. La lettre ا ل est prononcée.
Les lettres solaires initiales d'un nom assimilent l'article qui les précède et reçoivent ainsi la ***chadda***. La lettre ا ل est muette. Les caractères de la langue arabe n'appartiennent pas au code ASCII, d'où la nécessité d'utiliser un autre code qui prend en charge la langue arabe, ce code est l'**Unicode**, ce dernier permet de coder tous les caractères utilisés par la langue arabe en mode 16 bits.

3.2 Les éléments essentiels de la morphologie arabe

La langue arabe a une morphologie très riche et différente par rapport aux langues occidentales. L'analyse morphologique d'un mot arabe, consiste principalement à déterminer la structure générale de ce mot, s'il existe, et les autres éléments utilisés pour construire ce mot (les affixes, les modèles).

Les éléments essentiels de la morphologie de la langue arabe sont [1] :

3.2.1 Le schème «الوزن »

L'Arabe est une langue sémitique dont la morphologie fonctionne sur le croisement des racines (généralement trilitères) avec un nombre de formes déterminées pour produire sa gamme de vocabulaire. Ces formes, appelées moules ou schèmes, sont essentiellement constituées par une suite de voyelles et consonnes qui viendront se placer avant, après et entre les lettres de la racine afin de former les différentes formes des flexions verbales et nominales.
Une famille de mots peut ainsi être générée d'un même concept sémantique à partir d'une seule racine à l'aide de différents schèmes. Ce phénomène est caractéristique à la morphologie arabe.

Le **Tableau 1-4** donne quelques exemples de schèmes appliqués sur les deux mots « حمل », « كتب ». On peut ainsi dériver un grand nombre de noms, de formes et de temps verbaux [17].

Schèmes	KTB	كتب	HML	حمل	Notion de porter
فاعل	KâTiB	كاتب	HâMiL	حَامل	porteur
فعَل	KaTaBa	كتب	HaMaLa	حَمَلَ	A porté
مفعل	maKTab	مكتب	maHMaL	مَحْمَل	Brancard
فعل	KuTiBa	كتب	HuMiLa	حُملَ	A été porté

Tableau 1-4 : La dérivation de mots كتب écrire et حمل porter

La majorité des verbes arabes ont une racine composée de 3 consonnes. L'arabe comprend environ 150 schèmes ou patrons dont certains plus complexes, tel le redoublement d'une consonne ou l'allongement d'une voyelle de la racine, l'adjonction d'un ou de plusieurs éléments ou la combinaison des deux.

Une autre caractéristique est le caractère flexionnel des mots : les terminaisons permettent de distinguer le mode des verbes et la fonction des noms.

3.2.2 La racine «الجذر »

Une racine est purement consonantique, elle est formée par une suite de trois ou quatre (ou même cinq pour les noms) consonnes formant la base du mot. La racine est un élément important dans les langues dérivationnelles. En effet, à chaque racine correspond un champ sémantique et à l'aide de différents schèmes, on peut générer une famille de mots appartenant à ce champ sémantique [16].

Par exemple, la racine « كتب » (il a écrit) a la signification de base « écrire ». Plusieurs mots sont dérives à partir de cette racine [9].

	La racine "كتب"(écrire)			
Verbes	كتب	Il a écrit	تكتب	Il écrit
	كتبا	Nous avons écrit	يكتبون	Ils écrivent
	كتبت	Elle a écrit	تكتب	Tu écris
	تكتبون	Vous écrivez	نكتب	Nous écrivons
Noms	كاتب	Ecrivain	كتابة	Ecriture
	كتاب	Livre	مكتوب	Ecrit
	مكتب	Bureau	اكتتاب	Enregistrement

Tableau 1-5 : Quelque dérivation du verbe " كتب ".

3.2.3 Les affixes

Les affixes sont des lettres qui s'ajoutent au début (les préfixes) ou à la fin des mots arabes (les suffixes). En général, Ils sont utilisés pour accorder aux mots des éléments syntaxiques. Ils marquent l'aspect verbal, le mode, les propriétés transitives, etc. Leur nombre tourne aux alentours de ***150*** affixes [12].

❖ **Préfixes**

Dépendent des mots auxquels ils s'attachent. En effet, la plupart des mots arabes commencent par le préfixe « ال التعريف », « al altåryif » « l'article de définition » qui est utilisé en tant que terme déclaratif. Pour cela, il y a trois types de préfixes. Premièrement, les préfixes nominaux qui sont réservés pour les noms et les adjectifs. Deuxièmement, les préfixes verbaux qui sont réservés aux verbes. Et troisièmement, les préfixes généraux qui sont utilisés indépendamment du type des mots.

Le tableau (1-6) présente des exemples de chaque type de préfixes [14].

Type	Les préfixes			
	Nom en français	Signification	Nom arabe	Transcription
Préfixes nominaux	L'article de définition	Le	ال	Al (Lamltaarif)
	Les prépositions	Avec	ب	B
		pour	لـ	L
		comme	ك	K
	.....	.....	.....	.....
Préfixes verbaux	La particule du futur	Sera	سـ	S
	Les particules du subjonctif	Pour	ل	L
	.....	.....	.....	.....
Préfixes généraux	Les conjonctions de coordination	Et	ف	F
	L'article d'interrogation	Est-ce-que	أ	A

Tableau 1-6 : table des préfixes.

Les préfixes peuvent s'enchaîner dans un mot pour former des préfixes composés qui peuvent atteindre jusqu'à quatre lettres (<وبال , wabil, et avec le|la|les>, <وال , wal, et le|la|les>, <بال , bal, avec le|la|les>, <كال , kal, comme le|la|les>, etc.).

Dans ce cas, certains préfixes ne peuvent prendre que la première position, l'article d'interrogation <أ, a> par exemple, d'autres peuvent prendre n'importe quelle position, l'article de définition < ال التعريف , al altåryif, l'article de définition > exemple **(Tableau 1.6)**.

- **Suffixes**

  Il y a deux types de suffixes, les suffixes verbaux et les suffixes nominaux, les premiers dépendent de la transitivité et de la personne conjuguée **(Tableau 1-7)**

  Les suffixes nominaux indiquent la flexion du nom (nominatif, accusatif, et génitif), le genre (masculin et féminin), le nombre (singulier, duel et pluriel), etc [13].

Type	Nombre	Suffixes		
		Signification	Nom en arabe	Transcription
Première personne	Singulier	Moi/mon	ني	Nyi
	Duel/Pluriel	Nous/Notre	نا	Na
Deuxième personne	Singulier	Toi/ton	ك	K
	Duel	Votre/vous	كما	Kma
	Pluriel	Votre/vous	كم	Km
		Votre/vous	كن	Kn
Troisième personne	Singulier	Lui/son	ه	H
	Duel	Eux/leur	هما	Hma
	Pluriel	Eux/leur	هم	Hm
		Eux/leur	هن	hn

Tableau 1-7 : Un exemple des suffixes divisés selon leurs types [15].

3.3 Structure d'un mot

La définition du mot du point de vue du traitement automatique se heurte à des considérations syntaxiques et sémantiques.

Dans le domaine des langages formels, la transformation du flux de caractères représentant un texte en une suite d'unités mieux adaptées aux traitements ultérieurs, est habituellement appelée segmentation (tokenization).

Les unités produites, les segments (tokens), sont construits sur la base de définitions purement orthographiques. Le problème posé par de telles techniques pour des applications de traitements de langue est malheureusement l'absence de correspondance entre les segments identifiés et les unités textuelles élémentaires (les mots) manipulées dans le traitement linguistique. En arabe cette séquence de lettres est appelée le mot graphique.

Les mots sont séparés par des espaces et d'autres signes de ponctuation. Néanmoins, des prépositions sont agglutinées au mot (apparaissant après eux), faisant des limites invisibles entre le mot et la préposition.

Plusieurs types d'affixes sont agglutinés au début et à la fin des mots : antéfixes, préfixes, suffixes et post fixes. La représentation suivante schématise une structure possible d'un mot.

Note que la lecture et l'écriture d'un mot se fait de droite vers la gauche.

Enclitique suffixe Corps Schématique préfixe proclitique

Figure 1-2 : Montre une structure possible d'un mot.

- **Les proclitiques** sont des prépositions ou des conjonctions.
- **Les préfixes et suffixes** expriment des traits grammaticaux, tels que les fonctions de noms, le mode du verbe, le nombre, le genre, la personne…etc.
- **Les enclitiques** sont des pronoms personnels.
- **Le corps schématique** représente la base de mot « radicale »

Exemple :

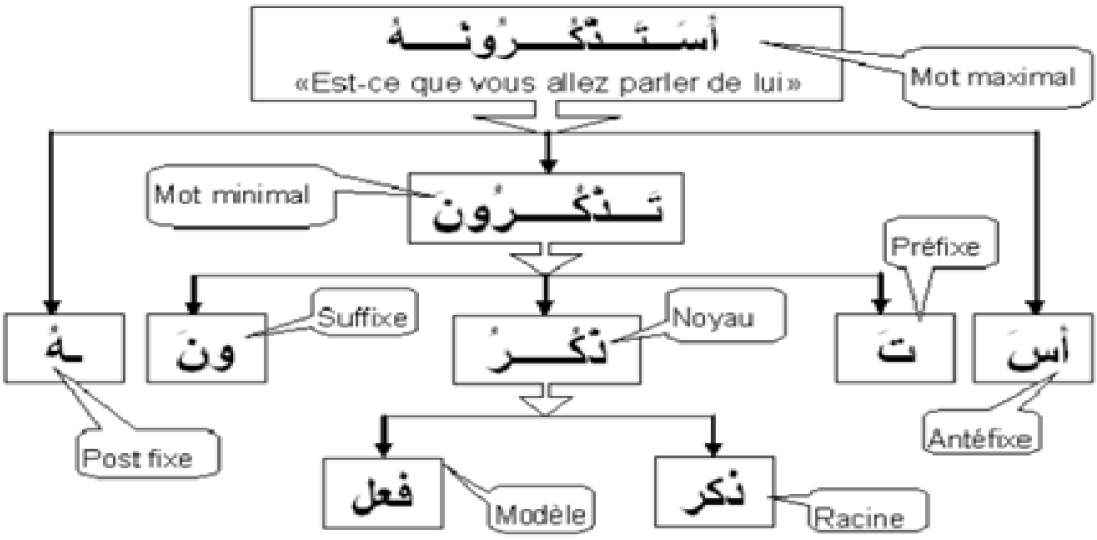


Figure 1-3 : La segmentation de mot « أَسَتَذْكُرُونَهُ ».

3.3.1 Catégories des mots

L'arabe considère 3 catégories de mots :

- **Le verbe :** entité exprimant un sens dépendant du temps, c'est un élément fondamental auquel se rattachent directement ou indirectement les divers mots qui constituent l'ensemble.
- **Le nom :** l'élément désignant un être ou un objet qui exprime un sens indépendant du temps.
- **Les particules :** entités qui servent à situer les événements et les objets par rapport au temps et l'espace, et permettent un enchaînement cohérent du texte.

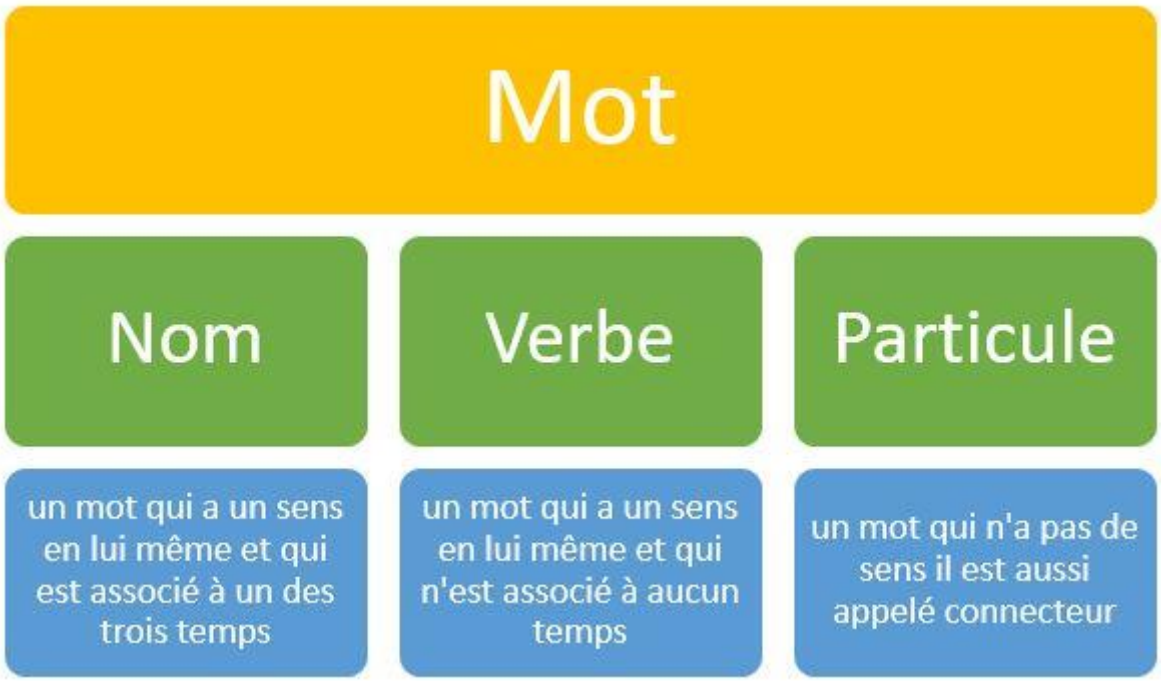


Figure 1.4 : Hiérarchisation du mot en Langue Arabe.

3.3.1.1 Morphologie Verbale :

La plupart des mots en arabe, dérivent d'une racine de trois lettres. Chaque racine est donc la source d'une famille de mots. Comme en français, le mot en arabe se déduit de la racine en rajoutant des suffixes ou des préfixes ou en appliquant un schème particulier.

La conjugaison des verbes dépend de plusieurs facteurs :

- Le temps (accompli, inaccompli).
- Le nombre du sujet (singulier, duel, pluriel).
- Le genre du sujet (masculin, féminin).
- La personne (première, deuxième et troisième)
- Le mode (forme active, forme passive)

Par exemple **:** ك + ت + ب, *K+T+B* donne le verbe كتب *KaTaBa* (écrire).

Dans tous les mots qui dérivent de cette racine, on trouvera ces trois lettres K, T, B. La conjugaison des verbes se fait en ajoutant des préfixes et des suffixes, un peu comme en français. La langue arabe dispose de trois modes :

L'accompli (الماضي) : correspond au passé et se distingue par des suffixes, par exemple pour le pluriel féminin on a كتبن **KaTaBna**, « ***elles ont écrit »*** et pour le pluriel masculin كتبوا

KaTaBuu, « ***ils ont écrit »***

L'inaccompli présent (المضارع) : présente l'action en cours d'accomplissement, ses éléments sont préfixés (يكتب yaKTuBu *il écrit*; تكتب taKTuBu, *elle écrit*).

L'inaccompli futur (الأمر) : correspond à une action qui se déroulera au futur et est marqué par l'antéposition de « س » *sa* ou « سوف » *sawfa* au verbe (سيكتب sayaKTuBu *il écrira,* يكتب سوف sawfa yaKTuBu *il va écrire*).

❖ Types du Verbe :

Lorsqu'aucune des consonnes radicales n'est pas une voyelle longue le verbe est dit «صحيح» (sain) dans le cas contraire le verbe est dit «معتل » (défectueux).

- **Le verbe sain «** الفعل الصحيح **»**

Tous les verbes dont aucune des lettres radicales n'appartient pas à l'ensemble [ا و ي] (l'ensemble des voyelles longues). Les verbes saints sont de trois types :

- S'il y a deux consonnes radicales identiques en deuxième et troisième, le verbe est dit redoublé «مضاعف » (mudaaf)
- Si une des trios consonnes radicales est une " أ " (a hamza) indépendamment de sa position on dit que le verbe est «مهموز».
- Si le verbe sain n'est pas redoublé ou hamzé, il est donc un verbe régulier «سالم»

- **Le verbe défectueux «**الفعل المعتل»

Tous les verbes dont au moins une de ces lettres d'origine appartient à l'ensemble [ا و ي] (L'ensemble des voyelles longues). Le verbe défectueux, à son tour, se divise en trois types selon la position des voyelles longues :

- Dans la première position le verbe est dit « مثال » (assimilé).

Exemple : «وَعَدَ »

- Dans la deuxième position le verbe est dit « أجوف » (creux).

Exemple : « قال »

- Dans la troisième position le verbe est dit « ناقص » (faible).

Exemple : « سعى » « رمى »

3.3.1.2 La Morphologie Nominale :

Les substantifs arabes sont de deux catégories, ceux qui sont dérivés de la racine verbale et ceux qui ne le sont pas comme les noms propres et les noms communs. Dans le premier cas, le fait que le nom soit dérivé d'un verbe, il exprime donc une certaine sémantique qui pourrait avoir une influence dans la sélection des phrases saillantes d'un texte pour le résumé.
La déclinaison des noms se fait selon les règles suivantes :

- **Le féminin singulier :** On ajoute le 'ة', Exemple : صغير *petit* devient صغير ة *petite.*
- **Le féminin pluriel :** De la même manière, on rajoute pour le pluriel (ت , ا)
  Exemple صغير *petit* devient صغيرا ت *petites.*
- **Le masculin pluriel :** Pour le pluriel masculin on rajoute les deux lettres ين ou ون dépendamment de la position du mot dans la phrase (sujet ou complément d'objet)

Exemple : الراجع *revenant* devient الراجعين ou الراجعون *revenants.*

- **Le Pluriel irrégulier :** Il suit une diversité de règles complexes et dépend du nom. **Exemple :** طفل *un enfant* devient أطفال *des enfants.*

Le phénomène du pluriel irrégulier dans l'arabe pose un défi à la morphologie, non seulement à cause de sa nature non concaténâtes, mais aussi parce que son analyse dépend fortement de la structure comme pour les verbes irréguliers.

On distingue trois catégories des noms arabes :

❖ Les noms primitifs

Ce sont des noms qui ne se dérivent pas d'une racine verbale.

Exemple : أخ, كراسي, كبش

❖ Les noms dérivés

Ce sont des noms qui se dérivent à partir d'une racine selon des schèmes prédéfinis, le nombre des noms dérivés de la même racine varie selon le statut de la racine dont ils se rattachent. Les noms dérivés de la langue arabe sont :

- **Le nom d'agent «** الفاعل **»**

C'est un nom dérivé d'un verbe pour designer qui a fait l'action.

Exemple : كتب (il a écrit) كاتب (écrivain)

- **Qualificatif de supériorité «**اسم التفضيل**»**

C'est un nom signifie que deux choses ont un caractère commun mais un est plus qualifié que l'autre.

Exemple : « محمد أذكى من علي »

- **Qualificatif assimilé «** الصفة المشبهة **»**

C'est un nom dérivé qui a le sens du non d'agent mais le qualificatif assimilé ne se dérive qu'à partir d'un verbe non transitif trilitère pour signifier que le qualificatif est permanant et confirmé.

Exemple : « جريح »

- **Nom de patient «** اسم مفعول **»**

C'est un nom dérivé d'un verbe pour designer qui a subi l'action. Il ne se dérive qu'à partir des verbes transitifs.

Exemple : «مسحوقٌ»

- **Nom de temps «** اسم الزمان **»**

C'est un nom dérivé d'un verbe pour indiquer le temps de déroulement de l'action.

Exemple : « مغرب »

- **Nom de lieu «** اسم المكان **»**

C'est un nom dérivé d'un verbe pour indiquer le lieu du déroulement de l'action.

Exemple : « مسبح »

- **Nom d'instrument «** اسم الآلة **»**

C'est un nom dérivé d'un verbe pour designer l'outil utilisé pour réaliser l'action.

Exemple : « مطرقة »

- **Qualificatif intensifié «** صيغة مبالغة **»**

C'est un nom dérivé d'un verbe. Il porte le sens intensifié du nom d'agent.

Exemple : « علّام »

❖ Les pronoms

Les pronoms sont considérés comme un ensemble des noms particuliers dans la langue arabe, et échappent à toute règle de dérivation. Dans cet ensemble, on distingue :

- **Les pronoms personnels :** Il existe deux types de pronom dans la langue arabe, les pronoms personnels isolés et les pronoms personnels liés (voir le **Tableau1.8**).
  Le pronom personnel séparé remplace le sujet dans une phrase arabe.

	les pronoms isolés			les pronoms liés		
Sémantique	arabe	prononciation	française	arabe	prononciation	française
1er singulier	أنا	Ana	je	ـي	-Y	je
2ème masculin singulier	أنتَ	Anta	Tu (masculin)	ـكَ	-Ka	toi (masculin)
3ème féminin singulier	أنتِ	Anti	Tu (féminin)	ـكِ	-Ki	Toi (féminin)
3ème masculin singulier	هو	Houwa	il	ـه	-Ho	lui
3ème féminin singulier	هي	Hya	elle	ـها	-Haa	elle
1er pluriel	نحن	Nahnou	nous	ـنا	-Naa	-
2ème masculin ou mixte pluriel	انتم	Antom	Vous (masculin ou mixte)	ـكم	-Kum	-
2ème féminin pluriel	انتن	Antonna	Vous (féminin)	ـكن	-Kunna	-
3ème masculin ou mixte pluriel	هم	Hom	ils	ـهم	-Hum	-
3ème féminin pluriel	هن	Honna	elle	ـهن	-Hunna	-

Tableau 1.8 : Les pronoms personnels en Langue Arabe

Les pronoms démonstratifs « أسماء الإشارة » : Ce sont des pronoms exprimant une idée de démonstration (voir le **Tableau 1.9**). Ils permettent d'indiquer que l'objet représenté se trouve, soit dans le texte, soit dans l'espace ou le temps.

Pronom	prononciation	français
هذا	Hadha	Ceci, Celui-ci
هذه	hadhi-hi	Celle-ci
هؤلاء	haoulai	Ceux-ci
ذلك	dhalika	Celui-là
تلك	tilka	Celle-là
أولئك	oulaika	Ceux-là

Tableau 1.9 : Les pronoms démonstratifs en Langue Arabe.

- **Les pronoms Relatifs «** الأسماء الموصولة**» :** Ils se rapportent aux noms ou aux pronoms personnels qui les précèdent et qui nous désignons par antécédente (Voir **le Tableau 1.10**).

Pronom	prononciation	Français
الذي	alla-dhi	Qui
الذي	alla-dhi	Lequel
التي	alla-ti	Laquelle
الذين	Alla-dhin	Lesquels
اللواتي	alla-lawati	Lesquelles

Tableau 1.10 : Les pronoms relatifs en langue Arabe.

3.3.1.3 Les particules

Ce sont principalement les mots outils comme les conjonctions de coordination et de subordination. Les particules sont classées selon leur sémantique et leur fonction dans la phrase, on en distingue plusieurs types (introduction, explication, conséquence, …). Elles jouent un rôle important dans l'interprétation de la phrase [5]. Elles servent à situer des faits ou des objets par rapport au temps ou au lieu, elles jouent également un rôle clé dans la cohérence et l'enchaînement d'un texte. Comme exemple de particules qui désignent un temps (بعد ، قبل ، منذ : *pendant, avant, après)*, un lieu (حيث : *où)*, ou de référence (الذين : *ceux)*, ….
Les particules peuvent avoir des préfixes et suffixes ce qui rajoute une complexité quant à leur identification.

Certains linguistes définissent les particules comme des mots non fléchis, et si l'on suit cette définition, on peut dire que de nombreuses langues occidentales comme l'Anglais et l'Allemand disposent de particules. Cependant, ce mot est normalement utilisé pour désigner certains types de mots que l'on trouve dans les langues orientales comme le Chinois, le coréen et le Japonais.

Dans les langues qui utilisent très souvent les particules, comme l'arabe, on peut considérer qu'elles remplacent certaines prépositions et certains articles, même si elles jouent en général un rôle encore plus important. Par exemple, elles peuvent aussi indiquer qu'une phrase entière est interrogative ou emphatique.

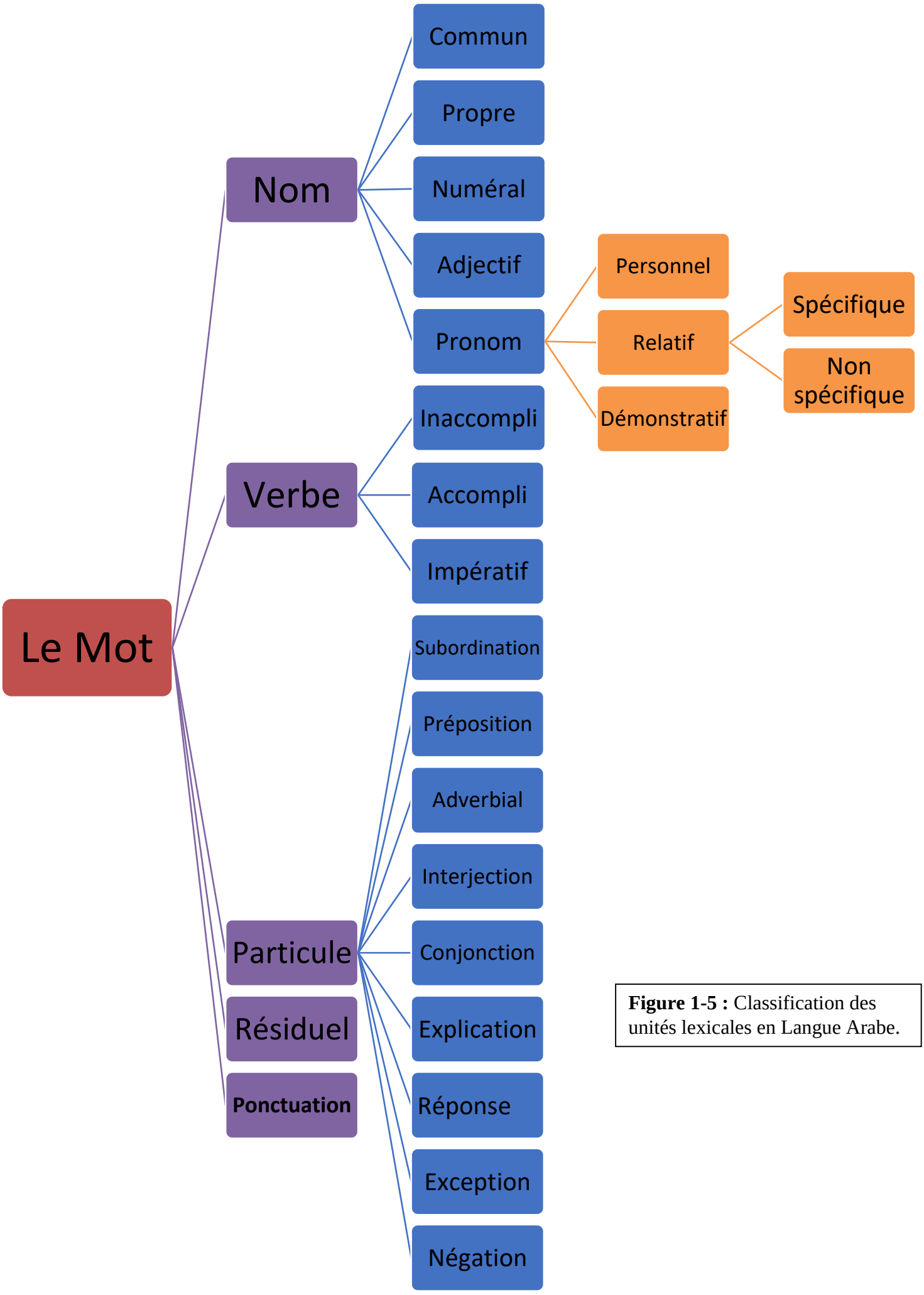


Figure 1-5 : Classification des unités lexicales en Langue Arabe.

4. Les particularités de la langue arabe

4.1 Voyellation

En Arabe écrit, les voyelles (signes diacritiques) sont omises et le résultat de cette omission est que les mots tendent à avoir un risque de générer une certaine ambiguïté à deux niveaux :

- Ambiguïté au niveau du sens du mot.
- Difficulté à identifier sa fonction dans la phrase, (différencier entre le sujet et le complément, ce qui n'aurait pas été le cas avec un texte voyellé.

Figure 1-6 : Exemple sur l'effet du mot non voyelle « العلم » sur les extraits.

L'ambiguïté vient du mot العلم *la science* ou *drapeau* alors que voyellé on aura العِلْمُ pour *la science* et العَلَمُ pour *le drapeau*. Cette ambiguïté pourrait, dans certains cas, être levée soit par une analyse plus profonde de la phrase ou des statistiques (par exemple il est plus probable d'avoir « العلم الوطني» *le drapeau national* que la *science nationale*).

De plus la capitalisation n'est pas employée dans l'arabe ce qui rend l'identification des noms propres, des acronymes, et des abréviations encore plus difficile [7].

Comme la ponctuation est rarement utilisée, on doit ajouter une phase de segmentation de phrase pour l'analyse d'un texte.

4.2 Agglutination

Contrairement aux langues latines, en arabe, les articles, les prépositions, les pronoms, etc. collent aux adjectifs, noms, verbes et particules auxquels ils se rapportent. Comparé au français, un mot arabe peut parfois correspondre à une phrase française.

Exemple : le mot arabe « أتتذكروننا » correspond en Français à la phrase "Est-ce que vous vous souvenez de nous ?".

Cette caractéristique peut engendrer une ambiguïté au niveau morphologique. En effet, il est parfois difficile de distinguer entre une proclitique ou enclitique et un caractère original du mot. Par exemple, le caractère "و" dans le mot "وصل " (il est arrivé) est un caractère original alors que dans le mot « وفتح » (et il a ouvert), il s'agit d'une proclitique [8].

4.3 Irrégularité de l'ordre des mots dans la phrase

L'ordre des mots en arabe est relativement libre. D'une manière générale, on met au début de la phrase le mot sur lequel on veut attirer l'attention et l'on termine sur le terme le plus long ou le plus riche en sens ou en sonorité. Cet ordre provoque des ambiguïtés syntaxiques artificielles, dans la mesure où il faut prévoir dans la grammaire toutes les règles de combinaisons possibles d'inversion de l'ordre des mots dans la phrase.

Ainsi par exemple, on peut changer l'ordre des mots dans la phrase **(tableau1-11)** pour obtenir trois phrases ayant le même sens.

Verbe + sujet + complément	فعل + فَاعل + متمم	Est allé le garçon à l'école	ذهب الولدُ إلَى المدرسة
sujet + verbe + complément	فَاعل + فعل + متمم	Le garçon est allé à l'école	الولدُ ذهب إلَى المدرسة
complément + verbe + sujet	متمم + فعل + فَاعل	A l'école est allé le garçon	إلَى المدرسة ذهب الولدُ

Tableau 1-11 : Exemple de combinaisons possibles d'inversion de l'ordre des mots dans la phrase.

4.4 Absence de ponctuation régulière

La langue arabe n'est pas appuyée principalement sur les signes de ponctuations et les marqueurs typographiques ; il est à noter que ces derniers ne sont pas utilisés de façon régulière dans les textes arabes actuels, et même dans le cas où ils y figurent, ils ne sont pas gérés par des règles précises d'utilisation.

Par ailleurs, nous pouvons trouver tout un paragraphe arabe ne contenant aucun signe de ponctuation à part un point à la fin de ce paragraphe. Ainsi, il convient de noter que la présence des signes de ponctuation ne peut pas guider la segmentation comme c'est le cas pour d'autres langues latines, telles que le français ou l'anglais. Ainsi, la segmentation de textes arabes doit être guidée non seulement par les signes de ponctuations et les marqueurs typographiques mais aussi par des particules et certains mots tels que les conjonctions de coordination, etc. [11].

4.5 Détection de racine

Pour détecter la racine d'un mot, il faut connaître le schème par lequel il a été dérivé et supprimer les éléments flexionnels (antéfixes, préfixes, suffixes, post fixes) qui ont été ajoutés. On utilise pour cela la liste de préfixes et de suffixes. Plusieurs d'entre eux ont été utilisés pour la lemmatisation de mots arabes ils ont été déterminés par un calcul de fréquence sur une collection d'articles arabes de l'Agence (AFP).

Préfixes							
لا	في	لل	كم	بم	وت	بت	وال
با	وا	لي	فم	له	ست	يت	فال
	فا	وي	ال	وم	نت	مت	بال

Suffixes							
ا	ة	ين	ية	هم	ته	وه	ات
	ه	يه	تك	هن	تم	ان	وا
	ي	ية	نا	ها	كم	تي	ون

Tableau 1-12 : La liste de Préfixes et suffixes les plus fréquents

L'analyse morphologique devra donc séparer et identifier des morphèmes semblables aux mots préfixés comme les conjonctions wa-‘و’et fa- ‘ف’ , des prépositions préfixées comme bi-‘ ب ’ et li- ‘ ل ’, l'article défini ‘ال ’ , des suffixes de pronom possessif.
La phase d'analyse morphologique détermine un schème possible. Les préfixes et suffixes sont trouvés en enlevant progressivement des préfixes et des suffixes et en essayant de faire correspondre toutes les racines produites par un schème afin de retrouver la racine [10].

Lorsqu'un mot peut être dérivé de plusieurs racines différentes, la détection de la racine est encore plus difficile, en particulier en absence de voyelles [10].

4.6 Le caractère ‘_’

Les typographes font un usage fréquent du caractère ‘-’ (appelé Kashida), qui permet l'allongement du trait au milieu des mots, pour une meilleure lisibilité, pour limiter les espaces blancs sur une ligne justifiée, voir pour des raisons purement esthétiques. Or cet usage peut nuire aux analyses automatiques : ce caractère ne fait pas partie de l'alphabet arabe, il est considéré comme un intrus par le système d'analyse automatique. Il faut donc recourir à un sous-programme particulier afin de l'éliminer. Exemple : le mot الكتاب : peut-être écrit de plusieurs façons : الكتاب , الكتاب , الكتاب , …etc.

4.7 Mots étrangers translittérés en arabe

Les translittérations en arabe de mots étrangers posent un problème, puisqu'ils n'ont pas de racine en arabe. Les mots translittérés sont considérés comme inconnus par l'analyseur.
Quelques items étrangers méritent une attention particulière en raison de leurs fréquences élevées.

Exemple : دولار , أورو …etc.

5. Difficultés du traitement automatique de la langue arabe

5.1 La segmentation de textes

La segmentation d'un texte est une étape fondamentale pour son traitement automatique ; son rôle est de découper le texte en unités d'un certain type qu'on aura défini et repéré préalablement. En effet, l'opération de segmentation d'un texte consiste à délimiter les segments de ses éléments de base qui sont les caractères, en éléments constituants différents niveaux structurels tels que : paragraphe, phrase, syntagme, mot graphique, mot-forme, morphème, etc.

Toutefois, les particularités de la langue arabe, rend la segmentation arabe toujours différente, il n'y a pas de majuscules qui marquent le début d'une nouvelle phrase. De plus, les signes de ponctuation, ne sont pas utilisés de façon régulière. D'après l'étude réalisée par certaines particules comme "et | و ", "donc | ف", etc. jouent un rôle principal dans la séparation de phrases et peuvent être déterminantes pour guider la segmentation.

5.2 L'analyse morphologique

La morphologie est un niveau essentiel dans les systèmes de traitement automatiques de la langue. L'opération de l'analyse morphologique tient à étudier la forme d'un mot (unités lexicales) en faisant une analyse interne de la structure de ce dernier. Le but étant de décomposer un mot à des éléments plus petits (préfixes, suffixes, etc.) selon des règles de combinaison relatives à ces derniers. À proprement parler, l'analyse morphologique ne fait que la séparation et l'identification des morphèmes semblables aux mots préfixés (comme les conjonctions "wa | و " et "fa | ف ", etc.), des prépositions préfixées (comme "bi | ب . " et "li | ل ", l'article défini "ال ", etc.), des suffixes de pronom possessif.

La phase d'analyse morphologique détermine un schéma possible. Les préfixes et suffixes sont trouvés en enlevant progressivement des préfixes et des suffixes et en essayant de faire correspondre toutes les racines produites par un schème afin de retrouver la racine.

Le problème principal de cette analyse réside dans l'agglutination et l'absence de voyellation. Pour l'agglutination et contrairement aux langues latines, en arabe, les pronoms, les prépositions, les articles, les conjonctions, et autres particules collent aux noms, verbes,

adjectifs et particules auxquels ils se rapportent. Comparé au français, un mot arabe peut parfois correspondre à une phrase française.

Cette caractéristique engendre une ambiguïté morphologique au cours de l'analyse. Ainsi, la reconnaissance des unités lexicales qui composent une unité morphologique n'est pas toujours facile à détecter. Le problème est de reconnaître que la bonne segmentation réside ainsi, dans la difficulté de distinction entre une proclitique ou enclitique et un caractère original du mot. Par exemple, le caractère "و" dans le mot "il est arrivé | وصل " est un caractère original alors que dans le mot "et il a ouvert | وفتح ", il s'agit plutôt d'une proclitique [11].
L'absence de voyellation pose un autre problème important. En effet, les mots non voyellés engendrent beaucoup de cas ambigus au cours de l'analyse (e.g. le mot non voyellé "فصل " pris hors contexte peut être un verbe au passé conjugué à la troisième personne du singulier "il a licencié | فَصَلَ ", ou un nom masculin singulier "chapitre/ saison | فَصلٌ ", ou encore une concaténation de la conjonction de coordination "puis | ف " avec le verbe "صل " : impératif du verbe lier conjugué à la deuxième personne du singulier masculin).

5.3 L'orthographe de la langue arabe

L'alphabet arabe se compose de : 25 consonnes et 6 voyelles classées en trois voyelles longues (ي, و, ا) (a, w, y) et trois voyelles courtes écrites comme des signes diacritiques (a, u, i).

Les lettres arabes changent de forme en fonction de leur position dans le mot. De plus, dans la langue arabe les signes diacritiques sont utilisés en dessus et en dessous des lettres. Ces signes diacritiques sont le **Sukun** pour marquer des lettres muettes (c.-à-d. absence de voyelle courte), la gémination ou l'incorporation **Chaddah** pour indiquer une lettre doublée et le **Tanwin** pour indiquer la marque syntaxique de noms singuliers indéfinis.

5.4 L'étiquetage grammatical

L'étiquetage grammatical est l'opération qui consiste à attribuer à chacun des mots d'un texte la catégorie (non, verbe, adjectif, article défini, etc.) qui est la sienne dans le contexte où il apparaît.
La difficulté de l'étiquetage grammatical s'amplifie lorsque les textes visés se présentent sous leur forme non pas voyellée, mais partiellement voyellée ou encore totalement non voyellée, ce qui correspond au cas le plus courant.
Dans ces conditions, le but général de l'étiquetage grammatical consiste à répondre à la question suivante : Comment associer aux différents mots qui composent un texte l'étiquette qui leur convient, compte tenu du contexte où ils occurrent ?

Le tableau (1-13) suivant présente le problème d'ambiguïté grammaticale rencontrée lors de l'attribution catégorique d'un mot non voyéllé "ktb | كتب ", qui admet au moins cinq étiquettes grammaticales qui sont les suivantes :

Exemple de voyelation	*Etiquettes grammaticales*
كُتُبٌ | **kutubun : des livres**	Substantif, masculin, pluriel
كَتبٌ | **katbun : un écrit**	Substantif, masculin, singulier
كَتَبَ | **kataba : il a écrit**	Verbe 3éme personne masculin, singulier de l'accompli actif
كُتِبَ | **kutiba : il a été écrit**	Verbe 3éme personne masculin, singulier de l'accompli passif
كَتِّب | **kattib : fais écrire**	Verbe à l'impératif, 2éme personne masculin, singulier

Tableau 1-13 : Exemple d'étiquettes grammaticales attribuées selon la voyellation [11].

5.5 L'analyse syntaxique

L'analyse syntaxique permet d'associer à un énoncé sa ou ses structures syntaxiques possibles, en identifiant ses différents constituants et les rôles que ces derniers entretiennent entre eux. Toutefois, l'analyse syntaxique prend en entrée le résultat de l'analyse lexicale (éventuellement de l'étiquetage morpho-syntaxique) et fournit en sortie une structure hiérarchisée des groupements structurels et des relations fonctionnelles qui unissent les groupements.
Enfin, il est à signaler que les ambiguïtés vocaliques et grammaticales, relatives à la non voyellation des mots, pose des difficultés au niveau de l'analyse syntaxique. Ainsi, une phrase, en absence de la voyellation, peut être interprétée et traduite selon plusieurs interprétations qui sont toutes syntaxiquement correctes.

6. Conclusion

Dans ce chapitre nous avons présenté et montré certains aspects et caractéristiques de la langue arabe qui nous semblent importants pour la suite de ce travail. Nous avons commencé par une présentation de la langue arabe ainsi que de ses variétés, aussi la structure d'un mot arabe puis, nous avons relevé les particularités de la langue arabe, telles que l'absence de ponctuation régulière, l'ambiguïté due à l'absence de voyelles et l'agglutination des mots et en dernier lieu nous avons montré la difficulté du traitement automatique de cette langue qui reste très riche de par sa diversité morphologique et grammaticale.

Nous allons poursuivre dans le chapitre suivant par une présentation des différentes approches d'étiquetage grammatical de la langue arabe puis une analyse de quelques étiqueteurs existants.

Chapitre 2 : Travaux antérieurs

1. Introduction

Dans ce chapitre nous allons aborder le principe de l'étiquetage grammatical le plus couramment utilisé qui fait intervenir des règles grammaticales qui portent sur les successions permises ou non de deux, trois ou n étiquettes grammaticales. Parce que ne permettant pas de résoudre l'ambiguïté dans tous les cas, ces règles se sont vu adjoindre des poids statistiques afin de choisir les résolutions les plus probables. Ces règles peuvent être lues de plusieurs façons, on peut dire par exemple qu'après telle étiquette, ce sont telles ou telles étiquettes qui peuvent suivre mais si l'on considère la dernière étiquette, on peut également dire qu'elle dépend de celles qui la précèdent. Ainsi c'est la formulation probabiliste utilisant les sources de Markov comme modèle.

2. L'étiquetage grammatical

L'étiquetage grammatical [1] pour la langue arabe reste toujours un sujet d'intérêt pour de nombreux chercheurs du fait de son rôle de brique de base dans de nombreuses applications du TAL. Bien que de nombreux systèmes d'analyse aient été réalisés selon des méthodes différentes, les pistes d'amélioration sont encore très ouvertes.

3. Méthodes d'étiquetage

Les différentes méthodes utilisent toutes les mêmes informations pour étiqueter un mot dans un texte : son contexte et sa morphologie.

Ce qui diffère, c'est la façon de représenter ces éléments et de hiérarchiser ces informations.

Il existe deux grands types d'étiqueteurs :

3.1 Etiqueteurs symboliques

- Les étiqueteurs symboliques sont ceux qui appliquent des règles grammaticales qui leur ont été fournies par des experts humains.

  Dans ce type d'étiqueteur il y a très peu d'automatisation, c'est le concepteur qui manipule toutes les règles d'étiquetage généralement implémentées sous forme de transducteurs ou des automates. La conception n'est pas automatisé mais l'étiqueteur, une fois les règles élaborées, fournit un bon étiquetage automatique qui atteint plus de 95% de rappel et de précision.

  La conception d'un tel étiqueteur est longue et coûteuse. De plus, les étiqueteurs ainsi conçus ne sont pas facilement portables, c'est-à-dire qu'ils ne sont efficaces que pour une langue donnée et un domaine donnée (exemple : le droit, la médecine, etc...).

---

[1] **L'opération qui consiste à attribuer à chacun des mots d'un texte sa catégorie grammaticale**

3.2 Etiqueteurs avec apprentissage

- Les étiqueteurs avec apprentissage automatique, divisés à leur tour en **2** catégories :

  (i) Ceux qui sont **supervisés** autrement dit, qui apprennent à partir de corpus pré-étiquetés tel l'étiqueteur de Brill (De Loupy, 1995, l'étiqueteur APT de Khoja (Khoja, 2001) [20] et l'étiqueteur POS Tagger (Diab, M., Hacioglu, K., et Jurafsky, D. 2004)

  (ii) Ceux qui ne le sont pas **-non supervisés-** qui apprennent à partir de corpus bruts sans informations additionnelles.

  Qu'ils soient supervisé ou non, les étiqueteurs avec apprentissage peuvent être regroupés en trois familles : système à base de règles, système statistique ou neuronal

Les deux types d'étiqueteurs symbolique et avec apprentissage, ont chacun leurs avantages et leurs inconvénients. L'un et l'autre ne seront pas utiliser dans le même type de situation.

Toutefois, dans l'approche symbolique une formulation probabiliste est utilisée pour la résolution de l'ambiguïté dans le choix de la règle grammaticale. Ainsi les règles, se sont vu adjoindre des poids statistiques afin de choisir les résolutions les plus probables.

La plupart des étiqueteurs **(taggers**, en anglais) développés pour l'arabe emploient des lexiques en utilisant des analyseurs morphologiques. La sélection de l'étiqueteur le plus approprié pour une application donnée est tout à fait difficile car ces derniers n'ont été évalués que sur des corpus de leur choix en raison du manque de benchmarks **(testeurs)** en langue arabe.

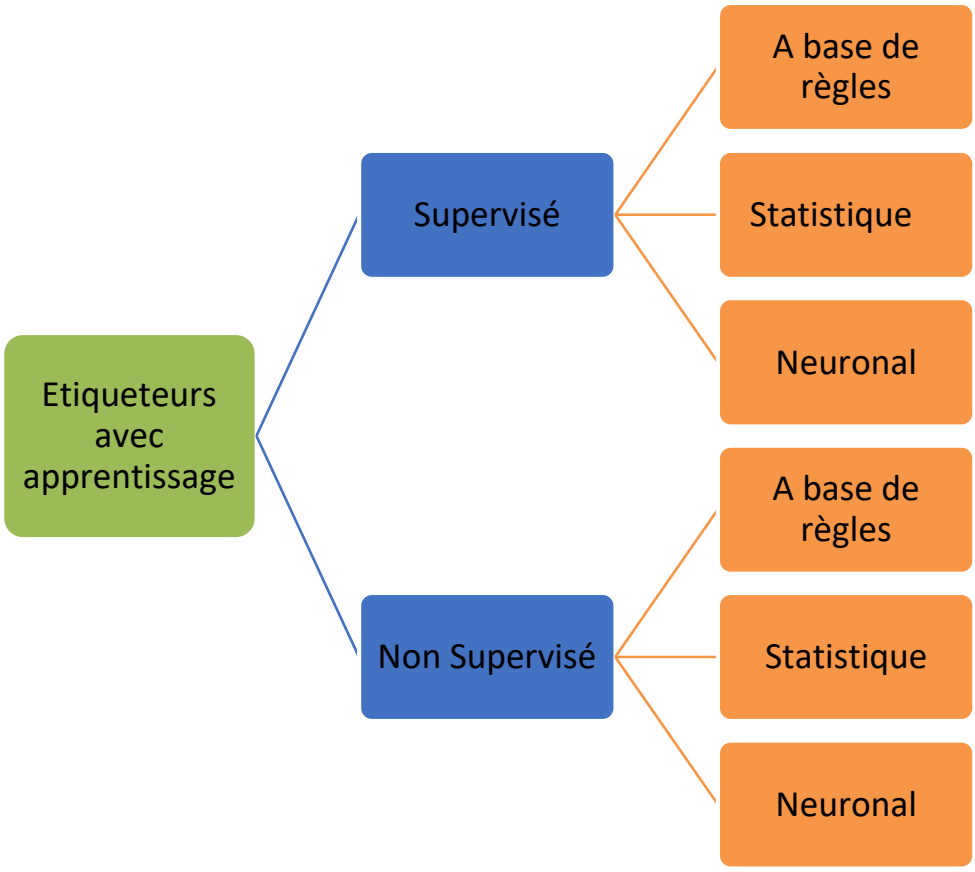


Figure 2.1 : Les différentes méthodes d'étiquetage avec apprentissage automatique

3.2.1. Etiquetage non supervisé

Au contraire des étiqueteurs supervisés, les étiqueteurs non supervisés ne nécessitent pas de corpus préalablement étiqueté pour la phase d'entraînement. Ils utilisent une analyse distributionnelle afin de regrouper automatiquement les mots en groupes ou classes de mots, c'est-à-dire en catégories grammaticales. Comme le note (Thibeault, 2004) [26] « En général, le regroupement des mots en classes se fait en fonction de la similarité des contextes. Plus précisément, le fait que des mots soient interchangeables dans des contextes formellement similaires détermine s'ils font partie d'une même catégorie ou classe. » **Exemple :** une classe de mots, une catégorie de mots.

3.2.2. Etiquetage supervisé

Les étiqueteurs supervisés sont entraînés sur des corpus préalablement étiquetés, ce qui permet de préparer toutes les données nécessaires pour l'étiquetage. Ces données sont créées à partir de dictionnaires permettant d'attribuer à chaque mot un ensemble de critères : catégorie, lemme, fréquence moyenne d'apparition du mot, parfois des statistiques sur les étiquettes du mot en contexte et des règles gammaticales pour faciliter l'analyse du mot par la suite.

L'étiqueteur de Brill (Brill, 1993) [22], l'étiqueteur APT de Khoja (Khoja, 2001) [20] et l'étiqueteur POS Tagger (Diab, Hacioglu & Jurafsky, 2004) [18] font partie de cette famille.

Les étiqueteurs supervisés ont tendance à donner de meilleurs résultats si on les utilise pour étiqueter le même type de texte que ceux sur lesquels ils ont été entraînés. Toutefois, l'étiqueteur de Brill échappe un peu à la norme. Il est en fait très portable du fait que le corpus nécessaire à l'entraînement n'a pas à être très gros.

De plus, son approche a été conçue dans une perspective multilingue, ce qui fait que le logiciel peut être entraîné en quelques heures pour tout type de texte et pour toute langue écrite avec un alphabet.

Cet étiqueteur applique des règles pour faire l'étiquetage, et a été déjà adapté sur la langue arabe (Haddad, Ben Ghezala & Ghenima, 2007) [24]. Par contre, TreeTagger est basé sur des méthodes statistiques qui sont plus efficaces que celles de Brill, mais il n'est toutefois pas disponible pour la langue arabe.

Notons aussi qu'il n'y a pas si longtemps, les corpus étiquetés nécessaires à l'entraînement des étiqueteurs supervisés étaient plutôt rares, dispendieux et pas nécessairement disponibles pour toutes les langues et tous les genres de texte.

Mais, depuis quelques années, plusieurs corpus ont été créés et rendus disponibles (EAGLES ,1996 [2]), (Penn TreeBank Arabic, 2004 [3]) et (corpus Buckwalter, 2002 [4]), il est donc de plus en plus facile de travailler dans le cadre d'une approche supervisée. Comme il a été précisé précédemment, quel que soit le type des étiqueteurs supervisés ou non, les étiqueteurs avec apprentissage peuvent être regroupés en trois familles qu'on va détailler dans ce qui s'en suit.

4. Catégorisation des étiqueteurs avec apprentissage

4.1. Etiquetage à base de règles

Ces types d'étiqueteurs s'appuient sur des règles grammaticales ou morphologiques, soit pour affecter une étiquette à un mot, soit pour définir les transitions possibles entre les différentes étiquettes Par exemple, une règle contextuelle peut dire qu'un mot X ambigu ou inconnu précédé d'un nom est un verbe. Cette règle servira à désambiguïser un mot en arabe comme «ذَهب» « *hb», qui peut être un nom **« de l'or »** ou un verbe **« aller ».**

Plusieurs étiqueteurs exploitent la morphologie pour faciliter la désambiguïsation. Par exemple, une règle qui tient compte de la morphologie peut spécifier qu'un mot ambigu ou inconnu qui se termine par «هم» «hom» et qui est précédé par un verbe est un nom.

4.2. Etiquetage statistique

Ce type d'étiquetage caractérise les étiqueteurs qui utilisent des fréquences et des calculs de probabilité. La forme la plus simple d'étiqueteur statistique attribue les catégories pour chaque mot en se basant sur les catégories les plus fréquentes dans un texte de référence. Cette méthode est dite robuste, c'est-à-dire qu'elle donne toujours des résultats, même si elle produit beaucoup d'étiquettes erronées, car le même mot recevra toujours la même étiquette, quel que soit son contexte.

---

2 **Expert Advisory Group on Language Engineering Standard**

3 http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2005T20

4 http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2002L49

Il existe une autre approche cherchant à calculer la probabilité qu'une certaine séquence d'étiquettes apparaisse.

Cette approche est appelée approche par **« n-grammes ».** Les algorithmes les plus connus pour implémenter une approche de ce type sont des algorithmes de programmation dynamique, comme l'algorithme de Viterbi 1967 [5], forward/backwar, et de Baum Welch du Modèle de Markov caché (HMM).

Ces algorithmes identifient, pour une suite de mots donnés, la séquence d'étiquettes maximisant la probabilité des transitions en se basant sur les fréquences observées dans un corpus d'entrainement.

Algorithme de Viterbi ou (régle du petit poucet) :

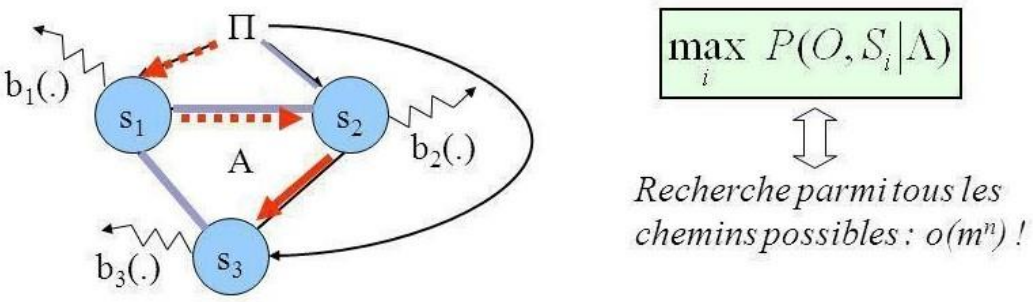


Figure 2.2 : Explication de l'algorithme de Viterbi

- Chaque symbole est émis par un seul état caché.
- La séquence d'états la plus probable pour expliquer la séquence d'observations à l'instant **t** dépend seulement de la séquence la plus probable à l'instant **t-1.**
- On peut trouver la séquence en procédant de proche en proche.

---

5 http://perso.telecom-paristech.fr/~vallet/dom_com/bacuvier/cadreviterbi.html

Pour estimer la probabilité que l'étiquette X soit suivie de l'étiquette Y, on peut se baser sur les fréquences observées dans un corpus d'entrainement, avec la formule suivante :

Prob (x|y) = freq(x , y) / freq(y).

Outre ces algorithmes, on trouve les modèles de langue, les n-classes, les SVM (Support Vector Machines) dont se base essentiellement les analyseurs connus : AMIRA (Diab, M., 2009) et MADA (Habash, N., & Rambow, O., 2005) [19] etc…

4.3. Etiquetage à base de réseaux de neurones

Un réseau de neurone peut être implémenté sous la forme d'un arbre où chaque nœud représente un neurone interconnecté avec les autres neurones par des liens inhibiteurs ou activateurs.

Un **RNN** (réseau de neurones récurent) est au minimum composé d'une succession de trois couches de neurones : une couche d'entrée au temps **t** notée **x(t)**, une couche cachée **h(t)** (aussi appelée couche de contexte), et une couche de sortie **y(t)**. Chaque neurone de la couche d'entrée est relié à tous les neurones de la couche cachée par les matrices des poids **U** et **W**. La matrice des poids **V** connecte tout neurone de la couche cachée à chaque neurone de la couche de sortie, voir **(Figure 2.3).** La couche d'entrée est formée par la concaténation de la représentation vectorielle **w(t)** du mot courant, et de la couche cachée au temps précédent **h(t-1)** (information de la première des couches cachées, dans le cas où on utilise plusieurs). La première étape est donc d'associer à chaque mot **w** (appartenant aux vocabulaires des langues source et cible) une représentation vectorielle spécifique.

Si on arrive à construire un espace de représentation commun, où un mot source et sa traduction cible possèdent des représentations vectorielles proches, nous pourrons à partir de cette représentation commune utiliser l'annotateur morpho-syntaxique de type RNN pour annoter un texte en langue cible. En général, un mot source et sa traduction cible apparaissent le plus souvent ensemble dans les mêmes bi-phrases, et donc leurs empreintes distributionnelles sont proches. En associant à chaque mot (source, cible) son empreinte distributionnelle **Vw** de dimension **N** (nombre de biphrases dans le corpus parallèle) indique si le mot apparaît ou pas dans chaque bi-phrase **Phi** {i = 1, ..., N} du corpus parallèle :

Vw = 1 si w ∈ P hi

Vwi = 0 sinon

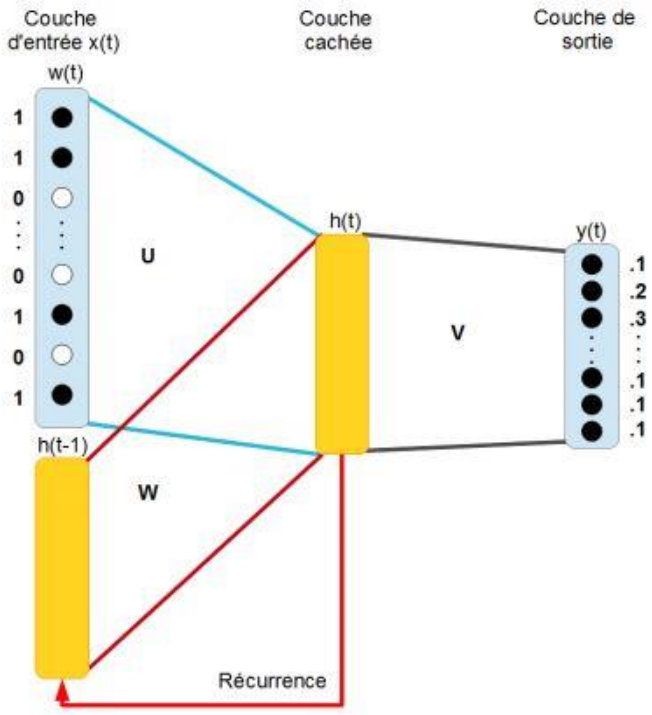


Figure 2.3 : Exemple de réseau de neurones récurent.

De plus, pour pouvoir transférer les annotations morpho-syntaxiques d'une langue à une autre, il est nécessaire que ces annotations soient décrites de la même manière dans les deux langues. (Petrov et al., 2012) définissent un ensemble de 12 étiquettes morpho-syntaxiques universelles à gros grain, communes au plus grand nombre de langues (Universal Tagset). Ces étiquettes universelles sont les suivantes : NOUN (noms), VERB (verbes), ADJ (adjectifs), ADV (adverbes), PRON (pronoms), DET (déterminants et articles), ADP (prépositions et postpositions), NUM (numéraux), CONJ (conjonctions), PRT (particules), « . » (symboles de ponctuations) et X (pour tout ce qui échappe aux autres catégories). Par conséquent, la couche de sortie comporte 12 neurones, chaque neurone correspondant à une étiquette morpho-syntaxique universelle. Il faut donc utiliser la fonction d'activation softmax sur la couche de sortie afin d'obtenir des scores assimilables à des probabilités, le mot w en entrée du réseau est annoté par l'étiquette la plus probable en sortie du réseau.

5. Analyse sur les étiqueteurs existants

5.1. AraMorph

Aramorph est un analyseur morphologique qui est entièrement développé en orienté objet et distribué par le LDC (Linguistic Data Consortium), ce programme informatique est donc une collection de classes indépendantes et d'objets.

AraMorph identifie la plupart des traits morpho-syntaxiques inclus dans le mot graphique. Il segmente les unités lexicales, repère les différents composants et atteste son appartenance à la langue. Pour cela, le système exploite le lexique DIINAR.1 pour éviter les analyses théoriquement possibles et inexistantes dans la langue. Par la suite, l'analyseur donne une liste des traits associés à l'unité lexicale en entrée. Il offre deux types d'options.

Le premier vise les traits morphosyntaxiques, le second concerne l'analyse des préfixes et suffixes. En plus des étiquettes morphosyntaxiques, il donne en sortie d'autres informations comme la base, l'unité lexicale minimale vocalisé ou non ainsi que la forme complète supposée vocalisée ou non.

Analyser les préfixes revient à décrire ses découpages possibles et d'examiner les compositions des clitiques. Ceci amène le système à faire la distinction entre les clitiques ayant la même forme mais appartenant à des catégories syntaxiques différentes

5.2. L'étiqueteur ASVM

C'est un analyseur gratuit développé en perl par l'équipe de Mona Diab en 2004. Il s'agit d'une adaptation à l'arabe du système anglais « Yamcha » qui a été entrainé sur le corpus annoté Treebank, en utilisant le modèle Support Vector Machine et en se basant sur 24 étiquettes.

L'étiqueteur ASVM est réalisé à partir de trois modules qui permettent de générer la sortie attendu :

- TOKrun.pl pour la tokenisation.
- LEMrun.pl pour la normalisation des mots féminins uniquement.
- POSrun.pl pour l'étiquetage.

D'après (Diab, Hacioglu & Jurafsky, 2004) [18] l'évaluation d'ASVM se fait selon le corpus TreeBank arabe qui se compose de 4 519 phrases. Le corpus est distribué comme suit : 119 phrases pour le développement, 400 phrases pour le test et 4 000 phrases pour l'apprentissage.

Les résultats obtenus de SVM-POS sont de 95.49% d'étiquettes correctes.

Par la suite, l'équipe de Mona Diab a réalisé une amélioration de cet étiqueteur (Diab, 2010) [25] sous le nom d'AMIRA 2.0 [6] (non téléchargeable), qui obtient des résultats plus performants au niveau de la segmentation (99.2%) et une précision de plus de 96% au niveau de l'étiquetage.

5.3. L'analyseur AlKhalil

C'est un programme open source, proposé par l'organisation arabe pour la culture et les sciences, développé dans le langage de programmation java, et peut être utilisé sur des défirent environnement (Windows, Linux et Mac OS et Solaris).

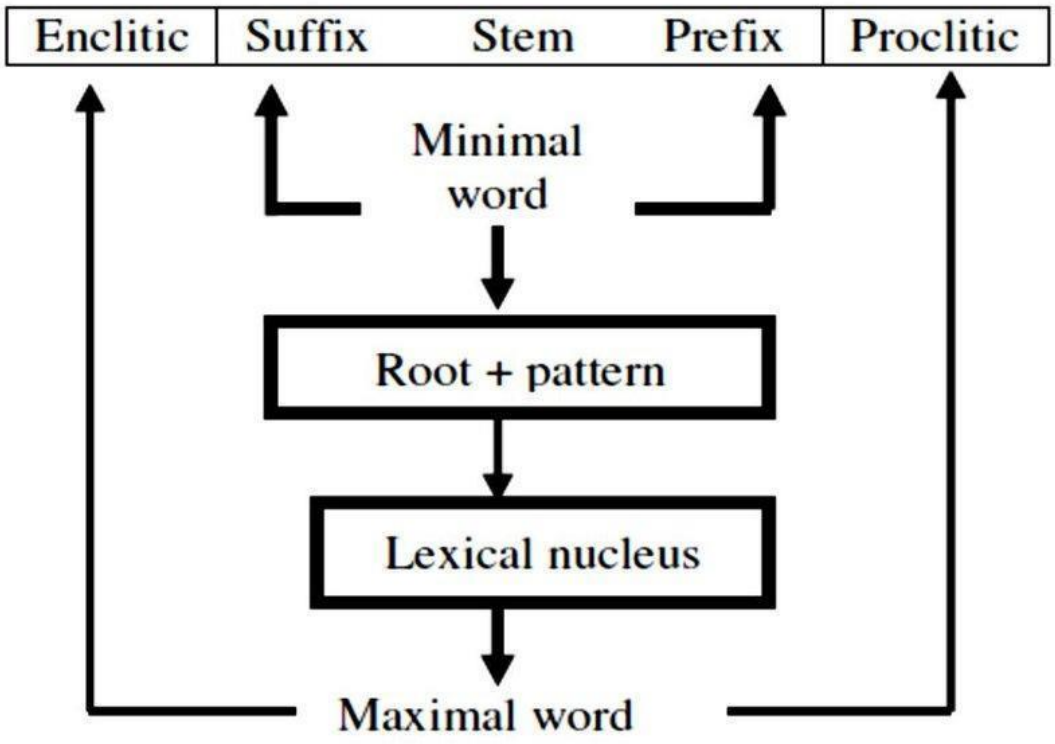


Figure 2.4 : Décomposition d'un mot avec AL Khalil.

[6] http://nlp.ldeo.columbia.edu/amira/

Cet analyseur repose sur une approche de la modélisation d'un très grand ensemble de règles morphologiques arabes et sur l'intégration de ressources linguistiques utiles à l'analyse, telles que la base de données des racines, les schèmes "الأوزان " vocalisés associés aux racines et les tables des proclitiques et des enclitiques. En sortie de l'analyse, le système produit un tableau très informatif contenant principalement la discrétisation du stem, sa catégorie grammaticale, ses racines possibles associées aux motifs correspondants, les proclitiques et les enclitiques

Alkhalil a remporté la première position, parmi les 13 systèmes d'analyse morphologique arabe à travers le monde, à un concours organisé par la ligue arabe pour l'éducation la culture et la science (ALECSO), il est considéré comme le meilleur système morphologique arabe.

5.4. L'étiqueteur APT de Khoja

Il s'agit d'une adaptation à l'arabe du système anglais BNC qui combine des données statistiques et des règles pour déterminer tous les traits morphologiques d'une unité lexicale. Cet analyseur est entrainé à partir d'un corpus contenant 50 000 mots en utilisant 131 étiquettes.

D'après (Khoja, 2001) le nombre élevé de tags permettrait de donner des résultats plus efficaces au niveau de l'étiquetage.

5.5. Etiqueteur de XEROX

La phase de segmentation pour cet analyseur est faite par un transducteur à états finis (Farghaly & Dichy, 2003) [23] en découpant la chaine d'entrée en unités lexicales qui correspondent à une forme fléchie ou une ponctuation, en donnant à chaque segment des étiquettes qui représentent le comportement morphologique de chaque unité lexicale et sa catégorie.

Cet étiqueteur regroupe 4930 racines et 400 modèles qui permettent de produire 90 000 lexèmes. Cet analyseur utilise des règles à large couverture, par contre il génère un taux assez élevé d'ambiguïtés lexicales. Il ne traite pas bien la phase de désambiguïsation.

5.6. L'analyseur de Sakhr

C'est un analyseur synthétiseur morphologique qui fournit l'analyse de base pour tout mot arabe. Cet analyseur couvre toute la langue arabe moderne [7] et classique [8]. L'analyseur identifie toutes les bases possibles d'un mot, c'est-à-dire trouver sa forme de base après l'extraction des suffixes et préfixes. D'après le site web de la société sakhr [9], cet analyseur donne des résultats qui atteignent une précision de plus de 90%.

5.7. L'analyseur de Buckwalter

L'analyseur morphologique de Tim Buckwalter est différent par rapport à d'autres algorithmes, en donnant en sortie un lemme et non pas une racine. Il est basé sur un ensemble de dictionnaires de préfixes (299 entrées), suffixes (618 entrées), lemmes (82,158 entrées) construits manuellement. Les lexiques sont complétés par trois tables de compatibilité entre les : préfixes-suffixes (1648 entrées), préfixes-stem (1285 entrées) et suffixes-stem (598 entrées) pour indiquer leurs combinaisons correcte et bien gérer les analyses. Il considère que le lemme est une forme de base et produit des informations sur les racines, et inclut des marques de voyelles brèves et des signes diacritiques en plus du glossaire anglais et l'étiqueteur grammatical.

L'analyseur est développé par LDC (Linguistic Data Consortium) il utilise son étiqueteur de Penn Arabic Treebank, et le Prague Arabic Dependency Treebank. Il est le plus célèbre parmi les analyseurs arabes et est considéré comme la ressource lexicale la plus respectable dans son genre.

---

7 **Le langage de la presse, de la littérature et de la correspondance formelle.**

8 **La langue du Coran, parlée au VIIe siècle.**

9 **http://www.sakhr.com**

6. Comparaison théorique entre les étiqueteurs morphologiques :

Chacun des analyseurs morphologiques de l'arabe, étudiés dans ce chapitre, offrent des avantages mais aussi souffrent de certains inconvénients. Nous en résumons dans ce qui suit leurs principales caractéristiques.

<table>
<tr><th>Etiqueteurs</th><th colspan="2">Approche</th><th>Langage</th><th>année</th><th>Avantage</th><th>Inconvénients</th></tr>
<tr><td rowspan="6">- Aramorph

- Al Khalil

- ASVM

- Buckwalter

- APT de Khoja

- Sakhr

- XEROX</td><td rowspan="6">Analyseur

Morphologique

à base de

Dictionnaire</td><td rowspan="3">A base

de

Lemme</td><td>JAVA</td><td>2003</td><td rowspan="6">- Basées sur des listes établies préalablement

- Sortie soit un lemme ou racine

- Traitant tous types des mots arabes

- Résultats précis

- Adéquats pour l’apprentissage</td><td rowspan="6">- Gourmands en espace mémoire.

- Charge CPU élevée.

- Non adéquats à la recherche d’information</td></tr>
<tr><td>JAVA</td><td>2000</td></tr>
<tr><td>PERL</td><td>2004</td></tr>
<tr><td>PERL</td><td>2002</td></tr>
<tr><td rowspan="3">A base

de

Racine</td><td>C ++</td><td>2001</td></tr>
<tr><td>JAVA</td><td>2001</td></tr>
<tr><td>C ++</td><td>1996</td></tr>
</table>

Tableau 2.1 : Tableau comparatif des étiqueteurs morphologiques

7. Conclusion

Les différentes recherches effectuées sur l'analyse grammaticale de la langue arabe montrent que c'est une langue très difficile à traiter à cause de l'agglutination et des ambiguïtés graphiques. Dans ce chapitre, nous avons présenté les différentes approches de l'étiquetage grammatical de la langue arabe et nous avons donné une analyse de quelques étiqueteurs existants.

Il nous apparait que l'arabe est une langue très riche morphologiquement, ce qui complique les traitements. Dans le chapitre qui suit, nous expliquerons les difficultés auxquelles nous avons été confrontés pendant la conception et la réalisation de notre analyseur grammatical.

CHAPITRE 3 : Conception et modélisation de la solution

1. Introduction :

Ce chapitre est consacré à la partie conception et modélisation de notre solution. Dans un premier temps, nous allons présenter les ressources utilisés ainsi que la base de données implémentée, enfin nous présenterons l'architecture générale proposée en détaillant les différents modules du programme.

En vue de la problématique que nous avons qui n'est autre que la difficulté pour un élève de maîtriser l'aspect grammatical d'une phrase dans une langue complexe comme l'arabe, le but de notre projet a été de développer une solution pour guider l'élève dans son apprentissage, et lui permettre de corriger et d'apprendre de ses erreurs.

Après une longue et fastidieuse phase de recherche, nous nous sommes penchés sur les différentes approches. Toutes sont différentes avec des avantages et des inconvénients. Mais après avoir analysé la nature de la langue arabe avec ses nombreuses spécificités qui amplifient la difficulté de l'analyse, nous avons fini par opter pour une approche d'étiquetage grammatical symbolique basé sur les schèmes et les règles grammaticales.

2. Programme ARIBLI

2.1 Description fonctionnelle :

Le premier défi auquel nous avons fait face a été le manque de ressources voir l'absence de ces derniers dans la langue arabe, ce qui nous a coûté énormément de temps et d'énergie, mais nous avons fini par avoir le minimum pour pouvoir donner un résultat encourageant.

Si l'on voudrait résumer le fonctionnement de notre programme, il serait préférable de le découper en quelques briques ou modules, qui sont interconnectés, et qui travaillent ensembles :

Et la brique la plus importante pour le bon fonctionnement du programme, reste bien sûr la base de données, et voici un diagramme de classes pour l'illustrer :

Figure 3.1 : Diagramme de classes du programme ARIBLI

Pour une meilleure compréhension du diagramme, nous allons présenter dans la figure suivante des exemples de valeurs que peuvent contenir les champs des tables. Nous ne présenterons qu'une partie de la BDD. La deuxième partie sera présentée avec le module de l'analyse grammaticale.

Suffixe	Exemple
Classe	« C » Pour « Commun »
Description	ضمير المخاطبين
Forme voyéllée	كُمَا
Forme non voyéllée	كما

Préfixe	Exemple
Classe	« N » Pour « Nominal »
Description	ال التعريف + حرف العطف
Forme voyéllée	وَال
Forme non voyéllée	وال

Nom Propre	Exemple
Type	اسم علم
Forme voyéllée	آدَم
Forme non voyéllée	آدم

Racine	Exemple
Forme non voyéllée	بلغ

Schème	Exemple
Type	Nominal
Sous Type	Masdar
Forme voyéllée	فِعال
Forme non voyéllée	فعال

Mot outil	Exemple
Type	أداة استفهام
Forme voyéllée	هَلْ
Forme non voyéllée	هل

Figure 3.2 : Exemples pour les champs des tables de la première partie de la BDD

Cette première partie regroupe les ressources nécessaires au bon fonctionnement de l'analyse morphologique, et voici la description des différentes tables :

- **Préfixes et Suffixes :** Regroupent 64 préfixes et 65 suffixes (pris de l'outil Al Khalil[2]) définis par une classe qui désigne si l'affixe est réservé aux noms, aux verbes ou bien s'il peut être utilisé pour les deux. Une description pour indiquer le rôle de l'affixe, et une forme voyéllée et non voyéllée pour la chaîne de caractère de l'affixe.
- **Nom propre :** Regroupe 8386 noms propres (pris de l'outil Al Khalil) définis par un type pour indiquer si c'est un nom propre ou si c'est une référence à un nom propre, une forme voyéllée et une forme non voyéllée.
- **Racine :** Regroupe 4750 racines (prises d'une base construite par Tim Buckwalter[3]) en tant que simples chaînes de caractères.
- **Mot Outil :** Regroupe 285 mots outils (prises de l'outil Al Khalil) définis par un type pour décrire le rôle du mot outil, et les formes voyéllées et non voyéllées.
- **Schème :** Regroupe une liste non exhaustive de 655 schèmes puisqu'elle ne comporte pas des schèmes comme ceux des verbes au présent ou au futur, la liste a été prise d'un fichier proposé par Mohamed Attia sur Source Forge[4], et que nous avons prétraité pour traduire les schèmes à partir d'une translittération de Buckwalter[5] en arabe puis nettoyé d'informations dont nous avons pas besoin. Chaque schème à un type pour désigner la catégorie grammaticale du mot, un sous type pour donner une information sur l'aspect grammatical, et les formes voyéllées et non voyéllées.

Ces ressources sont essentielles à l'analyse morphologique que nous allons détailler ci-dessous.

[2] Lien vers le projet sur Source Forge : https://sourceforge.net/projects/alkhalil/

[3] Liste des racines : https://www.angelfire.com/tx4/lisan/roots1.htm

[4] Lien vers le fichier sur Source Forge : https://sourceforge.net/projects/arabicpatterns/

[5] La translitération Buckwalter permet de faire correspondre chaque lettre (son) arabe à un caractère de langue anglaise.

2.2 Description de l'analyse morphologique

La figure suivante illustre le schéma du fonctionnement de l'analyse morphologique :

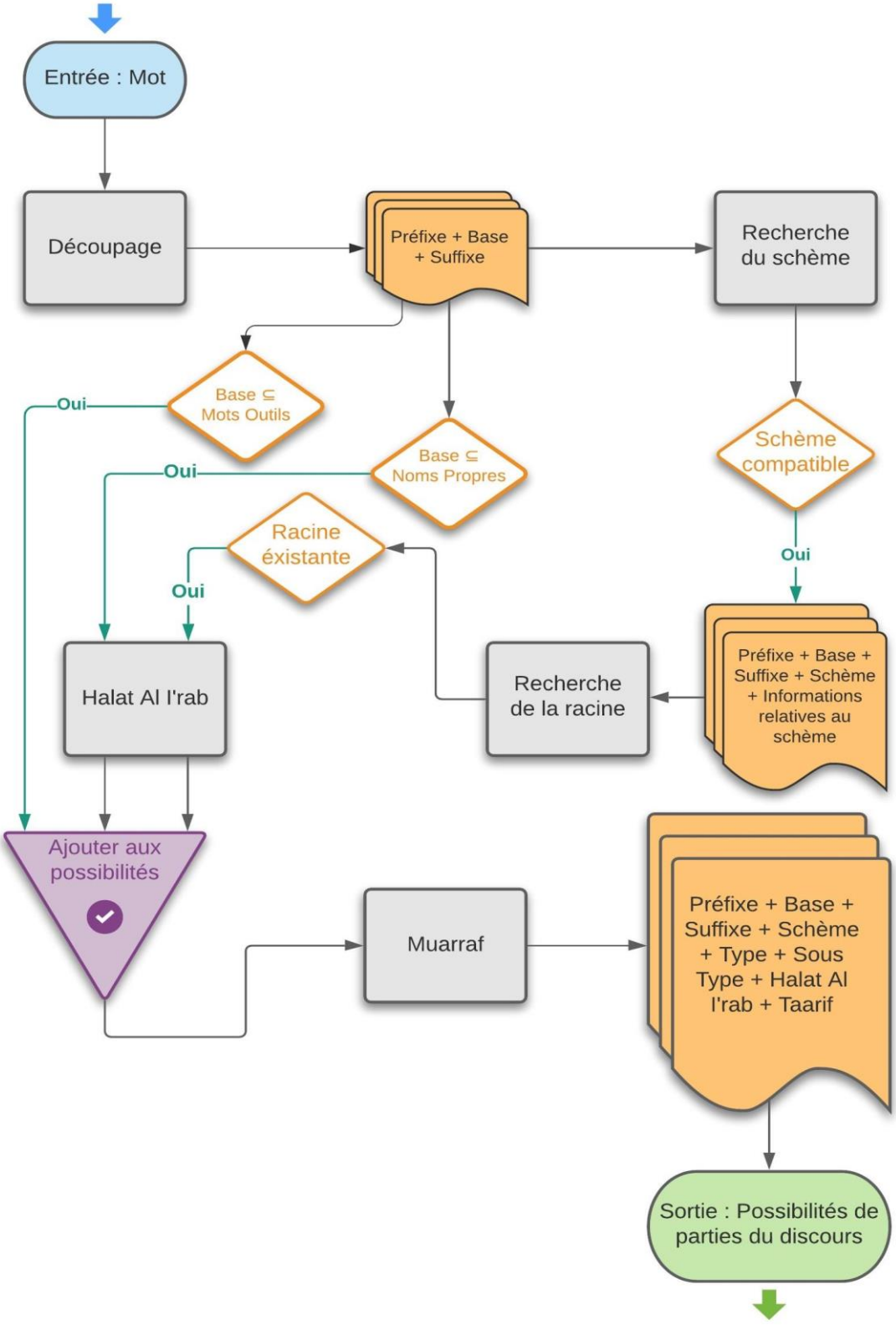


Figure 3.3 : Schéma générale de l'analyse morphologique

L'analyse morphologique prend en entrée un mot arabe voyellé de préférence, pour permettre le bon choix du schème que suit le mot (si c'est un mot flexionnel) et donne en sortie toutes les possibilités de catégories grammaticales d'après la forme et la voyellation du mot. Chaque module ci-dessous fait un travail précis, et l'analyse morphologique fait appel à tous les modules pour analyser un mot donné.

Pour une meilleure compréhension, nous allons prendre un exemple que nous allons suivre au fil des étapes :

- **Exemple :** «. صَدَقَ الرَسُولُ في كَلَامِهِ ».

2.2.1 Découpage : Pour chaque mot de la phrase, nous procédons à son découpage en (Préfixe, Base, Suffixe), le processus commence par l'analyse du début du mot graphique en le comparant avec les préfixes de la BDD, puis la fin avec les suffixes.
En sortie de ce module, nous nous retrouverons avec plusieurs combinaisons possibles pour chaque mot.

Pour le mot : 'كَلَامِهِ'

Combinaisons :

1 - Base : 'كَلَامِهِ', Préfixe : /, Suffixe : /

2 - Base : 'كَلَامِ', Préfixe : /, Suffixe : ه

3 - Base : 'لَامِهِ', Préfixe : كَ, Suffixe : /

4 - Base : 'لَامِ', Préfixe : كَ, Suffixe : ه

Nous remarquons qu'il y a deux combinaisons avec le préfixe « ك », et c'est évidemment une mauvaise interprétation de découpage du mot par le module. Mais ces fausses combinaisons seront éliminées au fur et à mesure des étapes.

2.2.2 Détection des entités spéciales (mots outils et noms propres) : Le module prend en entrée la base d'un mot après être découpé en (préfixe, base et suffixe), et essaie de détecter si la base est un mot outil ou bien un nom propre, la détection se fais grâce à une requête de sélection auprès de la BDD.

Dans notre exemple, le mot « في » est détecté en tant que mot outil après être découpé en (préfixe : /, base : « في », suffixe : /.

2.2.3 Recherche des schèmes : Pour retrouver le schème d'un mot, on va parcourir pour chaque combinaison (en sortie du module de découpage) tous les schèmes de la BDD qui ont la même taille par rapport à la base du mot.

Ensuite, nous vérifions si dans la position des lettres radicaux du schème (c'est-à-dire différents de **[**'ف', 'ع', 'ل'**]**) on retrouve les mêmes caractères de la base à la même position.

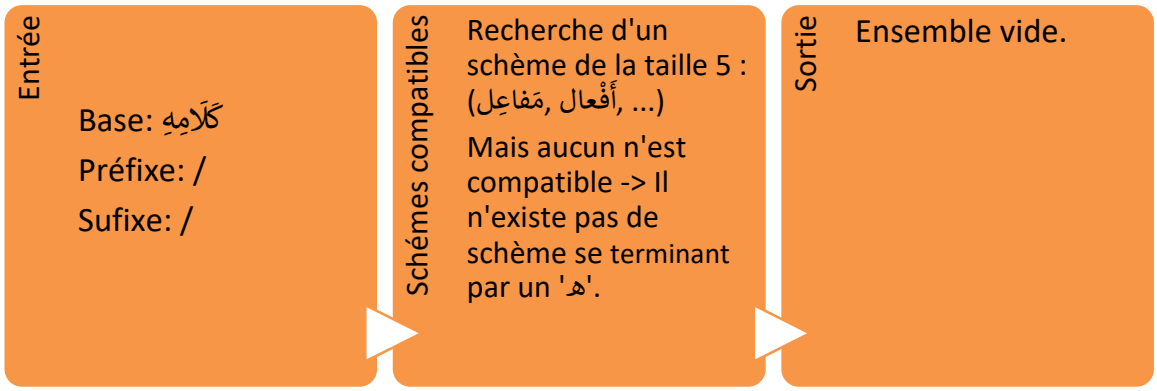
Puis, nous vérifions la voyellation des deux côtés, s'ils ont une voyellation compatible on considère que le schème est compatible avec cette base, sinon on élimine la combinaison, et on a déjà pour chaque combinaison une information grammaticale sur le mot (nom, verbe, etc…) grâce aux informations portées par le schème détecté.

Une voyellation compatible veut dire qu'il n y a pas une voyellation différente à une même position des deux côtés : Une voyellation partielle d'un mot A est compatible avec une voyellation entière d'un mot B si la voyellation à une position i dans les deux mots n'est pas différente.

Exemple : « فَعَل » est compatible avec « قَلَمٌ », puisque quand la voyellation est présente (lettres n°1 et n°2 du schème), elle est identique à la voyellation du mot « قَلَمٌ ».

Nous procédons ensuite à une batterie de tests, pour éliminer les schèmes verbaux si les affixes sont d'une classe nominale, et les schèmes nominaux si les affixes sont d'une classe verbale. Voici une petite illustration sur le fonctionnement simplifié du module :

Pour la combinaison (1 - Base : 'كَلَامِهِ', Préfixe : /, Suffixe : /) :



Pour la combinaison (2 - Base : 'كَلَامِ', Préfixe : /, Suffixe : ه) :

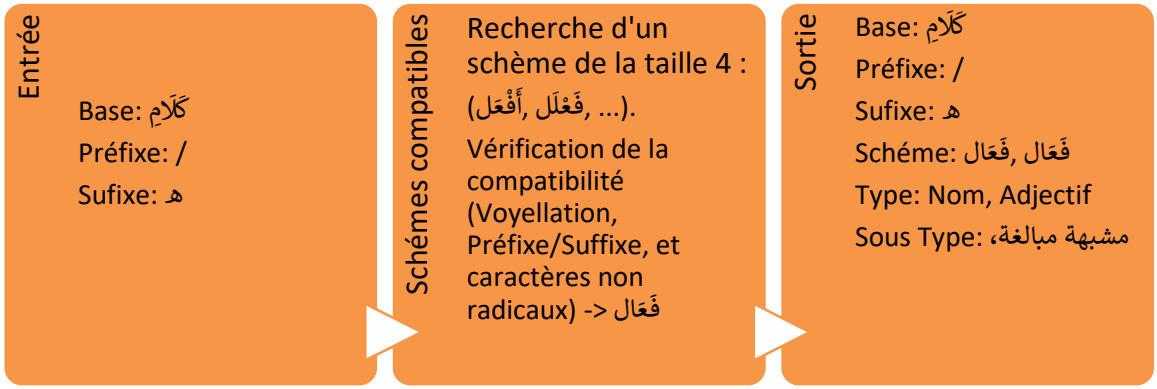


Figure 3.4 : Illustre la recherche du schème du mot

Nous remarquons que nous nous retrouvons avec deux possibilités de schèmes pour le même mot (combinaison), avec la même forme et la même voyellation mais qui différent dans le rôle grammatical (type, et sous type). Donc, nous filtrons les mauvaises combinaisons en sortie du module de découpage, mais nous créons d'autres combinaisons à cause de l'incapacité à choisir le bon schème.

De plus, parfois ce module véhicule en sortie d'autres schèmes erronés, à cause des combinaisons de découpage reçus en entrée (détection d'un préfixe qui en réalité fait partie de la base, ou au contraire détection d'une base incluant un préfixe qui va être ignoré). Comme par exemple à la combinaison n°4 ou nous avons en sortie un découpage [Base : 'لَامِ', Préfixe : كَ, Suffixe : ه], l'analyse détectera un schème erroné qui est : « فَاع » (le schème existe mais ce n'est un schème adapté au mot). D'où le besoin d'utiliser le module suivant.

2.2.4 Recherche de la racine : Après l'obtention des combinaisons avec des schèmes compatibles, on peut à présent vérifier si le mot est juste et accepté dans l'Arabe Standard tout simplement en vérifiant si la racine existe dans la base de données, il faut cependant différencier racine et base, la base dans notre cas est le mot nettoyé des préfixes et suffixes, la racine par contre ne suit pas un schème, elle est d'ailleurs non voyéllée, et généralement à trois caractères seulement. Pour extraire la racine à partir de la base nous supprimons les caractères non radicaux (d'après leur position dans le schème) ainsi que la voyellation.

Exemple : Si l'on suit l'exemple précédent, nous avons en entrée :

1- **[Base :** « كَلَامِ », **Schème :** « فَعَال »**]**, après suppression de la lettre **n°3** « ا » de la racine, vu que c'est une lettre non radicale dans le schème, nous aurons comme racine : « كلم » qui est présente dans la base des racines. Et la combinaison est acceptée.

En revanche concernant :

2- **[Base : « لَامِ », Schème : « فَاع »],** après suppression de la lettre **n°2** « ا », nous aurons comme racine : « لم » qui n'est pas présente dans la base des racines. Et la combinaison est rejetée

2.2.5 Muarraf : Un module indépendant qui permet de déterminer si un nom est **« معرف » (défini),** cela se fais grâce à une batterie de tests sur les informations récoltés sur le mot, le plus simple exemple reste le test sur les préfixes comportant un **« ال ».** La fonction en charge de ce module n'est pas parfaitement précise, l'identification de l'information de **« التعريف »** est liée à un tas de paramètres linguistiques que nous ne maitrisons malheureusement pas.
Un exemple assez compliqué à mettre en place est l'identification par ajout d'un nom défini (الاسم المضاف إلى اسم معرف) comme dans la phrase : « كُتُبُ زَيْدٍ », ici le mot « كُتُبُ » est identifié, mais cette information ne peut être détectée que dans l'analyse du mot suivant.

2.2.6 Halat al i'rab : Un module indépendant qui donne « الحالة الإعرابية » d'un nom ou un verbe, et encore une fois, un module qui doit être développé et enrichis par les tests linguistiques nécessaires. Pour le moment, nous identifions l'information grâce à la voyellation de fin du mot, plus quelques cas spéciaux comme par exemple pour le duel d'un nom : « المثنى وما يلحق به : يرفع بالألف ، وينصب ويجر بالياء ». Mais les cas spéciaux sont nombreux, ce qui complique le traitement.

Récapitulatif de l'analyse morphologique :

Le premier traitement effectué sur le mot est le découpage en (préfixe, base et suffixe). Après avoir découpé le mot et avoir obtenu les combinaisons possibles, nous testons pour chaque combinaison si la base existe dans la BDD en tant que Mot Outil puis en tant que Nom Propre, si elle existe, nous ajoutons la possibilité à une liste que l'on va retourner à la toute fin des tests et qui représente toutes les possibilités du rôle grammaticale du mot.

Ensuite nous allons passer les combinaisons en sortie du module de découpage en entrée du module de recherche du schème, où nous allons détecter tout schème compatible avec la base (même taille, lettres non radicales identiques, et voyellation compatible).

Nous nous retrouvons encore avec plusieurs combinaisons que nous allons filtrer cette fois-ci avec le module de recherche de la racine, en testant pour chaque combinaison si la racine extraite à partir de la base (en respectant le schème) existe dans la BDD. En sortie, nous aurons plusieurs combinaisons.

Nous passons chaque sortie ou possibilité au module de détection de " الحالة الإعرابية" sauf ceux qui représentent des mots outils. Et de la même manière nous passons chaque possibilité au module de détection de " التعريف".

Pour enfin retourner les possibilités détectées pour le mot en entrée.

2.3 Description de l'analyse grammaticale et de l'enrichissement

Voici des exemples de la deuxième partie de la base de données ci-dessous :

Etat	Exemple
Label	مفعول به منصوب وعلامة نصبه الفتحة الضاهرة
Début	Vrais/Faux
Fin	Vrais/Faux

Transition	Exemple
Etat de provenance	Nominal
Etat de destination	Masdar
Préfixe du mot	… / حرف العطف / حرف جر
Suffixe du mot	… / ضمير الغائبين / ضمير المخاطب
Type du mot	فعل من الشكل 1 / اسم/ صفة/ اسم علم/ فعل من الشكل 2/…
Sou type du mot	إسم مفرد / إسم مؤنث/ إسم مكان/ مشبهة/ مبالغة/ إسم أداة/…
حالة الإعراب	الرفع / النصب / الجر / الجزم
Identifié	Vrais/Faux
Occurrence de la transition	[0 : +∞]

Figure 3.5 : Exemples pour les champs des tables de la deuxième partie de la BDD

La figure suivante illustre le fonctionnement général du programme ARIBLI avec l'analyse grammaticale et l'enrichissement :

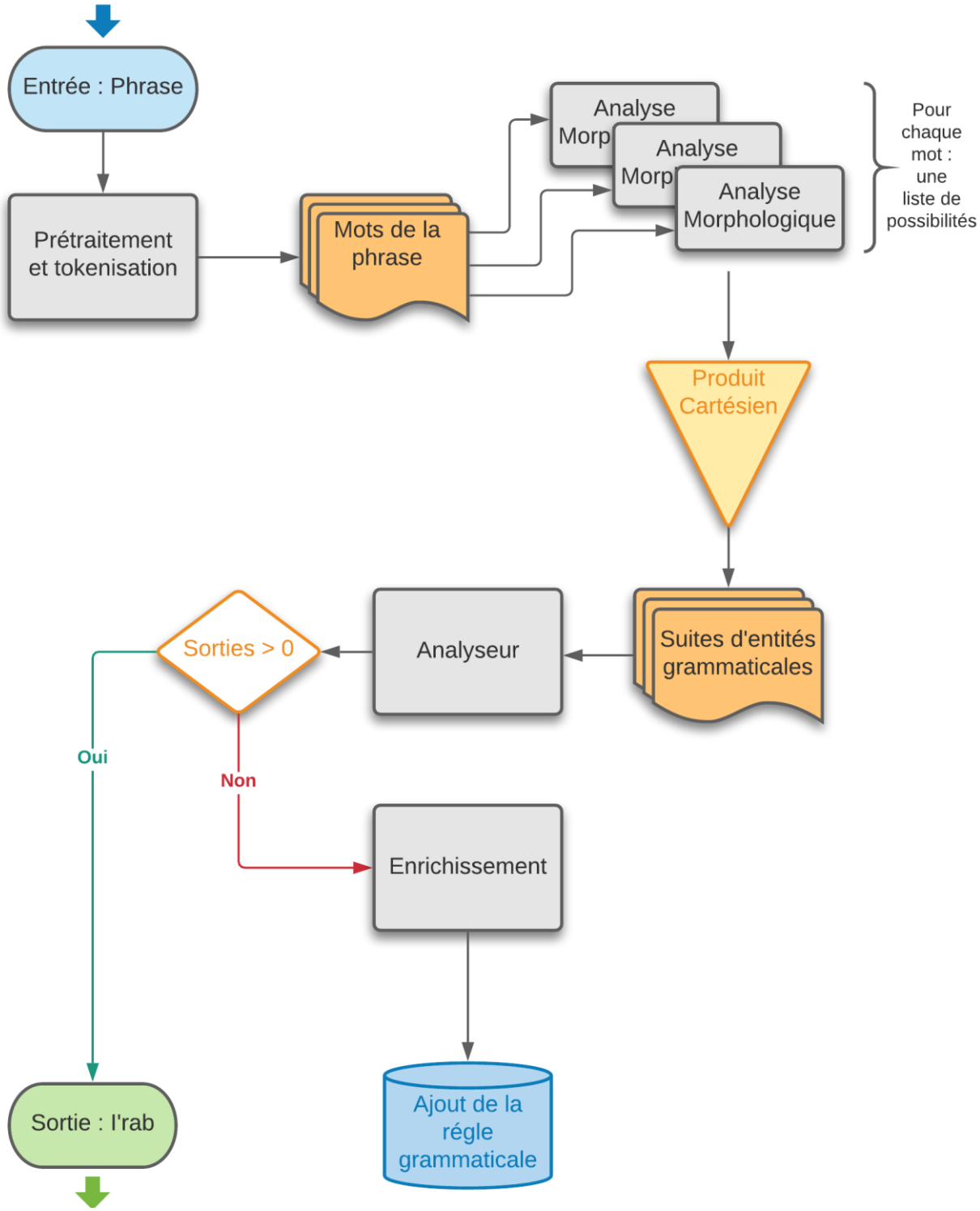


Figure 3.6 : Schéma général de l'analyse grammaticale et de l'enrichissement

Et nous continuons avec les modules restants :

2.3.1 Analyseur : Certainement le module essentiel pour pouvoir identifier réellement ce que nous cherchons (الإعراب).

Pour pouvoir réaliser un analyseur qui puisse nous donner le résultat recherché, nous avons besoin en premier lieu d'un formalisme de grammaire pour pouvoir décrire des règles grammaticales.

Cependant nos recherches nous ont menés vers le constat que la complexité des formalismes qui sont certes plus adaptés pour la langue arabe nuirait à la phase de développement où l'on va devoir travailler sur beaucoup de règles grammaticales.

Il faut préciser que ces règles doivent être compatibles avec le reste des informations extraites des autres modules, et donc il faudrait à la fois maitriser l'un de ces formalismes puis procéder à l'intégration des ressources et de nos modules pour les adapter au formalisme.

Mais en réalité, un mécanisme basique qui pourrait facilement s'adapter à une grande partie des caractéristiques de la langue arabe n'est autre qu'un automate à états fini.

Cependant il reste encore un problème, un analyseur comme nous l'avons dits, a besoin de règles grammaticales, et le manque de ressources auquel nous avons été confronté nous a forcé à aller vers une approche différente qui pourrait résoudre le problème.

Nous avons pensés à une solution qui nous donnerait tout de même une analyse sans avoir à entrainer un modèle au préalable sur un grand corpus annoté. La solution en question se résume à l'enrichissement de l'automate au fur et à mesure de son utilisation.

Avant d'aborder l'approche d'enrichissement nous allons rappeler ce qu'est un automate et comment il fonctionne :

Définition d'un automate :

« De façon très informelle, un automate est un ensemble "d'états du système", reliés entre eux par des "transitions" qui sont marquées par des symboles. Étant donné un "mot" fourni en entrée, l'automate lit les symboles du mot un par un et va d'état en état selon les transitions. Le mot lu est soit accepté par l'automate soit rejeté… Un automate est définis par un quintuplé (Q, Σ, δ, q0, F) constitué des éléments suivants :

- Un alphabet fini (Σ)
- Un ensemble fini d'états (Q)
- Une fonction de transition ($\delta : Q * \Sigma \rightarrow Q$)
- Un état de départ ($q0 \in Q$)
- Un ensemble d'états finaux (ou acceptant) $F \subseteq Q$ »

Dans le contexte d'un analyseur qui devra donner une analyse spécifique **(**إعراب**)** :

- L'alphabet est l'ensemble des possibles parties du discours d'un mot arabe (par exemple : « un nom, masculin, identifié, et marfu' », « un verbe, type 1, مبني على الفتح» ou «un mot outil, حرف جر»).

- Un mot est donc une suite de parties du discours dans une phrase, ce qui représente une règle grammaticale.

- Les états en plus de leur nécessité dans un parcours de l'automate portent l'information finale dont nous avons besoin qui n'est autre que (الإعراب) qui est stockée dans une chaine de caractères.

- La fonction de transition est l'ensemble de transitions qui définissent les pas dans une règle grammaticale
  Un exemple de transition : « START » -> (nom, identifié, marfu') -> « مبتدأ ».
  Ce qui se traduit en : « A partir de l'état initial si nous trouvons dans le premier mot un nom, identifié et marfu' l'état courant devient un deuxième état qui décrit l'analyse d'un « مبتدأ ».
  Dans l'exemple précédent, les informations (nom, identifié, marfu') ne sont qu'une partie des informations qui définissent une transition, il reste encore la catégorie du préfixe et celle du suffixe s'ils existent, et surtout « Halat al i'rab ».

Ce qui donne une meilleure précision dans le parcourt de l'automate et donc de l'analyse.

- L'état de départ est un état unique nommé « START » qui désigne concrètement le début d'une phrase.

- L'ensemble d'états finaux est l'ensemble d'états pouvant accepter d'être à la fin d'une phrase.

Un automate décrivant des règles grammaticales de la langue arabe ne peut être déterministe. En effet, la langue arabe comme nous l'avons dit dans le premier chapitre, comporte beaucoup d'ambiguïtés. Ces ambigüités peuvent être levés par une compréhension sémantique poussée car il faudrait prendre en compte le contexte de la phrase, et parfois le contexte de l'auteur, de sa localité, du temps… etc

2.3.2 Enrichissement des règles grammaticales: Nous avons précisés que les phrases en entrée de notre système sont considérés comme justes, et au vue de cette considération, le simple fait de faire appel à l'analyseur sur une phrase donnée, permettrais de donner de possibles résultats d'analyse si elles existent déjà sur la base, sinon d'enrichir cette base en demandant un peu d'aide à l'utilisateur pour une meilleure interprétation de la phrase par le système.

Le résultat de l'analyseur sera une liste de propositions pour une analyse juste, à cause de la multitude des sorties au niveau morphologique, ce qui veut dire que la précision du résultat n'est pas très bonne. Pour remédier à ce problème, nous avons intégrés l'ajout d'un paramètre dans les transitions de l'automate qui pourra pondérer les règles grammaticales et permettre à l'analyseur de prendre une décision au moment où il se retrouve avec plus d'un résultat en sortie.

Concrètement, la partie d'enrichissement se résume à l'ajout de transitions dans l'automate si elles n'existent pas encore. L'utilisateur pourra guider le système en choisissant parmi les potentiels transitions (partis du discours détectés en sortie de l'analyse morphologique), ce qui donnera plus de précision car sans cela, nous auront à ajouter une transition et un état pour chaque possibilité, et nous nous retrouverons avec des règles fausses.

Ainsi nous évitons d'augmenter le nombre de résultats probablement faux. D'autre part, toujours dans l'objectif d'améliorer la précision, la pondération que nous avons abordés en haut prendra comme information le nombre de fois ou la règle grammaticale est proposée pour une suite donnée d'analyses morphologiques et validée par un expert.

Exemple : Toujours avec la phrase « صَدَقَ الرَسُولُ في كَلَامِهِ ».

Nous admettons qu'il existe déjà dans l'automate une règle grammaticale qui reconnais les deux premières entités de la phrase « صَدَقَ الرَسُولُ » qui font référence à un verbe au passé suivis d'un nom et l'analyse de la règle est : « فعل ماضي – فاعل مرفوع », l'état de l'automate avant l'analyse de la phrase est décrit dans la figure suivante :

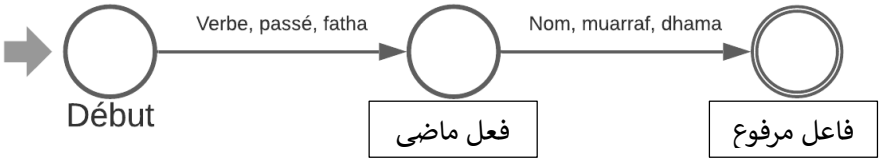


Figure 3.7 : Représentation de l'automate avant l'analyse de la phrase

Après avoir analysé toute la phrase, le but de l'enrichissement est de rajouter les états et les transitions non existantes pour qu'il puisse reconnaître la règle grammaticale dans une prochaine analyse. Voici l'état de l'automate après l'enrichissement :

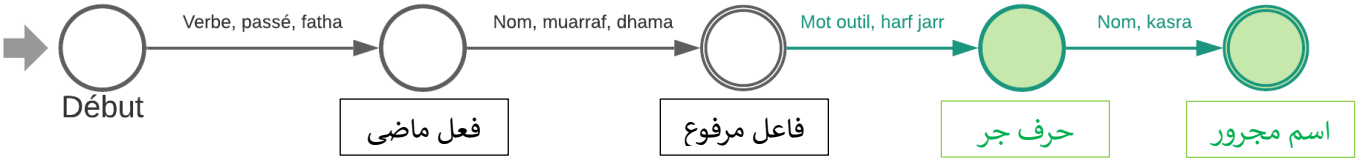


Figure 3.8 : Représentation de l'automate après l'enrichissement

Ainsi, chaque fois que l'analyseur prend en entrée une phrase avec la règle (« Verbe, passé, fatha », « Nom, muarraf, dhama », « Mot outil, harf jarr », « Nom, kasra ») il reconnaitra la règle, et donnera « الإعراب » qui est stocké dans les états de la règle.

3 Récapitulatif :

A présent, nous avons vus tous les modules avec leur fonctionnement détaillé : nous allons ici résumer l'approche du programme :

Nous rappelons que l'entrée du programme est une phrase arabe syntaxiquement correcte, et le but du programme est d'analyser la phrase, ses entités, et l'enchaînement de ses derniers pour donner en sortie : (الإعراب). Le programme sera capable d'analyser une phrase non voyéllée mais donnera un résultat ambiguë.

Après un prétraitement et le découpage de la phrase en mots, chaque mot sera analysé et étiqueté par rapport aux potentielles catégories grammaticales auxquelles il pourrait appartenir (Analyse morphologique). Ensuite, nous procédons à la distribution des possibles catégories de chaque mot sur toute la phrase avec un produit cartésien, pour avoir une liste de suites de catégories grammaticales, puis nous allons tester chaque suite avec l'analyseur pour tenter de reconnaître la règle grammaticale que suit la phrase en entrée.

Le programme pourra donner plusieurs résultats qui seront ordonnés par pertinence. En effet, les utilisateurs avec un certain type pourront choisir entre les résultats pour ainsi incrémenter la note d'une règle grammaticale par rapport aux autres quand l'analyseur propose plusieurs résultats.

A défaut de ne pas reconnaître la règle grammaticale que suit l'enchaînement des entités de la phrase en entrée, le but est de faire apprendre au programme la règle en question avec une étiquette représentant (الإعراب) pour chaque entité (rôle grammatical).

Pour cela le seul moyen est de faire appel à un expert, pour nous l'expert serait sans doute un professeur d'arabe. Ce choix est motivé par la maîtrise du domaine par un professeur, mais aussi par la nature de la problématique posée au départ. En effet, la solution que nous proposons profitera aux élèves pour leur permettre d'apprendre à maitriser le processus de (الإعراب) qui est directement lié à la grammaire de la langue, ce qui les aidera dans leur apprentissage de la voyellation adéquate pour chaque entité de la phrase. Et le but d'un professeur d'arabe est de faciliter cet apprentissage et de le rendre accessible voir divertissant pour ses élèves.

4 Conclusion :

Nous avons vu dans ce chapitre la conception du programme avec l'approche que nous avons choisie, une approche symbolique basée sur les règles grammaticales implémentée à l'aide d'automates à états finis. On peut dire que l'approche est hybride puisque l'on trouve dans la solution un aspect statistique avec la mise en place d'une pondération des règles, mais l'apprentissage est un apprentissage continu, qui n'est pas réalisé au préalable sur un corpus d'apprentissage.

Cette approche prend les avantages des deux types d'étiqueteurs, et l'adapte au besoin du projet qui est assez différent de l'utilisation des étiqueteurs grammaticaux à la base, puisque la sortie recherché n'est pas une simple étiquette grammaticale, mais un (إعراب), les deux sont intimement liés mais restent pour autant différents. Notamment dans les ressources, car il n'existe pas à notre connaissance un corpus annoté avec des analyses grammaticales de type (إعراب).

Dans le chapitre suivant, nous aborderons l'implémentation de la solution, avec les outils utilisés et nous présenterons l'interface du programme avec un déroulement basique.

CHAPITRE 4 : Application et résultats

1. Introduction :

Dans ce dernier chapitre, nous allons présenter la partie implémentation de notre projet. Nous irons tout droit d'abord à la présentation de l'environnement de développement en décrivant le langage de programmation utilisé ensuite détailler les différents outils et ressources utilisés, puis nous montrerons les différentes étapes de déroulement de notre application, l'interface graphique ainsi que les résultats obtenus.

2. Langage Python

Python est un langage de programmation. Il est l'un des langages de programmation les plus intéressants du moment. Facile à apprendre. Il s'agit d'un langage de programmation interprété, qui ne nécessite donc pas d'être compilé pour fonctionner. Un programme **« interpréteur »** permet d'exécuter le code Python sur n'importe quel ordinateur. Ceci permet de voir rapidement les résultats d'un changement dans le code. En revanche, ceci rend ce langage plus lent qu'un langage compilé comme le C.

Python a été créé au début des années 1990 par Guido Van Rossum au CWI (centre national de recherches) aux Pays-Bas dans le cadre du projet Amoeba, un système d'exploitation distribué qui avait besoin d'un langage de script. Guido faisait précédemment partie de l'équipe en charge de développer le langage ABC dans ce même laboratoire, et Python en est fortement inspiré.

Python a ensuite été rendu public, pour devenir un logiciel libre en 2000 (version 1.6.1) pour évoluer jusqu'à la version 3.9.0 actuelle. Son devenir est toujours géré par Guido, entouré d'une équipe de contributeurs, et ses intérêts protégés par la Python Software Foundation, association créée en 2001 et détentrice du langage.

Python dispose d'un type dynamique de système automatique de gestion de la mémoire. Il prend en charge plusieurs paradigmes de programmation, y compris orienté objet, impératif, fonctionnel et procédural, et dispose d'une grande bibliothèque complète et standard.

2.1 Caractéristiques du langage Python :

- Python est portable, non seulement sur les différentes variantes d'Unix ou de Windows, mais aussi sur les autre systèmes d'exploitation : MacOS, BeOS, NeXTStep, MS-DOS.

- Python est libre et gratuit même pour les usages commerciaux.

- La syntaxe de Python est très simple et, combinée à des types de données évolués (listes, dictionnaires,...), conduit à des programmes à la fois très compacts et très lisibles.

- Python gère ses ressources (mémoire, descripteurs de fichiers...) sans intervention du programmeur, par un mécanisme de comptage de références.

- Il est introspectif. un grand nombre d'outils de développement, comme le debugger ou le profiler, sont implantés en Python lui-même.

- Il est réflectif. il supporte la métaprogrammation, par exemple la capacité pour un objet de se rajouter ou de s'enlever des attributs ou des méthodes, ou même de changer de classe en cours d'exécution.
- Python est orienté-objet. Il supporte l'héritage multiple et la surcharge des opérateurs. Dans son modèle objets, et en reprenant la terminologie de C++, toutes les méthodes sont virtuelles.

- Python intègre, comme Java ou les versions récentes de C++, un système d'exceptions, qui permettent de simplifier considérablement la gestion des erreurs.

- La bibliothèque standard de Python, et les paquetages contribués, donnent accès à une grande variété de services : chaînes de caractères et expressions régulières, protocoles Internet (Web, News, FTP, CGI, HTML...), persistance et bases de données, interfaces graphiques.

3 PyCharm

- PyCharm, est un environnement de développement intégré (IDE) utilisé pour la programmation en Python. Il fournit une analyse de code, un débogueur graphique, un testeur d'unité intégré, une intégration aux systèmes de contrôle de version (VCS) et prend en charge le développement Web avec Django. PyCharm est développé par la société tchèque JetBrains.
- Il fonctionne sur plusieurs plates-formes Windows, Mac OS X et Linux. PyCharm a une édition professionnelle, publiée sous une licence propriétaire et une édition communautaire.
- PyCharm permet de compléter le code de manière intelligente, d'inspecter le code, de mettre en évidence à la volée les erreurs et de les corriger rapidement, ainsi que la refacturation du code automatiquement et offre des fonctionnalités de navigation avancées.
- Il s'intègre à IPython Notebook, dispose d'une console Python interactive et ainsi que de nombreux packages scientifiques, notamment matplotlib et NumPy.
- Outre Python, PyCharm offre un support de premier ordre pour divers frameworks de développement Web Python, des langages de gabarit spécifiques, JavaScript, CoffeeScript, TypeScript, HTML / CSS, AngularJS, Node.js, etc…

4. Analyse grammaticale d'un texte en langue arabe :

4.1 Choix du langage de programmation :

Nous avons choisi le langage python parce qu'il nous a permis de traiter les chaines de caractères d'une façon simple et efficace grâce à des fonctions prédéfinis. De plus, python reste le langage le plus prisé pour le TAL ce qui explique le nombre important de bibliothèques traitant le domaine.

4.2 Matériel utilisé :

Le matériel hardware utilisé pour le développement et les tests est résumé dans le tableau ci-dessous :

Machine 1	Machine 2
Nom de l'ordinateur : Acer Aspire E 15 Processeur : Intel(R) Celeron N2940 QuadCore @ 1.83Ghz Mémoire installée (RAM) : 4.00 Go Type du système : Système d'exploitation 64 bits Windows 10 professionnel	Nom de l'ordinateur : LENOVO - PC Processeur : Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz 2.70 GHz Mémoire installée (RAM) : 4.00 Go Type du système : Système d'exploitation 64 bits Windows 10 professionnel

Tableau 4.1 : Matériel hardware utilisé lors du développement et des tests

4.3 Bibliothèques utilisés :

Pour réaliser notre analyseur, nous avons fait appel aux packages suivants :

- **PyArabic**

  C'est un package python installable facilement via la commande « pip install », et qui regroupe un tas de ressources et de méthodes pour la gestion de texte arabe. En effet, le premier défi qui nous a fait face, est le traitement des caractères arabes voyéllés, et cette bibliothèque disponible sur ce lien et développé par le professeur Taha Zerrouki, un chercheur algérien passionné par le TALA, nous a grandement facilité le développement de quelques spécificités de la langue arabe.

- **Itertools**

  Nous avons fait appel à cette bibliothèque pour une seule méthode qui réalise le produit cartésien d'une liste de listes.

  Nous en avons eu besoin pour avoir le produit cartésien des résultats des différentes analyses de chaque mot d'une phrase, pour en faire une liste de compositions, ou chaque composition sera une suite de parties du discours.

- **Django**

  C'est un Framework réputé pour permettre une mise en production d'une application web le plus rapidement possible, qu'elle soit grande ou petite, avec un tas d'avantages permettant de gagner beaucoup de temps. Notamment la maintenabilité, la popularité du Framework et donc une grande communauté très active, une mise en place très facile d'une interface d'administration, et surtout un ORM complet permettant une communication avec la base de données…

  De plus, nous avons eu à travailler auparavant sur le Framework ce qui justifie le choix de ce dernier.

- **SQLite**

  Une bibliothèque écrite en C qui offre un moteur de base de données complet et très léger, c'est une bibliothèque standard de python, et il est intégré de base avec Django. SQLite est le moteur de base de données le plus utilisé au monde, son avantage est qu'il n y a pas besoin d'un serveur pour faire tourner le moteur de base de données, un simple fichier binaire contient toute la base de données.

Les versions utilisées sont disponibles sur un fichier dans le repository GitHub du projet.

4.4. Description de l'interface graphique d'ARIBLI :

Dans cette section nous allons présenter l'interface graphique de notre système.

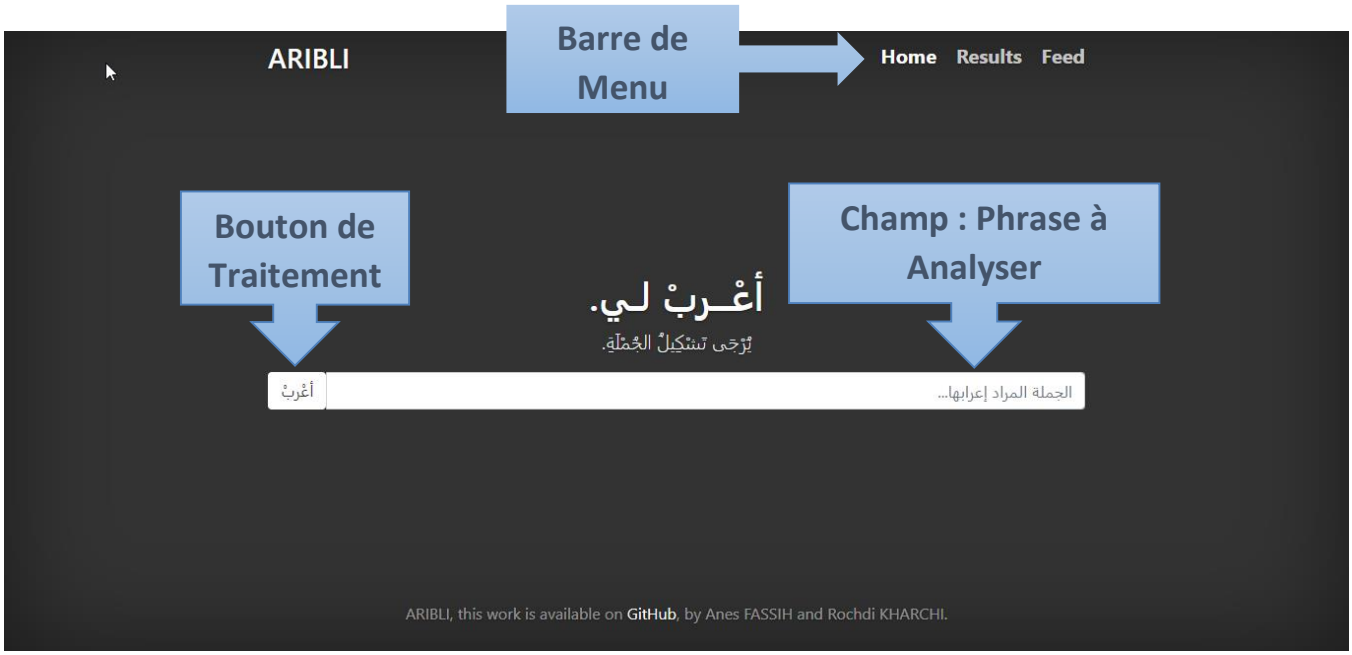


Figure 4.1 : Interface graphique du système « ARIBLI »

La fenêtre principale de notre analyseur est représentée dans la **Figure 4.1** . Elle contient les éléments suivants :

- Une zone de texte qui permet de saisir la phrase à Analyser.
- Une barre de Menu qui représente les trois interfaces de notre application :
  - *Home*
  - *Results*
  - *Feed*
- Un bouton de Traitement (Analyse Grammaticale)

Bouton	Fonction
أعْرِبْ	*Attribuer à chaque entité de la phrase le rôle grammatical qui lui convient.*

Tableau 4.2 : Description du bouton de traitement.

4.5. Déroulement

Dans ce qui suit, nous allons expliquer les différentes étapes de déroulement de notre système qui nécessite tout d'abord en entrée une phrase en langue arabe syntaxiquement correcte et voyellée de préférence, le programme pourra identifier en sortie le rôle grammatical de chaque entité de cette phrase ou bien le cas échéant, proposer une approche interactive d'enrichissement.

Étape 1 : Saisie du texte arabe

Dans la zone de texte nous allons introduire la phrase à analyser, elle doit être correcte du point de vue syntaxique et voyellée de préférence.

La figure ci-dessous explique cette étape :

Figure 4.2 : Saisie du texte

Étape 2 - 1 : Analyse reconnue

Après la saisie d'une phrase, trois scénarios peuvent se produire, le premier que nous allons expliquer ici est celui ou l'analyseur trouve une règle grammaticale qui convient à la phrase donnée en entrée.

En conséquence un résultat sera affiché dans l'interface intitulée **« Results »** comme dans la figure ci-dessous :

Figure 4.3 : Interface Results

Dans le contexte d'un apprentissage continu de l'automate, un message en bas suggère à l'utilisateur de choisir entre les propositions en entrée quand ils existent pour donner une meilleure pondération à la bonne phrase, et éviter l'erreur une prochaine fois. Mais aussi de proposer un autre résultat si par malheur un résultat est faux.

Étape 2 - 2 : Analyse non reconnue

Le deuxième scénario implique que la phrase exprime une règle grammaticale non présente dans l'automate, et l'utilisateur se retrouvera dans l'interface **« Feed »** illustrée ci-dessous :

Figure 4.4 : Interface Feed

L'interface comporte un message expliquant la procédure à suivre pour l'enrichissement et la phrase en question pour permettre un meilleur suivi. L'interface comporte également un formulaire de trois champs :

- Le premier est une sélection multiple qui regroupe les analyses éventuelles (si un état existe dans l'automate avec une description grammaticale équivalente à l'état actuel).

Ce champ quand il est rempli, cache automatiquement les deux autres, car on ne fait qu'avancer dans l'automate sans ajouter d'état ni de transitions. S'il n'existe pas d'états adéquats, le champ sera automatiquement caché et l'utilisateur sera invité à remplir les deux autres champs qui décrivent :

- La transition à choisir parmi les analyses morphologiques détectées sur le mot.
- Une zone de texte pour introduire la chaine de caractères désignant « الإعراب ».

Le bouton « قدم » permet d'avancer dans la phrase jusqu'à la fin ou l'utilisateur se verra emmené vers la page d'accueil où il pourra à nouveau introduire une nouvelle phrase à analyser.

Étape 2 - 3 : Mot non reconnu

Parfois l'analyse morphologique d'un mot ne donne aucun résultat à cause notamment du manque de quelques schèmes, racines. Dans ce cas-là, puisque l'analyse morphologique d'un mot ne peut être faite, le programme ne peut ni avancer dans l'automate, ni apprendre une nouvelle règle grammaticale, c'est un état critique du programme que nous devons à tout prix éviter ! Voici ce que l'utilisateur voit dans ce cas :

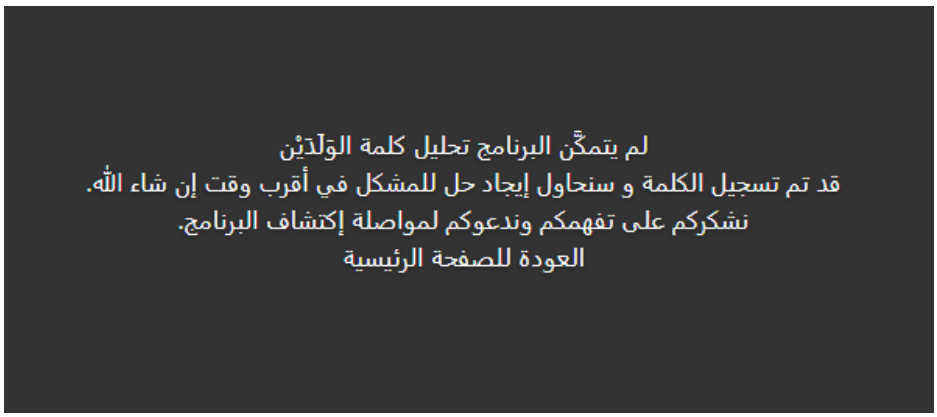


Figure 4.5 : Interface d'analyse impossible

L'interface comporte un message expliquant la situation à l'utilisateur, et un lien pour revenir vers la page d'accueil. Il est mentionné aussi que le mot en question a été ajouté à la BDD. En effet, nous avons mis en place une table pour regrouper ces exceptions quand elles se produisent. Le but derrière cet ajout est de pouvoir suivre en temps réel ces exceptions et essayer de les régler simplement en rajoutant un schème ou une racine non existante.

A long terme, cette action sera faite par un type d'utilisateur qui sera l'administrateur du programme, il aura accès à l'interface d'administration ou il pourra voir les différentes exceptions, mais aussi aux ressources ou il pourra ajouter la ressource manquante. Dans le cas où c'est un problème au niveau du code, ce sera aux développeurs de le régler.

Voici un aperçu de l'interface d'administration du programme.

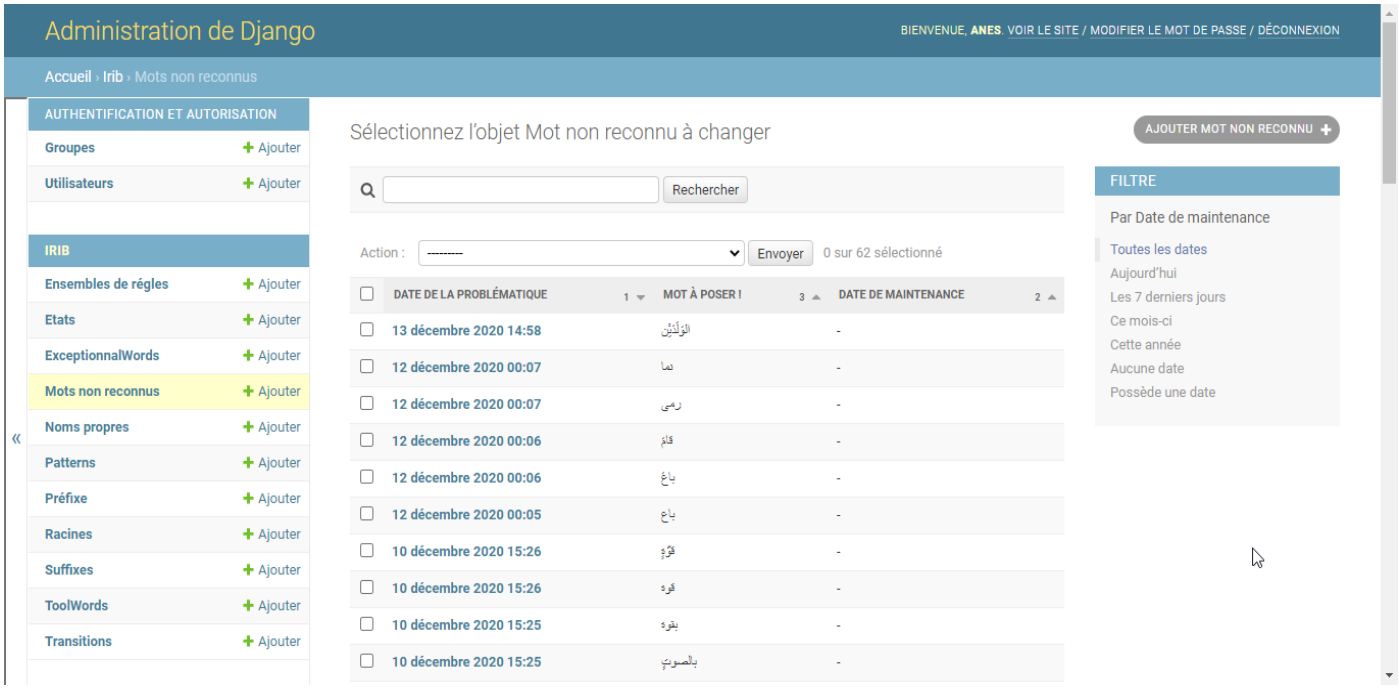


Figure 4.6 : Interface d'administration

Cette interface permet l'accès à toute la base de données de l'application, en commençant par les groupes et les utilisateurs, deux tables pré-intégrés par Django, et qui permettent de contrôler l'accès des utilisateurs par groupe ou par individu. D'autres tables comme celle des schèmes, des racines ou des transitions de l'automate sont aussi présentes et peuvent aider l'administrateur à déboguer le programme dans des situations critiques.

4.6. Tests et résultats

Il est nécessaire de présenter les résultats de quelques tests pour pouvoir valider la solution. Cependant, la solution proposée, n'est pas finale, et nécessite un approvisionnement des ressources ainsi que l'intégration de quelques spécificités de la langue arabe, donc il est clair, que le résultat n'est pas parfait. Aussi, de par la nature de l'approche que nous avons choisie (enrichissement continue), le résultat est par définition variable, et non constant.
Théoriquement, chaque mot non reconnu pourra être reconnu après l'ajout du schème ou de la racine manquants dans la BDD. Aussi, chaque règle grammaticale manquante sera rajoutée, et même pour un « i'rab » faux, la règle grammaticale qui porte ce dernier se verra petit à petit de moins en moins présentée à l'utilisateur au profit d'un « i'rab » juste, ceci grâce à la pondération des règles grammaticales.

5. Conclusion

Dans ce chapitre, nous avons présentés les outils, langages et bibliothèques utilisés lors du développement de la solution, ainsi que les différentes interfaces du programme ARIBLI avec un déroulement. Pour finir, nous avons expliqués pourquoi nous ne pouvons donner de résultats.

Conclusion générale et perspectives

Conclusion générale :

La langue arabe est une langue riche ce qui fais sa beauté et sa splendeur, c'est une langue d'histoire, de religion et de science. Mais sa richesse fait d'elle une langue difficile à apprendre pour un jeune élève, et c'est dans ce contexte que nous avons essayés dans ce projet de pallier à cette problématique. En effet, Al I'rab reste l'un des meilleurs moyens d'apprendre la structure d'une phrase et donc de pouvoir comprendre son sens mais aussi de pouvoir reproduire cette structure de phrase avec les concepts désirés.

De la même manière que pour un jeune élève, une machine trouve également des difficultés à comprendre une langue riche et pleine de spécificités. Ce qui nous emmène aux difficultés rencontrés lors de la réalisation de ce projet. Le premier souci rencontré a été le choix entre les différentes approches possibles pour la résolution du problème. Puis le manque de ressources nécessaires à l'accomplissement de la solution. Une fois les ressources trouvées, il a fallu les adapter à notre besoin. Mais le plus dur a été de devoir intégrer une grande partie des spécificités de la langue ce qui demande une maitrise de cette dernière.

Parmi les approches que nous pouvions prendre, le choix de l'une de ses dernières s'est fait plus ou moins malgré nous à cause des contraintes que nous avions. L'approche numérique nécessite un corpus annoté d'une manière spécifique à notre besoin, une approche symbolique nécessite une base de règles ce que nous n'avions pas. Et c'est ainsi que nous nous somme retrouvé à prendre les avantages des deux approches, l'utilisation de règles grammaticales en utilisant les schèmes, et la notion d'apprentissage pour pallier le manque de règles et de précision de l'analyse.

La solution que nous proposons se résume à la mise en place :

- D'une analyse morphologique qui permet de détecter les possibilités de catégories grammaticales d'un mot en se basant sur son schème et sa voyellation, ou sur des ressources quand il ne s'agit pas d'un mot flexionnel.
- Et d'une analyse grammaticale qui se base sur des règles qui indiquent l'agencement des entités d'une phrase et que le programme va apprendre au fur et à mesure de son utilisation.

Perspectives :

Nous avons pu mettre en place une application répondant au besoin et à la problématique posés dans ce projet, mais le plein potentiel de l'application ne peut être atteint qu'après avoir réglés quelques points faibles de celle-ci. En effet, choisir ce thème était un sérieux défi à prendre de par la multitude de difficultés qui pouvaient se ressentir dès le premier jour. Cependant, nous restons satisfaits du travail que nous avons pu réaliser, mais surtout optimistes car cette application s'inscrit dans une philosophie d'entraide, notamment avec l'approche d'enrichissement, mais aussi avec la publication du code en open source sur la plateforme GitHub pour ainsi permettre à ceux qui s'y connaissent sur la question de pouvoir contribuer à améliorer cette application.

Les axes d'amélioration de l'application sont les suivants :

- L'augmentation de ressources nécessaires au bon fonctionnement de l'application, notamment les schèmes et les racines.
- Les ressources avec lesquelles nous avons travaillés sont de multiples sources, et ça nous a posé un petit problème qui est le suivant : Les racines et les schèmes ne sont pas compatibles entre eux par rapport à l'approche choisie pour l'extraction de la racine. Le problème survient sur tous les mots qui ont une racine à deux lettres, or, dans les ressources que nous avons pu récolter, la taille des racines est au minimum à trois lettres. Pour l'instant, la solution de l'ajout des racines par l'administrateur régle le problème, mais ce n'est pas envisageable de garder cette approche au vu de la quantité de racines à deux caractères qui existent.
- Une meilleure précision. Nous en avons parlés lors de la description des modules morphologiques dans le chapitre 3, les deux modules (Halat Al I'rab, et Ta'rif) n'ont pas une très bonne précision, par manque de connaissances linguistiques, nous n'avons pas pu traiter toutes les conditions nécessaires, et c'est un axe important d'amélioration car il est directement lié à l'analyse grammaticale. En effet, si l'analyseur se trompe dans la détection d'une information aussi importante que (Halat Al I'rab, ou Atta'rif), il va automatiquement se tromper dans le choix de la règle grammaticale et donc de Al i'rab.

Le lien vers le repository GitHub est ici : https://github.com/anesfassih/IRIBLI

Références bibliographiques :

[1] M. A. Chéragui, Y. Hoceini et M. Abbas, “A Morphological Analysis of Arabic Language based on Multicriteria Decision Making: TAGHIT System”, IEEE, International Conference On Machine and Web Intelligence, (2010).

[2] M. Aljlayl and O. Frieder, On Arabic Search: Improving the Retrieval Effectiveness via a Light Stemming Approach, *In 11th International Conference on Information and Knowledge Management (CIKM), November 2002, Virginia (USA).*

[3] Larkey L. S., Ballesteros L. and Connell M., Improving Stemming for Arabic Information Retrieval: Light Stemming and Cooccurrence Analysis, *In Proceedings of the 25th Annual International Conference on Research and Development in Information Retrieval (SIGIR 2002), Tampere, Finland, August 2002.*

[4] J. Leclerc, L'aménagement linguistique dans le monde, *http://www.tlfq.ulaval.ca/axl/monde/famarabe.htm* .

[5] De l’´évaluation en Traitement Automatique des Langues mémoire d’Habilitation `a Diriger des Recherches École Doctorale d’Informatique Université Paris-Sud Patrick Paroubek Ingénieur de Recherche au LIMSI-CNRS (groupe ILES).

[6] (Aloulou et al. 02, Baccour et al. 03, Boualem 93, Debili 01, Dichy et al. 02, Gaubert 01, Jaccarini 97.)

[7] Hammo B., Abu-Salem H., Lytinen S., Evens M., QARAB: A Question Answering System to Support the Arabic Language, *Workshop on Computational Approaches to Semitic Languages. ACL 2002, July 2002, Philadelphia, PA.*

[8] A. Chalabi : Sakhr Web-based Arabic<>English MT engine, *ACL/EACL 2001 Workshop on Arabic Language Processing, Toulouse July2001(http://www.elsnet.org/arabic2001/chalabi.pdf)*

[9] Traitement automatique des langues pour l’accès au contenu des documents?Christian Jacquemin?, Pierre Zweigenbaumy? LIMSI-CNRS BP 133 91403 ORSAY Cedex FRANCE jacquemin@limsi.fr et http://www.limsi.fr/Individu/jacquemi/ y DIAM

[10] Journal of Computational Information Systems4:3(2008) 1169-1180 Available at http://www.JofCI.org 1553-9105/ Copyright © 2008 Binary Information Press March, 2008 Automatic Keyword Extraction from Documents Using Conditional Random Fields.

[11] Mohamed Hedi Maaloul. Approche hybride pour le résumé automatique de textes. Application _a la langue arabe.. Traitement du texte et du document. Universit_e de Provence - Aix-Marseille I, 2012. Fran_cais. <tel-00756111v1>.

[12] JONES, K. (1972). A Statistical Interpretation of Term Specificity and its Application in Retrieval.

[13] CLAVEAU, V. (2012). Vectorisation, Okapi et Calcul de Similarité pour le TAL : pour Oublier Enfin le TF-IDF. *In Proceedings of the Joint Conference JEP-TALN-RECITAL 2012, volume 2 : TALN*.

[14] BARKER, K. et CORNACCHIA, N. (2000). Using Noun Phrase Heads to Extract Document Keyphrases. *In Proceedings of the 13th Biennial Conference of the Canadian Society on Computational Studies of Intelligence : Advances in Artificial Intelligence*.

[15] TOMOKIYO, T. et HURST, M. (2003). A Language Model Approach to Keyphrase Extraction. *In Proceedings of the ACL 2003 workshop on Multiword expressions : analysis, acquisition and treatment-Volume 18*.

[16] DING, Z., ZHANG, Q. et HUANG, X. (2011). Keyphrase Extraction from Online News Using Binary Integer Programming. *In Proceedings of 5th International Joint Conference on Natural Language Processing*.

[17] LIU, Z., LI, P., ZHENG, Y. et SUN, M. (2009). Clustering to Find Exemplar Terms for Keyphrase Extraction. *In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing Volume 1*

[18] Diab, M., Hacioglu, K. & Jurafsky, D. (2004). Automatic tagging of Arabic text: From raw text to base phrase chunks. In *Proceedings of North American Association for Computational Linguistics* (NAACL, pp. 149–152).

[19] Habash, N. & Rambow, O. (2005). Arabic Tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop. In *Proceedings of the Association for Computational Linguistics* (ACL, pp. 573–580)

[20] Khoja, S. (2001). APT: Arabic part-of-speech tagger. In *Proceedings of the North American Association for Computational Linguistics Student Workshop* (pp. 20–25).

[21] Abbes, Ramzi. (décembre 2004). *la conception et la réalistion d'un concordancier électronique pour l'arabe.* Thèse de doctorat en sciences de l'information, Lyon,ENSSIB/INSA.

[22] Brill.Eric. (1993). Tagging an unfamiliar text with minimal human supervision. *In proceedings of the Fall Dymposium on Probabilistic Approch to Natural Language*.

[23] Farghaly.A and Dichy.J. (2003). Roots & Patterns VS Stems plus Grammair-Lexis specification :On what basis should a multilingual lexical database centred on arabic be built *Acte de la 9ème MT conference, Workshop on Machine translation for semitic language :issues and approaches ;New Orleans, Louisiana, USA* .

[24] Haddad. Ahmed, Ben Ghezala. Henda and Ghenima. Malek (2007). Conception d'un catégoriseur morphologique fondé sur le principe d'Eric Brill dans un contexte Multi-Agents. 26 th conference on Lexis and Grammar, Bonifacio.

[25] Mona T.Diab. (2010). Second Generation AMIRA Tools for Arabic processing: Fast and Robust Tokenization, pas Tagging and Base phrase chunking. *Center for Computational Learning Systems Columbia University* , 285-288.

[26] Thibeault, M. (2004). La catégorisation grammaticale automatique : adaptation du catégoriseur de Brill au Français et modification de l'approche . *Université Laval Québec, Canada* , 11.

[27] Viterbi. (1967). Error bounds for convolutional codes and an asymptotically optimal decoding algorithm . *IEEE Transavtions Information an theory*, pp260-269.

[28] Traitement_de_larabe_écrit_et_Web_arabe_lapport_de_léquipe_lyonnaise_SILAT_Systèmes_dinformation_Ingénierie_Linguistique_arabes_et_Terminologie

[29] Abbes, R. 2004. " La conception et la réalisation d'un concordancier électronique pour l'arabe ". Thèse de doctorat en sciences de l'information, Lyon, ENSSIB/INSA.(Corpus DIINAR.1) http://www.opengrey.eu/item/display/10068/763061

[30] Souhir Gahbiche-Braham , Amélioration des systèmes de traduction par analyse linguistique et thématique Application à la traduction depuis l'arabe , thèse de doctorat en informatique, soutenue le 30 Septembre 2013, École Doctorale d'Informatique, Université Paris Sud.

[31] Larkey L. S., Ballesteeros L. and Connell M., improving stemming for Arabic information retrieval: light stemming and co-occurrence analysis, in proceeding of the 25th annual international conference on research and development in information retrieval (SIGIR 2002), tampere, finland, august 2002,pp.275-282.

[32] Polly Lingual (En Ligne) Cours et Leçons, Classification Parties du discours en Langue Arabe/Français ; Available : https://pollylingu.al/ar/fr/types/433

[33] Alexandrc ANDREEWSKY, Christian FLUHR A learning method for natural language processing and application to information retrieval, IFIP Congress, August 1974.

[34] Imago Mundi, L'Encyclopédie gratuite en ligne, " Texte sur L'Analyse Grammaticale ". http://www.cosmovisions.com/analysegrammaticale.htm

[35] Slim MESFAR," Analyse Morpho-Syntaxique Automatique et Reconnaissance des entités nommées En Arabe Standard", thèse de Doctorat présenté en 24 Novembre 2008, Université De Franche-Comté.

[36] Siham Boulaknadel. Traitement Automatique des Langues et Recherche d'Information en langue arabe dans un domaine de spécialité : Apport des connaissances morphologiques et syntaxiques pour l'indexation. Autre [cs.OH]. Université de Nantes, 2008. Français.

[37] Philippe Baptiste et Luc Maranget, Les bases de la programmation et de l'algorithmique. École Polytechnique, Consulté le 06/11/2020. Français. http://gallium.inria.fr/~maranget/X/421/poly/automate.html

[38] PETROV S., DAS D. & MCDONALD R. (2012). A universal part-of-speech tagset. In Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12), Istanbul, Turkey.