République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieure et de la Recherche Scientifique

Université Ahmed Draia - Adrar

Faculté des Sciences et de la Technologie

Département des Mathématiques et Informatique

Mémoire de fin d'étude, en vue de l'obtention du diplôme de Master en informatique

Option : Réseaux et systèmes intelligents

Thème

Mise en place d'un chunker (grammaire en tronçons) des textes arabes

Préparés par

Fatima BELHADJ et Rekia MOULOUDI

Encadré par

Mr. Mouhamed Amine CHERAGUI

Année Universitaire 2015/2016

Résume

L'utilité très vaste du Traitement Automatique des Langages Naturels, nécessite de plus en plus des résultats plus robuste et plus précis. Afin d'obtenir ces résultats, l'analyse d'une langue doit atteindre un niveau plus précis que le lexical et le morphologique, comme le niveau syntaxique et sémantique qui permettent de fournir des informations contextuelles attribuant au bon traitement de la donnée exprimées en langage naturel.

Notre objectif dans ce mémoire est de réaliser un chunker qui transmit les informations attachées aux parties du discours à un ensemble de règles pour construire les différents types des tronçons (chunks), en se fondant sur l'étude de la grammaire de la langue arabe.

Mots clés : Traitement Automatique des Langages Naturels, Traitement Automatique de la Langue Arabe, étiquetage morpho-syntaxique, Grammaire en tronçons, parenthésage syntaxique.

Abstract

The vast utility of automatic natural language processing, requires increasingly more robust and more accurate results. To achieve these results, the analysis of a language must reach a specific level as the lexical and morphological, as syntactic and semantic level that can provide contextual information attributing to the good treatment of the data expressed in language natural.

Our goal in this paper is to provide a chunker who transmitted the information attached to the parts of speech to a set of rules for constructing different types of sections (chunks), based on the study of grammar of the Arabic language.

Key words: Automatic Natural Language Processing, Automatic Arabic Language Processing, POS Tagging, Grammar sections, chunking.

ملخص

الاستعمال الواسع للمعالجة الآلية للغات الطبيعية يتطلب المزيد من الدقة والفعالية في النتائج. ومن اجل الوصول إلى هذه النتائج، يجب ان يصل تحليل اللغة إلى مستوى ادق من المستويين المعجمي و الصرفي، كأن يبلغ المستوى النحوي والدلالي لتوفير معلومات سياقية تساعد في المعالجة الجيدة للمعلومات المعبر عنها باللغة الطبيعية. يرسل المعلومات المتعلقة بوحدات الكلام إلى مجموعة من القواعد من اجل بناء انواع مختلفة من المقاطع (أجزاء)، اعتمادا على قواعد اللغة العربية.

الكلمات المفتاحية: المعالجة الآلية للغات الطبيعية، المعالجة الآلية للغة العربية، التحليل الصرفي والنحوي، قواعد التقسيم النحوي، التجميع النحوي.

Remerciements

Au nom d'Allah le plus grand merci lui revient de nous avoir guidé vers le droit chemin, de nous avoir aidées tout au long de nos années d'étude.
Nous adressons notre profond remerciement à Monsieur ***CHERAGUI Mohammed Amine*** *notre encadreur qui nous a aidé.*
Nous exprimons nos reconnaissances à tous les enseignants de la faculté des Sciences et de la Technologie. Merci pour la bonne formation de base que nous avons bénéficiés auprès de vous
nous adressons nos remerciements les plus chaleureux à Mles : ***Noura*** *Tiuoririne et* ***Zohra*** *Kibir.*
nos remerciements pour toutes les personnes qui ont aidé de près ou de loin par le fruit de leur connaissance pendant toute la durée de notre parcours éducatif.

MERCI

Dédicace

Je dédie ce modeste travail à :

Ma mère qui m'a donné la vie, la tendresse, qui s'est sacrifiée pour mon bonheur Et ma réussite.

Mon père, pour l'éducation qu'ils m'ont prodigué

Mes frères et ma sœur

Mes chères amies

Mes sincères dédicaces pour : RekiaMouloudi.

Tous mes enseignants depuis primaire jusqu'à mon cursus

Toute la promotion 2015/2016

Belhadj Fatima

Dédicace

Je dédie ce travail :

A Dieu Le Tout Miséricordieux, ton amour, ta miséricorde grâces à mon endroit m'ont fortifiée dans la persévérance et l'ardeur au travail.

A toutes les personnes qui me sont chères :

Mes parents, qui ont toujours prête à se sacrifier pour le bonheur de ses enfants, Merci pour tout.

A l'esprit de ma mère que Dieu lui reposer et la mettre dans son paradis.

A mon compagnon de ma vie (mon mari), qui m'encourage toujours.

Mes sœurs et frères et ses enfants, Qui je le sais ma réussite est très importante pour vous. Que Dieu vous paye Pour tous vos bienfaits.

Mes amies de ma vie: Fatouma, Rekia, Zeyneb, Aicha, Yamina.

Mes sincères dédicaces Pour ma soeur :Fatima BELHADJ.

A tous mes enseignants depuis primaire jusqu'à mon cursus universitaire.

A toute les étudiants de l'informatique promotion 2015/2016

Rekia MOULOUDI

Table des matières …….. I

Liste des tableaux…… V

Liste des figures……..……..……... VII

Introduction générale ……..…….. 01

Chapitre 1 : Traitement Automatique la Langue Arabe….. 03

1. Introduction……..…….…… 03

2. Historique du traitement automatique de la langue arabe (TALA). 03

3. Particularités de la langue arabe……..……..…….. 05

4. Niveaux de traitement du langage naturel……..……..…...… 06

4.1 Traitement morpho lexical……..…….…….... 07

4.2 Traitement syntaxique……..……..……..... 07

4.3 Traitement sémantique……..……...…..... 07

4.4 Traitement pragmatique……..…..... 07

5. Problèmes du traitement automatique de la langue arabe……..... 08

5.1 L'absence de voyellation dans la majorité des textes arabes…..…..... 08

5.2 La Structure complexe des mots…..…… 08

5.3 La nature agglutinante de la langue arabe…… 09

5.4 Le problème de la flexibilité……..…..... 10

5.5 L'ambigüité…….……..……..... 11

6. Applications concernant le TALA……...…… 12

6.1 Analyseur Sakhr……..... 12

6.2 Ara-Conc…… 12

6.3 ASVM…… 12

6.4 BuckWalter…… 12

6.5 MADA…… 13

6.6 MorphArabe…… 13

6.7 Racineur de larkey…… 13

6.8 Racineur de Khoja…… 13

7. Conclusion…… 13

Chapitre 2 : Le Chunking…… 14

1. Introduction…… 15

2. Les types de segmentation…… 15

2.1 L'itémisation (Tokenization ou Word Segmentation)…… 15

2.2 La segmentation morphologique …… 16

2.3 Le chunking…… 16

3. C'est quoi la grammaire en tronçons (Chunking) ? …… 17

4. Pour quoi La grammaire en tronçons ?…… 18

5. Les étapes du chunking…… 19

5.1 L'étiquetage morpho-syntaxique…… 20

5.1.1La segmentation…… 20

5.1.2 L'analyse morphologique…… 20

5.1.3 La désambiguïsation…….……... 20

5.2 Parenthésage syntaxique……..... 22

5.2.1 Le découpage en groupes …….... 22

5.2.2 Etiquetage des groupes …….…….. 22

6. Exemples de chunkers…….…… 23

7. conclusion…….…… 24

Chapitre 03 : Conception et réalisation de « ArabChunk ».... 25

1. Introduction……..... 25

2. Le principe de chunker «ArabChunk»…… 25

3. Architecture générale de chunker «ArabChunk»……...… 25

4. La phase d'étiquetage morphosyntaxique……..……. 27

5. La phase de parenthésage syntaxique……..…… 27

5.1 Étude de la grammaire en tronçons……..…… 27

5.1.1 Les types de phrases…….…..... 27

5.1.2 Les liens syntaxiques…… 28

5.2 Les type des tronçons……..…….…….. 31

5.3 Classification à base de tag……..……..…… 33

5.4 Classification à base de règle……..…….…… 35

6. Comment construisant un tronçon…..……..... 37

6.1 Les tronçons nominaux qui commencent par un nom indétermi….…… 37

6.2 Les tronçons nominaux et verbaux qui commencent par des particules…… 38

6.3 Les tronçons relatifs…… 39

6.4 Les tronçons conjonctifs…….…... 40

7. Conclusion ……...… 41

Chapitre 04 : Application et résultats……..……..….. 42

1. Introduction…… 42

2. Environnement de travail …… 42

2.1 Environnement hard…….…… 42

2.2 Présentation de langage de programmation python……..……. 42

2.3 Plateforme……..…… 43

3. Interface graphique de chunker « ArabChunk »…….…….. 44

4. Evaluation…….…… 46

6. Discussion…… 49

7. Conclusion……. 50

Conclusion générale…….……..… 51

Références……..…… 52

Annexe …….. 55

Liste des tableaux

Chapitre 01

Tableau 01 : Exemple de génération des mots d'une même racine……..…….....05

Tableau 02 : Quelques voyellation possible du mot non voyellé « **كتب** »…….……08

Tableau 03 : Segmentation du mot complexe «**أتتفكروننا** »…….……..09

Tableau 04 : La première segmentation du mot « **أوحل** »…….…….……09

Tableau 05 : La deuxième segmentation du mot « **أوحل** »…….…….....09

Tableau 06 : La troisième segmentation du mot « **أوحل** »…….….....09

Tableau 07 : Exemple des lemmes possibles lors de la flexion du verbe **« يعد »**…….….....10

Tableau 08 : Les conjugaisons possibles du verbe **« أرسل »** par modification des voyelles courtes seulement……...……..……..….....10

Tableau 09 : Deux significations possibles pour le mot « **ذهب** » comme nom ou verbe…...……11

Tableau 10 : Exemple d'ambigüité provoqué par les proclitiques…….…….....11

Tableau 11: Deux lemmes différents pour le même mot « **درس** » par l'absence du chadda.…....11

Chapitre 02

Tableau 12 : Exemple du chunking de la phrase en arabe « **نجح الطالب في الامتحان** »……...…18

Tableau 13: Exemple d'étiquetage morphosyntaxique d'une phrase en arabe…….....21

Tableau 14 : Exemple de parenthésage syntaxique d'une phrase en arabe……23

Chapitre 03

Tableau 15 : Exemples des adjectifs……..28

Tableau 16 : Exemples des annexions…….28

Tableau 17 : Exemple des prépositions avec des cas génitifs…….....29

Tableau 18 : Exemple des prépositions avec des cas génitifs…….....29

Tableau 19 : Les pronoms relatifs……!……...30

Tableau 20 : Exemples des phrases relatifs…….....30

Tableau 21 : Exemples des phrases relatifs…….....30

Tableau 22 : Les déférents types de tronçons…….....32

Chapitre 04

Tableau 23 : Exemple de chunking…….…….....46

Tableau 24 : Evaluation qualitatif de « **ArabChunk** »…….....47

Liste des figures

Chapitre 01

Figure 01: Les niveaux de traitement du langage naturel……....06

Chapitre 02

Figure 02 : Les types de segmentation ……..15

Figure 03: Exemple d'itémisation de la phrase « **أكل الولد التفاحة** » …….....16

Figure 04: Exemple de la segmentation morphologique du mot « **سأكتبها** » …….....16

Figure 05: Exemple de Chunking de la phrase « **قرأ التلميذ في الدفتر الكبير** »…… …..17

Figure 06 : Le chunking de la phrase arabe « **ظهر الإسلام في شبه الجزيرة العربية** »…..……19

Figure 07 : Exemple de déduction contextuelle pour enlever l'ambiguïté sur le mot « **ذهب** »...21

Figure 08 : Exemples de chunkers……..23

Chapitre 03

Figure 09 : Schéma générale du chunker «**ArabChunk**»……..…….....26

Figure 10 : Les différents types des tronçons……..…….....31

Figure 11 : Schéma générale de classification à base de tag…….....33

Figure 12 : Les différents types des mots …….…..33

Figure 13: Schéma générale de classification à base de tag…...……....34

Figure 14 : Schéma générale de classification à base de règle ……...35

Figure 15 : Exemple de classification à base de règle …….…...36

Figure 16 : Les chemins possibles d'un tronçon qui commence par un nom indéterminé...…...37

Figure 17: Les chemins possibles d'un tronçon qui commence par un particule ……..38

Figure 18 : Le tronçon relatif ……..……....39

Figure 19 : Le tronçon conjonctif …….……..39

Figure 20 : Exemple de découper une phrase utilisant « **ArabChunk** »…….....40

Chapitre 04

Figure 21 : Interface graphique de chunker « **ArabChunk** »……..44

Figure 22 : Description de barre d'outils……...……..……...45

Figure 23 : Téléchargement d'un texte……...45

Figure 24 : Résultat de chunking …… .46

Figure 25 : Représentation graphique de test de première catégorie de corpus Khaleej…….....47

Figure 26 : Représentation graphique de test de deuxième catégorie de corpus Khaleej……...48

Figure 28 : Représentation graphique de test de troisième catégorie de corpus Khaleej……....48

Introduction générale

Le traitement automatique des langages naturels (TALN) est la discipline qui vise au développement des programmes et techniques capables de traiter automatiquement les langages naturels, comme : la traduction automatique, l'extraction d'information, la correction automatique et l'indexation des documents.

Malgré des nombreuses recherches le traitement automatique de la langue arabe reste difficile à maitriser, à cause de la richesse morphologique, et les caractéristiques linguistiques comme la voyellation, l'agglutination et la flexibilité.

L'analyse est la base du TALN, elle étudie une langue sur différentes niveaux : lexical, morphologique, syntaxique, sémantique et pragmatique. L'étiquetage est une analyse morphosyntaxique des langues, qui permet de définir les catégories et la fonction des mots dans une phrase. Ensuite, vient l'analyse syntaxique qui étudier les liens syntaxique entres les mots pour former des phrases ou structures correctes, à travers des règles grammaticales qui dirige cette formation.

Notre travail, porte sur l'étude de la langue arabe, ses caractéristiques linguistiques et ses règles, dans le but de réaliser un chunker des textes arabes. Premièrement, nous avons passé par une analyse morphosyntaxique des unités lexicales arabes effectuées par l'outil AMIRA 2.0, en second lieu, nous avons fait une analyse syntaxique permettant de former des tronçons par regroupement des unités à base d'une grammaire en tronçons, afin de mieux comprendre un texte et de fournir un outil qui permettra d'autres traitements comme la traduction ou la correction automatique de la langue arabe.

Pour réaliser cette étude, nous avons organisé ce travail en quatre chapitres :

- Le premier chapitre concernant le « traitement automatique du langage arabe », présente la langue arabe, ses propriétés, ses particularités linguistiques comme l'agglutination, la voyellation et la flexibilité, les niveaux de traitement automatique d'une langue. Ainsi que les problèmes rencontrés lors de traitement. Enfin, un survole sur les outils de traitement automatique de la langue arabe.
- Dans le deuxième chapitre, nous avons défini le « Chunking », nous définissant le chunking et la grammaire en tronçons, ainsi que son utilité. Ensuite, nous avons présentés les deux grandes étapes pour aboutir au chunking d'un texte: l'analyse morphosyntaxique et le parenthésage syntaxique. Enfin, quelques exemples d'outils chunking.
- Le troisième chapitre met l'accent sur « la conception et la réalisation », il présente l'architecture générale du logiciel « **ArabChunk** ». Ainsi que les règles qui permettent la construction des tronçons.

- Le quatrième chapitre est intitulé « application et résultats », il expose les interfaces graphiques du logiciel développé. Ainsi qu'une discussion des résultats obtenus.

1. Introduction

Le Traitement Automatique des Langues Naturelles (TALN) est une discipline situé à la frontière de la linguistique, l'informatique et l'intelligence artificielle, elle concerne la conception et le développement de programmes et techniques informatiques capables de traiter de façon automatique des données exprimées dans une langue naturel. Les applications liées au TALN ont fait l'objet d'une attention toute particulière depuis plusieurs décennies. Bien que certaines langues ont été réputées privilégiées comme le français et l'anglais, d'autres voient se poursuivre les recherches et les travaux afin de proposer des outils robustes de traitement.

La langue arabe malgré sa position de cinquième langue mondiale avec plus de 50 000 sites arabes sur le web et plus de 320 millions locuteurs, n'a encore aucun outil capable de traiter de façon robuste la totalité de ses caractéristiques linguistiques.

Dans ce chapitre nous allons présenter tout d'abord quelques caractéristiques de la langue arabe, ensuite les différents niveaux de traitement du langage naturel, ainsi que les problèmes faisant obstacle pour le traitement automatique de cette langue, puis nous citons quelques outils du TALA.

2. Historique du traitement automatique de la langue arabe (TALA)

- **En 1961,** L'article de David Cohen dans la revue de l'Association pour le Traitement Automatique des Langues naturelles (ATALA) et intitulé « Essai d'une analyse automatique de l'arabe », ou il propose un schéma général des mots graphiques maximaux entièrement vocalisés consiste à décomposer le mot graphique en racine, schème, base, préfixes, suffixes, antéfixes, postfixes. Cette unité décomposable est donc le mot maximal. La concaténation des préfixes, de la base et des suffixes formant ce que D. Cohen appelle mot minimal [ZOU, XX]. Ces travaux ont été poursuivis et améliorés dans les travaux de SAMIA[1] et les travaux de DIINAR[2];
- **En 1970**, David Cohen présente son travail de « Etudes de linguistique sémitique et arabe » [ZOU, XX];
- **En 1989**, Dichy.J, Hassoun.M, ont réalisées le travail de « Simulation de modèles linguistiques et Enseignement Assisté par Ordinateur de l'arabe », dans le cadre des travaux de l'équipe SAMIA I [ABB, 04] ;
- **En 1991,** L'équipe SAMIA a engagé un partenariat avec l'IRSIT (Institut Régional des Sciences de l'Informatique et des Télécommunications de Tunis). Cette collaboration a abouti à la réalisation du dictionnaire électronique DIINAR.1« DIctionnaire INformatisé de l'ARabe » qui est l'une des bases lexicales les plus importantes dans le domaine du TALN traitant la langue arabe [ABB, 04].

[1] SAMIA : **Sy**nthèse et **Analyse M**orpho-syntaxiques **I**nformatisées de l'**A**rabe.
[2] DIINAR : **DI**ctionnaire **IN**formatisé de l'**Ar**abe.

- **En 1998**, partant d'une étude comparative du problème de la voyellation automatique et de celui de la ré-accentuation automatique, propose un système de restitution des voyelles. Ce système commence par associer, à chaque unité lexicale reconnue dans le texte, l'ensemble de ses vocalisations potentielles. Ensuite, il résout ou réduit le plus possible l'ambiguïté de voyellation de chaque unité en faisant intervenir des connaissances de nature morphologique et syntaxique. Plus tard, ce travail a abouti sur un système interactif d'aide à la voyellation, baptisé VOWEL, qui a été intégré dans le logiciel de traitement de texte Microsoft Word [NEL, 05].
- **Au début des années 2000**, Y. Gal a utilisé un modèle de Markov caché (Hiden Markov Model – HMM) basé sur un apprentissage effectué sur des textes totalement voyellés, notamment le Coran [SAK, 05] ;
- **En 2002**, l'analyseur Buckwalter (the Arabic Morphological Analyser V1) est développé par LDC (Linguistic Data Consortium) ;
- **En 2004**, Mona Diab, Kadri Haciglu et Daniel Jurafsky ont realisés le « Automatic Tagging of Arabic Text : From Raw to Base Phrase Chunks [DIA, 04] ;
- **En 2005**, Nelken et Shieber ont utilisé un transducteur à états finis pondéré basé sur un apprentissage réalisé sur les corpus du LDC de M. Maamouri et Al [MES, 03];

  Dans la même année, un analyseur morphologique de l'arabe appelé MADA (Morphological Analyzer and Disambiguator of Arabic) été développé par Habash. Cet analyseur réalise la segmentation, la diacritisation, la lemmatisation, l'étiquetage grammatical et l'analyse morphologique [AHM, XX];

- **En 2006**, Safadi et Al ont développés une technique de voyellation non-contrôlée. Ils s'appuyaient sur une combinaison de catégorisation grammaticale automatique et de règles écrites manuellement [SAF, 06].
- **En 2008**, Siham Boulaknadel présente son travail de « 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 » [SIH, 08];
- **En 2010**, la 2ème génération de AMIRA, l'outil du traitement de l'arabe : Tokenization Rapide et Robuste, Tagging de la phrase de base de Chunking, de Mona Diab [DIA, 10] ;
- **En 2011**, La publication du travail de Mona Ali Mohammed et Nazlia Omar « Rule Based Shallow Parser for Arabic Language » [MON, 11];
- **En 2012**, le travail de Ahmed Hamdi a l'Université de Marseille intitulé « Apport de diacritisation dans l'analyse morphosyntaxique de l'arabe » [AHM, XX] ;
- **En 2014**, une méthode de lissage d'une approche morphosyntaxique pour la voyellation automatique des textes arabes réalisée par Amine Chennoufi et Azzeddine Mazroui à l'Université de Mohammed I, Ouejda, Maroc [AZZ, 14] ;

Conjointement, le travail de M. Maamouri sur la voyellation [MAA, 06] venait soutenir cette dernière théorie après une étude entreprise sur les annotations du treebank de la langue arabe élaborée à l'université de Pennsylvanie. Cette étude a révélé la forte liaison entre voyellation, catégorisation grammaticale et annotation.

3. Particularités de la langue arabe

Toute langue a ses propres caractéristiques qui la diffèrent des autres langues, soit par l'alphabet ou par les règles d'écriture ou d'utilisation. La langue arabe possède les caractéristiques et les particularités suivantes [MOU, 07]:

- L'alphabet de la langue arabe compte 28 consonnes ;
- La langue arabe compte trois voyelles long (ا،و،ي), et trois voyelles courts (َ،ُ،ِ) ;
- La langue arabe s'écrit et se lit de droite à gauche ;
- Les lettres arabes changent de forme de présentation selon leur position au début, au milieu ou à la fin : la lettre « ع » par exemple : (عـ : au début , ـعـ : au milieu, ع: à la fin) ;
- Un mot arabe s'écrit avec des consonnes et des voyelles. Les voyelles sont ajoutées au-dessus ou au-dessous des lettres (voyelles courts) ;
- La langue arabe comprend trois catégories de mots: verbes, noms et particules ;
- Les verbes et les noms sont le plus souvent dérivés d'une racine à trois consonnes ;
- Une famille de mots peut être générée à partir d'une seule racine à l'aide de différents schèmes.

  Exemple : pour la racine « **كتب** », en utilisant les schèmes (**فَعَلَ, مَفْعُول, فَاعِل**) on peut dériver les mots suivants :

Racine	Schème	Mot
كتب	**فَعَلَ**	كَتَبَ
كتب	**مَفْعُول**	مَكْتُوب
كتب	**فَاعِل**	كَاتِب

Tableau 01 : Exemple de génération des mots d'une même racine

- L'arabe est une langue flexionnelle. Elle emploie, pour la conjugaison du verbe et la déclinaison du nom, des indices d'aspect, de mode, de temps, de personne, de genre, de nombre et de cas, qui sont en général des suffixes et préfixes [BLA, 75]. Par exemple, pour le verbe « **ذَهب** » (aller), les formes à l'accompli sont repérables à l'aide de leurs suffixes tel que « **ذهبتُ** » (je suis allé) ou de leurs préfixations tel que « **أذهبُ** » (je vais) ; Aussi la fonction des noms à l'aide des suffixations tels que « **رجلانِ** » (deux hommes au nominatif) ou « **رجليْنِ** » (deux hommes à l'accusatif ou génitif) [MES, 08].

- La morphologie de l'arabe est dérivationnelle : Tout verbe a dans son sillage des formes déverbales qui lui sont associées et avec lesquelles il entretient des relations morphologiques, syntaxiques et sémantiques stables. Le nombre et la nature de ces formes varient selon le statut du verbe [MES, 08].

  Exemple, le verbe وَدَّ (aimer) admet cinq noms verbaux différents qui expriment une «affection » avec une certaine nuance sémantique : « وَدُّو », « وُدُّو », « وِدَادٌ », « وِدَادَةٌ » et « مَودّةٌ ».

4. Niveaux de traitement du langage naturel

Le traitement d'un langage naturel passe par quatre (04) niveaux: morpho-lexical, syntaxique, sémantique et pragmatique.

Ces niveaux d'analyse interagissent dynamiquement au cours du traitement d'un document écrit [EVA,15].

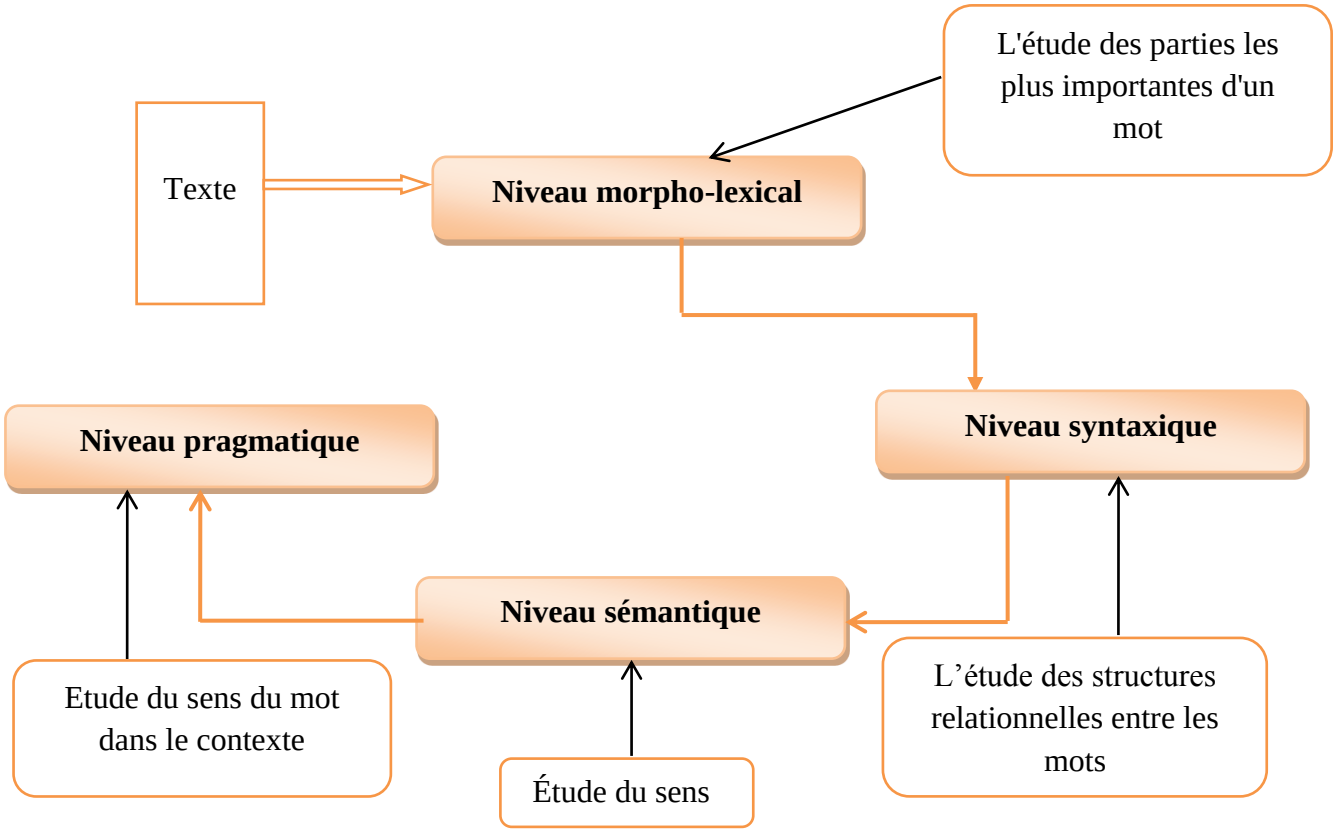


Figure 01: Les niveaux de traitement du langage naturel

4.1 Traitement morpho lexical

Le but principal de cette phase est de vérifier l'appartenance d'un mot donné au domaine linguistique choisi (ou bien à la langue étudié) et de pouvoir disposer ainsi de tous les renseignements le concernant pouvant servir à l'analyse syntaxique [CHE ,15].

Le traitement morpho-lexical porte sur l'unité élémentaire identifiable, en l'occurrence le morphème, à ce niveau, on s'intéresse à deux concepts essentiels qui sont : l'analyse qui a pour rôle d'associer à un mot graphique un ensemble d'unités morphologiques entrant dans sa composition (proclitiques, préfixes, base, suffixes, enclitique) , la synthèse qui permet de générer des mots ou des phrases par le biais d'un ensemble de règles de dérivation, de flexions et d'adaptations[CHE ,15].

Exemple : le mot arabe « **رفع** » après une analyse morpholo- lexical en déduirons qu'il appartient au lexique arabe. Par contre le mot « **رمن** » n'est pas un mot arabe.

4.2 Traitement syntaxique

L'étude de la syntaxique porte sur l'agencement des mots et la formation correcte des phrases à travers une grammaire qui régit la langue. Les systèmes qu'on qualifie de système syntaxique permettant d'une part, l'identification des différentes parties d'une phrase et de leurs relations indépendamment des contenus sémantiques et d'autre part d'extraire les renseignements contenus dans les différentes parties de la phrase et les insérer dans les applications particulières[CHE, 15].

Exemple : la structure « **ضرب نام الولد** » n'est pas correcte syntaxiquement. Mais, la structure « **كسر القط المزهرية** » est correcte.

4.3 Traitement sémantique

L'analyse sémantique joue un rôle important dans l'étude du langage naturel, dans le sens où elle consiste à extraire la signification des structures de surface (l'étude du sens du mot hors contexte de la phrase ou du texte), et vise aussi à enlever les ambiguïtés qui restent après le traitement syntaxique et ainsi traiter les problèmes relatifs à la correspondance structure sens [CHE, 15].

Exemple : la phrase « **قرأ الباب الكتاب** », cette phrase est syntaxiquement correcte mais n'a pas de sens, par contre « **قرأ الطفل القصة** », est correcte syntaxiquement et sémantiquement.

4.4 Traitement pragmatique

Ce type de traitement permet de lever les ambiguïtés qui ne peuvent pas être éliminées par le traitement sémantique, à cause de certains problèmes ayant un lien avec le contexte dans lequel la phrase est prononcé (donner un sens au mot par rapport au contexte dans lequel il se trouve), c'est-à-dire, il se charge de placer le mot dans le contexte de l'ensemble des connaissances en faisant recours à des informations hors-contexte (géographie, sport, travail, …etc.) [CHE, 15].

5. Problèmes du traitement automatique de la langue arabe

Depuis plusieurs décennies, de nombreux travaux de recherche ont été menés pour étudier les phénomènes linguistiques spécifiques à la langue arabe tels que la non vocalisation et l'agglutination. La difficulté de ces phénomènes est apparue sous forme d'une complexité de mise en place dès qu'il s'agissait d'un passage de l'état d'un prototype théorique à celui d'un système réellement utilisable dans des applications à large échelle. A cause des propriétés morphologiques et syntaxiques de la langue arabe le traitement automatique doit donc faire face à :

5.1 L'absence de voyellation dans la majorité des textes arabes

L'analyse automatique de la langue arabe est confrontée à de nombreux problèmes en raison de l'absence des voyelles (ou signes diacritiques) dans la plupart des textes. Ce qui entraine plusieurs cas d'ambiguïté lexicale en arabe à cause de la nature polysémique des mots non voyellés [MES, 08].

Le mot non voyellé « **كتب** » (ktb) par exemple peut avoir 17 voyellations différentes parmi lesquelles:

Mot sans voyelles	**Voyellation 1**	**Voyellation 2**	**Voyellation 3**	**Voyellation 4**
كتب	كَتَبَ	كُتِبَ	كُتُبٌ	كَتَّبَ
Ktb	Kataba « il a écrit »	Kutiba « Il a été écrit »	Kutubun « des livres »	Kattaba « a fait écrire »

Tableau 02 : Quelques voyellation possible du mot non voyellé « **كتب** »

Ce mot simple non voyellé (**ktb**) possède au total : 17 voyellations potentielles, neuf catégories grammaticales possibles et deux découpages acceptables [MES, 08].

Compte tenu des proportions importantes des mots ambigus en arabe, ce phénomène est très fréquent et nécessite de sérieuses études. .

5.2 La Structure complexe des mots

Le mot graphique en arabe peut provenir d'une structuration assez complexe, auquel cas il est désigné de « mot maximal ». Cette appellation a été attribuée par D. Cohen [COH, 61] à un mot graphique décomposable en : proclitique(s), forme fléchie, enclitique(s). La forme fléchie, désignée de mot minimal, qui est le noyau lexical du mot graphique, les autres constituants étant des extensions.

Comparé au français, un mot en arabe peut correspondre à une phrase française complète. Par exemple le mot « **أتتفكروننا** », peut être traduite en « est-ce que vous vous souvenir de nous ? ».

La segmentation correcte de ce mot se fait sous la forme suivante :

Enclitique	Suffixe	Forme fléchie	Préfixe	Proclitique
نا	ونـ	تفكر	تـ	أ

Tableau 03 : Segmentation du mot complexe « **أتتفكروننا** »

5.3 La nature agglutinante de la langue arabe

L'ensemble des morphèmes collés à l'unité lexicale offre plusieurs informations morpho-syntaxiques , ce qui augmente considérablement le taux d'ambigüité en introduisant des ambiguïtés supplémentaires au niveau de la segmentation des mots.

Un mot peut avoir plusieurs segmentations possibles en proclitique(s), corps schématique et enclitique(s) [GHO, 11].

Exemple : Le mot « **أوحل** » peut avoir trois segmentations possibles :

- **La première segmentation :**

corps schématique	Préfixe	Proclitique
حَل Verbe à l'accompli (ouvrir, résoudre)	و Conjonction de coordination	أ Article d'interrogation

Tableau 04 : La première segmentation du mot « **أوحل** »

- **La deuxième segmentation :**

Corps schématique	Proclitique
وحَل Verbe à l'accompli (se coincer) ou un substantif (boue)	أ Article d'interrogation

Tableau 05 : La deuxième segmentation du mot « **أوحل** »

- **La troisième segmentation :**

Corps schématique
أوحَل Verbe à l'accompli (faire coincer)

Tableau 06 : La troisième segmentation du mot « **أوحل** »

5.4 Le problème de la flexibilité

Lors de la flexion des verbes, des successions d'opérations morphologiques et orthographiques (comme la suppression de caractères ou l'assimilation) produisent fréquemment des formes fléchies homographes qui peuvent appartenir à deux ou plusieurs lemmes [COH, 70]. L'exemple suivant montre qu'une forme verbale simple peut être interprétée comme appartenant à cinq lemmes.

Le mot	1er lemme	2ème lemme	3ème lemme	4ème lemme	5ème lemme
يعد	**(أعاد)يُعِد** (Il refait)	**(وعد)يَعِد** (Il promet)	**(عد) يَعُدُّ** (Il compte)	**(عاد)يَعُد** (Il retourne)	**(أعد) يُعِدُّ** (Il prépare)

Tableau 07 : Exemple des lemmes possibles lors de la flexion du verbe « **يعد** »

Ainsi, beaucoup de paradigmes flexionnels se fondent sur une légère modification au niveau des voyelles courtes. Ces modifications restent sans effet en cas d'absence de voyellation [COH, 70]. Plusieurs exemples sont à noter, L'exemple ci-dessous, montre l'ambiguïté entraînée par la conjugaison du même verbe à l'accompli, voix active et voix passive et à l'impératif :

Le mot	La 1ère voyellation	La 2ème voyellation	La 3ème voyellation
أرسل	**أَرْسَلَ** (a envoyé)	**أُرْسِلَ** (est envoyé)	**أَرْسِلْ** (envoie)

Tableau 08 : Les conjugaisons possibles du verbe « **أرسل** » par modification des voyelles courtes

L'ordre des mots dans une phrase simple en arabe est relativement flexible. Nous disposons, généralement, d'un libre choix du terme que nous voulons mettre en valeur, en tête de phrase ; la même phrase peut être ordonnée comme :

- ✓ Verbe + sujet + complément.

  Exemple : **سافر الرئيس إلى بريطانيا** (le président s'est rendu en Grande Bretagne).
- ✓ Sujet + verbe + complément.

  Exemple : **الرئيس سافر إلى بريطانيا** (c'est le président qui s'est rendu en Grande Bretagne).
- ✓ Complément + verbe + sujet.

  Exemple : **إلى بريطانيا سافر الرئيس** (c'est en Grande Bretagne que le président s'est rendu).

L'ordre relativement libre des mots d'une phrase arabe ne vient que s'accumuler aux problèmes morpho syntaxiques spécifiques à l'arabe et limiter considérablement les éventualités de résolution de ceux-ci. En effet, la prise en compte de toutes les règles de combinaisons possibles des constituants de la phrase provoquerait, sans doute, des ambiguïtés syntaxiques artificielles supplémentaires [COH, 70].

5.5 L'ambigüité

L'ambiguïté est un problème central de l'analyse morpho-syntaxique de l'arabe. Les analyseurs se trouvent fréquemment confrontés à des situations d'ambiguïtés à tous les niveaux de l'analyse que ce soit au niveau lexical, syntaxique ou sémantique. Outre la richesse des constructions syntaxiques et leurs interprétations multiples auxquelles les analyseurs sont confrontés, cette ambiguïté est due, essentiellement, à l'ambiguïté des segmentations en unités lexicales et l'homographie polycatégorielle [ATT, 06].

Le problème réside au niveau du traitement de façon robuste et réaliste. En effet, après une première phase de segmentation du texte en unités lexicales, il est convenu de chercher dans le lexique les interprétations correspondantes à chacune d'entre elles. A chaque interprétation, nous associons une catégorie syntaxique reconnue par la grammaire [MES, 08].

En outre, beaucoup de mots en arabe sont homographiques: ils ont la même forme orthographique, bien que la prononciation soit différente. De cette homographie, lorsqu'elle est majorée par d'autres phénomènes (absence de voyellation, morphologie flexionnelle et agglutinante, etc.) découle un taux d'ambiguïté assez élevé [MES, 08].

Le lexique arabe contient des mots homographes qui, sans flexion préalable, peuvent avoir différentes prononciations, des significations différentes et généralement des catégories grammaticales différentes [COH, 70]. Par exemple, le mot « ذهب » peut considérer comme :

Le mot	**La 1ère signification**	**La 2ème signification**
ذهب	**ذهب** (or)	**ذهب** (il est allé)

Tableau 09 : Deux significations possibles pour le mot « **ذهب** » comme nom ou verbe

Les proclitiques peuvent engendrer deux formes homographiques.

Le mot	**1ère forme**	**2ème forme**
علمي	**علمي** Adjectif (scientifique)	**علم+ ي** (mes connaissances)

Tableau 10 : Exemple d'ambigüité provoqué par les proclitiques

Certains lemmes sont différents uniquement par un redoublement, au moyen de la lettre chadda (signe diacritique), sans que cela ne soit explicite à l'écrit. Les deux formes, ci-dessous, diffèrent uniquement au niveau du redoublement de la syllabe du milieu.

Le mot	**1er lemme**	**2ème lemme**
درس	**دَرَسَ** (il a étudié)	**دَرَّسَ** (darrasa- il a enseigné)

Tableau 11: Deux lemmes différents pour le même mot « **درس** » par l'absence du chadda

6. Applications concernant le TALA

Les applications de traitement automatique de la langue arabe sont l'ensemble des recherches et développements visant à modéliser et reproduire, à l'aide de machines, la capacité humaine à produire et à comprendre des énoncés linguistiques dans des buts de communication. L'objectif de cette section est de recenser les principaux outils de TAL en langue arabe, à savoir les analyseurs morphologiques, les concordanciers et les racineurs.

6.1 Analyseur Sakhr

C'est un analyseur développé par (Chalabi, 2004) qui permet de traiter aussi bien l'arabe moderne que l'arabe classique. Il permet de définir la forme de base par suppression de tous les préfixes et les suffixes du mot à traiter et donne les traits morphologiques de ce dernier. D'après le site web de la société sakhr, cet analyseur donne des résultats qui atteignent une précision de plus de 90% [GHO, 11].

6.2 Ara-Conc

Ara-Conc est un concordancier développé pour l'arabe par Aljlayl.M et Frieder.O en 2002. Il a pour objectif de donner les contextes et fréquences, et permettre l'exploration du corpus selon les traits proposés par l'analyse morphologique et selon les informations graphiques qui se trouvent dans le texte. La concordance finale arabe tourne autour du trio : unité lexicale, position et analyse morphologique. L'outil prend en entrée un texte ou un ensemble de textes [GHO, 11].

6.3 ASVM

L'analyseur ASVM est un logiciel libre, développé en Perl par l'équipe de Mona Diab [ABB, 04] à la Leland Stanford Junior University en 2004. Il s'agit d'une adaptation à l'arabe du système anglais YamCha basé sur les Support Vector Machines. Les données probabilistes ont été acquises pendant une phase d'entrainement sur le corpus annoté Arabic TreeBank [GHO, 11].

6.4 BuckWalter

BuckWalter est un analyseur développé par LDC (Linguistic Data Consortium) en 2002. Il permet de segmenter chaque unité lexicale en une séquence du type préfixe-stem-suffixe. Le préfixe est une combinaison de 0- 4 caractères, le suffixe est composé de 0 à 6 caractères et le stem comprend un à plusieurs caractères. Il est constitué principalement de trois lexiques : préfixes (548 entrées), suffixes (906 entrées), et stem (78839 entrées).

Les lexiques sont complétés par trois tables de compatibilité utilisés pour couvrir toutes les possibilités de combinaisons préfixe-stem (2435 entrées), suffixe-stem (1612 entrées) et préfixe-suffixe (1138 entrées).

Ainsi, l'analyseur donne en sortie l'unité lexicale, sa catégorie morphosyntaxique et sa traduction anglaise [ATT, 06].

6.5 MADA

MADA (Morphological Analyzer and Disambiguator of Arabic) réalisé par Habash en 2005, est un analyseur morphologique de l'arabe. Cet analyseur réalise la segmentation, la diacritisation, la lemmatisation, l'étiquetage grammatical et l'analyse morphologique. Les données d'apprentissage de MADA proviennent du corpus the Penn Arabic Treebank PATB [BES, 01], le corpus d'apprentissage contient 120 000 mots alors que 12 000 mots ont été utilisés pour l'évaluation.

6.6 MorphArabe

C'est un analyseur développé en langage orienté objet par l'équipe lyonnaise SILAT en 2002. C'est un programme informatique qui présente une collection de classes et d'objets. Il est entrainé sur le corpus Dinar.1. Il identifie les traits morphosyntaxiques des mots. D'après (Abbes, 2004), le moins ambigu des marqueurs est la racine. L'ajout de nouveaux traits augmente la discrimination dans l'analyse et offre plus de solutions [GHO, 11].

6.7 Racineur de larkey

L'approche de [DIA, 04] est une analyse morphologique assouplie. Elle consiste à essayer de découvrir les préfixes et les suffixes ajoutés à l'unité lexicale : par exemple le duel (ان) dans (معلمان, deux professeurs),le pluriel des noms masculins (ين, ون) dans (معلمون, des professeurs) et féminins (ات)dans (مسلمات musulmanes) , la forme possessive (نا , هم , كم) dans (كتابهم, ses livres) et les préfixes dans les articles définis (ال , وال , بال, كال, فال).

6.8 Racineur de Khoja

Le racineur de Khoja [KHO, 01] est un logiciel développé par Shereen khoja au sein de l'université de Lancaster en 2001[KHO, 01], a été utilisé dans le cadre d'un système de recherche d'information développé à l'Université du Massachusetts [LAR, 02]. Consiste à détecter la racine d'une unité lexicale, d'une part, il faut connaître le schème par lequel elle a été dérivée et supprimer les éléments flexionnels (préfixes et suffixes) qui ont été ajoutés, d'autre part comparer la racine extraite avec une liste des racines préalablement conçue.

7. Conclusion

Ce chapitre a pour but de présenter les caractéristiques de la langue arabe, commençant par la description de ses propriétés linguistiques : la voyellation, l'agglutination et la flexibilité. Ensuite, nous avons présenté les niveaux de traitement d'un langage naturel, puis les problèmes rencontrés lors du traitement automatique de la langue arabe comme l'absence de voyellation, la flexibilité et l'ambiguïté. Finalement, nous avons recensé quelques applications concernant le traitement automatique de la langue arabe à savoir les analyseurs morphologiques et les racineurs de la langue l'arabe.

1. Introduction

L'analyse est une tâche centrale dans les applications du TALN, elle peut être superficielle ou profonde. Ces deux types ne nécessitent pas la même quantité d'information et ne donnent pas le même résultat.

Il existe plusieurs niveaux d'analyse auxquels on peut s'arrêter pour repérer les différents éléments constituants le texte et en définir les frontières. On peut s'arrêter au niveau de la phrase, au niveau de la proposition ou à celui du syntagme. Mais on peut arriver aussi au niveau du mot graphique, au niveau des unités lexicales ou aller au-delà de celles-ci pour arriver aux unités de base qui compose ces mots qui sont les morphèmes.

Une analyse syntaxique permet de définir des sous structures dans une phrase, comme les structures nominales, verbales et prépositionnelles [ABN, 91] [RAM, 95]. Ces structures sont les résultats du chunking. Au cours de ce chapitre, nous allons définir c'est quoi le chunking, pour quoi on l'utilise, ses étapes, ainsi que certains exemples de chunkers.

2. Les types de segmentation

La segmentation est une étape fondamentale dans le traitement automatique d'une langue, son rôle est de découper un texte en unités d'un certain type qu'on aura définies et repérées préalablement. Selon la visée de l'analyse à entreprendre : lexicale, morphologique ou syntaxique, on peut généralement parler de trois grands types d'applications de segmentation :

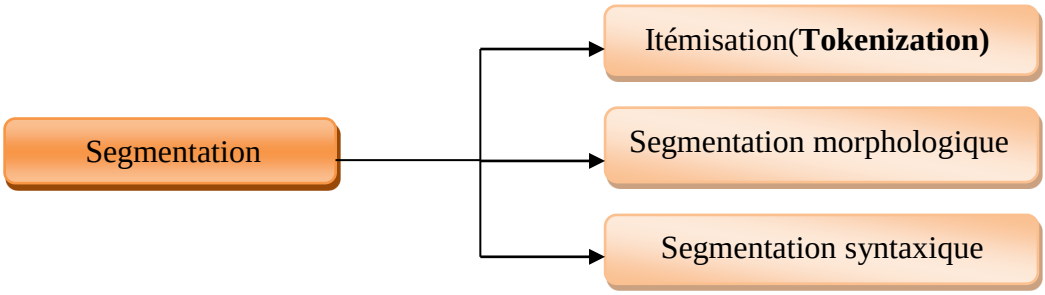


Figure 02 : Les types de segmentation

2.1 L'itémisation (Tokenization ou Word Segmentation)

C'est la segmentation d'un texte en mots ou items lexicaux (tokens). Ce type de segmentation est aussi appelé segmentation lexicale [ZOU, 02].

Exemple : pour la phrase **« أكل الولد التفاحة »,** l'itémisation consiste à séparer tous les mots de cette phrase, Comme la montre la figure suivante :

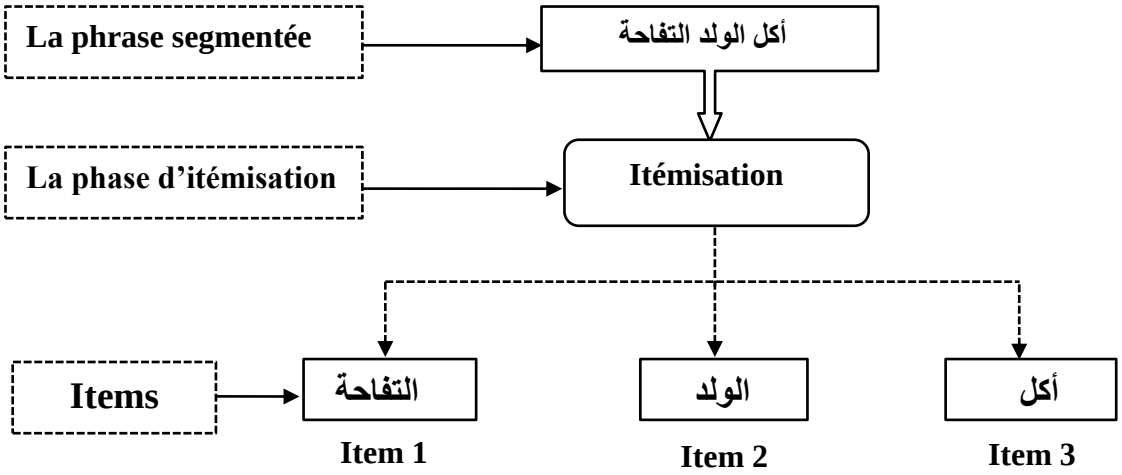


Figure 03: Exemple d'itémisation de la phrase « **أكل الولد التفاحة** »

2.2 La segmentation morphologique

Cette segmentation va plus loin que la segmentation lexicale en cherchant à isoler les différents constituants des items lexicaux en unités distinctes, plus petites, qui sont les morphèmes [ZOU, 02].

Exemple : pour le mot **« سأكتبها »,** la segmentation morphologique est comme suite **س/أكتب/ها** en séparant les clitiques et la base du mot.

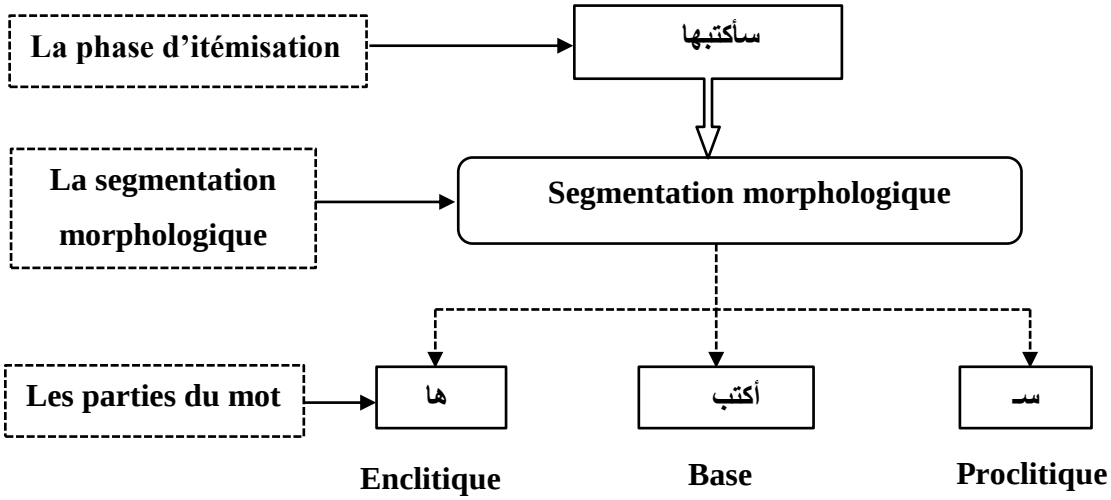


Figure 04: Exemple de la segmentation morphologique du mot « **سأكتبها** »

2.3 Le chunking

Le chunking consiste à isoler les différents constituants du texte en unité indépendantes, supérieures aux mots, comme les propositions, les syntagmes… etc. Ce type de segmentation est aussi appelé segmentation syntaxique [ZOU, 02].

Exemple : Soit la phrase « **قرأ التلميذ في الدفتر الكبير** », le chunking nous donne la segmentation suivante : **(قرأ) GV التلميذ (GN) (في الدفتر الكبير) GP.**

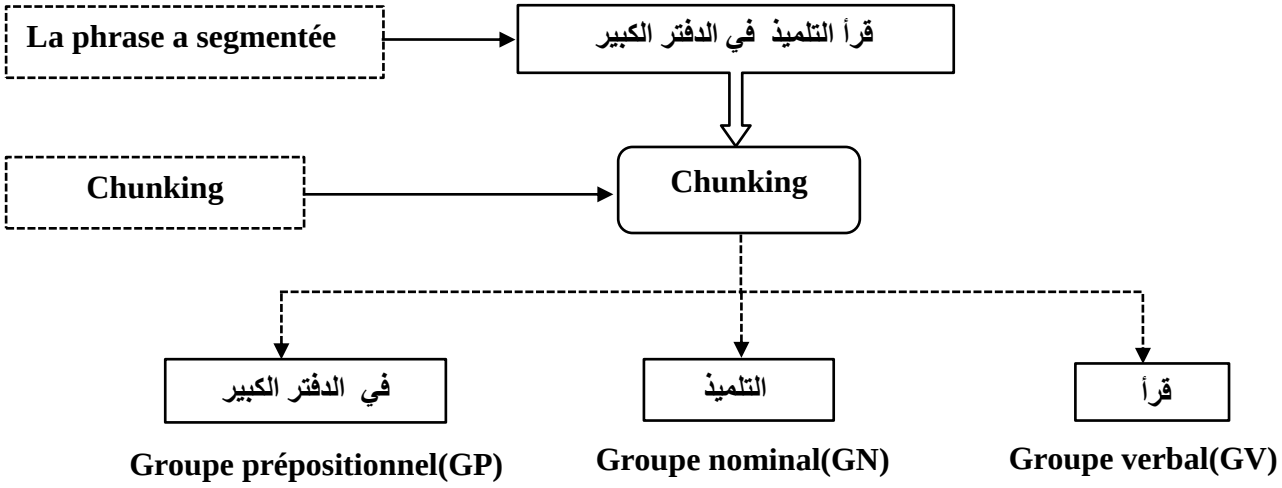


Figure 05: Exemple de Chunking de la phrase « **قرأ التلميذ في الدفتر الكبير** »

3. C'est quoi la grammaire en tronçons (Chunking) ?

une grammaire en tronçons est proposée dans un contexte qui est fondée sur une analyse superficielle et non-exhaustive du texte, cette grammaire consiste à diviser la phrase en groupes de mots non récursifs, baptisés **chunks** en anglais [ABN, 91], **tronçons** en français [BOU, 97], sans nécessairement les mettre en relation les uns avec les autres. Les mots appartenant à un même tronçon se caractérisent par des liens syntaxiques forts : ainsi, leur ordre dans le tronçon est rigide comparé à l'ordre des tronçons dans la phrase, qui est relativement flexible.

Les grammaires en tronçons proposées dans la littérature, qu'elles soient probabilistes ou par règles, intègrent des termes coordonnés ou certains syntagmes prépositionnels, prenant ainsi une décision de rattachement.

La notion de **chunk** peut définir comme un groupe syntaxique minimal non récursif. Chunks sont des mots (ou ensembles de mots) qui composent correctement des sous-structures syntaxiques. Il se compose d'un mot de tête et ses mots modificateurs voisins (tels que les adjectifs, les adverbes et/ou modificateurs) [FER, 12].

Exemple : Considérons l'énoncé suivant : « **نجح الطالب في الامتحان** ».

Le chunking, segmente cette phrase en différents groupes selon les liens syntaxiques définis par la grammaire en tronçons de l'arabe. Cette grammaire définis les groupes suivants : le groupe verbal, le groupe nominal et le groupe prépositionnel.

Donc pour l'énoncé précédent, on obtient le chunking suivant :

Tronçon	Type de tronçon	L'étiquetage adéquat
نجح	Verbale	**GV**
الطالب	Nominal	**GN**
في الامتحان	Prépositionnel	**GP**

Tableau 12 : Exemple du chunking de la phrase « **نجح الطالب في الامتحان** »

4. Pour quoi La grammaire en tronçons ?

La grammaire en tronçons peut être utilisé pour des fins diverses. Elle peut servir à la découverte d'unités de traduction, à l'extraction d'information ou à la génération automatique d'index : dans ce domaine, la plupart des efforts sont concentrés sur l'identification des groupes nominaux de base [RAL 95]. Comme les techniques utilisées dans [RAL, 95] (issues de Brill [BRI, 93]), des arbres de classification et de régression (CART) ont également été appliqués dans [HIR, 96], pour positionner des frontières intonatives [PHI, 01].

De plus, le chunking est appliqué pour le parenthésage prosodique et à la génération de la prosodie .En utilisant les informations attachées aux parties du discours et un jeu de règles, pour assigner des frontières et des mouvements prosodiques en synthèse de la parole à partir du texte [PHI, 01].

La phrase arabe est caractérisée par une grande variabilité au niveau d'ordre de ses mots, c'est la propriété de flexibilité qui laisse le sens d'une phrase correcte malgré le changement de positions des mots. Cette flexibilité, provoque des ambiguïtés syntaxiques artificielles. Pour cela, il faut préciser dans la grammaire toutes les règles de combinaisons possibles d'inversion de l'ordre des mots dans la phrase qui sont réalisées par des linguistes.

Si on fait le changement de positon des mots d'une phrase il peut changer le sens de la phrase ou bien elle n'aura pas de sens. La grammaire en tronçons garde la rigidité et la cohérence des mots au sein d'un même tronçon, donc le sens de la phrase ne va pas changer lors du changement de positions des tronçons. **Exemple** : Pour la phrase en arabe suivante, dans l'ordre **verbe+ sujet+ complément** : « **ظهر الإسلام في شبه الجزيرة العربية** ».

Le chunking de cette phrase est de cette forme : **GV (ظهر) GN (الإسلام) GP (في شبه الجزيرة العربية).**

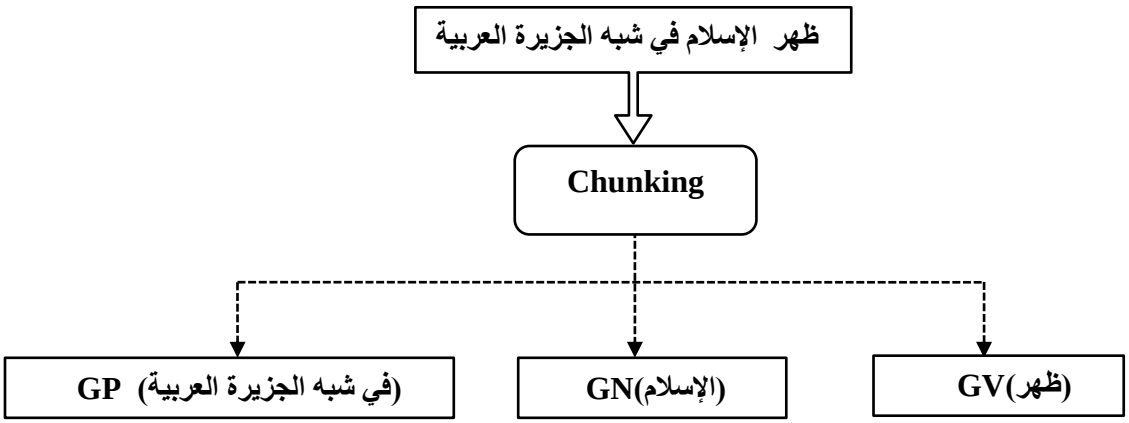


Figure 06 : Le chunking de la phrase arabe « **ظهر الإسلام في شبه الجزيرة العربية** »

Deux autres possibilités d'expression peuvent être adoptées, sans faire changer le sens de la phrase grâce au découpage de la phrase en tronçons:

- Dans l'ordre **sujet+ verbe+ complément** : (الإسلام) **GN** (ظهر) **GV** (في شبه الجزيرة العربية) **GP**.
- Dans l'ordre **complément** + **verbe+ sujet :** (في شبه الجزيرة العربية) **GP** (ظهر) **GV** (الإسلام) **GN**

L'exemple montre le rôle de la grammaire en tronçons en conservant le même sens de la phrase malgré la flexibilité des composantes de la phrase. Ce qui réduit ou élimine le problème de flexibilité de la langue arabe.

5. Les étapes du chunking

Le rôle de l'analyse syntaxique est de reconnaître la structure de la phrase : identification du sujet et des objets liés au verbe, reconnaissance des groupes… etc. [SAB, 89].

Les connaissances syntaxiques, appelées aussi grammaire, expriment les relations entre les catégories de mots.

Le problème rencontré lors d'une analyse syntaxique est celui de l'ambiguïté, soit au niveau des mots (la morphologie qui peut proposer plusieurs étiquettes pour un token) soit au niveau des phrases (la syntaxe qui propose plusieurs structures pour une phrase). cette ambiguïté suit une fonction exponentielle qui augmente avec le nombre de mots dans la phrase [VER, 98].

Pour réduire ce double problème d'ambiguïté (flexibilité des mots et des phrases), des nouvelles méthodes ont été développées, fondées sur un étiquetage morpho-syntaxique, ces méthodes permettent d'introduire des informations sur le contexte immédiat des mots.

5.1 L'étiquetage morpho-syntaxique

L'étiquetage morpho-syntaxique a pour but d'associer une étiquette grammaticale à chaque mot de la phrase (simples, composés, noms propres, abréviations) et leurs traits (genre et nombre). La première étape est alors de définir un jeu d'étiquettes. En effet, l'étiqueteur morpho-syntaxique doit fournir au module de découpage en tronçons toutes les informations grammaticales dont il a besoin pour mener à bien son processus [YAM, 05].

La taille d'un jeu d'étiquettes varie en fonction de la finesse des informations linguistiques représentées. Elle varie de quelques dizaines à quelques centaines d'étiquettes. Cette taille dépend à la fois de la langue traitée et de choix stratégiques concernant l'étiqueteur (information requise dans l'étiquetage, performance a priori, moins il y a d'étiquettes parmi lesquelles il faut choisir, plus la tâche d'étiquetage est facile[1].

Remarque

Certaines langues dont la morphologie est très riche nécessitent plus d'étiquettes, d'autres moins, par exemple le jeu d'étiquettes utilisé dans le corpus annoté américain PennTreebank ne compte que 36 étiquettes[1.]

L'étiquetage morphosyntaxique peut être vu comme la composition de 3 fonctions :

5.1.1 La segmentation

Le but de la segmentation est de découper le texte en phrases puis en mots distincts (Tokenization).

5.1.2 L'analyse morphologique

Cette fonction fait attribuer des étiquettes aux mots (tokens) au moyen des informations lexicales (qui associe toutes les étiquettes possibles pour un mot donné) qui rendent compte la nature du mot (verbe, nom, particule), leur état déterminé/indéterminé , car ce choix est étroitement lié au regroupement en tronçons [BAL, XX].

5.1.3 La désambiguïsation

La désambiguïsation consiste à sélectionner l'étiquette la plus adéquate parmi celles identifiées dans la phase d'analyse morphologique, en fonction du contexte du mot. Les règles de déduction contextuelle (locale), interviennent après l'analyse morphologique ont pour tâche de vérifier l'étiquetage établi, tenant compte le contexte d'apparition des mots. les seules déductions fiables sont celles qui n'utilisent que les contraintes locales fortes [GIG, 98] dont le contexte est restreint à un seul tronçon. Ceci suppose que les frontières de tronçons soient connues au préalable. Parce que l'ordre des tronçons à l'intérieur d'une phrase obéit à des contraintes relativement souples, comparé à celles agissant sur les mots à l'intérieur de ces groupes.

[1] http://www.technolangue.net/ Etiquetage Morphosyntaxique.html, Patrick Paroubek (consulté le 10/04/2016.)

Exemple : voilà l'exemple suivant de deux phrases distinctes qui contient le même mot « **ذهب** », dont sa fonction est totalement différente relativement au contexte de chacune des phrases :

« **ذهب الأولاد في نزهة** » et « **اشتريت عقدا من ذهب** ».

Il s'agit, pour le mot "**ذهب**" de le catégoriser en tant que verbe dans la première phrase et en tant que nom dans la seconde. Cette distinction est faite grâce à la désambiguïsation en exploitant la base des déductions contextuelle pour le mot "**ذهب**".

Pour la première phrase :

Pour la deuxième phrase :

Figure 07: Exemple de déduction contextuelle pour enlever l'ambiguïté sur le mot « **ذهب** »

Exemple d'étiquetage morphosyntaxique (POS Tagging)

Pour la phrase en arabe suivante « **وضع الولد الصغير محفظته فوق الطاولة** », L'étiquetage morpho-syntaxique supposons que le jeu d'étiquettes contient les étiquettes associées aux unités de la phrase est comme suite:

- La segmentation : **وضع/ الولد / الصغير/ محفظته/ فوق/ الطاولة**
- L'étiquetage morpho-syntaxique (POS Tagging) :

وضع الولد الصغير محفظته فوق الطاولة						
Les segments de phrase	**وضع**	**الولد**	**الصغير**	**محفظته**	**فوق**	**الطاولة**
Les étiquettes	V	Nds	Jds	Nis+ Prp	Pr	Nds

Tableau 13 : Exemple d'étiquetage morphosyntaxique d'une phrase arabe

La séquence de tokens étiquetée est ensuite passée au module suivant de parenthésage syntaxique qui a pour tâche de découper cette séquence en tronçons. Mais le problème est de savoir quelles sont les séquences d'étiquettes susceptibles d'appartenir à un même tronçon.

5.2 Parenthésage syntaxique

Le parenthésage syntaxique consiste à ajouter des informations syntaxiques telles que la division en groupes, le marquage des relations de dépendances…etc.

La tâche d'annotation syntaxique est d'autant plus difficile à mener à bien qu'elle peut être appréhendée de différentes façons et fournir plusieurs types d'informations. Une synthèse des niveaux d' d'analyse contient deux niveaux principales est définit comme suite [ELI, 03] :

5.2.1 Le découpage en groupes

Cette tâche est obligatoire si on veut étiqueter syntaxiquement un corpus, en effectuant le regroupement des unités étiquetées précédemment dans la phase d'étiquetage morphosyntaxique [ELI, 03].

5.2.2 Etiquetage des groupes

Ce niveau consiste à assigner des étiquettes aux groupes précédemment définis. Il est généralement considéré comme faisant partie de toute tâche d'étiquetage syntaxique [ELI, 03].

Ces deux niveaux sont les seules qui se trouvent dans tout étiquetage syntaxique, mais il existe d'autres niveaux qui sont :

- ✓ **Marquage des relations de dépendance** : qui consiste a indiqué les relations entretenues entre les groupes.
- ✓ **Etiquetage fonctionnel** : ce niveau consiste à indiquer qu'elles fonctions grammaticales ou syntaxiques sont remplis par les groupes (sujet, verbe, objet, etc.).
- ✓ **Indications des sous –classifications** : ce niveau consiste à fournir des informations supplémentaires qui entrent en ligne de compte dans les phénomènes d'accord, de flexion, etc. (genre, nombre, temps, etc.).
- ✓ **Marquage du rang :** ce niveau consiste à noter l'enchâssement des éléments [ELI, 03].

Le parenthésage syntaxique bien qu'elle soit liée d'une certaine façon à l'étiquetage morphologique (qui constitue très souvent une phase pré requis aux opérations de l'étiquetage syntaxique), est beaucoup plus difficile à mener à bien, à cause de certaines difficultés :

- ✓ **La complexité de la tâche** : car les unités a étiquetées ne sont pas prédéfinis, donc il faut segmenter l'énoncé puis l'étiquetées.
- ✓ **L'absence de consensus** : Il n'existe pas de consensus sur les unités ou segments syntaxique (sauf le syntagme nominal ou prépositionnel), sur les étiquettes à assigner à ces segments et sur les relations qu'entretiennent entre eux [ELI, 03].

Exemple de parenthésage syntaxique

Le parenthésage syntaxique de la phrase « وضع الولد الصغير محفظته فوق الطاولة » après le découpage en groupes et l'association des étiquettes a chacun d'entre eux, en supposons que parmi les groupes qu'on peut constituer dans l'analyse on a le groupe nominal, verbal et prépositionnel, est comme suite :

وضع الولد الصغير محفظته فوق الطاولة						
Les segments de la phrase	وضع	الولد	الصغير	محفظته	فوق	الطاولة
Les étiquettes	V	Nds	Jds	Nis+Prp	Pr	Nds

GV **GN** **GN** **GP**

Les troncons

Tableau 14 : Exemple de parenthésage syntaxique de la phrase « وضع الولد الصغير محفظته فوق الطاولة »

6. Exemple de chunker

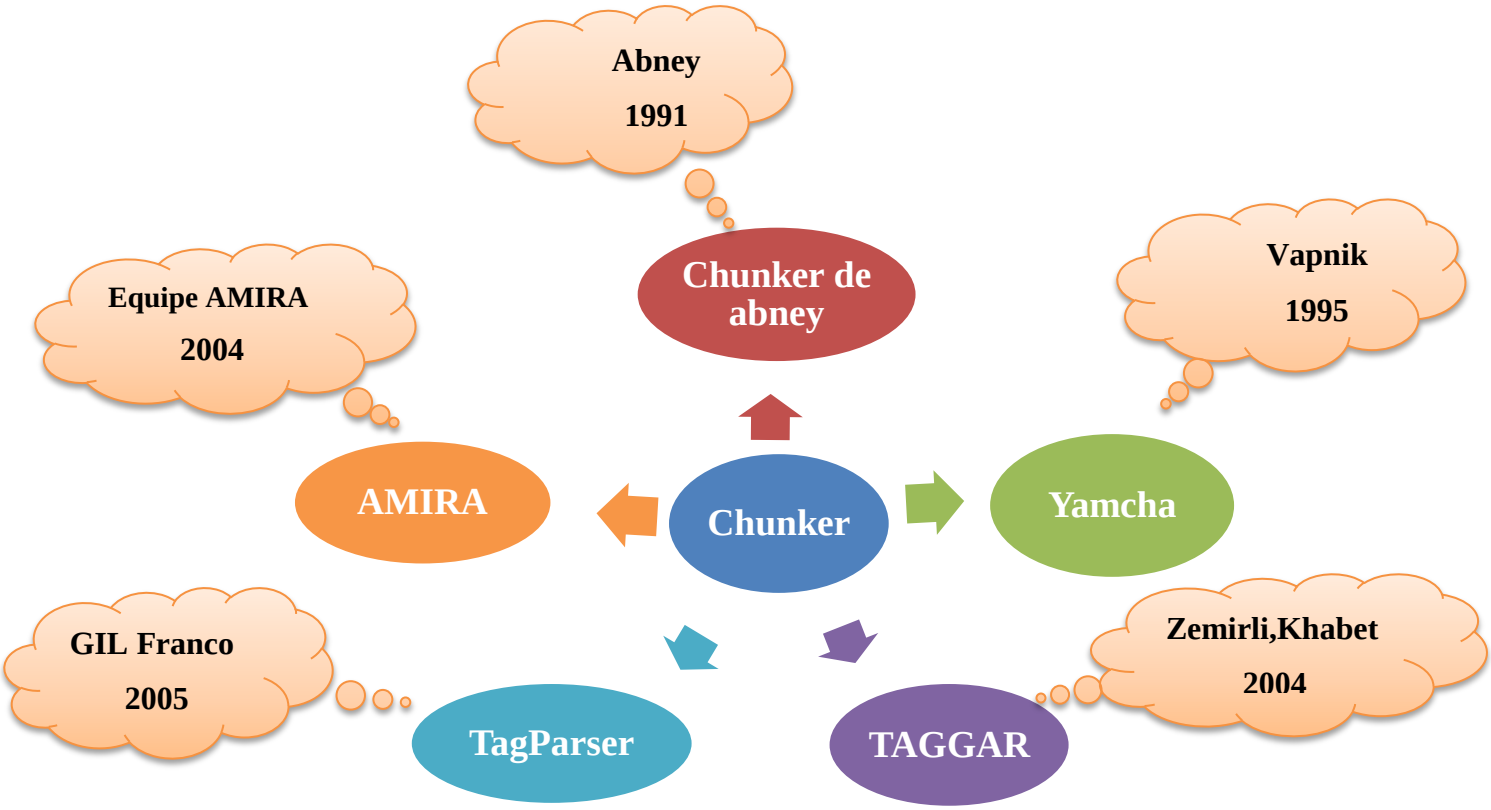


Figure 08 : Exemples de chunkers

7. Conclusion

Dans ce chapitre nous avons présenté le Chunking qui a pour rôle le découpage des longues chaînes d'information en chunks (tronçons) moins long, moins complexe et facile à comprendre. Il se base sur une grammaire qui dirige le découpage en tronçons en définissant les règles de regroupement des unités lexicales dans un tronçon.

Pour arriver au Chunking d'un texte un étiquetage morphosyntaxique des unités lexicales permet de réduire l'ambigüité, puis un parenthésage syntaxique de ces unités selon la grammaire en tronçons définis dans le langage étudié sont effectués successivement afin de générer des tronçons. Ces tronçons peuvent être utilisés pour la traduction automatique, l'indexation ou pour la synthèse de parole.

1. Introduction

Le développement d'une application concernant l'automatisation de la langue arabe a suscité ses dernières décennies l'intérêt de nombreux chercheurs. Donnant ainsi naissance à une panoplie de produits, mais aussi la création de plusieurs firmes[1].

La mise en place de tels outils considérés dans leurs majorités comme étant fortement complexe, ainsi que leur réussites dépendent essentiellement de la performance de plusieurs modules complémentaires, tel que la segmentation syntaxique.

L'analyse syntaxique fait habituellement suite à une analyse morpho-lexicale qui découpe le texte en un flux de lexèmes. Connaître la structure syntaxique d'un énoncé permet d'expliciter les relations de dépendance entre les différents lexèmes, puis de construire une représentation du sens de cet énoncé.

Ce chapitre est consacré à la description de l'architecture de notre chunker, ainsi que les défirent phase de développement de cet outil.

2. Le principe de chunker «ArabChunk»

Le **chunker "ArabChunk"** est une application en traitement automatique de la langue arabe qui permettant de diviser la phrase en groupes syntaxiques. Chaque groupe de tronçon a une désignation spécifique.

3. Architecture générale de chunker «ArabChunk»

La réalisation de notre application est divisée en deux phases principales :

- L'étiquetage morphosyntaxique.
- Le Parenthesage syntaxique.

[1] Xeros, Sakhr, Synapse.

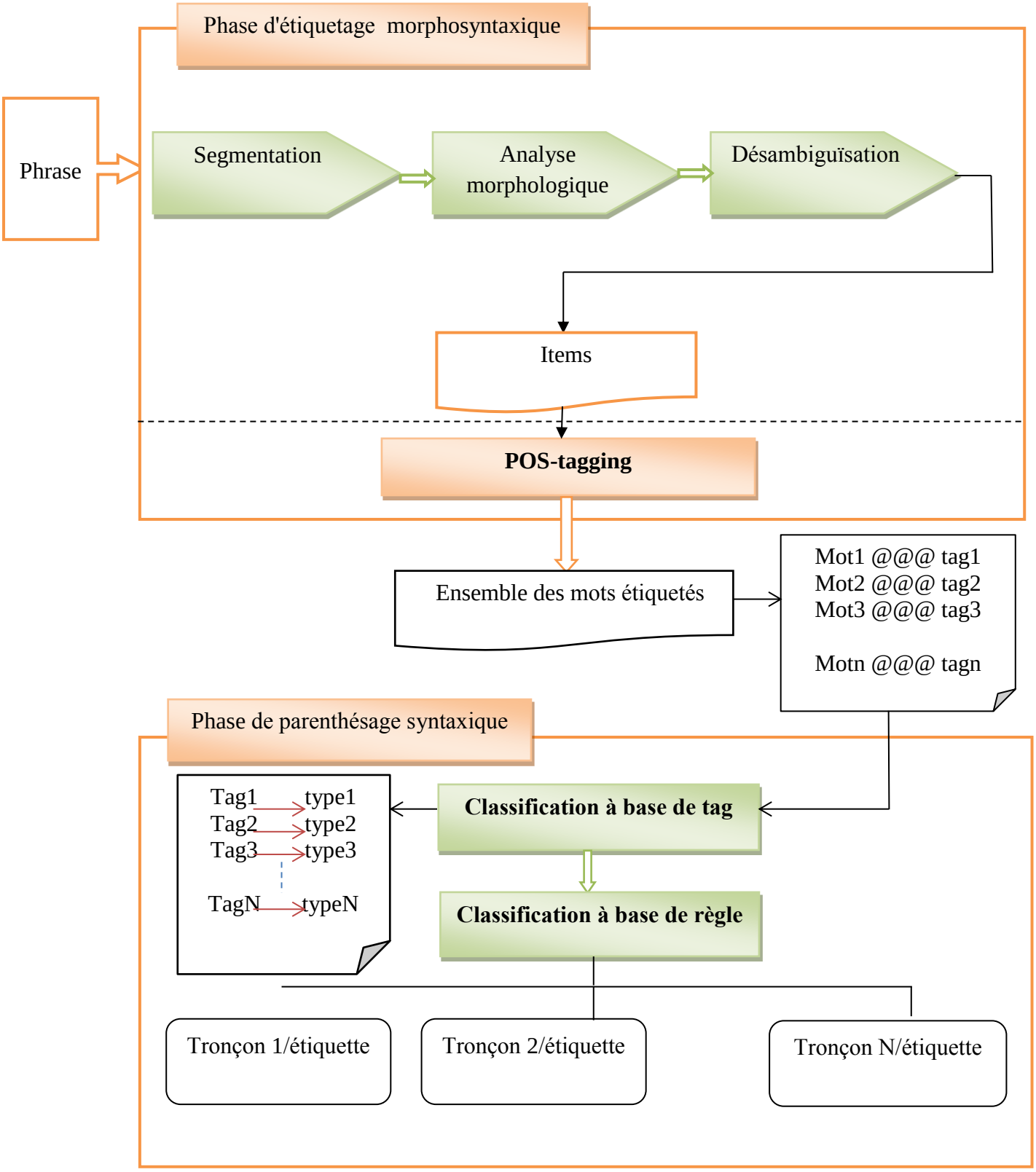


Figure 09 : Schéma générale de l'outil **ArabChunk**

4. La phase d'étiquetage morphosyntaxique

L'étiquetage morphosyntaxique constitue une étape essentielle pour réaliser notre application car il permet d'identifier la catégorie grammaticale à laquelle appartiennent les mots de la phrase. Pour cela nous avons choisi l'outil AMIRA 2.0 (C'est un analyseur gratuit développé en perl par l'équipe de Mona Diab en 2010 à l'université de Columbia New York) comme un étiqueteur, car il obtient des résultats performants au niveau de la segmentation (99.2%) et une précision de plus de 96% au niveau de l'étiquetage [GHO,11].

- **La segmentation :** est la première étape Effectuée par l'outil, pour découper le texte en phrases puis en mots distincts;
- **Analyse morphologique :** cette étape consiste à attribuer des étiquettes aux mots (tokens) au moyen des informations lexicaux (qui associer toutes les étiquettes possibles pour un mot donné);
- **Désambiguïsation :** cette phase consiste à sélectionner l'étiquette la plus adéquate parmi celles identifiées dans la phase d'analyse morphologique**.**

5. La phase de parenthésage syntaxique

Avant d'expliquer la phase de parenthésage ou groupement syntaxique, il faut premièrement présenter notre étude sur la grammaire en tronçons.

5.1 Étude de la grammaire en tronçons

5.1.1 Les types de phrases

Il existe deux types de phrase arabe :

- Phrase nominal : sujet +attribut/prédicat.
  Exemple : الطالب جميل
- Phrase verbal : verbe +sujet+(ou sans) complément d'objet (direct ou indirect).
  Exemple : جلس الطالب

5.1.2 Les liens syntaxiques

Rappelons que le groupement syntaxique consiste à diviser la phrase en groupe ayant des liens syntaxique fort, comme :

a. L'adjectif (المركب الوصفي)

En arabe, l'adjectif est un mot qui permet de décrire un nom .L'adjectif est appelé « **نعت** » (le qualifiant) ou « **صفة** » (la description) qui suit en genre, en nombre et en état (défini / indéfini) le qualifié « **المنعوت** » [NED ,XX].

Exemple	**رجل فقير.**	**المدينة الكبيرة.**
Traduction	Un homme pauvre.	La grande ville.

Tableau 15: Exemples d'adjectifs

b. La possession / annexion (الإضافة)

L'annexion ou Al-idafa est traduit littéralement par l'ajout ou l'annexion. Il permet de déterminer un nom en l'annexant a un autre en remplissant la fonction de complément de nom, ce groupe nominal se compose ainsi des deux termes dits [NED,XX]:

- Premier terme : mudhaf « **مضاف** ».
- Deuxieme terme : mudhaf ilayhi « **مضاف اليه** ».

Le premier terme de l'annexion ne prend jamais d'article défini, car il sera déterminé par le deuxième mot qui est à son tour déterminé par :

- Un pronom affixe 'Dhamir motasil'.

  Exemple : كتابي = ي+كتاب
- l'article défini 'ال'.

  Exemple : الكتاب

Un nom déterminé peut être aussi :

- Un pronom personnel 'Dhamir monfasil' : أنت، هم، هما، هي، هو, .نحن انا....etc.
- Un pronom démonstratif : ذلك، هؤلاء، هذه، هذا...etc.
- Un nom propre : محمد، علي

Exemple	**كتاب المدرس.**	**قلم محمد.**
Traduction	Le livre de l'enseignant.	Le stylo de Mohamed.

Tableau 16 : Exemples des annexions

c. Les prépositions et le cas génitif (الجار و المجرور)

Les prépositions en arabe sont des unités qui permettent de lier deux mots, phrase ou plus entre eux. Les mots précédés par une préposition passent du cas nominatif au cas génitif «اسم مجرور»[2].

Les prépositions sont :{من، إلى، عن ، على ، في ، رب ، الكاف ، اللام ، الباء}.

Exemple	إلى مسجد.	على الكتاب.
Traduction	Vers une mosquée	Sur le livre.

Tableau 17 : Exemple des prépositions avec des cas

d. L'adverbe de temps et lieu (ظروف الزمان و المكان)

Les adverbes de temps et lieu se sont des noms qui permettent d'indiquer le temps/lieu dans lequel l'action se réalise. Ils s'emploient comme des moudaf-annexé, et comme les prépositions elles aussi ont pour conséquence de rendre le mot suivant au cas génitif[3].

Les adverbes de temps :{ صباح، ظهر، عصر، مساء، ساعة، دقيقة، سنة، لحظة، ثانية، برهة،يوم،أمس، غد، الآن، عشية، ليل نهار دهر، فجر}.

Les adverbes de lieu :{ قدام، خلف، يمين، شمال، شرق، غرب، جانب مكان، ناحية، منطقة، فوق، تحت، أعلى، أسفل، أمام، وراء، بين}.

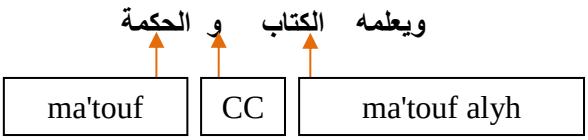
Exemple	صباح الأربعاء.	تحت الطاولة.
Traduction	Mercredi matin.	Sous la table.

Tableau 18 : Exemple des prépositions avec des cas

e. Conjonction de coordination (العطف)

Conjonction de coordination ou « العطف » est un procédé syntaxique qui permet de coordonner deux noms, le premier appel ma'touf alyh « معطوف عليه » alors que le deuxième ma'touf « معطوف ».les trois conjonctions de coordination les plus reconnus sont : «ثم , ف , و» [NED,XX].

Exemple :



[2] http://al-dirassa.com/le%C3%A7on/les-prepostions-et-le-cas-genitif (consulté le 01/05/2016)
[3] http://al-dirassa.com/le%C3%A7on/les-adverbes (consulté le 01/05/2016)

f. Les pronoms relatifs et la phrase relative

❖ **Les pronoms relatifs :** an Arabe, les pronoms relatifs qui servent à former la phrase relative sont les suivant :

Nombre	Singulier	Duel	Pluriel
Masculin	الذي	اللذان	الذين
féminin	التي	اللتان	اللاتي

Tableau 19 : Les pronoms relatifs

On peut aussi utiliser les pronoms relatifs invariables «**من** et **ما** ».

❖ **La phrase relative :** après le pronom relatif, on place la phrase relative «**صلة الموصول**», qui contient un prénom de rappel «**ضمير عائد**», visible ou caché.

Exemple	جاء الذي أكرمته	الذين امنوا
Traduction	Celui que j'ai honoré est revenu	A ceux qui ont cru
Pronom de rappel	'ه' visible	'هو' caché

Tableau 20 : Exemples des phrases relatifs

g. Le substitutif (البدل)

La substitution est le fait de remplacer un nom par autre .les substitutifs s'emploient comme les adjectifs car ils suite en genre, en nombre le mot précède«**المبدل منه**» .Les substitutifs sont toujours des mots déterminés.

Exemple	المعلم احمد.	هدا الرجل تاجر.
Traduction	L'enseignant Ahmed.	L'homme est un commerçant.

Tableau 21 : Exemples des phrases relatifs

Remarque : si deux mot sont défini par l'article 'AL', le deuxième n'est pas un substitutif.

h. Participe actif (اسم الفاعل)

Le participe actif désigne celui qui fait l'action. On construit le participe actif sur le schème **فاعل**: [NED,XX].

Exemple : à partir de «**كتب**»on obtient «**كاتب**»

i. Les particules et leurs relations avec les noms et les verbes

En plus des prépositions et des adverbes, il existe aussi d'autres particules qui peuvent accompagner les noms et les verbes. Ces particules peuvent être classées dans les catégories suivantes :

- ✓ Les particules de négation
- ✓ Les particules d'interrogatives
- ✓ Les particules subjection
- ✓ Les particules d'interjection

5.2 Les types de tronçons

D'après l'étude des liens syntaxiques, nous avons défini huit (08) types de tronçons :

- **Le tronçon verbal :** soyez sous la forme d'un verbe ou un verbe précéder par une particule.
- **Le tronçon nominal :** soyez sous la forme d'un nom, un nom déterminé par annexion ou un nom +adjectif(s).le nom peut être précédée par une particule.
- **Le tronçon prépositionnel :** qui compose d'une préposition +le nom génitif.
- **Le tronçon adverbial :** qui compose d'un adverbe de (temps/lieu) +le complément.
- **Le tronçon conjonctif :** qui compose d'une conjonction de coordination + le nom 'ma'touf'.
- **Le tronçon adjectif isolé :** est la description qui ne suite pas le mot précédé en état (défini / indéfini).
- **Le tronçon relatif :** qui compose d'un pronom relatif + une phrase verbal.
- **Le tronçon participe :** soyez sous la forme d'un participe actif.

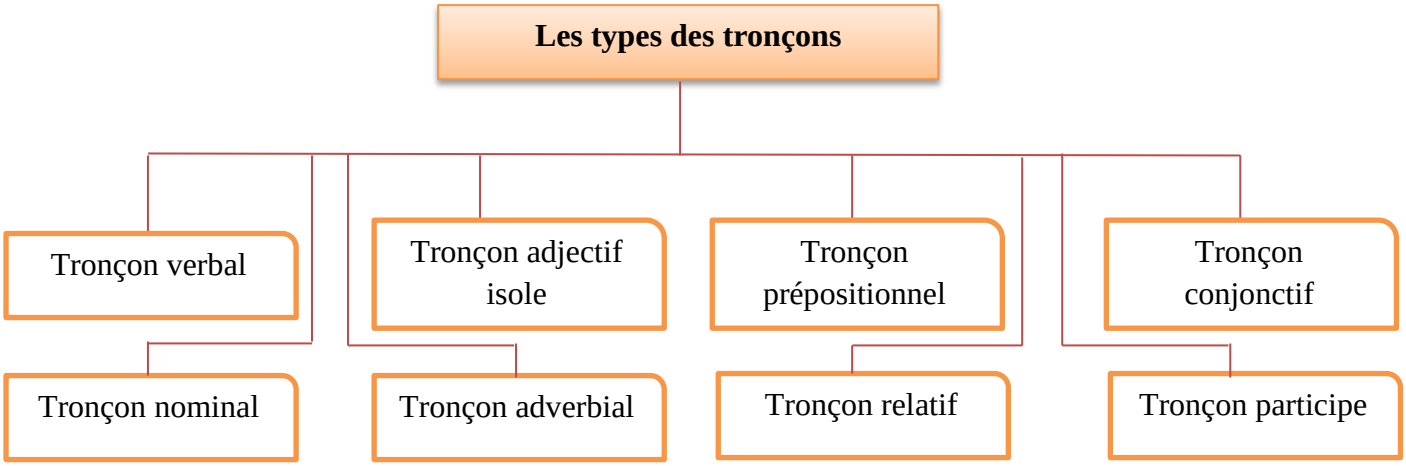


Figure 10 : Les différents types de tronçons

Le tableau suivant résume tous les types de tronçons avec des exemples :

Type de tronçon	Désignation	Composition	Exemple
Verbal	**TV**	Verbe simple	خرج
		particule+ verbe	لم يذهب
Nominal	**TN**	Nom + adjectif	خطئا فادحا
		Nom simple	وزير
		Nom déterminé par annexion	وزير الأعلام
		Nom introduit par une particule	أين خالد
Prépositionnel	**TP**	Préposition + nom génitif	في البيت
Circonstanciel	**TD**	Circonstance + le complément	صباح الأربعاء
Conjonctif	**TC**	Conjonction de coordination+ nom	خرج عمر و محمد
Adjectif	**TADJ**	Adjectif isolé	فريق الشباب قوي
Relatif	**TRV**	Pronom relatif+ une phrase	الطفل الذي خرج
Participe	**TPRP**	Participe actif	واصفا، قائلا

Tableau 22 : Les déférents types de tronçons avec exemples

La phase de parenthésage ou regroupement syntaxique, est divisée en deux phases complémentaires :

- Classification à base de tag.
- Classification à base de règle.

5.3 Classification à base de tag

C'est la première étape dans la phase de groupement syntaxique, elle consiste à ajuster le type de chaque mot selon le tag.

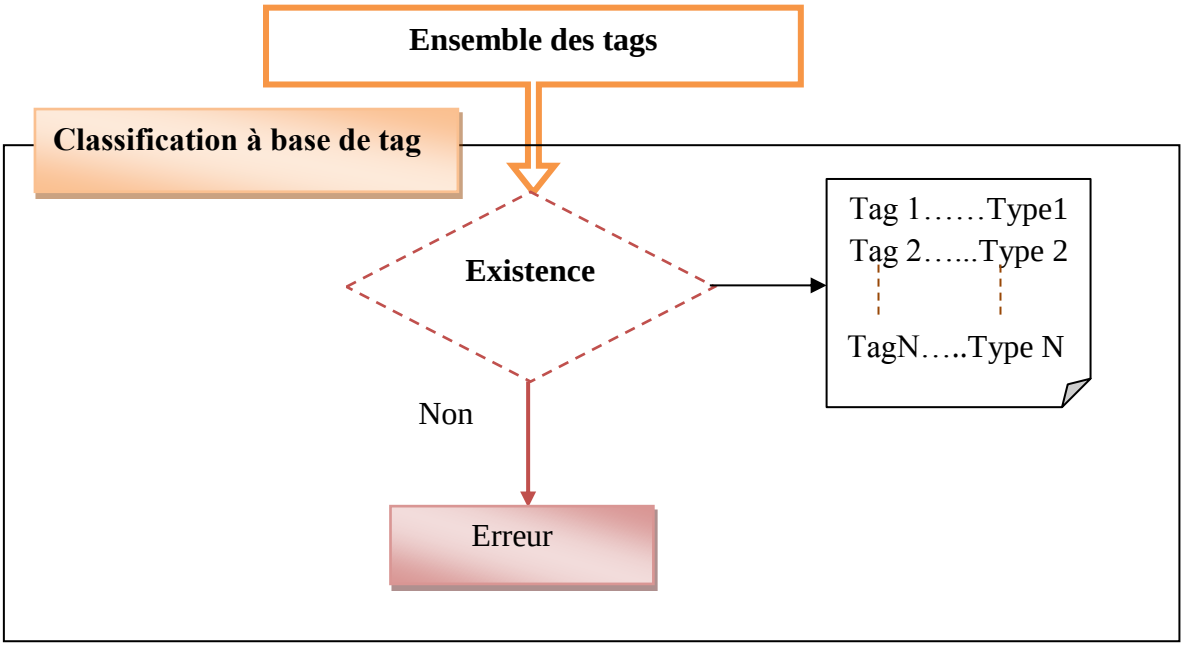


Figure 11 : Schéma générale de classification à base de tag.

A partir de l'étiquetage morphosyntaxique, nous avons défini neuf (09) types de groupes de tags, à savoir :

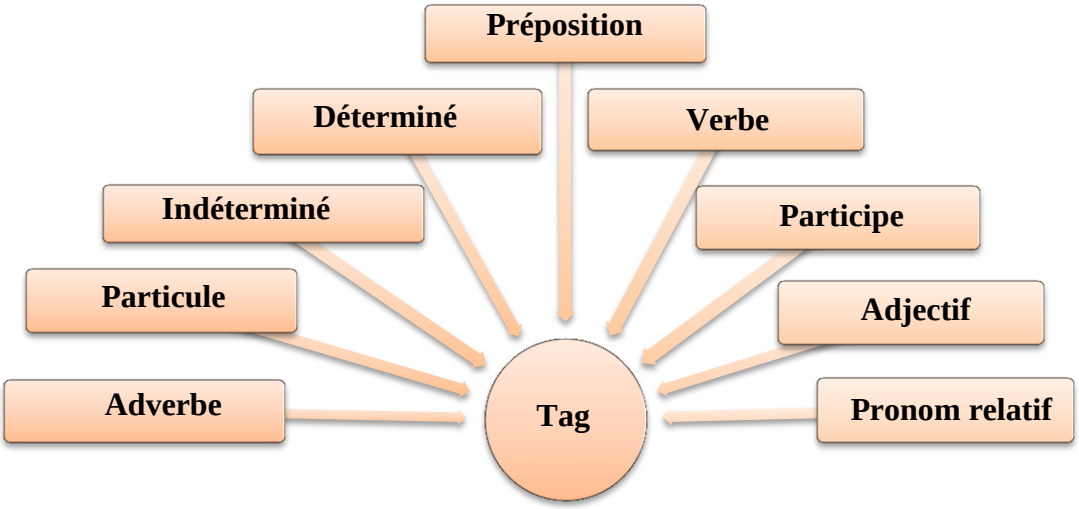


Figure 12 : Les différents types de mots

L'exemple suivant présente la classificatrion de mot «وزيرة» à partir de leur tag.

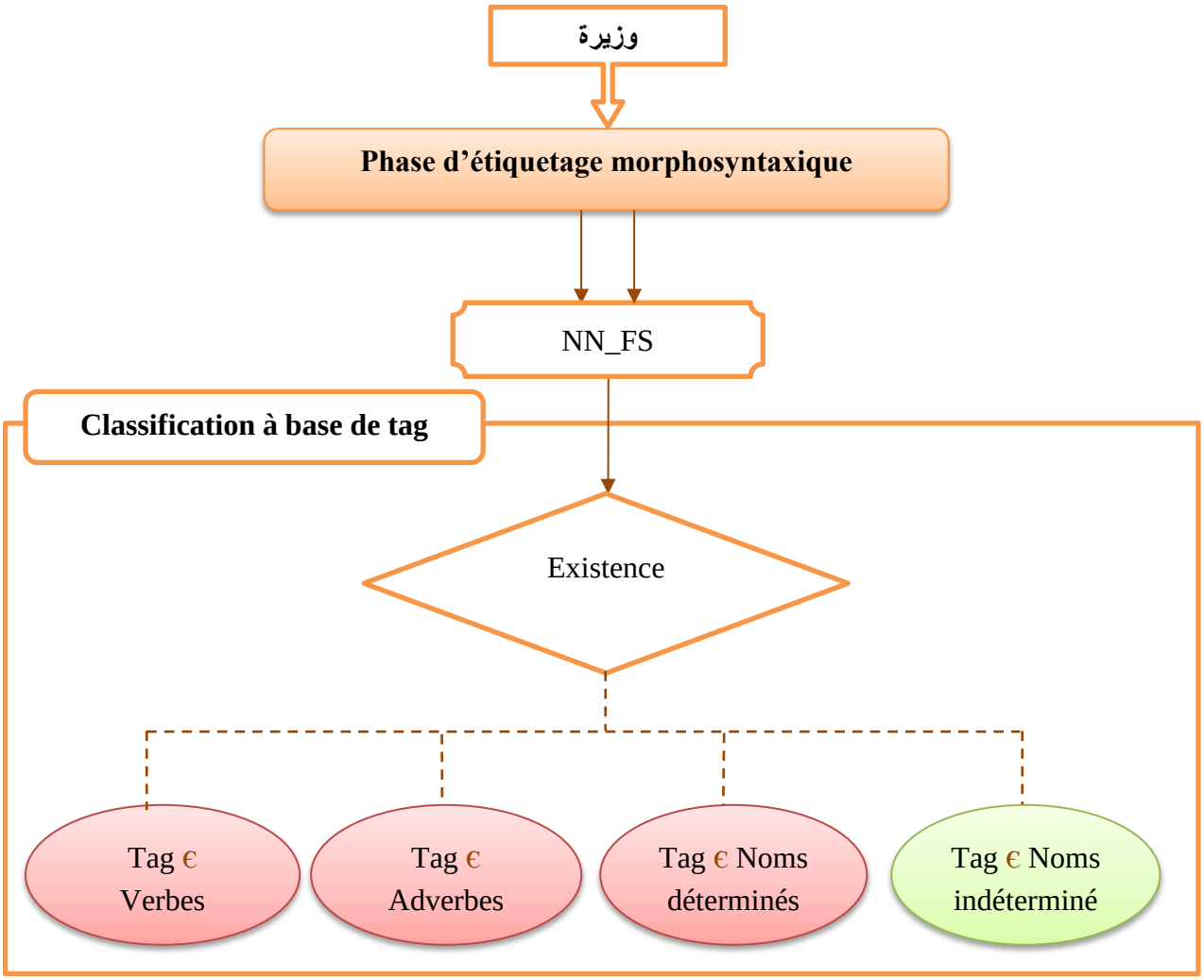


Figure 13 : Exemple de classification à base de tag

Le tag 'NN_FS[4]' est classifié dans le type des noms indéterminés. Cela lui permet d'être classé dans la phase suivante pour former un tronçon.

[4] N : nom
N : indéterminé
F : féminin
S : singulier

5.4 Classification à base de règle

C'est la dernière phase de chunking, elle consiste à regrouper les mots ayant des liens syntaxiques forts les uns avec les autres. Si le tag de mot est classifié dans la phase précédant (classification à base de tag), il est reclassé en conformité avec la règle qui correspond avec lui. À cause de l'étude de la grammaire du la langue arabe, L'ensemble des règles syntaxique que nous avons défini respectent la relation entre les mots qui constituent le tronçon. Chaque règle exprime un certain type de tronçon (verbal, nominal, prépositionnel…etc.)

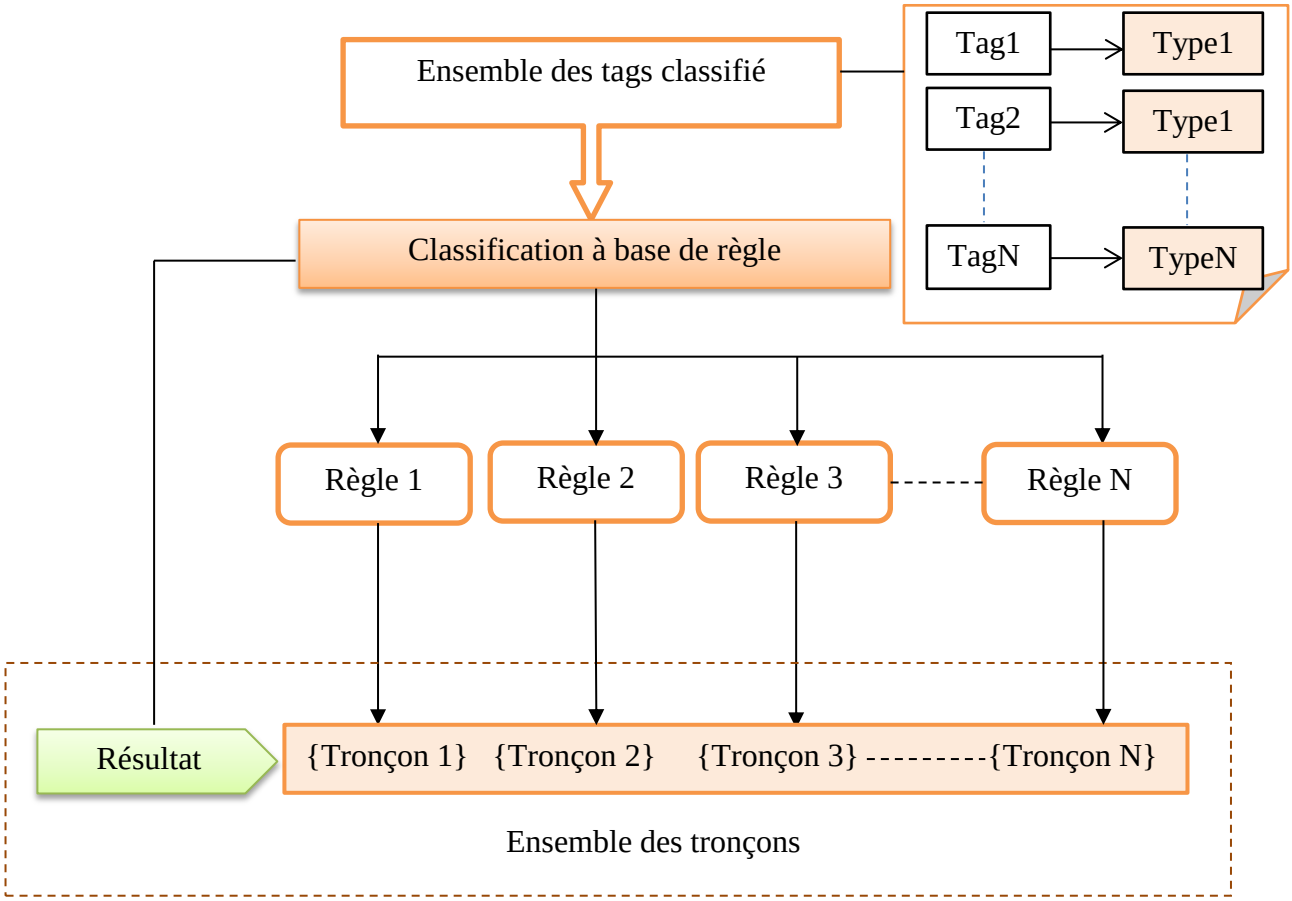


Figure 14 : Schéma générale de classification à base de règle.

L'exemple suivant représente une règle de regroupement syntaxique. Cette règle est défini par un Algorithme qui consiste à coupler les deux termes d'annexion (moudaf ,moudaf ilayh) « **مضاف اليه، مضاف**» telle que, le deuxième n'a pas suivi par un adjectif ou un substitutif .

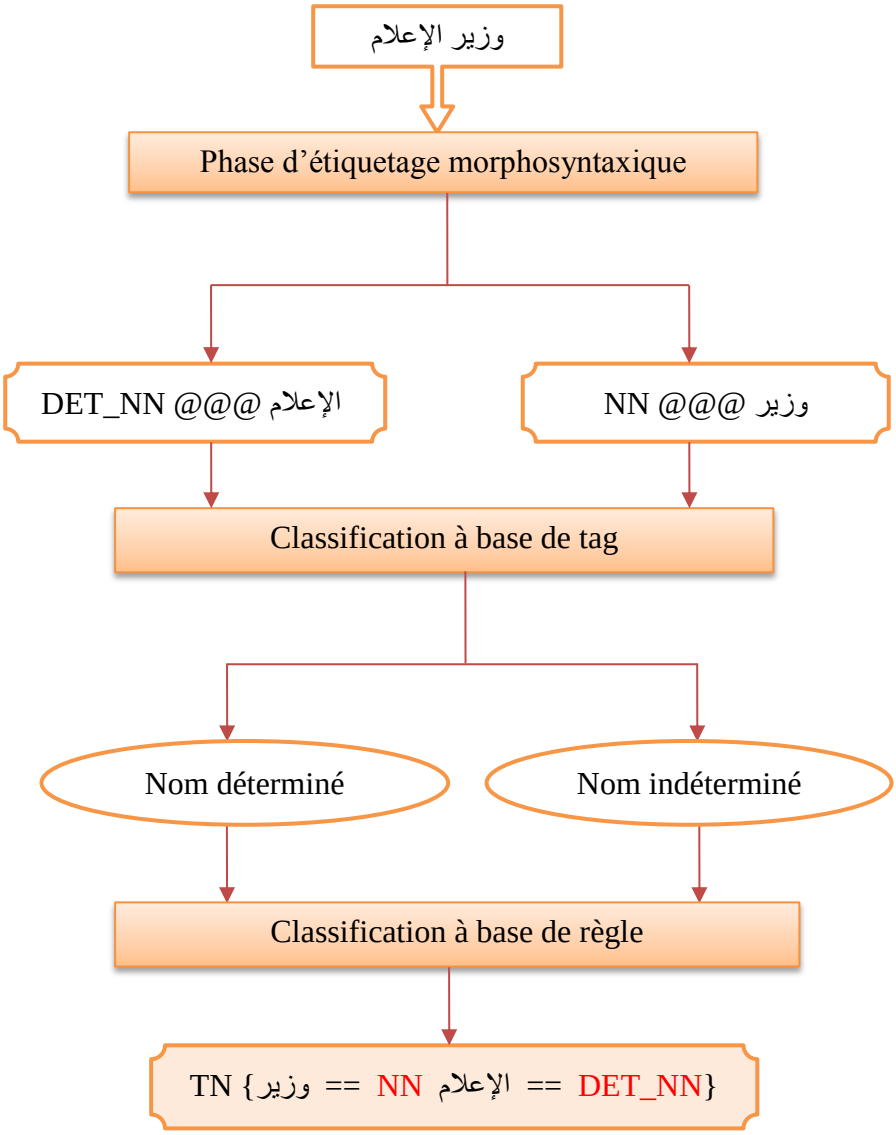


Figure 15 : Exemple de classification à base de règle

Le mot « **وزير** » est déterminé par le mot « **الإعلام**», donc le couple {**وزير الإعلام**} est un tronçon nominal.

6. Comment construisant un tronçon

6.1 Les tronçons nominaux qui commencent par un nom indéterminé

Comme tous les types de mots que nous avons mentionnés dans l'étape de classification à base de tag, le mot indéterminé est classifié à base de règle comme un tronçon nominal, mais ce mot peut être suivre par des mots qui sont associer les uns avec les autres par des liens syntaxiques.

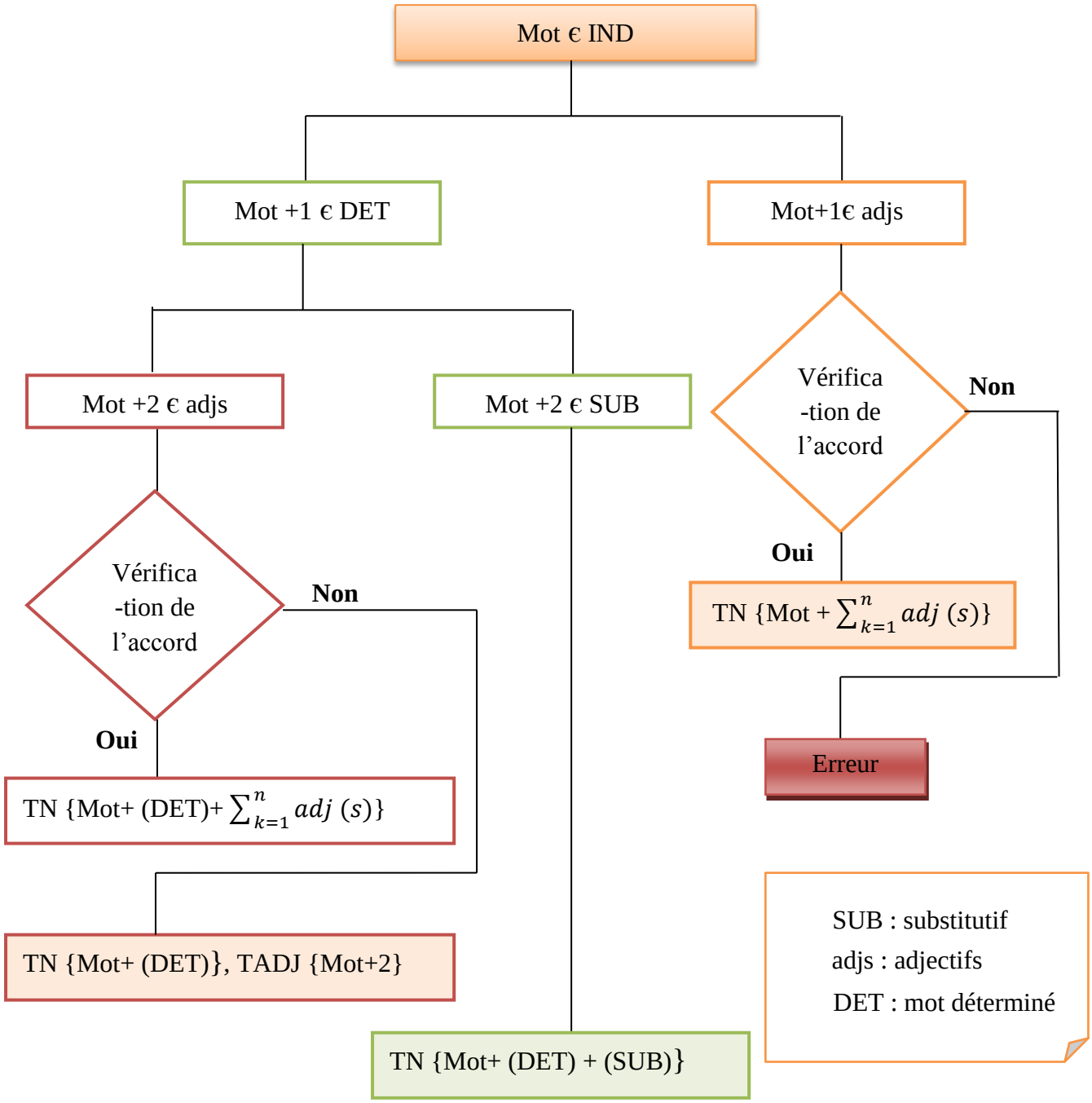


Figure 16 : Les chemins possibles d'un tronçon qui commence par un nom indéterminé

Un nom indéterminé peut accepter un ou plus d'un adjectif(s), mais il faut que ce(s) adjectif(s) sont conforme avec le qualifié pour former avec lui un tronçon nominal, sinon l'adjectif doit être isolé.

Le tronçon nominal regroupe aussi le nom indéterminé et leur déterminant, mais on doit vérifier que ce dernier accepte les mots qui sont en accord avec lui syntaxiquement (les appartenances[5]).

6.2 Les tronçons nominaux et verbaux qui commencent par des particules

Comme nous avons mentionné précédemment, les particules sont des unités qui peuvent introduit les verbes et forment avec eux des tronçons verbaux, et en plus des prépositions et adverbes les particules peuvent introduit aussi les noms et forment avec eux des tronçons nominaux.

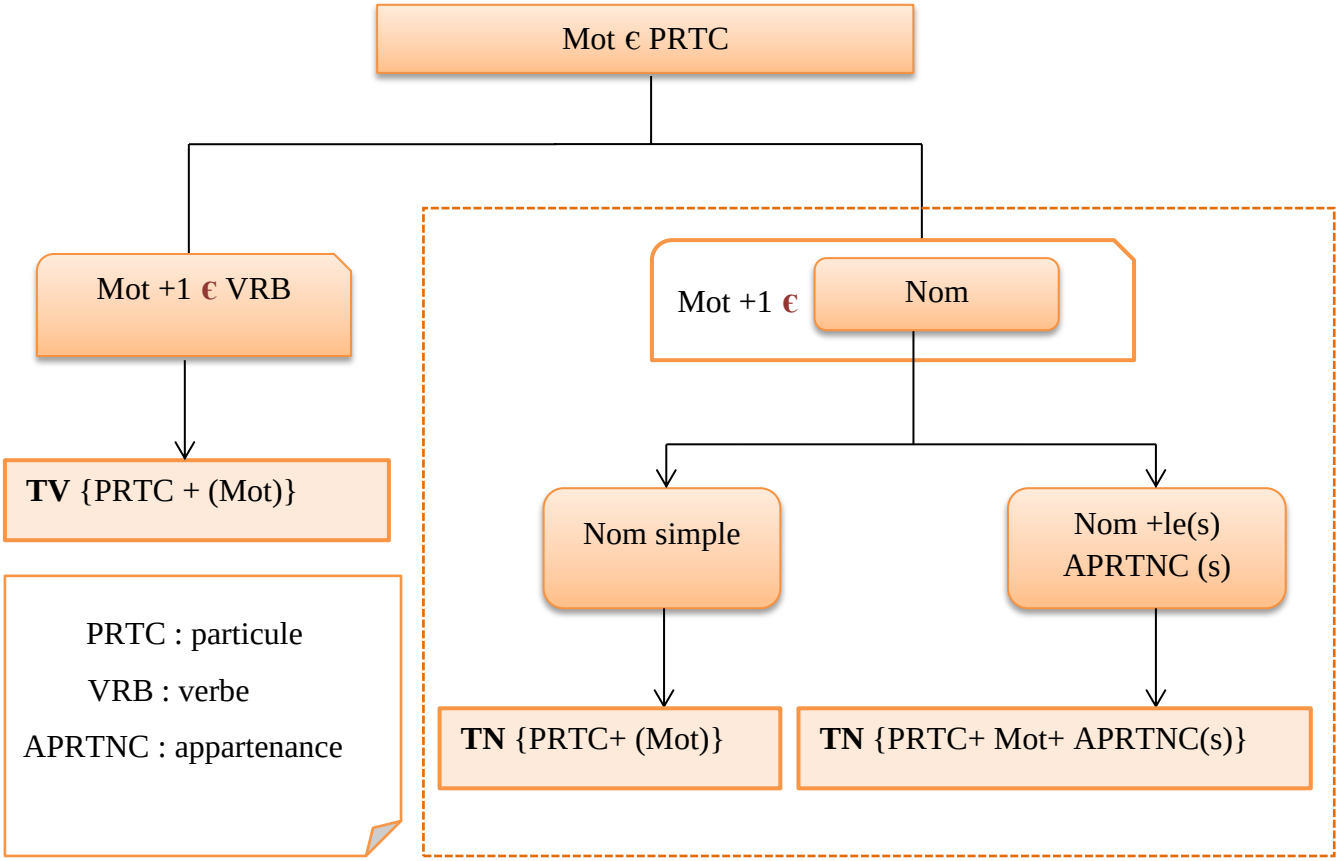


Figure 17 : Les chemins possibles d'un tronçon qui commence par un particule

[5] Les appartenances sont des adjectifs, des substitutif ou des déterminants

6.3 Les tronçons relatifs

Le pronom relatif et la phrase relative constituent ensemble un tronçon Ce que nous avons appelé TRV. Rappelons que le pronom de rappel de la phrase relative peut être visible ou caché.

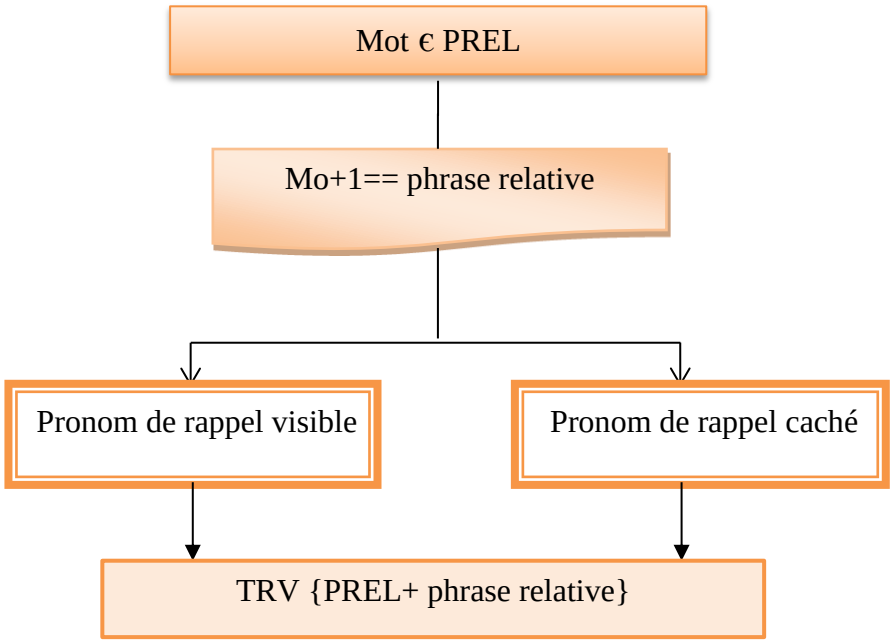


Figure 18 : Le tronçon relatif

6.4 Les tronçons conjonctifs

Dans ce type de tronçon nous vérifions que le premier terme de coordination "ma'touf Alayh" est conforme avec le deuxième "ma'touf" pour coupler ce dernier avec la conjonction de coordination.

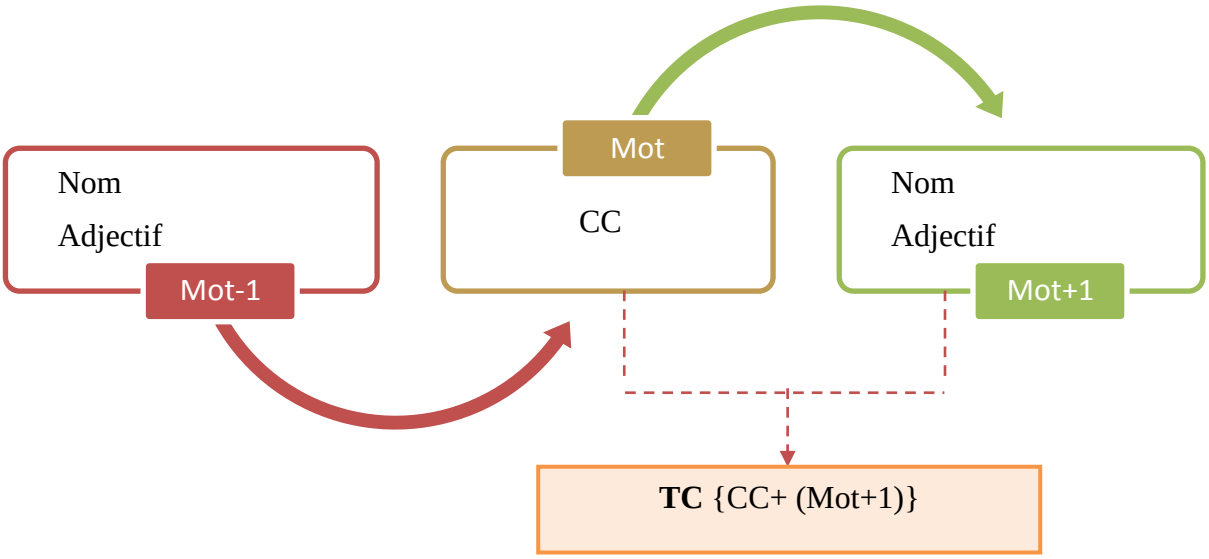


Figure 19 : Le tronçon conjonctif

7. Exemple de découpage d'une phrase utilisant ArabChunk

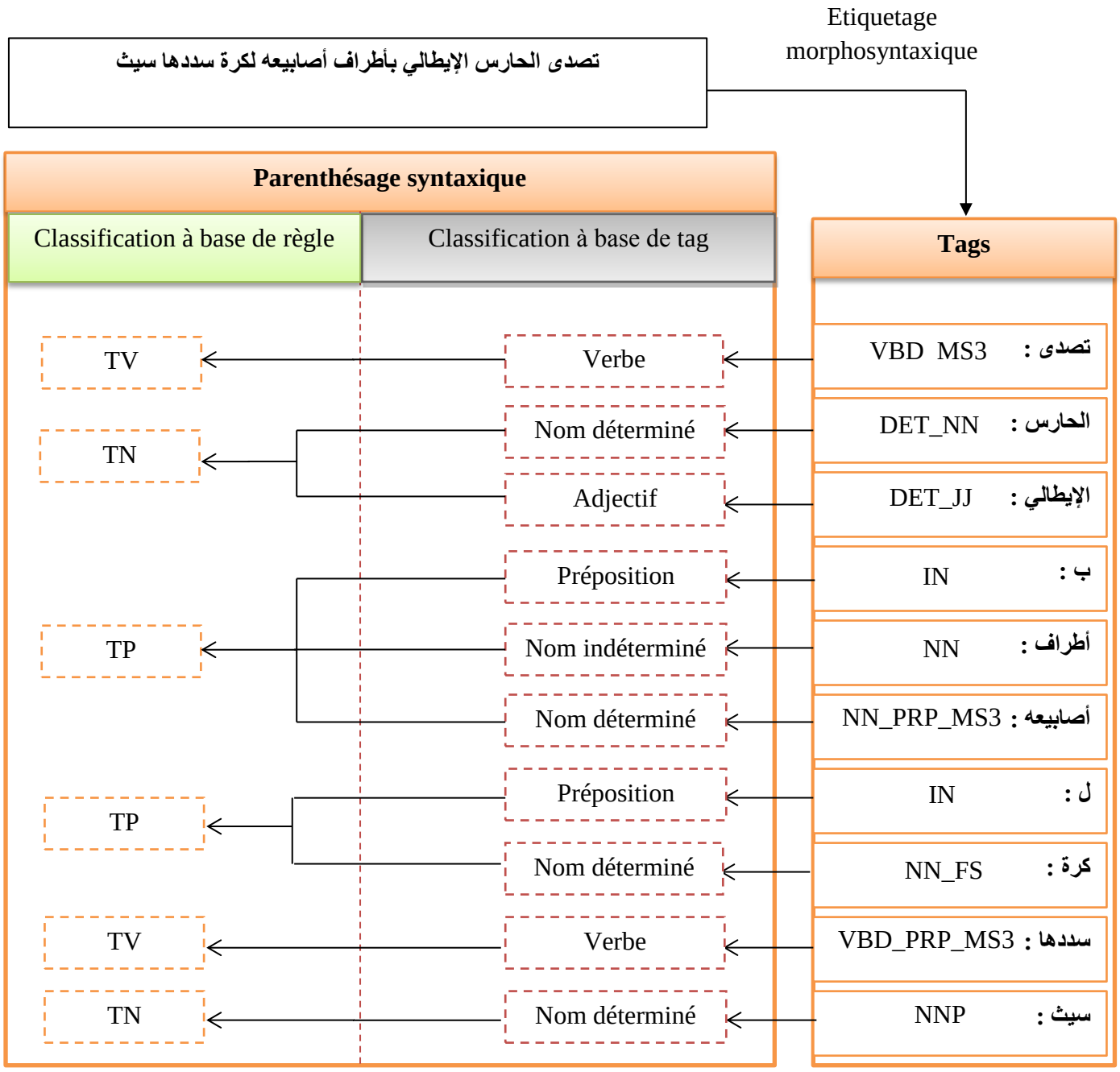


Figure 20 : Exemple de découper une phrase utilisant **ArabChunk**.

8. Conclusion

La conception représente une phase primordiale et déterminante pour produire une application de haute qualité.

Dans ce chapitre, nous avons présenté le nouveau outil de chunking pour les textes arabes non vocalisé, en présentant tout d'abord l'architecture de notre chunker **"ArabChunk".** Ensuite nous détaillons le principe de la grammaire en tronçons, ainsi que les étapes que nous avons employé pour formaliser les défirent types de tronçons.

1. Introduction

Dans ce chapitre nous allons aborder la dernière partie qui représente la partie réalisation de notre projet, en se basant sur les mécanismes évoqués précédemment dans la phase de conception.

Ce chapitre est composé de deux parties : la première présente l'environnement de développement alors que la seconde partie concerne les interfaces graphiques de notre logiciel, ainsi que les résultats des tests effectués.

2. Environnement de développement

2.1 Environnement hard

- Hôte : DELL INSPIRON N3500.
- Microprocesseur : Intel **I3** Inside.
- RAM : 4GO.

2.2 Présentation de langage de programmation python

a. Histoire

Python est un langage objet interprété de haut niveau, créé au début des années quatre-vingt-dix par **Guido Van Rossum** au **Centrum voor Wiskunde à Informatica**, Amsterdam. . En 1995, **Rossum** poursuivit le développement de Python à la **Corporation for National Research Initiatives** de **Reston** (Virginie). Et en 2000, **Rossum** créa l'équipe **BeOpen PythonLabs** qui, en octobre de la même année est incorporée à **Zope Corporation** puis à la société **Digital Creations**. En 2001, la **PSF** (Python Software Foundation) est créée. Il s'agit d'une organisation à but non lucratif détenant les droits de propriété intellectuelle de Python.
Elle est sponsorisée en particulier par **Zope Corporation**. Python est distribué sous forme de logiciel libre [XAV.XX].

b. Description

On distingue plusieurs classes parmi les langages informatiques selon la syntaxe qu'ils proposent ou les possibilités qu'ils offrent. Python est un langage :

- interprété : car il est directement exécuté sans passer par une phase de compilation qui traduit le programme en langage machine.
- 2. orienté objet : car il propose des fonctionnalités avancées et automatiques ainsi que des structures de données complexes telles que des dictionnaires, éloignées des types numériques standards.

- 3. de haut niveau : des structures de données complexes telles que des dictionnaires, éloignées des types numériques standards.
- 4. modulaire : Python est assez intuitif, être à l'aise avec ce langage revient à connaître tout autant sa syntaxe que les nombreux modules disponibles, eux-mêmes écrits en Python.
- 5. à syntaxe positionnelle : En Python, seule l'indentation permet de marquer le début et la fin d'un tel bloc, ce procédé consiste à décaler les lignes vers la droite pour montrer qu'elles appartiennent au même bloc d'instructions [XAV.XX].

2.3 Plateforme

Linux est un système d'exploitation complet et libre, qui peut être utilisé en lieu de systèmes d'exploitation commercialisés, tels que Windows, de Microsoft. Il est accompagné de nombreux logiciels libres complémentaires, offrant un système complet aux utilisateurs.

Le système peut être utilisé sur des serveurs (LAN[1] ou serveurs web), sur des PC ou encore sur des smartphones.

Linux est disponible en plusieurs versions, téléchargeables gratuitement sur le net, et nommées "**distributions**", telles que **Debian** ou **Ubuntu**[2].

[1] Le terme **LAN** (acronyme de Local Area Network) désigne un **réseau local**, constitué d'un ensemble d'ordinateurs (et de périphériques) connectés au moyen de liaisons physiques (câbles, wifi, CPL, ...) et utilisant des protocoles de communication communs.(consulté le 02/05/2016)

[2] http://www.mosaique-info.fr/glossaire-web-referencement-infographie-multimedia-informatique/l-glossaire-informatique-et-multimedia/425-linux-definition.html.(consulté le 02/05/2016)

3. interface graphique de chunker « ArabChunk »

L'image ci-dessous représente l'interface de notre logiciel. Elle se compose d'une barre d'outils, un bouton de traitement (chunking) et trois (03) zones de texte, les deux qui sont au sommet pour l'entrée/ sortie des phrases, tandis que la troisième en bas consacré aux informations statistiques (la fréquence des différents type des mots et des tronçons).

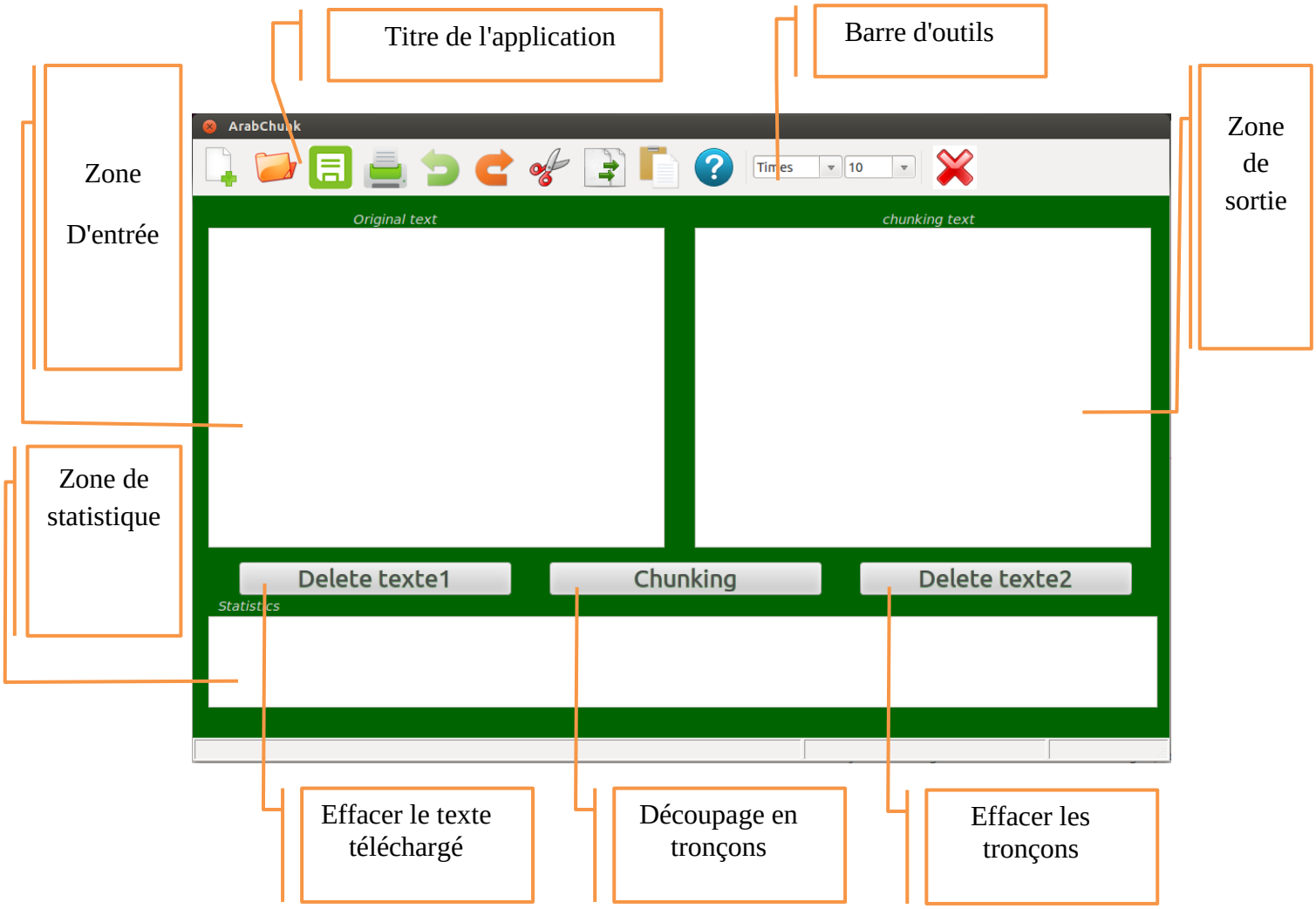


Figure 21 : Interface graphique de chunker **ArabChunk**.

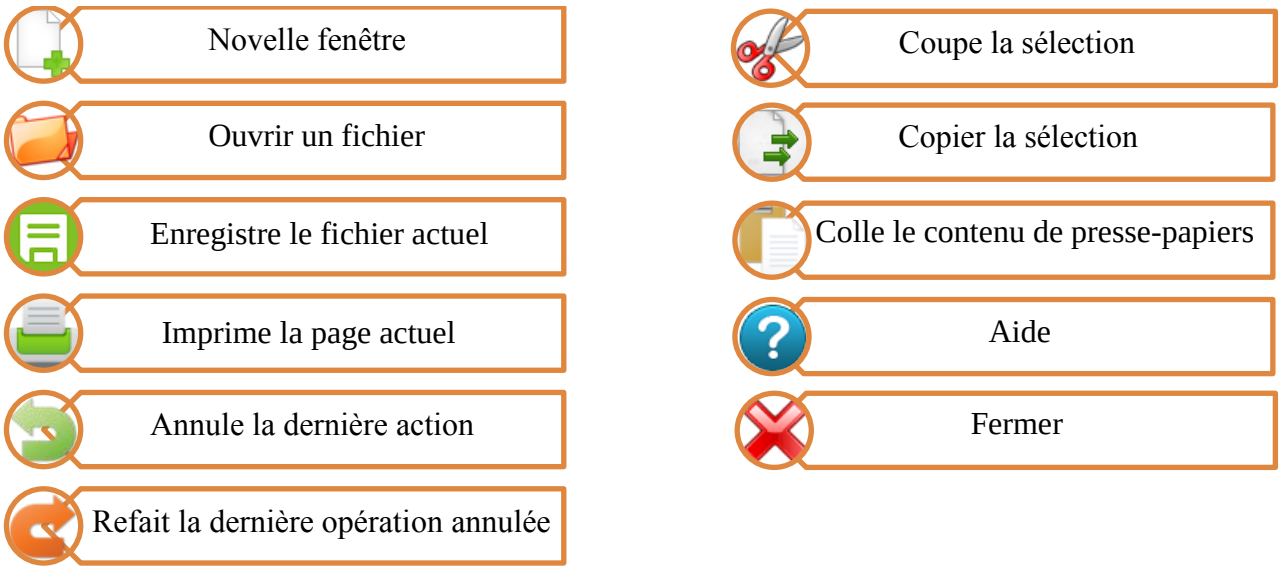


Figure 22 : Description de barre d'outils.

L'image ci-dessus représente la manière d'introduire un texte dans la zone d'entrée par le menu ouvrir de la barre d'outils.

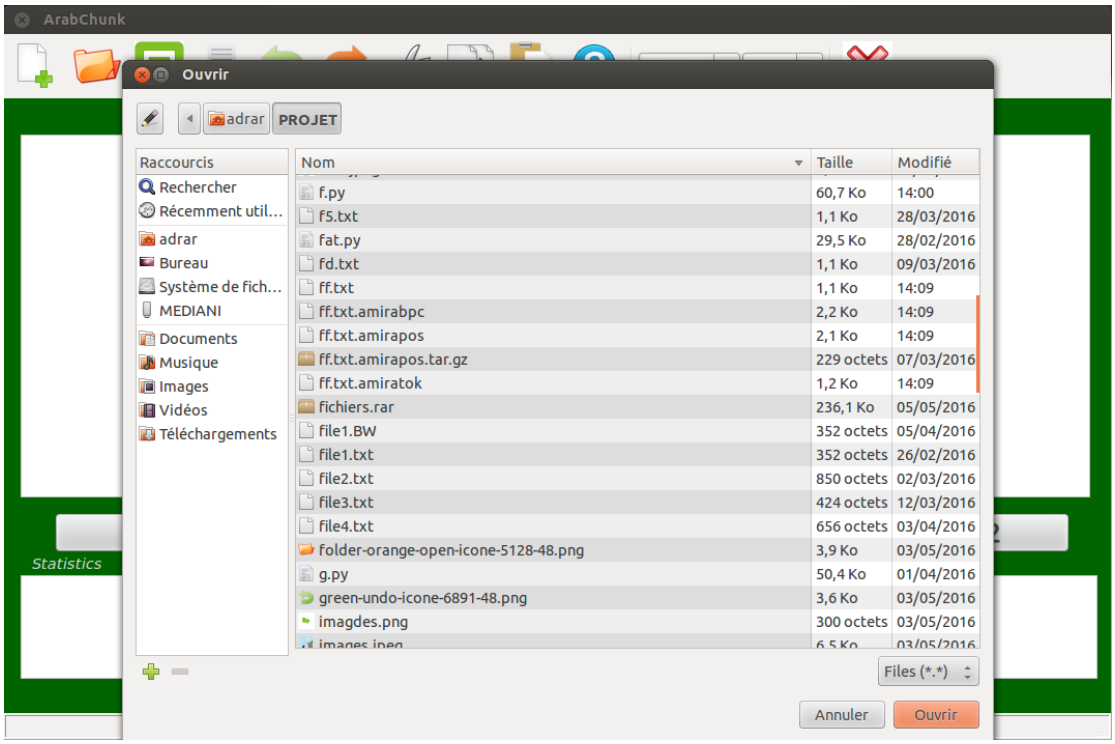


Figure 23 : Téléchargement d'un texte.

La deuxième zone de texte affiche les tronçons de la phrase en entrée à partir du bouton chunking.

Figure 24 : Résultat du chunking.

La segmentation de la phrase **« خسرت اسكتلندا امام استراليا في مباراة دولية ودية في كرة القدم مساء الاربعاء في غلاسكو»** est sous la forme suivant :

.	في غلاسكو	مساء الاربعاء	في كرة القدم	في مباراة دولية ودية	امام استراليا	اسكتلندا	خسرت
out	**TP**	**TD**	**TP**	**TP**	**TD**	**TN**	**TV**

Tableau 23 : Exemple du chunking

5. Evaluation

L'évaluation d'un système est une étape cruciale dans le processus de développement, car elle permet de mettre en évidence les points forts et les limites de ce système, afin de dégager des pistes pour l'amélioration.

Nous avons évalué la qualité des tronçons corrects à partir d'un test fait sur les trois (03) catégories du corpus «Khaleej» qui sont représenté dans le tableau suivant:

Catégorie	Texte	Nb des Phrases	Nb des mots	Taux de groupe syntaxique corrects	Taux moy
International News	Texte 01	04	72	100%	96.86%
	Texte 02	08	149	96.13%	
	Texte 03	09	187	95.04%	
	Texte 04	04	64	97.22%	
	Texte 05	03	57	95%	
	Texte 06	04	114	97.82%	
Sport	Texte 01	04	53	93.75%	90.52%
	Texte 02	06	137	89.04%	
	Texte 03	06	132	92.39%	
	Texte 04	05	109	86.77%	
	Texte 05	05	122	90.69%	
Local News	Texte 01	06	104	90.68%	86.84%
	Texte 02	03	60	89.44%	
	Texte 03	06	150	83.61%	
	Texte 04	05	102	83.66%	

Tableau 24 : Evaluation qualitatif de **ArabChunk**

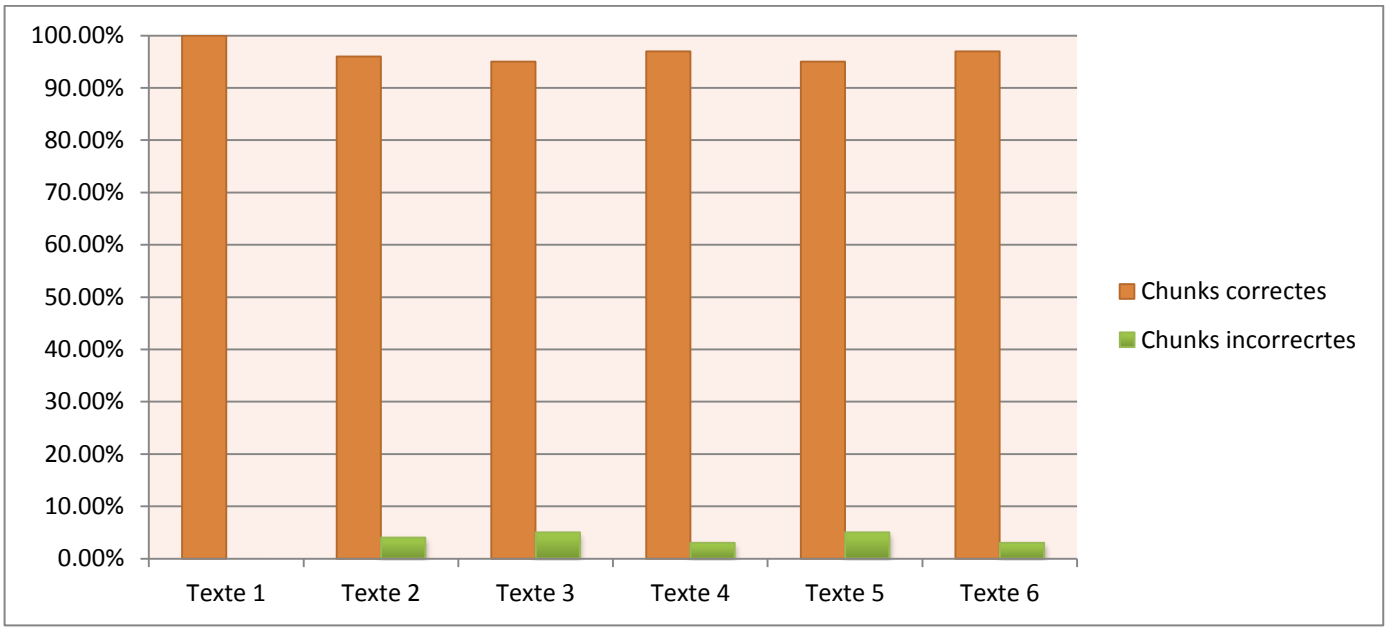


Figure 25: Représentation graphique de test de première catégorie de corpus Khaleej

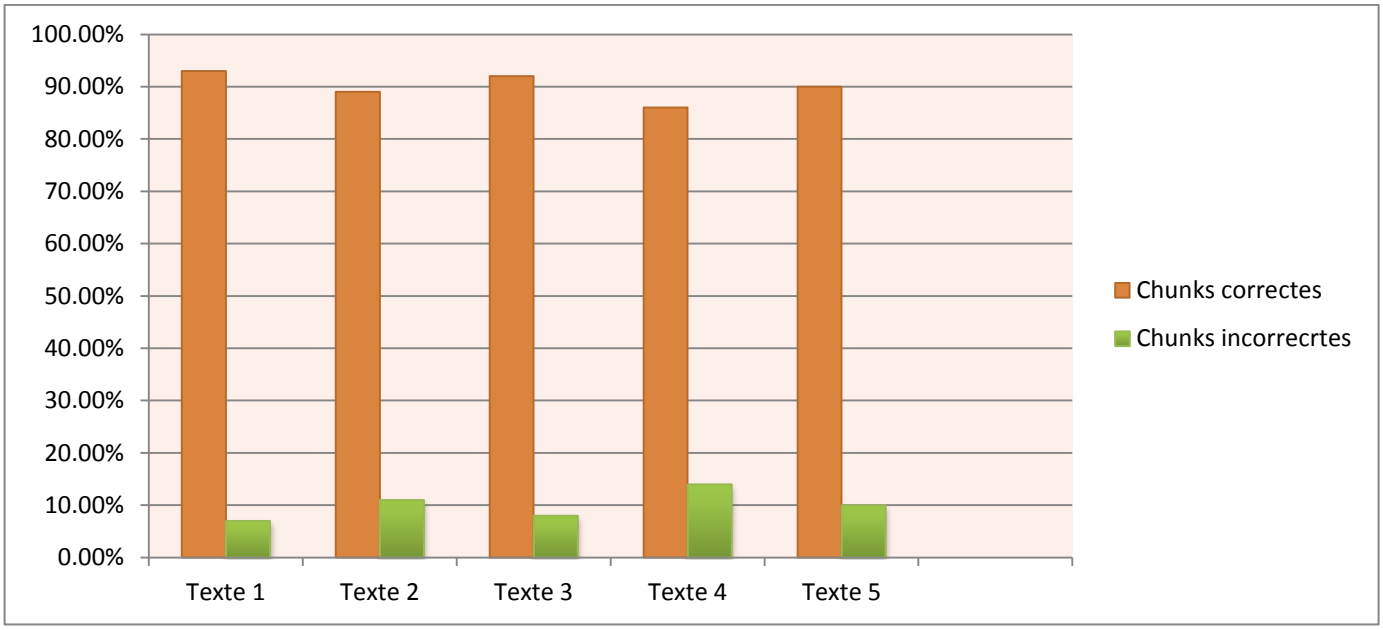


Figure 26 : Représentation graphique de test de deuxième catégorie de corpus Khaleej

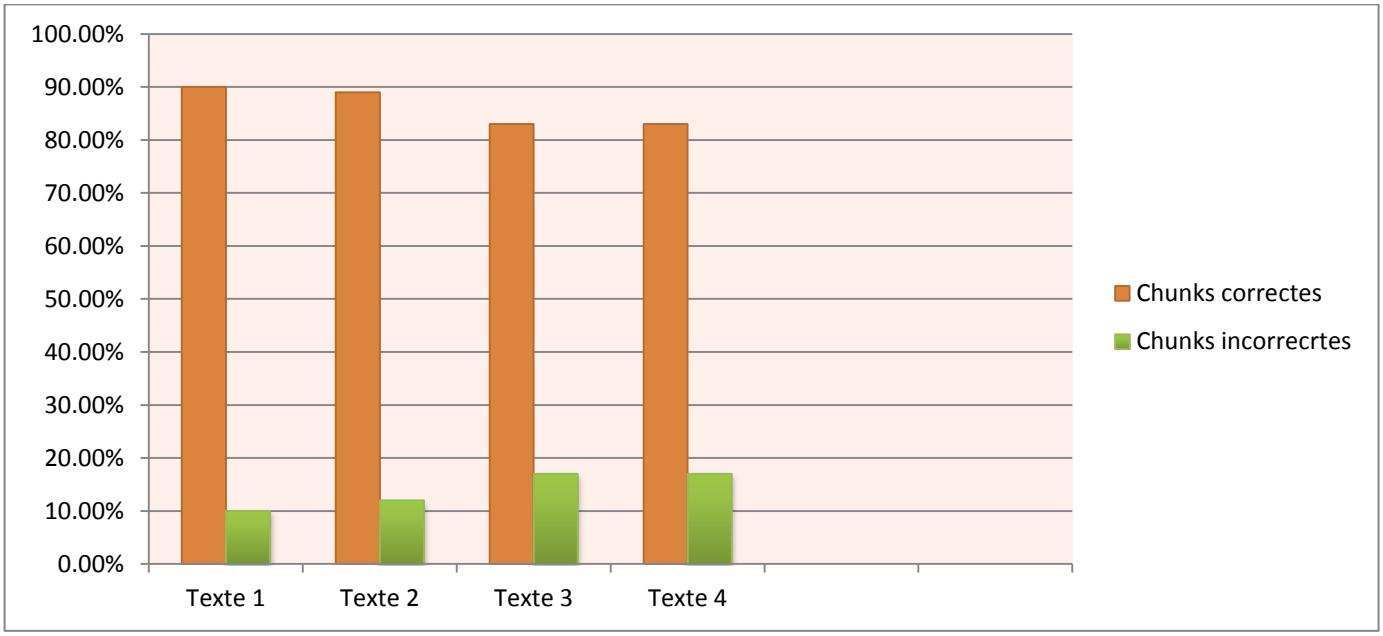


Figure 27 : Représentation graphique de test de troisième catégorie de corpus Khaleej

Pour la phase d'évaluation nous utilisons le taux de précision de la manière suivante:

$$\text{Pi} = \frac{\text{Nombre des tronçons corrects}}{\text{Nombre de tous les tronçons}} * 100 \text{ , } \text{P moy} = \frac{\sum_{k=i}^{n} Pi}{\text{n}}$$

Pi = Le taux des tronçons corrects

6. Discussion

La finalité de notre chunker est le groupement syntaxique des mots pour construire des tronçons, on se basant sur la méthodologie de segmentation qui respecte la grammaire de langue arabe. À partir de là, les performances de notre logiciel doivent être observées au niveau du chunking.

Après l'évaluation de la qualité de notre chunker **ArabChunk**, nous avons obtenu un taux de 91.40% de tronçons corrects, cela nous permet de dire que nous avons développé un outil qui peut offrir des bons résultats. Mais comme tous les outils de traitement automatique des langues naturelles, il y a toujours des problèmes entravant le logiciel à être parfait.

Dans notre cas, les résultats du chunking dépendent directement des résultats de l'étiquetage morphosyntaxique. Pour cela, il existe des situations ou une étiquette est erronée, ce qui affecte négativement sur les résultats de chunking. A titre d'exemple, dans le passage ci-dessous les crochets représentent les frontières théoriques et les parenthésage les frontières obtenues après l'analyse automatique.

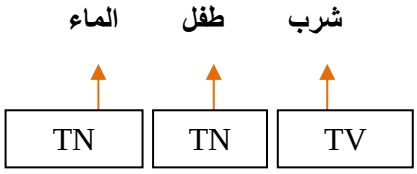
كما اعتقلت [قوات الاحتلال (الاسرائيلية)] عددا من الفلسطينيين في مناطق مختلفة من الضفة الغربية المحتلة منهم اثنان في نابلس وخمسة في الخليل و واحد في رام الله.

و أطلقت النار على برج المراقبة العسكري[قرب مستنوطنة رفيح (يام)] غرب رفح جنوب القطاع.

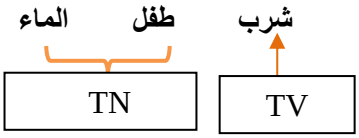
Volontairement, certaines erreurs d'étiquetage sont tolérées, si elles n'influencent pas sur la segmentation en tronçons, comme le montre l'exemple suivant: **ولم يبلغ عن وقوع اصابات [أو أضرار]**, la structure **[أو أضرار]** est considéré comme tronçon nominal, tandis qu'il est un tronçon conjonctif ,mais elle est correcte au niveau de la segmentation.

D'autre part, la reconnaissance des tronçons nominal pose un problème lorsque le sujet d'une phrase est indéterminé alors que l'objet et déterminé, le résultat est sous forme d'un couple de mot + leur déterminant.

L'exemple de« **شرب طفل الماء** » théoriquement, les tronçons en sortie de cette phrase doit être les suivants:



Mais pratiquement le résultat de notre application donne le découpage ci-dessous:



7. Conclusion

Dans ce chapitre, nous avons évalué notre chunker ArabChunk de texte arabe non vocalisé, ou nous avons exposé les résultats des tests qui sont fait sur les trois (03) catégories de corpus Khaleej, nous considérons que ces résultats sont satisfaisant avec un taux de 91.40% de tronçons corrects. Mais le 09% manquantes sont principalement due à l'étiquetage morphosyntaxique, c'est à dire les mots mal étiquetés qui conduisent à des faux résultats au niveau du chunking. Un autre problème au niveau de la segmentation syntaxique elle-même, surtout dans la construction des tronçons nominaux.

Conclusion : Bilan et Perspectives

Le traitement automatique de la langue arabe affronte de nombreuses difficultés venant principalement de sa richesse morphologique. Sa complexité n'a pas empêchée les chercheurs d'effectuer des travaux fructueux, et il y a toujours des nouvelles recherches pour rendre ce traitement plus robuste.

À travers notre étude, nous avons tenté de mettre l'accent sur un nouveau outil à savoir le chunking des textes arabes, basé sur une grammaire en tronçons. Pour cela, nous avons premièrement passé par un étiquetage morpho-syntaxique réalisé par l'outil AMIRA2.0, pour identifier la catégorie grammaticale des mots et leurs agencement dans la phrase. Deuxièmement, nous avons effectué un parenthésage syntaxique fondé sur des règles grammaticales de la langue arabe, afin de relie les unités ayant des liens syntaxiques forts dans un tronçon et lui donner une étiquette. Après des tests préliminaires sur 1612 mot du corpus Khaleej nous avons obtenus un résultat intéressant avoisinant les 91,40 % de tronçons corrects.

Afin d'améliorer la qualité de notre chunker « **ArabChunk** », nous pouvons signaler les perspectives suivantes :

- Améliorer l'étiquetage morpho-syntaxique, par l'utilisation d'un autre étiqueteur plus performent que AMIRA2.0.
- Spécifier les tronçons nominaux précédés par des particules selon la catégorie de particule comme les tronçons négatifs et interrogatifs.
- Réévaluer notre chunker sur d'autres corpus.

Les tags : ERTS_PERE

	Tag	Description (Arabe)
1	CC	حرف عطف
2	CC_PRP_D3	حرف عطف متصل بضمير مثنى الغائب
3	CC_PRP_FS3	حرف عطف متصل بضمير مفرد مؤنث الغائب
4	CC_PRP_MS3	حرف عطف متصل بضمير مفرد مذكر الغائب
5	CC_PRP_MP3	حرف عطف متصل بضمير جمع مذكر الغائب
6	CC_VBD_MD3	حرف عطف متصل بفعل ماض مع مثنى مذكر الغائب
7	CC_VBD_ MS3	حرف عطف متصل بفعل ماض مع مفرد مذكر الغائب
8	CC_PRP_P1	حرف عطف متصل بضمير جمع المتكلم
9	CJP	حرف نصب
10	DET	التعريف
11	DET_JJ	صفة معرفة مفرد مذكر
12	DET_JJCD	عدد ترتيبي معرف مفرد مذكر
13	DET_JJCD_FS	عدد ترتيبي معرف مفرد مؤنث
14	DET_JJ_FS	صفة معرفة مفرد مؤنث
15	DET_JJ_MP	صفة معرفة جمع مذكر
16	DET_JJ_PRP_MS3	صفة معرفة متصلة بضمير مفرد مذكر الغائب
17	DET_NN	اسم معرف
18	DET_NN_FS	اسم معرف مفرد المؤنث
19	DET_NNP	اسم علم معرف
20	DET_NN_PRP_MS3	اسم معرف متصل بضمير مفرد مذكر الغائب
21	DET_NNS_FP	جمع اسم مؤنث
22	DET_NNS_MD	مثنى اسم مذكر
23	DET_NNS_MP	جمع اسم مذكر
24	DET_NQ	حرف توكيد معرف
25	DET_NN_FS	مفرد اسم مؤنث
26	DT_F	اسم إشارة مؤنث
27	DT_MS	اسم إشارة مفرد المذكر
28	DT_MD	اسم إشارة مثنى المذكر
29	DT_MP	اسم إشارة جمع المذكر
30	DT_FP	اسم إشارة مثنى المذكر
31	IN	حرف جر
32	IN_PRP_FS3	حرف جر متصل بضمير مفرد مؤنث الغائب
33	IN_PRP_MS2	حرف جر متصل بضمير مفرد مذكر المخاطب
34	IN_PRP_MS3	حرف جر متصل بضمير مفرد مذكر الغائب
35	IN_PRP_MP2	حرف جر متصل بضمير جمع مذكر المخاطب
36	IN_PRP_MP3	حرف جر متصل بضمير جمع مذكر الغائب
37	IN_PRP_S1	حرف جر متصل بضمير مفرد المتكلم
38	IN_PRP_S3	حرف جر متصل بضمير مفرد الغائب
39	JJ	صفة
40	JJ_MP	جمع صفة مذكرة
41	JJCD	عدد ترتيبي مفرد مذكر
42	JJCD_FS	عدد ترتيبي مفرد مؤنث
43	JJCD_PRP_FS3	عدد ترتيبي متصل بضمير مفرد مؤنث الغائب
44	JJ_FS	مفرد صفة مؤنثة
45	JJ_PRP_MS2	عدد ترتيبي متصل بضمير مفرد مذكر المخاطب
46	NN	اسم نكرة
47	NNCD	رقم

48	NNCD_FS	مفرد رقم مؤنث
49	NNCD_MP	جمع رقم مذكر
50	NN_FS	مفرد اسم مؤنث
51	NN_FS_PRP_FS3	مفرد اسم مؤنث متصل بضمير مفرد مؤنث الغائب
52	NN_FS_PRP_MS2	مفرد اسم مؤنث متصل بضمير مفرد مذكر المخاطب
53	NN_FS_PRP_MS3	مفرد اسم مؤنث متصل بضمير مفرد مذكر الغائب
54	NN_FS_PRP_P1	مفرد اسم مؤنث متصل بضمير جمع المتكلم
55	NN_PRP_D3	مفرد اسم مذكر متصل بضمير مثنى الغائب
56	NN_PRP_FS3	مفرد اسم مذكر متصل بضمير مفرد مؤنث الغائب
57	NN_PRP_P1	مفرد اسم مذكر متصل بضمير جمع المتكلم
58	NN_PRP_S1	مفرد اسم مذكر متصل بضمير مفرد المتكلم
59	NN_PRP_MP2	مفرد اسم مذكر متصل بضمير جمع مذكر المخاطب
60	NN_PRP_MP3	مفرد اسم مذكر متصل بضمير جمع مذكر الغائب
61	NN_PRP_MS2	مفرد اسم مذكر متصل بضمير مفرد مذكر المخاطب
62	NN_PRP_MS3	مفرد اسم مذكر متصل بضمير مفرد مذكر الغائب
63	NNP	اسم علم
64	NNP_FS	اسم علم مفرد مؤنث
65	NNP_PRP_FS3	اسم علم متصل بضمير مفرد مؤنث الغائب
66	NNP_PRP_MP3	اسم علم متصل بضمير جمع مذكر الغائب
67	NNP_PRP_MS2	اسم علم متصل بضمير مفرد مذكر المخاطب
68	NNP_PRP_MS3	اسم علم متصل بضمير مفرد مذكر الغائب
69	NNP_PRP_P1	اسم علم متصل بضمير جمع المتكلم
70	NNS_FD	مثنى اسم مؤنث
71	NNS_FD_PRP_P1	مثنى اسم مؤنث متصل بضمير جمع المتكلم
72	NNS_FP	جمع اسم مؤنث
73	NNS_FP_PRP_MP3	جمع اسم مؤنث متصل بضمير جمع مذكر الغائب
74	NNS_FP_PRP_MS2	جمع اسم مؤنث متصل بضمير مفرد مذكر المخاطب
75	NNS_FP_PRP_MS3	جمع اسم مؤنث متصل بضمير مفرد مذكر الغائب
76	NNS_MD	مثنى اسم مذكر
77	NNS_MD_PRP_MP2	مثنى اسم مذكر متصل بضمير جمع مذكر المخاطب
78	NNS_MD_PRP_MS2	مثنى اسم مذكر متصل بضمير مفرد مذكر المخاطب
79	NNS_MD_PRP_MS3	مثنى اسم مذكر متصل بضمير مفرد مذكر الغائب
80	NNS_MP	جمع اسم مذكر
81	NNS_MP_PRP_MP3	جمع اسم مذكر متصل بضمير جمع مذكر الغائب
82	NQ	حرف توكيد
83	NQ_PRP_FS3	حرف توكيد متصل بضمير مفرد مؤنث الغائب
84	NQ_PRP_MS3	حرف توكيد متصل بضمير مفرد مذكر الغائب
85	PRP	ضمير
86	PRP_D3	ضمير مثنى الغائب
87	PRP_FP3	ضمير جمع مؤنث الغائب
88	PRP_FS3	ضمير مفرد مؤنث الغائب
89	PRP_MS2	ضمير مفرد مذكر المخاطب
90	PRP_MS3	ضمير مفرد مذكر الغائب
91	PRP_ MP3	ضمير مفرد مذكر الغائب
92	PRP_P1	ضمير جمع المتكلم
93	PRP_S1	ضمير مفرد المتكلم
94	PUNC	التنقيط

95	RB	حال
96	RP	حرف جزم
97	RP_PRP_FS3	حرف جزم متصل بضمير مفرد مؤنث الغائب
98	RP_PRP_ MP3	حرف جزم متصل بضمير جمع مذكر الغائب
99	UH	حرف جواب
100	VB_MS2	فعل امر
101	VBD_FS3	فعل ماض مع مفرد مؤنث الغائب
102	VBD_FS3_PRP_MS3	فعل ماض مع مفرد مؤنث الغائب متصل بضمير مفرد مذكر الغائب
103	VBD_FS3_PRP_MP3	فعل ماض مع مفرد مؤنث الغائب متصل بضمير جمع مذكر الغائب
104	VBD_MS3_PRP_MP3	فعل ماض مع مفرد مذكر الغائب متصل بضمير جمع مذكر الغائب
105	VBD_FS3_PRP_P1	فعل ماض مع مفرد مؤنث الغائب متصل بضمير جمع المتكلم
106	VBD_MD3	فعل ماض مع مثنى مذكر الغائب
107	VBD_MP3	فعل ماض مع جمع مذكر الغائب
108	VBD_MS3	فعل ماض مع مفرد مذكر الغائب
109	VBD_MS3_PRP_MS3	فعل ماض مع مفرد مذكر الغائب متصل بضمير مفرد مذكر الغائب
110	VBD_MS3_PRP_MP2	فعل ماض مع مفرد مذكر الغائب متصل بضمير جمع مذكر المخاطب
111	VBD_P1	فعل ماض مع جمع المتكلم
112	VBD_S1	فعل ماض مع مفرد المتكلم
113	VBD_S1_PRP_MP3	فعل ماض مع مفرد المتكلم متصل بضمير جمع مذكر الغائب
114	VBN_FS3	فعل مبني للمجهول مع مفرد مؤنث الغائب
115	VBN_MD3	فعل مبني للمجهول مع مثنى مذكر الغائب
116	VBN_MS3	فعل مبني للمجهول مع مفرد مذكر الغائب
117	VBP	فعل مضارع
118	VBP_FS3	فعل مضارع مع مفرد مؤنث الغائب
119	VBP_FS3_PRP_MS3	فعل مضارع مع مفرد مؤنث الغائب متصل بضمير مفرد مذكر الغائب
120	VBP_MP2	فعل مضارع مع جمع مذكر المخاطب
121	VBP_MP3	فعل مضارع مع جمع مذكر الغائب
122	VBP_MP3_PRP_MS3	فعل مضارع مع جمع مذكر الغائب متصل بضمير مفرد مذكر الغائب
123	VBP_MS3	فعل مضارع مع مفرد مذكر الغائب
124	VBP_MS3_PRP_FS3	فعل مضارع مع مفرد مذكر الغائب متصل بضمير مفرد مؤنث الغائب
125	VBP_MS3_PRP_MP3	فعل مضارع مع مفرد مذكر الغائب متصل بضمير جمع مذكر الغائب
126	VBP_MS3_PRP_MS2	فعل مضارع مع مفرد مذكر الغائب متصل بضمير مفرد مذكر المخاطب
127	VBP_MS3_PRP_MS3	فعل مضارع مع مفرد مذكر الغائب متصل بضمير مفرد مذكر الغائب
128	VBP_S1	فعل مضارع مع مفرد المتكلم
129	VBP_P1	فعل مضارع مع جمع المتكلم
130	VBP_P1_PRP_FS3	فعل مضارع مع جمع المتكلم متصل بضمير مفرد مؤنث الغائب
131	VBP_PRP_FS3	حرف مشبه بالفعل متصل بضمير مفرد مؤنث الغائب
132	VBP_PRP_MP3	حرف مشبه بالفعل متصل بضمير مفرد مؤنث الغائب
133	VBP_PRP_MS2	حرف مشبه بالفعل متصل بضمير مفرد مذكر المخاطب
134	VBP_PRP_MS3	حرف مشبه بالفعل متصل بضمير مفرد مذكر الغائب
135	VBP_PRP_S1	حرف مشبه بالفعل متصل بضمير مفرد المتكلم
136	VBP_S1_PRP_FS3	فعل مضارع مع مفرد المتكلم متصل بضمير مفرد مؤنث الغائب
137	VBP_S1_PRP_MS3	فعل مضارع مع مفرد المتكلم متصل بضمير مفرد مذكر الغائب
138	VN	اسم فاعل
139	VN_FS	اسم فاعل مفرد مؤنث
140	VN_MP	اسم فاعل جمع مذكر

141	WP	اسم موصول
142	WRB	حرف الشرط

Références

[ABN, 91] Abney.S, « Principle-based parsing: computation and psycholinguistics», 1st Edn, Springer, Dordrecht, ISBN: 0792311736, pp: 408, 1991.

[ABB, 04] ABBES. R, « La conception et la réalisation de concordancier électronique pour l'arabe », Thèse de Doctorat, Institut national des sciences appliquées de Lyon, Lyon, France, 2004.

[ABB, 08] Abbès Ramzi et Dichy Joseph, « Extraction automatique de fréquences lexicales en arabe et analyse d'un corpus journalistique avec le logiciel AraConc et la base de connaissances DIINAR.1 », in Heiden, Serge et Bénédicte Pincemain, Actes des JADT 2008, 9es journées internationales d'analyse statistique des données textuelles, Lyon 12-14-mars, Presses Universitaires de Lyon, 2 vol. , pp. 31-44, 2008.

[ALJ, 01] Aljlayl.M et Frieder.O, «On Arabic Search: Improving the Retrieval Effectiveness via a Light Stemmng Approach», In 11th International Conference on Information and Knowledge Management (CIKM), 340-347, 2002.

[ATT, 06] Attia. M, «An Ambiguity-Controlled Morphological Analyzer for Modern Standard Arabic Modelling Finite State Networks», Actes de la conférence internationale, The Challenge of Arabic for NLP/MT", The British Computer Society, London, 2006.

[BES, 01] BESANÇON. R, « Intégration de connaissances syntaxiques et sémantiques dans les représentations vectorielles des textes, Application au calcul de similarité sémantique dans le cadre du modèle DSIR ». Thèse de Doctorat, Ecole polytechnique fédérale de lausanne, Suisse, 2001.

[BOU, 97] BOULA. P DE MAREUIL, « Etude linguistique appliquée à la synthèse de la parole à partir du texte », thèse de doctorat, Université de Paris XI, Orsay, 1997.

[BAL, XX] BALOUL Sofiane & Philippe BOULA de MAREÜIL, « Un modèle syntactico-prosodique pour la synthèse de la parole à partir du texte en arabe standard voyellée ».

[CHE.15] CHERAGUI Mohamed Amin, « Introduction au traitement automatique du langage naturel », Module : Traitement automatique de la parole, support de cours 2015.

[COH, 61] Cohen David, « Essai d'une analyse automatique de l'arabe », In ATALA (revue de l'Association pour le Traitement Automatique des LAngues naturelles), Paris, 1961.

[COH, 70] Cohen David, « Etudes de linguistique sémitique et arabe ». Mouton, La Hague – Paris, 1970.

[DIA, 04] M. DIAB, K. HACIOGLU et D. JURAFSKY, « Automatic tagging of arabic text: From raw text to base phrase chunks». In Proceedings of NAACL-HLT, pages 149–152, Boston, USA, 2004.

[DIC, 89] Dichy Joseph, Hassoun Mohamed, « Simulation de modèles linguistiques et Enseignement Assisté par Ordinateur de l'arabe », travaux SAMIA I, Fondation postuniversitaire interculturelle (Conseil International à la Langue Française), Paris, 1989.

[ELI, 03] Elisabeth Delais-Roussarie et Jaques Durand, « Corpus et variation en phonologie du français », presse universitaire de Mrail, 2003.

Références

[FER, 12] Fériel Ben Fraj et Maroua Kessentini, **«** COMBINATORIAL CLASSIFICATION FOR CHUNKING ARABIC TEXTS **»,** RIADI Laboratory, Manouba University, Manouba, Tunisia, International Journal of Artificial Intelligence & Applications (IJAIA), Vol.3, No.5, September 2012.

[GHE, 98] M. GHENIMA, « Système de voyellation de textes arabes ». Thèse de Doctorat, Université Lyon2, Lyon, France, 1998.

[GIG ,98] Giguet. E, « Méthode pour l'analyse automatique de structures formelles sur documents multilingues », thèse de doctorat de l'Université de Caen, 1998.

[GAL, 02] Gal. Y, «An HMM Approach to Vowel Restoration in Arabic and Hebrew », Actes du workshop on Computational Approaches to Semitic Languages au meeting annuel de l'Association for Computational Linguistics (ACL-02), Philadelphie, USA, 2002.

[HAS, 87] M. HASSOUN, « Conception d'un dictionnaire pour le traitement automatique de l'arabe dans différents contextes d'application ». Thèse de Doctorat, Université Lyon1, Lyon, France, 1987.

[KHO, 01] S. KHOJA et G. GARSUDE, R.AND KNOWLES, «A tagset for the morphosyntactic tagging of Arabic». In Corpus Linguistics 2001 conference, pages 1–13, Lancaster, UK, 2001.

[LAR, 02] L.S. LARKEY, L. BALLESTEROS et M.E. CONNELL, «Improving stemming for Arabic information retrieval: light stemming and co-occurrence analysis», In Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval, pages 275–282, Tampere, Finland, 2002.

[MAA, 04] MAAMOURI, M, BIES, A. et BUCKWALTER, T. « The Pen Arabic Treebank: Building a Large-Scale Annotated Arabic Corpus », In EMAR Conference on Arabic Language Ressources and Tools, le Caire, 2004.

[MAA, 06] M. Maamouri, « Diacritization: A Challenge to Arabic Treebank Annotation and Parsing», Actes de la conférence international, The Challenge of Arabic for NLP/MT, The British Computer Society, London, 2006.

[MES, 03] Slim MESFAR, « VOWEL : outil interactif pour la voyellation de textes arabes », Mémoire de projet de fin d'études d'ingénieurs, Manouba, Tunisie, 2003.

[MES, 08] Slim MESFAR, « analyse morpho-syntaxique automatique et reconnaissance des entités nommées en arabe standard », Thèse de Doctorat, le 24 novembre2008.

[NEL, 05] R. Nelken et S. M. Shieber, «Arabic diacritization using weighed finite- state transducers », Actes du Workshop on Computational Approaches to Semitic Languages. Association for Computational Linguistics ACL, Michigan, USA, 2005.

[NED , XX] Nadjmeddine Khalfallah, « fiche grammatical », département d'Arabe-Université de lorraine.

Références

[PHI, 01]Philippe Boula de Mareüil, Christophe d'Alessandro, Frédéric Beaugendre, Anne Lacheret-Dujour, « Une grammaire en tronçons appliquée à la génération de la prosodie », LIMSI-CNRS Bâtiment 508 - Université Paris XI, BP 133 F91403 Orsay, 2001.

[RAM, 95] Ramshaw, L.A. and M.P. Marcus, «Text chunking using transformation-based learning», Proceeding of the 3rd ACL Workshop on Very Large Corpora, May 23-23, Computation and Language, Lance Ramshaw pp: 82-94, 1995.

[SAB, 89] SABAH. G, « L'intelligence artificielle et le langage : processus de compréhension », vol. 2, Hermès, Paris, 1989.

[SAF, 06] H. Safadi, O. Dakkak et N. Ghneim, « Computational Methods to Vocalize Arabic Texts », Actes du 2ème Workshop on Internationalizing SSML, Crète, Grèce, 2006.

[SLI, 08] Slim MESFAR, « analyse morpho-syntaxique automatique et reconnaissance des entités nommées en arabe standard », Le 24 novembre 2008.

[YAM, 05] Yamina TLILI-GUIASSA, **«** Memory-based-Learning et Base de règles pour un Etiqueteur du Texte Arabe **»,** Laboratoire de Recherche en Informatique, Université Badji M okhtar, Annaba Algerie, RÉCITAL 2005, Dourdon, 6-10 Juin 2005.

[ZAA, 01] R. ZAAFRANI, « Développement d'un environnement interactif d'apprentissage avec l"ordinateur de l'arabe langue étrangère ». Thèse de Doctorat, Université Lyon2, Lyon, France, 2001.

[ZEM, 04] Zemirli et Khabet, « TAGGAR : Un analyseur morphosyntaxique destiné à la synthèse vocale de textes arabes voyellés », 2004.

[ZOU, 02] Zoubeir Mouelhi, « AraSeg un segmenteur semi-automatiquedes textes arabes », ICAR - Université Lumière-Lyon2, AraSeg est une version que nous avons améliorée, d'un segmenteur développé en collaboration avec R. Zaafrani, voir (Zaafrani, 2002).