Universite Sidi Mohamed Ben Abdellah
Faculté Des Sciences Et Techniques Fès
Département d'Informatique

Projet de Fin d'Etudes

Master Sciences et Techniques
Systèmes Intelligents & Réseaux

Extraction lexicale et catégorisation morphologique et sémantique du dictionnaire de la langue arabe : « Alghani Azzahir »

Lieu de stage : Laboratoire Systèmes Intelligents et Applications
Istituto di Linguistica Computazionale de Pise

Réalisé par : KALLAL OMAR
KHALFI MUSTAPHA

Soutenu le 18 Juin 2014

Encadré par :

Mr. Arsalane Zarghili
Mme. Ilham Chaker
Mme. Ouafae Nahli

Devant le jury composé de :

Pr. Arsalane Zarghili
Pr. Ilham Chaker
Pr. Ahlame Begdouri
Pr. Med Chaouiki Abounaima
Pr. Ouafae Nahli

Année Universitaire 2013-2014

A nos chers parents :

Pour l'amour sans failles qu'ils ont pour nous

Pour l'éducation exemplaire qu'ils nous ont inculquée

Pour tout ce qu'ils ont fait pour nous

Et continuent toujours de faire ….

A nos chers sœurs et frères

Qui nous ont entourés de leur affection

Et nous ont scellés par leurs bénédictions

A ceux qui nous sont les plus chers

A tous nos amis

Nous dédions ce travail

À travers lequel on leur dit merci

Merci pour leur soutien et leur encouragement

Que dieu les garde

Omar & Mustapha.

Remerciements

Nous remercions, tout d'abord, DIEU qui garde toujours un œil bien veillant sur nous.

Nous exprimons notre grande gratitude auprès de nos encadrants à la FST ***M. ZARGHILI Arsalane*** *et* ***Mme. CHAKER Ilham****, pour leur encadrement, leur disponibilité et leurs conseils fructueux qu'ils nous ont prodigués le long de notre projet.*

Nous adressons nos remerciements à ***Mme. NAHLI Ouafae*** *notre encadrante de l'ILC de Pise en Italie, qui n'a jamais cessé de nous guider et de nous faire bénéficier de son grand savoir.*

Nous remercions aussi le chercheur ***Mr. ABOU EL AAZM Abdelghani*** *l'auteur du dictionnaire « Alghani Azzahir »,* pour sa confiance en nous, et pour toute sa serviabilité.

Nous remercions également tout le staff de la Faculté des Sciences et Techniques, plus particulièrement le Doyen et le corps enseignant pour leurs connaissances qu'ils ont bien voulu nous les partager.

Nos vifs remerciements s'adressent aussi à tous les membres de jury qui ont accepté de juger notre travail.

Un grand merci pour nos chères familles et aussi nos chers amis au sein de la FSTF et à toute personne ayant contribué au bon déroulement de ce stage de fin d'étude.

Résumé

Le présent rapport constitue le résultat d'un travail réalisé dans le cadre du projet de fin d'études, au sein du laboratoire Systèmes Intelligents et Applications à la FST-Fès en collaboration avec ILC de pise en Italie.

Notre stage qui a duré une période de 4 mois, est une initiation à la recherche, et qui a pour objectif la réalisation d'un système d'extraction automatique de données morphosyntaxiques et sémantiques pour la langue arabe à partir d'un dictionnaire numérisé.

En ce qui concerne le développement de l'application on a utilisé le langage Java pour faire notre traitement, et XML pour la représentation des données. Ensuite pour l'exploitation de ces données nous avons développé une application de recherche et de classification de ces données.

A travers ce document, nous allons décrire en détails le déroulement du travail du projet allant de la conception jusqu'à la réalisation de ce projet.

Mots clés : Analyseur Morphologique, Dictionnaire Electronique, Langue Arabe, TALN, TALA, LMF, Caractéristique morphosyntaxique, caractéristique sémantique, Extraction des paradigmes.

Abstract

The following report is the fruit of a hard work which lead to achieve our graduation project within a Laboratory of Intelligent Systems and Applications of the Faculty of science and technology of the University Sidi Mohamed Ben Abdellah-FEZ and in partnership with the Institute of computational linguistics of Pise in Italy.

Our project can be seen as a gateway to research on language processing of natural languages, in particular Arabic language, in the aim to develop a system for extraction of morph-syntactic and semantic characteristics using a digital dictionary.

Regarding the development of the application, we used the Java language to make our treatment, and XML for data representation. According to exploit these data we have developed a retrieval system including classification tools.

Through this document, we will describe with more details each part of this project.

Keywords : Arabic Language, Electronic Dictionary, Morphological Analyzer, Natural Language Processing (NLP), Arabic Language Processing, LMF, Feature morphosyntactic, semantic feature, extraction paradigms.

ملخص

هذا المشروع هو ثمرة عمل جاد على امتداد أربعة أشهر، داخل مختبر" الأنظمة الذكية والتطبيقات " التابع لكلية العلوم والتقنيات لجامعة سيدي محمد بن عبد الله بفاس وبشراكة مع معهد " اللسانيات الحاسوبية ببيزا بإيطاليا" ويمكن اعتباره مدخلا للبحث في موضوع المعالجة اللغوية للغات الطبيعية وخاصة منها اللغة العربية بهدف طرح نظام معلوماتي يمكننا من استخراج مميزاتها الصرفية والنحوية والدلالية باستعمال القاموس الرقمي.

لقد قمنا بتطوير هذا البرنامج باستخدام لغة البرمجة JAVA كما استخدمنا لغة التمثيل XML لتنظيم المداخل اللغوية. ولاستغلال هاته القاعدة المهمة أنجزنا تطبيقا على شكل قاموس رقمي للبحث عن طريق المداخل او أنواع الكلمات وتصنيفاتها.

من خلال هذا التقرير سنسلط الضوء على مختلف هاته المراحل.

Liste des abréviations

Abréviation	Désignation
API	Application Programming Interface
BAMA	Buckwalter Arabic Morphological Analyzer
CNR	Centre National de la Recherche
CNUCE	Centro Universitario di Nazionale Calcolo Elettronico
EDI	Environnement de développement integer
FST	Faculté des Sciences et Techniques
IHM	Interface Homme-Machine
ILC	Institut de linguistique computationnelle
ISO	International Organisation for Standardization
JDOM	Java Document Object Model
LMF	Lexical Markup Framework
LSIA	Laboratoire Systèmes Intelligents et Applications
MADA	Morphological Analysis and Disambiguation for Arabic
POS	Part Of Speech
RCD	Registre de catégorie de données
SAX	Simple API for XML
SGML	Standard Generalized Markup Language
TALN	Traitement Automatique des Langues Naturelles
URSS	Union des républiques socialistes soviétiques
XML	eXtensible Markup Language

Liste des figures

Liste des tableaux

Sommaire

Introduction générale

Le Traitement Automatique du Langage Naturel (TALN) regroupe à la fois la linguistique, l'informatique et l'intelligence artificielle. Cette discipline est apparue au début des années cinquante (Léon, 2002) aux États-Unis et est devenue un axe de recherche essentiel pour analyser et traduire la grande masse d'information qui évolue sans cesse. Cependant les enjeux cognitifs du traitement automatique des langues sont importants et varient selon les applications. De nos jours, il existe plusieurs applications de traitements des langues telles que la reconnaissance de l'écriture manuscrite, le résumé automatique, le traitement de la parole ou l'annotation sémantique, etc.

Les recherches dans le domaine de la langue arabe restent très modestes par rapport à l'anglais et aux langues Indo-Européens, suite à la complexité de la langue arabe. Ainsi qu'à la rareté des corpus et du contenu numérique de cette langue, elle a constitué 1% seulement du contenu numérique sur internet en 2009 (CLN, 2009). De plus, la recherche d'information s'est principalement développée à partir de l'anglais qui est aujourd'hui la langue dominante sur le web. Ce n'est que récemment qu'ont débuté les travaux sur d'autres langues très répandues comme l'arabe, mais pour lesquelles on manque aujourd'hui à la fois d'outils d'analyse et également de ressources linguistiques.

Dans l'optique d'enrichir le contenu numérique de la langue arabe, nous avons comme objectif de proposer et de représenter les entrées lexicales du dictionnaire arabe ***ALGHANI AZZAHIR*** du chercheur marocain ***Abdelghani Abou El Aazm*** sous un format intermédiaire XML dans la perspective de le représenter sous le format standard LMF. Par conséquent ceci nous permettra d'exploiter les différentes caractéristiques morphologiques, syntaxiques et sémantiques contenus dans ce dictionnaire, ainsi pour la contribution à la construction d'une base lexicale arabe complète.

Le présent rapport retrace le déroulement des étapes élaborées pendant la réalisation du projet, ces étapes sont axées autour de cinq points principaux :

Le premier chapitre décrit le contexte général du projet, et ce en présentant les Laboratoires d'accueil, puis en exposant la problématique, la définition de ce sujet ainsi que la planification du projet.

Le deuxième et troisième chapitre présentent un état de l'art sur la langue arabe, et quelques notions sur l'analyse morphosyntaxique de cette langue, puis une définition des modèles LMF.

Dans le quatrième chapitre on décrit les différentes étapes d'extraction et d'organisation des données morphosyntaxiques après la définition des ressources linguistiques utilisées au cours de ce stage.

Le dernier chapitre définit les outils que nous avons estimé les plus adéquats à utiliser. Ainsi une description de la modélisation, réalisation et des principales interfaces du système.

Enfin les annexes seront présentées comme compléments servant à expliquer de façon plus détaillée les différentes notions mises en jeu.

Chapitre 1 :

Contexte GENERAL DU PROJET

Ce premier chapitre présentera le contexte général du projet, il constitue un point de départ qui permettra de bien gérer le déroulement de toutes les phases d'étude qu'on entreprend par la suite.

Introduction

Dans ce chapitre nous présentons notre projet, l'objectif, et notre contribution, ensuite nous exposons les deux laboratoires d'accueil, et nous finissons par la présentation de la planification de notre projet.

1. Présentation du projet

Dans le domaine de l'ingénierie linguistique et de la connaissance, le problème des ressources lexicales et linguistiques s'est toujours posé. Néanmoins, l'avancée des techniques du Traitement Automatique des Langues Naturelles (TALN) l'a rendu plus sensible. Il nous faut maintenant pouvoir répondre à des besoins importants en termes de quantité, de qualité et de complexité. La complexité et la diversité des informations requises augmentent avec les exigences des outils de TALN ainsi qu'avec le développement de nouvelles applications (humaines ou machinales). Si la récupération semi-automatique d'information lexicale est une piste, elle ne pourra remplacer la création manuelle de dictionnaires.

Nous nous sommes donc intéressés à la construction d'outils pour les lexicographes et les lexicologues. Afin d'avoir une bonne compréhension des problèmes qui se posent, nous avons décidé d'informatiser un dictionnaire, contenant de nombreuses informations structurées, le dictionnaire **Alghani Azzahir** de la langue arabe. Le Dictionnaire étant un travail de lexicologie, il s'agit donc d'extraire l'ensemble d'entrées lexicales du dictionnaire en faisant l'étiquetage de chaque entrée.

L'objectif de ce travail est d'extraire les caractéristiques morphosyntaxiques et sémantiques, et de les présentées dans un format XML intermédiaire, regroupant le maximum d'informations sur une entrée lexicale, afin de migrer par la suite vers une représentation standard LMF (Lexical Markup Framework). Cette base lexicale générée peut être utilisée comme référence pour les analyseurs morphologiques de la langue arabes afin de confirmer leurs résultats.

Cette action a été menée en collaboration entre le Laboratoire Systèmes Intelligents et Applications (LSIA) de l'Université Sidi Mohammed Ben Abdellah - Facultés des Sciences et Techniques-Fès au Maroc et l'Institut de linguistique computationnelle (Istituto di Linguistica Computazionale - ILC) de Pise en Italie.

Nous présentons les outils et méthodes que nous avons adoptées pour la reconstruction du dictionnaire. Nous montrerons ensuite en détail l'outil développé, permettant l'extraction des ressources lexicale.

Nous présentons par la suite la stratégie que nous avons adopté pour l'informatisation du dictionnaire.

2. Présentation des Laboratoires d'accueil

2.1 Laboratoire Systèmes Intelligents & Applications : LSIA

Le laboratoire SIA, crée en 2011, est une unité de recherche du Centre d'Etudes Doctorales en Sciences et Techniques de l'Ingénieur domicilié à la Faculté des Sciences et Techniques de Fès et regroupant 19 laboratoires de recherche tous accrédités par l'Université Sidi Mohamed Ben Abdellah de Fès, et domiciliés à la Facultés des Sciences et Techniques, l'Ecole Supérieure de Technologie, la Faculté Polydisciplinaire de Taza et l'Ecole Normale Supérieure de Fès.

Le LSIA est composé de 15 enseignants-chercheurs du département d'Informatique de la FST de Fès et de 17 doctorants. Cette imbrication étroite entre enseignement et recherche, est un élément essentiel de la dynamique du laboratoire.

Les thématiques de recherche se situent au cœur des Sciences et Technologies de l'Information et de la Communication et s'articulent essentiellement autour des thématiques de recherche des enseignants chercheurs du laboratoire et assure une large couverture thématique présentant un atout très important pour le LSIA.

Le laboratoire est composé de 3 équipes de recherche :

- ✓ Systèmes de Communication et Traitement de Connaissances (SCTC)
  - ➢ **Thématiques de recherche :**
    - Traitement automatique de la parole.
    - Traitement des langues naturelles.
    - Intelligence Artificielle.
    - Reconnaissance de formes.
- ✓ enVironnement Intelligents & Applications (VIA)
  - ➢ **Thématiques de recherche :**
    - Adaptation au contexte dans un environnement ambiant
    - M-learning / Social learning.
    - Communautés de pratique.
    - Réseaux adhoc: performances et sécurité.
- ✓ Vision Artificielle & Systèmes Embarqués (VASE)

- **Thématiques de recherche :**
  - Traitement automatique de la langue Arabe.
  - Traitement et Analyse d'images.
  - Reconnaissance de formes.
  - Intelligence Artificielle.
  - Systèmes Embarqués et Théorie de codes.

2.2 L'Istituto di Linguistica Computazionale : ILC

L'Istituto di Linguistica Computazionale "Antonio Zampolli" (ILC-CNR) - travaillant dans le domaine de la linguistique computationnelle depuis 1967, quand une division de la linguistique computationnelle a été formée au Centro Universitario di Nazionale Calcolo Elettronico (CNUCE) - a été fondée comme une société indépendante Institut de la CNR en 1978.

ILC-CNR a été l'un des principaux promoteurs de la notion de ressources linguistiques comme l'élément central de l'infrastructure linguistique et a coordonné les initiatives majeures en matière de ressources linguistiques et la normalisation et a souvent été le promoteur de nouveaux «paradigmes» dans le domaine.

ILC-CNR a conçu et construit plusieurs types de corpus et de lexiques et les ontologies respectives, a développé une chaîne complète d'outils pour un traitement robuste de la langue italienne, pour l'acquisition d'informations à partir de corpus et de mot-sens homonymie et a développé les technologies pour plusieurs domaines d'application :

- ✓ Question-réponse,
- ✓ Recherche d'information, de text-mining,
- ✓ Extraction de la terminologie unilingue et multilingue,
- ✓ L'acquisition de l'ontologie et de la structuration,
- ✓ La récapitulation, le filtrage de documents sur le Web,
- ✓ La préservation du patrimoine culturel par traitement d'image numérique et les techniques de bibliothèques numériques, etc.

ILC-CNR fonctionne grâce à un personnel structuré :

- ✓ 22 unités de personnel avec des contrats à durée indéterminée.
- ✓ 7 unités de personnel avec des contrats à durée déterminée.

Et grâce à un personnel non structurées :

- ✓ Environ 20 unités de personnel entre les jeunes chercheurs, doctorants, etc.

Et finalement un autofinancement substantiel.

Les principales tâches de l'ILC-CNR sont :

- ✓ La promotion de la recherche fondamentale pour le progrès de la connaissance dans le secteur du traitement du langage naturel, sur des sujets dont l'analyse de l'état de l'art suggère la nécessité et la possibilité d'innovations importantes, favoriser la symbiose entre les différentes compétences disciplinaires concernées.
- ✓ L'étude des méthodes et outils innovants et le développement des technologies et des ressources linguistiques de base qui peuvent être utilisées et intégrées dans les différents types de services et dans les systèmes orientés vers les applications en vue de promouvoir le développement de l'industrie italienne du secteur, en particulier la réduction des coûts des activités de développement des «Start –up ».
- ✓ L'étude et le développement des méthodes et des modèles multimodales, à travers l'intégration des technologies linguistiques avec le traitement d'image et le traitement de la parole.
- ✓ L'étude et la réalisation des prototypes et des systèmes innovants pour l'utilisation des technologies de la langue à l'appui de recherches et d'applications dans le domaine des disciplines humanistes, de l'accès au patrimoine culturel et de la promotion de la langue italienne.
- ✓ La stimulation d'une relation constante avec l'industrie et le transfert de technologies vers l'industrie.
- ✓ La promotion et la participation aux activités et programmes de la Communauté européenne et, de manière générale, des organismes internationaux qui impliquent l'utilisation des technologies de la langue.
- ✓ Garantir la représentativité dans les principaux lieux scientifiques et professionnels internationales.
- ✓ Garantir une éducation interdisciplinaire appropriée dans la recherche et le développement technologique pour les jeunes chercheurs, au moyen de doctorat, des subventions et des contrôles.
- ✓ L'organisation des conférences, des ateliers internationaux et des réunions nationales sur des sujets stratégiques dans le secteur de la linguistique computationnelle afin de favoriser le transfert des connaissances scientifiques et la création de synergies entre les différentes communautés actives dans le secteur.

3. Planification

A la base de la durée du stage, nous avons établi un planning de travail, afin de bien maîtriser les ressources allouées au projet. Nous avons alors découpé le projet en tâches afin de planifier leurs exécutions et le temps alloué à chacune.

Nom	Date de début	Date de fin
Documentation	05/02/14	28/02/14
Etat de l'Art de la langue arabe	05/02/14	14/02/14
Annalyseur morphologique	17/02/14	28/02/14
Analyse et Conception du module d'extraction	03/03/14	07/03/14
Extraction et Organisation des Données	10/03/14	02/05/14
Organisation du texte libre	10/03/14	13/03/14
Organisation des données lexicales	14/03/14	21/03/14
Organisation des Glosses	24/03/14	18/04/14
Organisation et codification des symboles	21/04/14	02/05/14
Réunion & workshop	05/05/14	08/05/14
Réunion avec membre ILC de pise	05/05/14	05/05/14
workshop	08/05/14	08/05/14
Réalisation du module de recherche	12/05/14	06/06/14
Analyse et Conception du modue de recherche	12/05/14	16/05/14
Developpement du module de recherche	19/05/14	06/06/14
Préparation rapport de stage et présentation	15/04/14	18/06/14
Rapport de stage V1	15/04/14	06/06/14
Rapport de stage V2	09/06/14	13/06/14
Préparation de la soutenance	16/06/14	18/06/14

Figure 1: Planning du projet

Afin de mieux comprendre le déroulement de notre projet, nous présentons dans la figure suivante le diagramme de Gantt de la planification élaborée, qui montre les péripéties du projet dans l'ordre chronologique ainsi que les tâches réalisées.

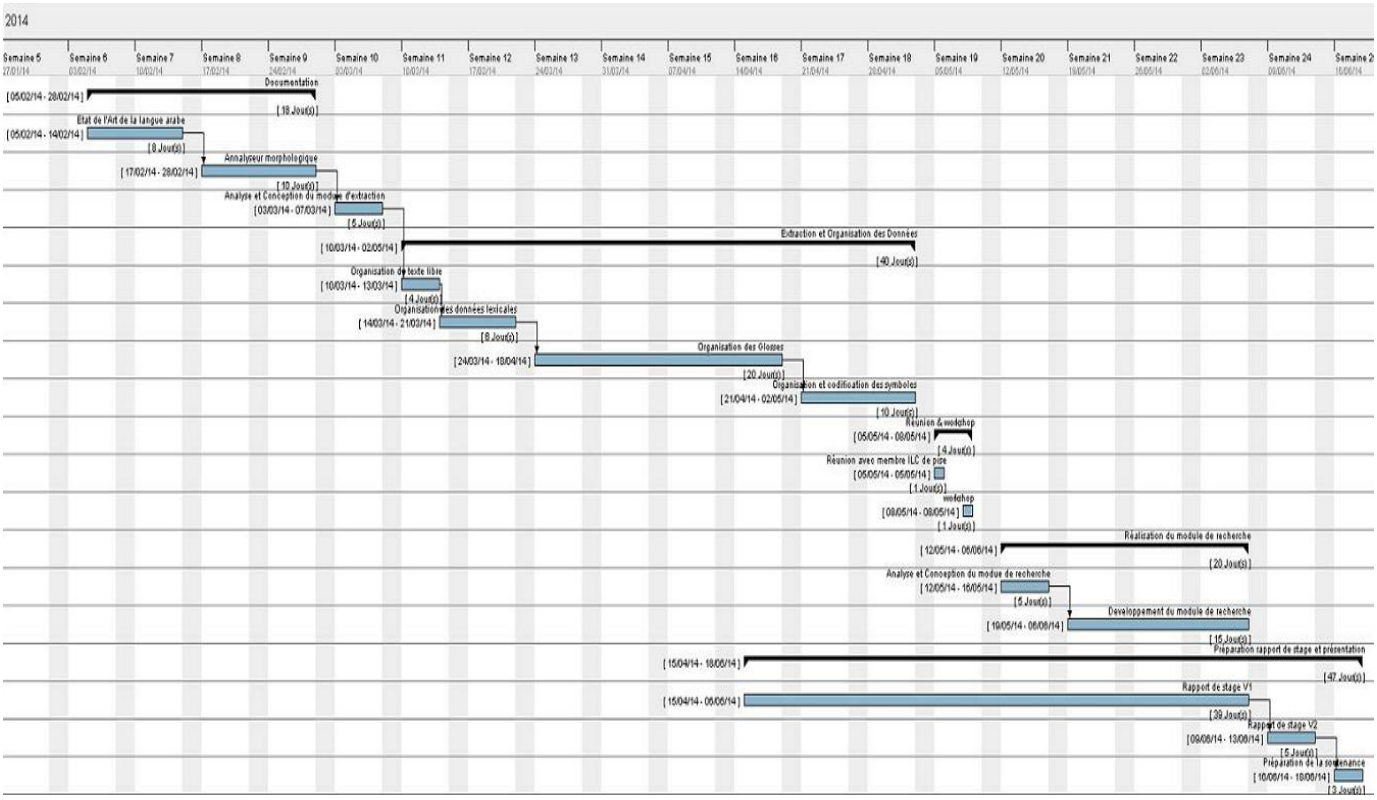

Figure 2: Diagramme de GANTT

Conclusion

Ce présent chapitre représentait un point d'entrée au projet, il présentait l'objectif et le processus de mise en œuvre du projet, la partie suivante est une présentation des caractéristiques morphosyntaxiques de la langue arabe.

Chapitre 2 :

Les caractéristiques morphologiques et lexicales de l'arabe

Ce chapitre présentera un état de l'art sur l'analyse morphologique, ainsi la définition de quelques notions de la langue arabe.

Introduction

La langue arabe est une langue dérivationnelle et flexionnelle. A l'origine, la langue arabe est la langue parlée par les Arabes. En plus, elle est la langue du Coran et de l'Islam. Du fait par la propagation de l'Islam et la diffusion du Coran, cette langue est parlée dans 22 pays alors que le nombre de ses locuteurs est plus de 280 millions.

Dans ce chapitre, nous commencerons par présenter les caractéristiques morphologiques de la langue arabe. Ensuite, nous décrirons le mécanisme de dérivation d'un mot arabe. Puis, nous présenterons les différentes catégories grammaticales. Aussi, nous préciserons les traits morphologiques des verbes et des noms arabes.

I. Les caractéristiques morphologiques d'un mot

La langue arabe s'écrit et se lit de droite à gauche. Le mot arabe s'écrit avec des consonnes et des voyelles. Les consonnes changent de forme de présentation selon leur position dans le mot (au début, au milieu ou à la fin). Les voyelles sont de deux types : les voyelles brèves et les voyelles longues. Elles sont nécessaires à la lecture et à la compréhension correcte d'un texte et permettent de différencier des paroles ayant les mêmes consonnes. Malgré l'importance de ces voyelles brèves, elles sont absentes dans la majorité des textes arabes ce qui peut engendrer des ambiguïtés de prononciation et de compréhension.

3. L'Alphabet

L'alphabet de la langue arabe comprend 29 lettres (Sibawayh -V4 de Haroun, 1982) dans lesquelles :

- Vingt-huit consonnes (حروف متحركة) (c-à-d peut être voyellée), et parmi ces dernière il y a la wāw "و" et yā' "ي", s'elles sont voyellées (nasiya نَسِيَ , bayt بَيْت , qawl قَوْل).
- Trois lettres d'allongement (حروف المد) 'alif "ا", wāw "و" et yā' "ي" lorsqu'elles présentent l'allongement d'une voyelle (bā'a بَاع, tasīru تَسِيرُ, taqūlu تَقُولُ).

Parmi ces 28 consonnes on trouve la Hamza "ء" qui se présente sous différentes formes en fonction de la vocale qui la précède et celle qui la suit, en sa position dans le mot (début, milieu, fin : Tableau1), elle peut s'écrire soit sur un support (أ ، ؤ، ئ) ou bien sur la ligne (ء), ce qui rend son orthographe un peu complexe.

Il faut noté que La *Hamza madd* a une seule forme de représentation : آ et résulte de deux combinaison **(Hamza+a+alif)** ou (**Hamza+a+Hamza+ sukūn**)

Pour la première combinaison, par exemple :

* "آدَبَ" : (Hamza+a+ alif)+(dal+a)+(bae+a)

Pour la deuxième combinaison :

* "آكُلُ" : (Hamza+a) + (Hamza+Sukun)+(kāf+ḍamma)+(lām+ ḍamma)

La *'alif* se comporte seulement comme une lettre d'allongement qu'on ne trouve jamais en tant que consonne de la racine.

Les deux consonnes (و،ي) sont considérées comme des consonnes défectueuses : "حروف العلة", par ce qu'elles subissent des changements en fonction des règles phonétiques, par exemple dans "نَسِيَ" où la yā' "ي" est une consonne et on peut trouver aussi "أَنْسَى" où la yā' est devenu lettre d'allongement "حرف مد".

La représentation graphique des consonnes est différente selon leur position dans le mot, ce qui engendre l'apparition de 100 graphies à partir des 28 consonnes.

Toutes les consonnes se lient entre elles sauf (ذ ، د ، ز ، ر ، و). Ces dernières ne se joignent jamais à gauche.

Forme	*Graphie selon la position*		
Isolée	*Initiale*	*Médiane*	*Finale*
ء	أ، إ	أ، ـأ، ؤ، ئ، ـئ، ء	
ب	بـ	ـبـ	ـب
ت	تـ	ـتـ	ـت، ة
ث	ثـ	ـثـ	ـث
ج	جـ	ـجـ	ـج
ح	حـ	ـحـ	ـح
خ	خـ	ـخـ	ـخ
د	د	د، ـد	
ذ	ذ	ذ، ـذ	
ر	ر	ر، ـر	
ز	ز	ز، ـز	
س	سـ	ـسـ	ـس
ش	شـ	ـشـ	ـش
ص	صـ	ـصـ	ـص
ض	ضـ	ـضـ	ـض
ط	طـ	ـطـ	
ظ	ظـ	ـظـ	
ع	عـ	ـعـ	ـع
غ	غـ	ـغـ	ـغ
ف	فـ	ـفـ	ـف
ق	قـ	ـقـ	ـق
ك	كـ	ـكـ	ـك
ل	لـ	ـلـ	ـل
م	مـ	ـمـ	ـم
ن	نـ	ـنـ	ـن
ه	هـ	ـهـ	ـه
و	و	ـو, و	ـو
ا	*Jamais*	ـا، ا	
ي	يـ	ـيـ	ـي

Tableau 1 : L'Alphabet de la langue arabe

En plus, on peut trouver d'autres représentations qui sont le résultat de concaténation de deux consonnes par exemple, lorsque une ل lâm est suivie d'une أ hamza, les deux lettres sont remplacées par la ligature."لأ"

4. Les Voyelles

Les voyelles ont une double fonction : l'une est morphologique ou sémantique et l'autre est syntaxique.

La langue arabe a deux séries de voyelles, les unes brèves ou courtes et les autres longues.

4.1 Les voyelles courtes

Les voyelles courtes ne sont pas comme les consonnes, elles sont rarement notées. Elles sont écrites seulement pour lever des ambiguïtés, dans les éditions du Coran ou dans les ouvrages didactiques. En effet, les voyelles jouent un rôle important dans les mots arabes, non seulement parce qu'elles enlèvent l'ambiguïté, mais aussi parce qu'elles donnent la fonction grammaticale d'un mot indépendamment de sa position dans la phrase.

Les voyelles courtes (ـُ , ـِ , ـَ) sont des diacritiques ajoutées au-dessus ou au-dessous des consonnes. Lorsque la consonne n'a aucune voyelle, on marquera une absence de voyelle représentée en arabe par une voyelle muette sukūn (ـْ) :

Voyelle courtes	Nom	Transcription
ـَ	فتحة /fatha/	a
ـِ	كسرة /kasra/	i
ـُ	ضمة /damma/	u

Tableau 2 : Les voyelles courtes

Notons que, le concept de " tanwīn " est réalisé phonétiquement par l'ajout de /an/, /un/, /in/ à la fin du mot, peut être sous trois formes (ٌ/un/, ٍ /in/, ً /an/) qui sont représentés par dédoublement des voyelles courtes. Il est ajouté seulement à la fin des mots indéterminés, par conséquent il n'apparaît jamais avec l'article de détermination "ال".

4.2 Les voyelles longues

Les voyelles longues sont formées par une des voyelles courtes et une des lettres d'allongement suivantes (ي ,و ,ا) :

Voyelles	lettres d'allongement	Transcription
ـَ	ا	ā
ـِ	ي	ī
ـُ	و	ū

Tableau 3 : Les voyelles longues

II. Mécanisme de dérivation

En arabe, la majorité des mots (lemme) sont construits sur la base d'une racine tout en respectant un paradigme : ceci concernant notamment les verbes, les noms et quelques particules.

1. La racine - الجذر

Une racine est purement consonantique, elle est formée par une suite de trois ou quatre (ou même cinq pour les noms) consonnes formant la base du mot.

La racine est un élément important dans les langues dérivationnelles. En effet, à chaque racine correspond un champ sémantique et à l'aide de différents paradigmes, on peut générer une famille de mots appartenant à ce champ sémantique, par exemple la racine [ك ت ب] peut engendrer quinze mots autour de la notion de l'« écriture » tels que «كاتِبٌ» [kātibun] (écrivain), «مَكْتَبٌ» [maktabun] (bureau), «مَكْتَبَةٌ» [maktabatun] (bibliothèque) etc.

2. Le Paradigme - الوزن

Le paradigme est un schème composé de trois consonnes [ف ع ل] ou la fā' correspond à la première consonne de la racine, la 'ayn la deuxième et la lām correspond à la troisième consonne de la racine. Elles sont vocalisées et qui peuvent être concaténées avec d'autres lettres (préfixe ou suffixe). Le paradigme joue un rôle très important dans le processus de génération des formes dérivées à partir d'une racine. Ce processus de génération consiste à remplacer les consonnes du paradigme par les consonnes de la racine en question, tout en gardant les mêmes voyelles et les mêmes lettres autres que celles de la racine, tout en respectant le même ordre des consonnes, autrement dit le paradigme peut être considéré comme une moule sur laquelle coule la racine.

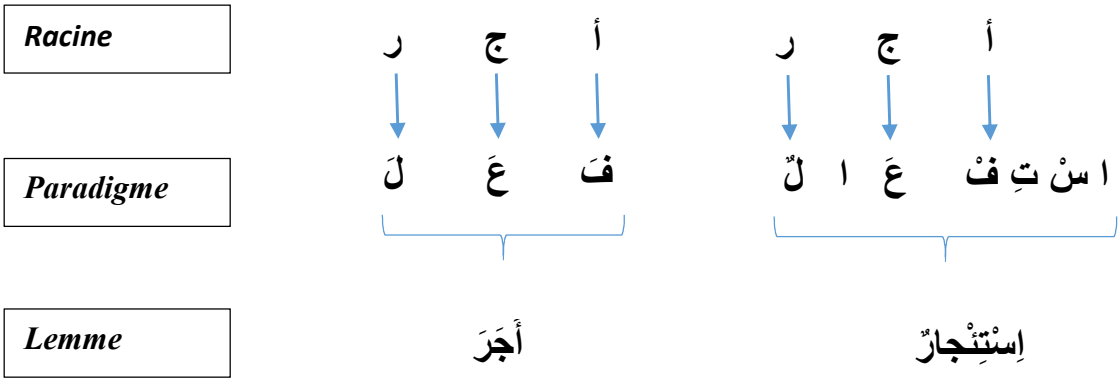

Figure 3 : Exemples de dérivation de la racine

Dans cette figure, le lemme est formé par le remplacement respectif des consonnes du paradigme par les consonnes de la racine, tout en gardant les autres composants du paradigme. On peut classer les paradigmes en deux catégories : des paradigmes verbaux et nominaux. Ainsi, à partir d'une racine, on peut générer des noms et des verbes selon la catégorie du paradigme utilisé.

3. Le lemme

Le lemme est l'entrée lexicale dans un lexique ou dans un dictionnaire. Il s'agit d'une forme entièrement vocalisée. Chaque mot est rapporté à son lemme qui est sa forme canonique qui dépend toujours de la catégorie grammaticale de ce mot : par exemple, si c'est un nom ou un adjectif il doit être au singulier et si c'est un verbe il doit être à l'accompli avec la troisième personne du singulier etc. Un lemme peut être formé par un mot simple ou un mot composé.

III. Les catégories grammaticales

Selon la théorie grammaticale arabe ancienne, le lexique de la langue arabe comprend trois catégories de mots : verbe, nom et particule (El-Dahdeh, 1996).

1. Verbe - فعل

Nous pouvons classer les verbes arabes selon plusieurs critères selon le nombre et la nature des consonnes de leurs racines et selon leurs schèmes aussi (El-Dahdeh, 1999).

Selon le nombre des consonnes de la racine, nous avons soit des verbes trilitères qui ont trois consonnes, soit des verbes quadrilitères qui ont quatre consonnes et sont peu nombreux.

Selon la nature des consonnes, nous avons soit des verbes sains qui ne sont pas formés par des lettres défectueuses (صحيح), soit des verbes défectueux (معتل) qui contiennent une ou deux lettres défectueuses qui causent des altérations importantes au cours de la conjugaison. Un verbe peut contenir la lettre *Hamza* ou *šadda*, qui peuvent engendrer des conjugaisons irrégulières.

Selon le schème et le nombre de consonnes qui constituent la structure verbale, nous avons soit des verbes nus (مجرد) qui sont formés seulement par les consonnes de leurs racines et des voyelles brèves, soit des verbes dérivés (مزيد) qui sont dérivés de trois consonnes de la racine par modification des voyelles, par redoublement de la deuxième lettre de la racine, par adjonction et même par intercalation d'affixes. Les verbes dérivés se conjuguent avec les mêmes préfixes et suffixes que le verbe nu. Les verbes trilitères peuvent être augmentés au maximum par trois lettres et les verbes quadrilitères par deux lettres. Alors, la longueur maximale d'un verbe arabe est de 6 lettres.

2. Nom - اسم

Les noms arabes regroupent les substantifs, les adjectifs et les pronoms, ainsi que d'autres noms invariables (Blachère et al., 1975). Les substantifs et les adjectifs sont créés en prenant pour origine tantôt un verbal tantôt un type nominal. Nous pouvons distinguer deux classes de noms : la première regroupe les noms conjugables ou semi conjugable qui peuvent avoir la forme duelle, plurielle etc. (arbre - شجرة) La deuxième classe regroupe les noms non conjugables qui gardent la forme quel que soit le contexte (Ahmad - أحمد). Les noms conjugables sont soit des noms primitifs qui échappent à toute dérivation, soit des noms dérivationnels qui sont formés à partir d'une racine.

- Les pronoms

  La classe des pronoms a été introduite en tant qu'extension du système de décomposition traditionnel du lexique de la langue arabe. Cette classe regroupe quelques formes qui étaient considérées comme des noms particuliers. Cet ensemble échappe à toute règle de dérivation. Les pronoms forment une liste fermée de mots. Dans cette liste, nous distinguons :

  Les pronoms démonstratifs : ils représentent une sous-catégorie de pronom expriment une idée de démonstration. Ils permettent d'indiquer que l'objet représenté se trouve, soit dans le texte, soit dans l'espace ou le temps, défini par la

situation d'énonciation, Ils existent deux sous-ensembles : les démonstratifs de de proximité et les démonstratifs d'éloignement.

Les pronoms relatifs : Ils se rapportent au nom ou au pronom personnel qui les précède et que nous désignons par antécédent.

Les pronoms personnels : servent à désigner les trois types de personnes grammaticales :

- La première personne, c'est-à-dire, l'énonciateur ou locuteur : "أنا" (je) ou "نحن" (nous).
- La deuxième personne, c'est-à-dire, le destinataire ou interlocuteur : "أنتَ" (tu, masculin), "أنتِ" (tu, féminin), "أنتما" (vous, duel), "أنتم" (vous, masculin, pluriel), "أنتن" (vous, féminin, pluriel).
- La troisième personne, c'est-à-dire, la personne absente, celle dont on parle : "هو" (il),"هي" (elle),"هما" (ils, duel), "هم" (ils), "هن" (elles).

3. Particule

Les particules sont des lemmes invariables et en nombre limité. Ils indiquent l'articulation de la phrase et ils servent à préciser les modalités des prépositions verbales et nominales [El-Dahdeh, 1996], [Blachère et al, 1975]. Malgré la difficulté de classer ces particules, on tenterait ce classement :

- ✓ Préposition : exemple (ب، ك، ل، عَنْ، حَتَّى).
- ✓ Particules de coordination : exemple (و، ف، ثُمَّ، أَو).
- ✓ Particules interrogatives : exemple (مَا، هَلْ، أ).
- ✓ Particules d'affirmation : exemple (نَعَم، بَلى، أجَلْ).
- ✓ Particules de négation : exemple (لما، لنْ، لم)
- ✓ Particules distinctive : exemple (أي).
- ✓ Particules relatives : exemple (ما).
- ✓ Particules de future : exemple (لَنْ، سَوْفَ، سَ)
- ✓ Particules conditionnelles : exemple (إنْ، لَوْ)

Conclusion

Dans ce chapitre, nous avons exploré les différentes caractéristiques morphologiques et syntaxiques de la langue arabe.

Les caractéristiques présentées dans ce chapitre est la base de la phase de l'extraction des données du dictionnaire, qui fera l'objet du chapitre 3, tant que la partie suivante présente un état de l'art sur le traitement automatique de la langue arabe.

Chapitre 3 :

Etat de l'Art : Traitement Automatique de la Langue Arabe

Ce chapitre présentera un état de l'art sur quelques analyseurs morphologiques existant de la langue arabe, après on présentera aussi les modèles de base et des extensions du standard LMF.

Introduction

L'arabe est la langue sémitique contemporaine la plus parlée de nos jours avec plus de 300 millions de locuteurs (Habash, 2010). Dans ce chapitre on va présenter un état d'art sur le traitement automatique de cette langue, suivi de quelques analyseurs morphologiques, et enfin on va présenter les modèle noyau et extension du LMF.

I. Etat de l'Art : Le traitement automatique de l'Arabe

Historiquement, les premiers travaux importants dans le domaine du TALN ont porté sur la traduction automatique (1954), avec la mise au point du premier traducteur automatique. Quelques phrases russes, sélectionnées à l'avance, furent traduites automatiquement en anglais. Bien que le vocabulaire ne comptât que 250 mots et 6 règles de la grammaire, cette expérience a déclenché de nombreux travaux dans ce domaine. C'est en effet l'époque où l'URSS remporte succès après succès dans la course à l'espace et où les militaires américains sont très désireux de suivre les publications techniques soviétiques, sans pour autant faire apprendre le russe à tous leurs ingénieurs.

Les premières recherches sur le traitement automatique de l'arabe ont commencé vers les années 1970 (Cohen, 1970) et concernaient notamment le lexique et la morphologie. Avec Internet, la diffusion de la langue arabe et la disponibilité des moyens de traitement de textes arabes, les travaux de recherche ont abordé des problématiques plus variées comme la traduction automatique. Les outils d'apprentissage sont de plus en plus disponibles et permettent de développer facilement des outils de traduction et d'extraction d'information.

Comme il a été déjà mentionné, l'arabe est une langue morphologiquement riche, l'analyse morphologique est une tâche importante qui permet à la fois de réduire le vocabulaire ainsi qu'à faciliter et améliorer les alignements en essayant d'avoir le même nombre de mots en source et en cible dans des traitements automatique, comme la traduction par exemple.

La segmentation est un processus résultant généralement d'une analyse du texte qui consiste essentiellement en une analyse morphosyntaxique et une analyse morphologique. Souvent, ces deux tâches sont liées. L'analyse morphosyntaxique consiste à détecter pour chaque mot du texte sa fonction grammaticale dans la phrase et l'étiqueter. Des outils d'analyse morphosyntaxique sont disponibles sur Internet comme Stanford Tagger 16.

(Darwish, 2002) Présente l'un des premiers travaux sur l'analyse morphologique de l'arabe où il présente une approche qui permet de construire rapidement un analyseur morphologique. L'analyseur produit la racine éventuelle pour chaque mot en arabe. Il est basé sur des règles dérivées automatiquement et statistiquement. Des approches de plus en plus performantes sont apparues par la suite.

(Lee, 2004) Utilise les étiquettes morphosyntaxiques du texte en arabe -segmenté- et qui sont alignées avec les étiquettes morphosyntaxiques du texte en anglais afin de décider de garder ou pas les segmentations.

AMIRA développée par (Diab, 2009) implémente une approche différente, où la séparation des clitiques est effectuée indépendamment de l'étiquetage morphosyntaxique.

(Marsi, 2005) utilisent l'apprentissage basé sur la mémoire (memory-based learning) pour l'analyse morphologique et l'étiquetage de l'arabe. Ils utilisent le k-plus-proche voisin et montrent que l'étiquetage morphosyntaxique peut être utilisé pour choisir l'analyse morphologique la plus appropriée.

Dans les travaux de (Kulick S. , 2010) la segmentation et l'étiquetage morphosyntaxique sont effectués simultanément en utilisant un classifieur, et sans utiliser d'analyseur morphologique.

Plus récemment (Kulick S., 2011) a fait une extension de l'approche, en distinguant entre les « tokens » classe ouvert (telle que nom, verbe, nom propre, etc.) et les « tokens » classe fermée (telle que préposition, pronom relatif, etc.), qui diffèrent dans leurs affixations morphologiques possibles et leur fréquences. Une liste de noms propres extraites de la base de données SAMA-v3.1 (Maamouri, et al., 2010) était utilisée comme trait pour l'aide à la classification.

MADA développé par (Habash, Rambow, et Roth, 2009) est l'outil d'analyse morphologique et de désambiguïsation pour l'arabe le plus utilisé. Cet outil effectue une analyse morphosyntaxique et choisit une proposition de segmentation parmi les propositions de mots segmentés proposés par BAMA (Habash et Rombow, 2005). D'autres outils de segmentation de l'arabe ont été développés initialement pour le prétraitement d'autres langues, ensuite ils ont été adaptés pour la langue arabe comme ***MorphTagger*** (Mansour, 2010) qui a été conçu d'abord pour l'étiquetage morphosyntaxique de ***l'hébreu*** (Mansour, Siman'an, et Winter, 2007) et adapté par la suite pour l'étiquetage et la segmentation de l'arabe. ***MorphTagger*** utilise également l'analyseur morphologique BAMA.

(El Isbihani, et al. , 2006) proposent trois méthodes de segmentation de la langue arabe : une méthode à base d'apprentissage supervisé, une méthode basée sur les fréquences, et une méthode fondée sur les automates à états finis. Ils montrent que la dernière approche donne les meilleurs résultats et est adaptable à différentes tâches.

Parmi les approches à base de règles, on cite ***G-LexAr (*** (Dbili, Achour, & Souissi, 2002) qui est un analyseur morphologique de l'arabe à base de règles. Il effectue à la fois voyellation, lemmatisation et segmentation d'un texte en arabe. Des travaux ont été également effectués pour la segmentation de l'arabe dialectal comme ceux de (Habash et Rambow, 2006) ou aussi ceux de (Mohamed et Oflazer, 2012)**.**

Une approche à base du Naïve Bayes propose une désambiguïsation des mots traduits de l'arabe vers l'anglais en utilisant des schémas de correspondances dans un corpus parallèle (Ahmed et Nürnberger, 2008).

(Shah, et al., 2010) proposent un modèle d'analyse lexicale utilisé dans plusieurs tâches entre autres l'annotation manuelle du texte en arabe.

(El Kassas et Kahane, 2004) utilisent un arbre de dépendance afin de présenter la structure syntaxique des phrases en arabe.

II. L'analyse morphologique

L'analyse morphologique arabe et l'un des outils qui permettent de résoudre la majorité des problèmes de la langue arabe, elle a été largement utilisée dans plusieurs domaines du Traitement automatique des langues naturelles (TALN) tels que la recherche documentaire, les dictionnaires électroniques, les systèmes de marquage, etc.

Plusieurs travaux ont été réalisés dans le but d'élaborer des analyseurs morphologiques de la langue arabe et qui peuvent être regroupés en trois approches *(* (Darwish, 2002)*;* (Yousfi, 2010)*)* :

- ✓ ***L'approche symbolique*** : Cette approche est basée sur la segmentation du mot en préfixes, infixes et suffixes dans le but d'extraire la racine du mot arabe. Plusieurs analyseurs morphologiques ont été élaborés et qui s'appuient sur cette approche ((Darwish, 2002); (Buckwalter, 2002) ; (Hegazi, 1986); (Beesly, 1998); (khoja, 1999) ; (Soudi, 2002)). Parmi les analyseurs les plus connus pour cette approche est celui de Buckwalter, ce dernier consiste à déterminer toutes les segmentations possibles du mot, puis à chercher les résultats dans les listes des radicaux, des suffixes et des préfixes, et vérifie ensuite si les morphologies

de chacun des éléments sont compatibles entre elles en examinant trois tables de correspondances : préfixe-radical, préfixe-suffixe, radical-suffixe.

- ✓ ***L'approche statistique*** : Cette approche calcule les possibilités et les probabilités qu'un préfixe, suffixe et un radical peuvent apparaître ensemble dans une base de données des mots (Goldsmith, 2001)
- ✓ ***L'approche hybride*** : cette approche combine entre les deux approches précédentes (Darwish, 2002)

Au début de ce stage, une période a été consacrée à étudier quelques analyseurs morphologiques de l'arabe, qui aident à son traitement automatique. On cite parmi ces derniers, les plus utilisés :

1. AraMorph (Buckwalter)

Buckwalter Arabic Morphological Analyzer - (Buckwalter, 2002-2004) est un analyseur morphologique de l'arabe qui a été développé par *Tim Buckwalter*. BAMA utilise une approche où les règles morphologiques et orthographiques sont intégrées directement dans le lexique. L'analyseur morphologique est représenté sous forme d'une grande base de données dans laquelle chaque mot en arabe est présenté avec toutes ses formes dérivées possibles « préfixe-racine-suffixe ». Chaque forme est donnée avec la version voyellée, l'ensemble des morphèmes (lemme, préfixes et suffixes) constituants de chaque mot, et toutes les étiquettes morphosyntaxiques (ou grammaticales) de chaque composante du mot.

Pour chaque mot, BAMA fournit un ensemble de toutes les segmentations possibles. Cet ensemble comprend un lemme sous la forme d'un identifiant unique, ainsi que pour chaque solution, l'ensemble des morphèmes constituants de chaque mot, leurs étiquettes grammaticales et la traduction correspondante en anglais. Toutes les propositions de segmentations sont proposées avec la translittération.

2. Alkhalil Morpho Sys

Alkhalil Morpho Sys (BOUDLAL, et al., 2010) est un analyseur morphosyntaxique des mots arabe standard. Le système peut traiter des textes non vocalisés ainsi que celles partiellement ou totalement vocalisés. Leur approche est basée sur la modélisation d'un très grand nombre de règles morphologiques arabes, ainsi que sur l'intégration des ressources linguistiques qui sont utiles à l'analyse, tels que la base de données racine, les modèles associés aux racines vocalisées et tables des proclitiques et enclitiques.

Alkhalil Morpho Sys est un logiciel open sources, actuellement dans sa version 2, ses développeurs essayent de rendre le produit sous forme d'une API à exploiter, car l'exploitation du produit mène les gens à changer pas mal de chose dans le code source.

III. LMF - Lexical Markup Framework

Lexical Markup Framework (LMF ou cadre de balisage lexical, en français) est le standard de l'Organisation internationale de normalisation (plus spécifiquement au sein de l'ISO/TC37) pour les lexiques du traitement automatique des langues. L'objectif est la normalisation des principes et méthodes relatifs aux ressources langagières dans le contexte de la communication multilingue et de la diversité culturelle.

L'objectif est de fournir un modèle commun pour la création et l'utilisation des ressources langagières, de gérer l'échange des données entre ces ressources et de permettre la fusion d'un grand nombre de ressources électroniques afin de constituer un vaste réseau de descriptions linguistiques.

Les différents types d'instanciation de LMF peuvent inclure des ressources monolingues, bilingues aussi bien que multilingues. Les mêmes spécifications valent pour les petits et grands lexiques, pour les structures simples comme complexes, pour les ressources lexicales de l'écrit comme de l'oral. Les descriptions couvrent aussi bien la morphologie, la syntaxe, la sémantique que les notations multilingues. Les langues ciblées ne se limitent pas aux langues européennes mais couvrent toutes les langues naturelles. LMF est capable de représenter la plupart des lexiques, incluant les lexiques WordNet et PAROLE.

Un lexique LMF se présente sous forme d'un méta-modèle noyau obligatoire et un ensemble d'extensions optionnelles qui décrivent les ressources lexicales spécifiques en réutilisant les composants du noyau (Francopoulo, 2006) Le méta-modèle noyau forme une structure hiérarchique des classes UML qui spécifie les notions de lexique, de l'entée lexicale, de forme et de sens. LMF fournit un mécanisme permettant de spécifier le contenu des classes du méta-modèle noyau à l'aide de descripteurs élémentaires sous forme de couples « Attribut-Valeur » définis par une autre norme ISO 12620, appelée catégories de données (RCD) (Romary.L, 2003) Les catégories de données reflètent les concepts de base linguistique, tels

que /PartOfSpeech/, /Genre/, /Nombre/ et ils sont stockés et gérés indépendamment de la structure hiérarchique du modèle de données.

1. Le modèle de base de LMF

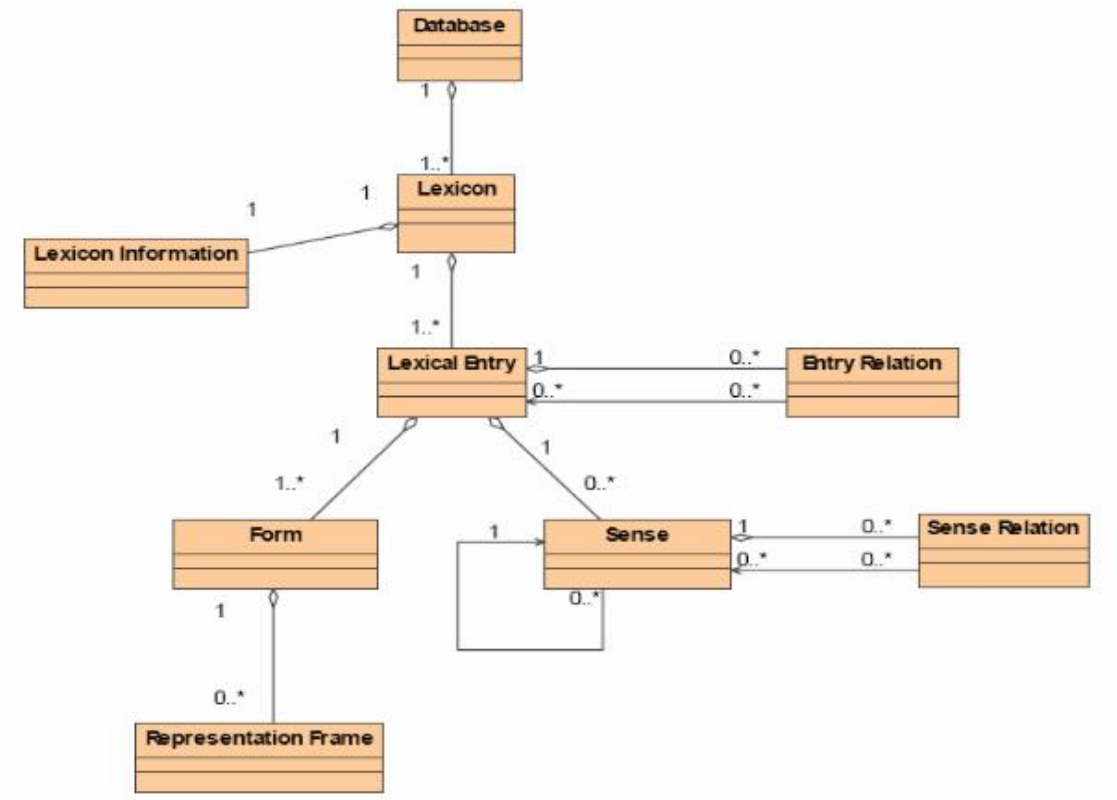

Figure 4: Modèle de base LMF

Le modèle de base ou noyau de LMF, présenté à la figure précédente, est une structure hiérarchique composée des Composants suivants :

- ✓ *Database* : la totalité de la ressource qui peut avoir un ou plusieurs *lexicon*.
- ✓ *Lexicon Information* : les informations administratives et autres attributs généraux concernant cette base de données tel que le numéro de la version, l'auteur etc.
- ✓ *Lexicon* : un lexique d'une langue donnée qui appartient à une seule *Database* et qui peut avoir un ou plusieurs *Lexical Entry.*
- ✓ *Lexical Entry* : représente un mot, un mot-multiple ou un affixe de la langue courante. C'est l'unité élémentaire dans une base lexicale qui porte l'information d'une partie du discours. Elle peut avoir un ou plusieurs *Form* et zéro ou plusieurs *Sense*.

- ✓ *Entry Relation :* présente une relation entre plusieurs entrées lexicales appartenant au même lexicon et qui peut avoir des attributs pour décrire cette relation.
- ✓ *Sense :* les attributs qui décrivent le sens du mot. Cette classe peut être partagée par plusieurs entrées lexicales. Cette classe a une relation réflexive et qui est composé par zéro ou plusieurs relations sémantiques.
- ✓ *Sense Relation :* les attributs qui décrivent une relation entre deux sens à l'intérieur d'une langue. Elle peut être liée au *Sense* à travers la composition ou la référence.
- ✓ *Form :* les valeurs orthographiques et phonologiques des unités lexicales avec des spécifications grammaticales. Cette classe peut avoir deux sous-classes de spécification : *lemmatisedForm* et *inflectedForm* présentées dans la Figure ci-dessous.

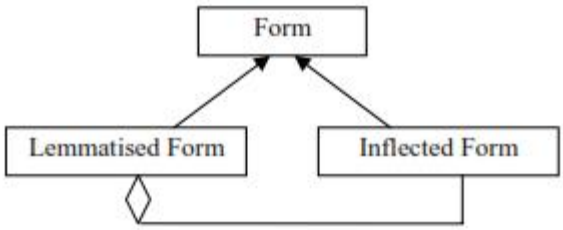

Figure 5: les sous classes de Form

- ➢ *LemmatisedForm :* une classe de spécification qui hérite toutes les propriétés de la classe *Form* et représente le lemme de cette entrée lexicale. Elle participe avec une seule *Lexical Entry*. Elle peut avoir zéro ou un paradigme et zéro ou plusieurs *InflectedForm*.
- ➢ *InflectedForm* : une forme fléchie correspond à une forme d'occurrence d'une *lemmatisedForm*.

- ✓ *Representation frame :* spécifie la représentation orthographique d'un mot s'il en a plusieurs telle qu'une graphie ou une transcription phonologique.

2. L'extension de LMF

Les fondateurs de LMF ont fait reposer le TALN sur cinq extensions. Ils ont utilisé les classes UML au niveau de la conception. Les classes dont la couleur est blanche appartiennent au noyau, les autres font partie de l'extension en cours.

2.1 Extension morphologique

L'extension morphologique est la partie obligatoire pour la plupart des applications de

TALN. Cette extension est traitée de deux manières différentes dans LMF. La première présente les formes fléchies, la deuxième fait référence aux paradigmes de flexion pour les générer. Les différentes classes relatives à cette extension sont présentées dans la Figure ci-dessous.

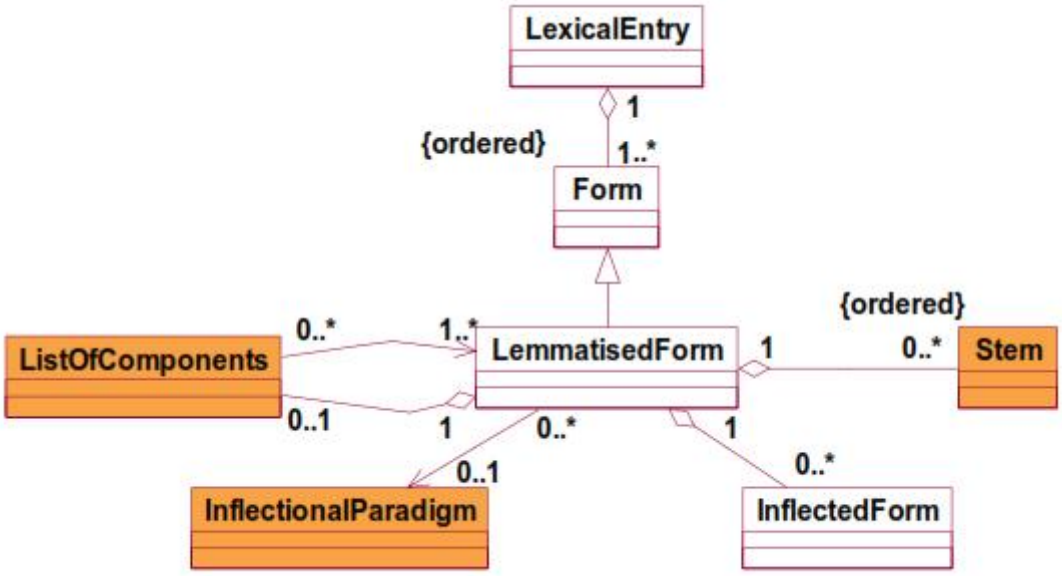

Figure 6: L'extension morphologique

Les trois classes de cette extension sont :

- ✓ *InflactionalParadigm* : factorisation d'un ensemble de structures communes à un grand nombre de mots. Par conséquent, un paradigme peut être utilisé avec plusieurs *LemmatisedForm*.
- ✓ *Stem* : représente un des éléments qui forme un mot (par exemple anticonstitutionnellement a deux Stems qui sont respectivement anti- et constitution) Elle peut participer avec un seul lemme qui peut avoir zéro ou plusieurs *Stem*.
- ✓ *List of Components* : indique l'ordre des mots dans le cas d'un mot multiple

2.2 Extension syntaxique

Cette extension est optionnelle, elle est liée à deux classes du noyau à savoir *LexicalEntry* et *Sense*, et à la classe de l'extension sémantique *SemanticArgument*. L'interaction entre ces classes avec les classes de cette extension est présentée dans le diagramme suivant :

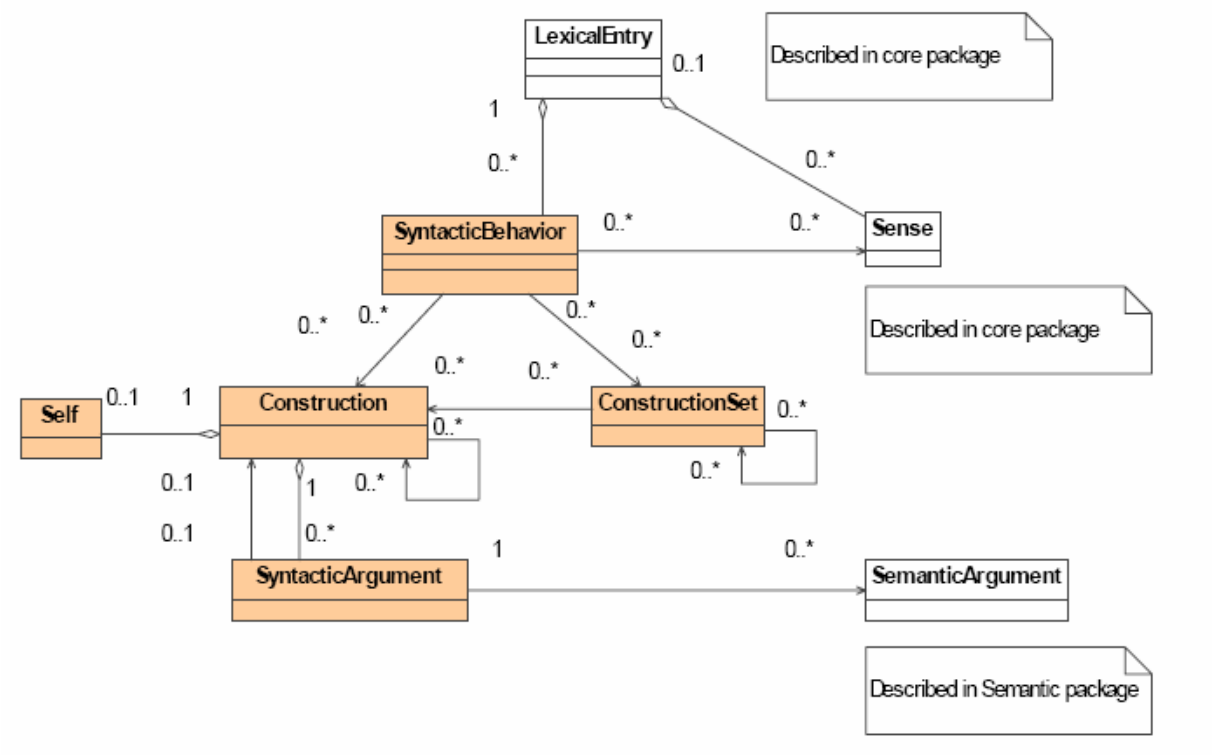

Figure 7: Le modèle syntaxique

Les cinq classes de cette extension sont :

- ✓ *SyntacticBehavior* : représente un des comportements possibles d'un ou de plusieurs sens.
- ✓ *Construction* : est partagée par tous les mots ayant le même comportement syntaxique dans la même langue. Il peut hériter des relations et des attributs d'une autre *Construction* plus générique par la relation de réflexion. Ainsi, il est possible d'intégrer une ontologie hiérarchique des constructions.
- ✓ *Self* : se réfère à l'entrée lexicale courante.
- ✓ *ConstructionSet* : regroupe un ensemble de construction syntaxique et une relation possible qui subit ces constructions. Il peut hériter des relations et des attributs d'une autre *ConstructionSet* plus générique par la relation de réflexion. Là, également, il y a lieu d'intégrer une ontologie hiérarchique des ensembles de constructions
- ✓ *SyntacticArgument* : décrit un actant syntaxique et peut être lié récursivement à une *Construction* pour décrire des arguments très complexes. Il permet la connexion avec un actant sémantique par *SemanticArgument.*

Le diagramme de l'extension syntaxique suivant est basé autour du composant «comportement syntaxique». Un comportement syntaxique est un patron de construction syntaxique qui peut être utilisé par plusieurs entrées lexicales permettant ainsi de factoriser le même comportement syntaxique utilisé par plusieurs entrées lexicales et d'éviter la redondance.

Un comportement syntaxique est décrit par l'ensemble des constructions syntaxiques permises éventuellement groupées dans des sous-ensembles de significations sémantiques disjointes. Un «Frame» représente synthétiquement un ensemble de structures syntaxiques possibles associées à un prédicat.

2.3 Extension sémantique

Cette extension assure les liens entre des définitions, des exemples, des *synsets* et des représentations prédicatives qui vont permettre la liaison entre les arguments sémantiques les arguments syntaxiques :

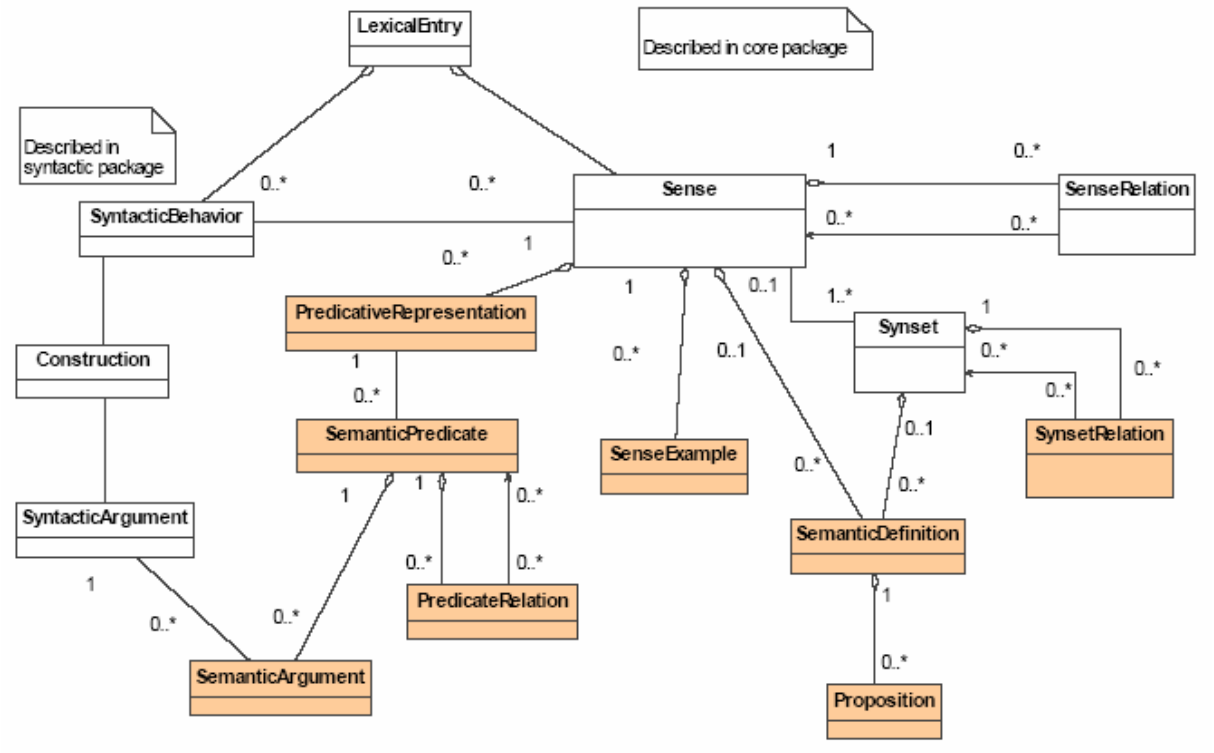

Figure 8: Le modèle sémantique

Les neuf classes de cette extension sont :

- ✓ *SenseExample* : décrit les utilisations d'une signification particulière.

- ✓ *SemanticDefinition* : permet la description narrative d'un *Sense* ou d'un *Synset*. Il n'est pas prévu pour l'usage d'un programme, mais il est fourni pour faciliter l'entretien humain.
- ✓ *Proposition* : assure le raffinement de la définition sémantique.
- ✓ *SemanticPredicate* : peut être utilisé pour représenter une signification commune entre des sens différents qui ne sont pas complètement des synonymes. Ces sens peuvent être liés à des entrées lexicales qui ont des parties de discours différentes.
- ✓ *PredicativeRepresentation* : décrit le lien entre le sens et le prédicat sémantique.
- ✓ *SemanticArgument* : est une classe consacrée pour le lien d'un actant sémantique avec un actant syntaxique exprimé par le moyen d'un *SyntacticArgument.*
- ✓ *PredicateRelation* : permet la description d'une relation entre deux ou plusieurs prédicats sémantiques.
- ✓ *Synset* : décrit une signification commune et partagée dans une même langue, autrement dit, il lie des synonymes.
- ✓ *SynsetRelation* : permet le lien entre deux *Synsets.*

On trouve aussi deux autres extensions : *Extension des annotations multilingues* et l'*Extension des modes de flexion* : les paradigmes de flexion cette dernière permet la description des paradigmes qui permettent la génération des formes fléchies. Tant que la première se base sur des liaisons inter-langues qui vont relier des sens par un sens intermédiaire *SenseAxis*. Le mécanisme de transfert permet de relier deux comportements syntaxiques de langues différentes par la classe *TransferAxis*.

Les fondateurs de LMF ont profité de l'expérience des projets précédents (Genelex, Eagles…) pour les langues indo-européennes. Ils ont proposé un modèle qu'ils considèrent complet, générique et extensible se basant sur d'autres normes telles que l'ISO 12620. Il est à noter que LMF a été testée conforme pour plusieurs langues. En effet, à l'origine Lexical Markup Framework est une proposition franco-américaine. Elle correspond par conséquent aux caractéristiques du français et de l'anglais. Par la suite, la conformité d'autres langues a été aussi testée (l'italien, l'espagnol…). En effet, il faut en étudier les possibilités d'application avec d'autres langues qui possèdent des structures différentes de celle de l'indo-européenne tel que l'arabe.

Conclusion

Dans ce chapitre, nous avons présenté un état de l'art sur le traitement automatique de la langue arabe, ainsi quelques analyseurs morphologiques. Et dans la deuxième partie nous avons défini le standard LMF et détaillé les composants du noyau ainsi des principaux extensions.

Le chapitre suivant sera consacré à l'étape d'extraction des données morphosyntaxiques et leurs organisations.

Chapitre 4 :

Extraction et Organisation des données

Ce chapitre présentera les différentes étapes d'extraction et d'organisation des ressources lexicales du dictionnaire de la langue arabe « AL Ghani Azzahir »

Introduction

La performance de la recherche d'information en langue arabe reste une problématique à cause des caractéristiques structurelles de la langue et parce que le problème a fait l'objet de beaucoup moins de travaux que les autres langues. En effet, les moteurs de recherche les plus utilisés et les plus célèbres, comme Google et Yahoo! n'ont pas les mêmes performances en arabe que dans les langues occidentales (SEO, 2010). Cette différence est due probablement au fait que le fonctionnement d'un moteur dépend de la nature et de la complexité de la langue traitée. Cette spécificité réside aux niveaux morphologiques et structurels de l'arabe par rapport aux langues occidentales. En effet, l'analyse morphologique d'un mot arabe consiste principalement à déterminer sa racine. Donc, les études menées sur les moteurs de recherche et sur les méthodes d'extraction des caractéristiques de la racine d'un mot arabe contribuent à identifier les problèmes et à améliorer les performances de ces moteurs en langue arabe.

Par ailleurs, la proposition des dictionnaires arabes a constitué l'essentiel des travaux effectués sur la morphologie de la langue arabe au fil de l'histoire (AL Hajjar, 2010)**.** Bien que ces dictionnaires sont disponibles sur internet sous formes des fichiers textes plats mais ils ne sont pas exploitables directement. Ceci s'explique par la rareté du contenu numérique arabe structuré sur le web par rapport à d'autres langues (WLP, 2007) (LJN, 2010)**.** Pour ces raisons, plusieurs tentatives de construction des dictionnaires électroniques ont été signalées ces dernières années (Almuajam, 2009)**,** (Sakher_Lexicons, 2009). En effet, la majorité de ces dictionnaires n'est pas structurée et n'offre que des services de navigation et de recherche dans plusieurs dictionnaires classiques.

Notre travail s'inscrit dans l'objectif de proposer et de représenter sous format XML les entrées lexicales du dictionnaire ***ALGHANI AZZAHIR*** « الغني الزاهر », qui présente les caractéristiques qu'il soit structuré et évolutif et ceci dans l'optique d'enrichir le contenu numérique de la langue arabe.

I. Ressource numérique Linguistique :

Figure 9: Dictionnaire Alghani Azzahir

Le dictionnaire « **AL GHANI AZZAHIR** » est le premier dictionnaire compilé par le chercheur marocain **Abdelghani ABOU EL AAZM,** et contient plus de 65 000 entrées.

Ce dictionnaire se distingue par :

- ✓ Le classement alphabétique selon la prononciation.
- ✓ Des données linguistiques (phonologie, morphologie, pragmatique, sémantique, syntaxe).
- ✓ Des figures de style (métaphores, analogies).
- ✓ Une méthodologie lexicographique novatrice.
- ✓ La prise en compte du patrimoine culturel avec une ouverture sur la création arabe contemporaine.
- ✓ La collecte des mots et l'exploration de leur sens à la lumière de leur évolution, de leur usage contextuel général et/ou spécialisé.
- ✓ Une approche genre qui se soucie de la parité entre les sexes.
- ✓ Entièrement voyellisé.
- ✓ Destiné à tous, et en particulier aux étudiants, enseignants, chercheurs et écrivains.
- ✓ Nombreuses illustrations, cartes géographiques et reproductions d'œuvres d'art.

- ✓ Néologismes, emprunts arabisés ou non, étymologie latine, ou autre, spécifiée
- ✓ Terminologie des religions, des civilisations, des sciences et des technologies

Figure 10 : Statistique du dictionnaire Alghani Azzahir

II. Extraction automatique des ressources

1. Généralité

Dans le processus de construction d'un lexique Digital, différents intervenants sont concernés :

- ✓ ***le lexicologue*** défini les informations qui seront contenues dans le lexique, spécifie leurs formes et donne les critères permettant de définir les unités du lexique.
- ✓ ***l'informaticien*** construit les outils spécifiques au lexique ainsi défini et met au point la méthodologie qui sera utilisée lors de la construction du lexique. Il construit de plus les interfaces nécessaires au lexicographe.
- ✓ ***le lexicographe*** construit le lexique selon les spécifications ainsi faites. Il va construire les unités du lexique et/ou compléter des unités déjà existantes.

2. L’organisation de la ressource lexicale :

Suivant notre objectif on est entrain de construire notre LMF mais par étape, d’abord il faut réunir toutes les entrées qui ont les mêmes caractéristiques :

- L'ID

- Le lemme est la vraie entrée lexicale.

- La part of speech (POS) est la nature de la parole qui peut être généralement verbe, nom ou particule.

Mais on peut avoir d’autres informations qui nous intéresseront dans le future (dérivé, transitif ou intransitive…)

Premièrement nous avons pris le document numérique original contenant les entrées lexicales de la lettre « HAMZA » (Word) et qui a été formaté en format Unicode (.TXT) où chaque entrée lexicale est organisée sur une ligne : Le retour à la ligne représente une nouvelle entrée lexicale.

Dans un premier temps nous avons travaillé sur la première partie du fichier numérique de la lettre HAMZA, afin de bien contrôler les résultats obtenus, cette dernière consonne se compose de trois parties et elle est considérée comme la lettre contenant le plus grand nombre des entrées lexicales. Dans cette première partie nous avons 3253 lignes qui représentent 3253 entrées lexicales.

Exemple d’entrée lexicale :

اِسْتَجْهَلَ – [ج ه ل] (ف: س. م). يَسْتَجْهِلُ، مص. اِسْتِجْهالٌ. 1."اِسْتَجْهَلَ الرَّجُلَ": وَجَدَهُ أَوْ عَدَّهُ جاهِلاً. 2."اِسْتَجْهَلَ الْخَطِيبَ": حَمَلَهُ عَلَى الْجَهْلِ. "مَنِ اِسْتَجْهَلَ مُؤْمِناً فَعَلَيْهِ إِثْمُهُ". 3."اِسْتَجْهَلَ الوَلَدَ": اِسْتَخَفَّهُ. 4."اِسْتَجْهَلَتِ الرِّيحُ الغُصْنَ": حَرَّكَتْهُ فاضْطَرَب.

Le travail préliminaire avait pour but de délimiter les parties constituantes de toute entrée lexicale. Ces parties ont été divisées en deux sous parties qu’on a nommées :

- ✓ Texte libre : contenant le lemme suivi de se racine et des informations morphosyntaxiques, forme imperfective, Masdar, pluriel, etc. ça dépend de chaque lemme. Il nous donne toutes les informations sur le lemme
- ✓ Glosse : contient les différentes définitions du lemme en plus des exemples ou citations.

Chaque partie qui suit un numéro est une glosse (glosse1, glosse2, …) qui sera détaillé dans la partie suivante de ce chapitre.

La première chose à extraire est le texte libre contenant les informations lexicales, morphologiques et syntaxiques, et la partie des glosses, et de les représenter dans des balises d'un fichier XML, après on détaille chacune des parties en faisant l'extraction de chaque information toute seule suivant son ordre dans l'entrée lexicale.

Dans notre cas la première consonne est toujours la Hamza, et on contrôle qu'il n'y a aucun lemme qui commence avec une autre lettre, ce qui signifie une erreur extraction. Ainsi que les cas qui ne rentrent pas dans cette organisation sont renvoyé vers un fichier d'erreur : *alef_1_error.xml,* afin d'être traités par la linguiste ***Nahli Ouafae***.

2.1 Extraction du lemme

Dans la ressource linguistique utilisée, le lemme est la première parole de chaque ligne suivie d'un trait, qu'on représente dans le fichier XML par la balise

<lemme> اِسْتَجْهَلَ *</lemme>*

2.2 Extraction de la racine

Parmi les avantages et les particularités du dictionnaire (*Alghani Azzahir)* utilisé est que chaque entrée a une racine et qui est représentée entre [ ].

<racine> ج ه ل *</racine>*

2.3 Extraction des données lexicales

Entre parenthèse on trouve les différentes données morphologiques et syntaxiques qu'on doit codifier. Pour l'instant on les met dans la même balise afin de pouvoir les utilisées après :

*<données_lexicales>*ف_س_م*</données_lexicales>*

Dans ce cas par exemple :

- ✓ ف = verbe
- ✓ س = six consonantes (pas très important en linguistique numérisée) mais ça nous permet de sélectionner tous les verbes de cette classe pour les travailler après en fonction de leur paradigme. La deuxième lettre représente le nombre des consonantes qui constituent le verbe :

  (ثلاثي ث , رباعي ر , خماسي خ , سداسي س)

- ✓ م = transitif

Symbole	Means	Morphology	POS
ث	فعل ثلاثي	3 Consonants	
ر	فعل رباعي	4 Consonants	
خ	فعل خماسي	5 Consonants	
س	فعل سداسي	6 Consonants	
مث	مثنى	dual	Noun
مؤ	مؤنث	female	Noun
(ة)	تأتي أمام كل مدخل يحتمل التأنيث مثال زائر، ة -أي زائر، زائرة، ثم بعد ذلك يأتي جمع المذكر وجمع المؤنث	Feminine suffixe	Noun
ج.	جمع	Plural	Noun
ج. ـون	جمع المذكر السالم	Masculine Sound Plural	Noun
ج. ـات	جمع المؤنث السالم	Feminine Sound Plural	Noun
مفع	إسم مفعول	Affected Theme	Noun
فا	آسم فاعل	Agent name	Noun
ص	صيغة	Pattern	
ف	فعل	Verb	Verb
مذ	مذكر	Masculine	Noun
مص	مصدر (يرد مرفوعا).	Masdar	Noun
مج	مبني للمجهول	Passive	Verb
مف	مفرد	Singular	Noun
لا	فعل لازم	Intransitif	Verb
م	فعل متعد	Transitif	Verb
مح	متعد بحرف	Transitif + preposition	Verb
مظ	متعد بظرف	Transitif + ADV	Verb
صف	صفة	Adjectif	ADJ
أ. تف	أَفْعَلُ التفضيل	Elative	Elative

Tableau 4 : Tableau récapitulatif des données morphosyntaxiques à extraire

2.3.1 Verbe - ف: فِعْل

Les données lexicales pour le verbe sont structurées comme suite :

<données_lexicales>ف: ر. م. مح</données_lexicales>

Pour toutes les balises qui commencent par ف on ajoute dans l'entrée une balise POS (Part-Of-Speech)

<POS> فِعْل </POS>

2.3.2 Syntaxe

Dans cette étape on signale dans le champ syntaxe si les verbes sont intransitifs " لازم ", transitif " متعد ", et ceux qui ont un complément d'objet indirect par l'intermédiaire d'une préposition (مح) : T_PREP (Transitif par préposition).

Dans ce cas il y a des verbes qui ont des comportements variés et peuvent être transitifs (T), intransitif (INT) et transitif indirect (T_PREP), en fonction de l'exemple ou ils sont utilisés.

<texte_libre> آيَدَ – [أ ي د] (ف: ر. لا. م). يُؤْيِدُ. مص. إيادٌ </texte_libre>

<DonneesLexicales> ف: ر. لا. م </DonneesLexicales>

<POS> VERB </ POS >

<syntaxe> T_INT </syntaxe>

2.4 Paradigme - Pattern

Du point de vue codification, ces données sont un peu vagues et il vaut mieux avoir le paradigme précis du verbe. Par exemple :

- Verbe de paradigme فَعَّلَ et de paradigme فَاعَلَ sont constitués de 4 consonantes رباعي mais ont des comportements syntaxiques et sémantiques très différents. Verbe de paradigme فَعَّلَ peut-être :
  - causatif (للتَّعْدِية) dans le sens « faire faire à quelqu'un », par exemple كَتَّبَ.
  - intensif (للتكثير) dans le sens « intensif » par exemple كَسَّرَ
- فَاعَلَ est un verbe de coparticipation où le sujet et le complément d'objet font la même action. كَاتَبَ : (للمُشارَكة)

Conclusion : l'information que les verbes ont 3,4 ou 5 consonantes ne suffit pas à arriver à la syntaxe et la sémantique verbale donc il vaut mieux codifier le paradigme فعل.

Pour codifier le paradigme du verbe.

- Verbe trilittère on change :

La première consonante du verbe (première consonante radicale) avec ف.

La deuxième consonante (deuxième consonante radicale) avec ع.

La troisième consonante (troisième consonante radicale) avec ل.

Exemple : كَتَبَ

<paradigme> فَعَلَ </paradigme>

Exemple : شَرِبَ

<paradigme> فَعِلَ </paradigme>

Exemple : كَبُرَ

<paradigme> فَعُلَ </paradigme>

a) Paradigme فَعَّلَ: (Form II)

La première consonante (la fãae) est une hamza avec fatha.

La deuxième consonante (la *'ayn*) a (shadda + fatha : عَّ).

La troisième (la lâm) a fatha.

<paradigme> فَعَّلَ </paradigme>

b) *Paradigme فَاعَلَ (Form III) :*

S'écrit suivant l'expression régulière suivante : C1(a?)AC2aC3a

Où :

C1 : la première consonne, C2 la deuxième…

a : la voyelle Fatha, (a?) expression régulière veut dire que l'existence de cette voyelle n'est pas obligatoire.

A : représente la voyelle longue Alif.

Donc dans notre cas, puisque on travaille sur la Hamza :

La première consonante (la fāae) est une hamza avec fatha + alif, devient : Alif madda آ

La deuxième et troisième consonantes ont fatha.

HaAC2aC3a → آ C2aC3a

c) *Paradigme أَفْعَلَ (Form IV):*

Ce paradigme commence par un préfixe (Hamza+fatha).

La première consonante (la fāae) est une Hamza avec absence de voyelle (sukūn).

La combinaison entre le préfixe et la première consonante donne une Alif madda آ . la deuxième et troisième consonantes ont comme voyelle une fatha.

HaHC2aC3a → آ C2aC3a

On note que les verbes de paradigme فَاعَلَ et les verbes de paradigme أَفْعَلَ ont la même forme morphologique dans la lettre Hamza : آعَلَ

Exemple 1 :

<texte_libre> آوَبَ – [أ و ب] (ف: ر. م. مح). يُؤاوِبُ، مص. مُؤاوَبَةٌ. <texte_libre/>

Exemple 2 :

<texte_libre> آيَدَ – [أ ي د] (ف: ر. لا. م). يُؤْئِدُ. مص. إيادٌ. <texte_libre/>

Dans les deux exemples le lemme " آوَبَ " et le lemme " آيَدَ " ont la même structure morphologique : Hamza madda suivie de deuxième consonne (la *'ayn*) avec fatha et d'une troisième consonne (la lâm) avec fatha.

Le facteur discriminant entre ces deux formes est la forme imperfective (المضارع), Les verbes de paradigme فَاعَلَ ont la forme imperfective de paradigme يُفَاعِلُ : yuC1aAC2iC3u

<Imperfective> يُؤاوِبُ </ Imperfective >

<paradigme> فَاعَلَ </ paradigme>

Les verbes de paradigme أَفْعَلَ ont la forme imperfective de paradigme يُفْعِلُ : yuC1C2iC3u

<Imperfective> يُؤْئِدُ </ Imperfective > yuC1C2iC3u

<paradigme> أَفْعَلَ </ paradigme>

Dans l'extraction automatique du paradigme on se base sur la forme imperfective afin de distinguer entre ces deux formes.

d) Paradigme انْفَعَلَ (Form VII)

Ce paradigme est écrit suivant l'expression suivante :

(prefixe : alif + (i?) + n + sukūn) + (C1+a) + (C2+a) + (C3+a) **ex: اِنْسَحَبَ**

On ajoute la balise du paradigme à tout verbe qui s'écrit sous la forme de l'expression indiquée

<paradigme> انْفَعَلَ </ paradigme>

e) Paradigme افْتَعَلَ (Form VIII)

L'expression du paradigme *افْتَعَلَ* est la suivante :

(prefixe : alif + (i?) + (C1+sukūn) + (TAe+a) + (C2+a) + (C3+a) **ex: اِمْتَثَلَ**

<paradigme> افْتَعَلَ </ paradigme>

Il y a des verbes de ce paradigme qui ont une dāl entre C1 et C2 de la racine :

Ex. اِزْدَحَمَ la racine est (ز ح م)

La forme à codifier est : **alif + (i?) + C1+sukūn + d+(a) + C2+(a) + C3+(a)**

f) Paradigme اِفْعَلَّ (Form IX)

L'expression du paradigme est la suivante :

alif + (i?) + C1+sukūn + C2+(a) + C3+(ṣadda **+a)**

Exemple :

<lemme>اِجْلَخَّ</lemme>

<racine1>ج ل خ</racine1>

<paradigme>اِفْعَلَّ</paradigme>

g) Paradigme اسْتَفْعَلَ (Form X)

(prefixe: alif + (i?) + s + sukūn + TAe + a) + (C1+ sukūn) + (C2+a) + (C3+a)

<paradigme> اسْتَفْعَلَ</ paradigme>

2.5 Nom Verbal المصدر

Nom Verbal المصدر **:** Représente l'infinitif substantivé c'est-à-dire le nom de l'action, il est un nom abstrait formé sur la même racine que le verbe auquel il est associé et exprime le

même contenu sémantique que le verbe. Un verbe peut avoir plus qu'un nom verbal. Par exemple, le verbe " وَدَّ " admet quatre noms verbaux différents :

وُدًّا – وِدَادا - وِدادةً - مَوَدَّةً

<texte_libre> آيَدَ – [أ ي د] (ف: ر. لا. م). يُؤْيِدُ. مص. مُؤايَدَةٌ </texte_libre>

<masdar1> مُؤايَدَةٌ </masdar1>

2.6 Inflexion

Dans cette partie on extrait pour les lemmes ses formes flexionnelles existantes :

- ✓ Singulier : مفرد
- ✓ Masculin : مذكر
- ✓ Féminin : مؤنث
- ✓ Dual : مثنى
- ✓ Pluriel : جمع

Pour le pluriel on trouve 3 types :

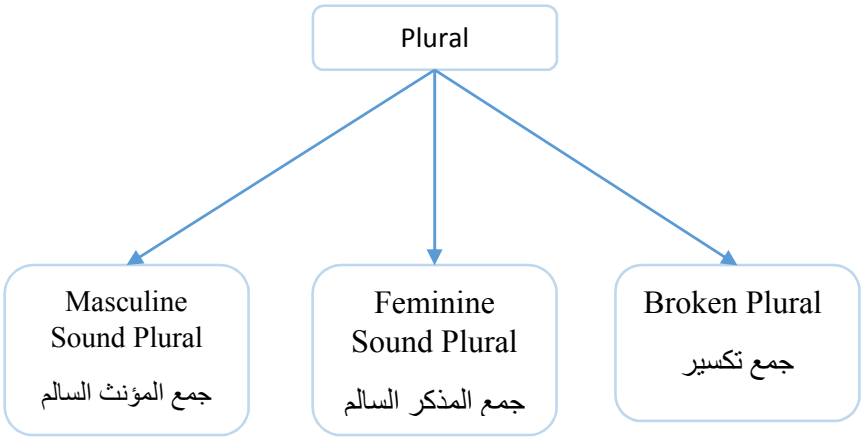

Figure 11: Pluriel du mot arabe

On indique ces informations en étiquetant les lemmes avec une balise d'inflexion :

```
<Entree_lexical>
<texte_libre>آبِقٌ، ةٌ – ج. أُبّاق، أُبَّق. [أ ب ق] (فا. مِنْ أَبَقَ)</texte_libre>
<lemme> آبِقٌ، ةٌ </lemme>
<racine1> أ ب ق </racine1>
<pos>Noun</pos>
<inflection>
add_femminile_prefix
<broken_plural>أُبَّق</broken_plural>
</inflection>
<glosse1>عَبْدٌ هارِبٌ مِنْ سَيِّدِهِ. "كَما يَلوحُ السِّجْنُ لِلْآبِقِ الطَّلِيقِ" (ن. محفوظ)</glosse1>
<DonneesLexicales>أَبَقَ مِنْ. فا</DonneesLexicales>
<ID>11</ID>
</Entree_lexical>
```

Dans ce cas la structure du texte libre est la suivante :

<texte_libre> آيَدَ – [أ ي د] (ف: ر. لا. م). يُؤْيِدُ. مص. مُؤايَدَةٌ </texte_libre>

<lemme> آيَدَ </lemme>

<racine1> أ ي د </racine1>

<paradigme> أَفْعَلَ </paradigme>

<DonneesLexicales> ف: ر. م </DonneesLexicales>

<syntaxe> T_INT </syntaxe>

<Imperfective> يُؤْيِدُ </ Imperfective >

<masdar1> مُؤايَدَةٌ </masdar1>

3. Organisation des glosses

Dans le but de pouvoir faire un traitement de la sémantique de la langue arabe on doit d'abord faire des prétraitements sur les glosse afin de les organisés dans un premier temps.

Parmi l'une des particularités du dictionnaire « Alghani Azzahir » est que chaque définition du lemme contient des exemples simples ou des citations (Poésie, Coran …)

Toute les glosses sont faite de deux parties :

- ✓ Glosse : le texte de la glosse.
- ✓ Exemple : texte entre guillemets " " qui peut être un exemple simple, citation, Coran, poésie… et dans ce cas l'exemple est suivi de parenthèse ( ).
- ✓ Citation : texte entre ( ) qui peut être le nom de auteur, poète ou Coran.

- Exemple simple :

```
<glosse1>"إحْتَجَّ بِكَلامِهِ": إتِّخَاذُهُ حُجَّةً. "إحْذَرْ خُصومَةَ الأَهْلِ والوَلَدِ واحْتَجِجْ عَلَيْهِمْ بِالحُجَجِ" (ع. الله بن المقفع)<glosse1/>
                                                                                    <definition>
                                                                    </ "إتِّخَاذُهُ حُجَّةً" "=val "sens"=att feat>
                                                                                    <statement>
                                                                    </ "إحْتَجَّ بِكَلامِهِ"=val "example"=att feat>
                                                                                    <statement/>
                                                                                    <statement>
                                                    </ "إحْذَرْ خُصومَةَ الأَهْلِ والوَلَدِ واحْتَجِجْ عَلَيْهِمْ بِالحُجَجِ"=val "example"=att feat>
                                                                    </ "ع. الله بن المقفع"=val "citation"=att feat>
                                                                                    <statement/>
                                                                                    <definition/>
<glosse2>"احْتَجَّ المُتَّهَمُ": قَدَّمَ حُجَجاً. "ظلّ الْمُتَّهَمُ يَحْتَجُّ بِحُجَجِهِ عَلَى بَراءَتِهِ"<glosse2/>
                                                                                    <definition>
                                                                    </ "قَدَّمَ حُجَجاً" "=val "sens"=att feat>
                                                                                    <statement>
                                                                    </ "احْتَجَّ المُتَّهَمُ"=val "example"=att feat>
                                                                                    <statement/>
                                                                                    <statement>
                                                    </ ظلّ الْمُتَّهَمُ يَحْتَجُّ بِحُجَجِهِ عَلَى بَراءَتِهِ" "=val "example"=att feat>
                                                                                    <statement/>
                                                                                    <definition/>
```

Figure 12: Exemple citation

- Poésie :

Les glosses contenant un exemple de poésie ont été contresignés suivant la structure suivante.

(شعر) première partie. (شعر) deuxième partie. (شعر) (poete).

Elles doivent être codifiées come suite:

< feat att = "example" val = " **première partie** **deuxième partie** " />

< feat att = "citation" val = poète />

```
                                                                    إحْتَجَنَ مالَ غَيْرِهِ": إقْتَطَعَهُ، سَرَقَهُ، إخْتَمَّ نَفْسَهُ بِهِ.<glosse4>
(شعر) "فَيا عَجَبَ الرَّهْنِ لِلقائِلا (شعر) تِ مِنْ آخِرِ اللَّيْلِ ماذا احْتَجَنْ" (شعر) (الأعشى).<glosse4/>
                                                                                    <definition>
                                                                    </ "إقْتَطَعَهُ، سَرَقَهُ، إخْتَمَّ نَفْسَهُ بِهِ" "=val "sens"=att feat>
                                                                                    <statement>
                                                                    </ "إحْتَجَنَ مالَ غَيْرِهِ"=val "example"=att feat>
                                                                                    <statement/>
                                                                                    <statement>
</ "تِ مِنْ آخِرِ اللَّيْلِ ماذا احْتَجَنْ" "فَيا عَجَبَ الرَّهْنِ لِلقائِلا"=val "example"=att feat>
                                                                    </ "الأعشى"=val "citation"=att feat>
                                                                                    <statement/>
                                                                                    <definition/>
```

Figure 13: Exemple Poésie

- Coran :

Des fois l'exemple est un verset Coranique et pour éviter les erreurs, ces derniers sont mis entre # #.

```xml
<glosse4>"يَخْتَسِبُ عِنْدَ الله خَيْراً": يُقدِّمه. #ومَنْ يَتَّقِ الله يَجْعَلْ لَـهُ مَخْرَجاً وَيَـرْزُقْـهُ مِنْ حَيْثُ لاَ يَخْتَسِبُ# (قـرآن).</glosse4>
<definition>
  <feat att="sens" val="يُقَدِّمهُ" />
  <statement>
    <feat att="example" val="يَخْتَسِبُ عِنْدَ الله خَيْراً" />
  </statement>
  <statement>
    <feat att="example" val="ومَنْ يَتَّقِ الله يَجْعَلْ لَـهُ مَخْرَجاً وَيَـرْزُقْـهُ مِنْ حَيْثُ لاَ يَخْتَسِبُ" />
    <feat att="citation" val="قـرآن" />
  </statement>
</definition>
```

Figure 14: Exemple des Verset Coranique

Conclusion

Dans ce chapitre, nous avons présenté la ressource linguistique utilisée durant ce projet et les différentes informations morphosyntaxiques et sémantiques extraites ainsi que leurs organisations dans le format XML intermédiaire adopté.

Le chapitre suivant sera consacré à la définition des outils utilisés dans la phase du développement du système.

Chapitre 5 :

Conception et réalisation de l'application

Ce chapitre contient deux parties : la première passe en revue les différentes technologies et outils utilisés pour la réalisation du projet.

La deuxième partie représente l'architecture du système et la présentation de quelques interfaces graphiques permettant l'utilisation de l'Application développée.

Introduction

Les dictionnaires sont très utiles pour des utilisations humaines et machines. Mais, en raison de la richesse des langues naturelles, ces dictionnaires sont complexes, leur construction et leur entretien coûtent très cher.

Ces dictionnaires utilisent divers structures sans aucun concept linguistique unifié. Par conséquent, la réutilisation de ces dictionnaires est difficile quand le défi est de fusionner l'échange et l'intégration dans des applications diverses donc une représentation uniforme peut être une solution pour ces défis.

Afin d'exploiter les données extraite, nous avons développez une application permettant la recherche par lemme, ainsi la classification des données selon les verbes, noms, paradigme, etc.

Le présent chapitre fournit une description globale de l'architecture technique de notre solution ainsi que les principaux outils de développement qui sont utilisés dans ce projet.

I. Technologie et outils de développement

Pour réaliser ce travail, nous avons fait appel à deux langages importants dans le monde informatique actuel, à savoir XML et JAVA. La force de XML réside dans sa capacité de décrire n'importe quel domaine de données grâce à son extensibilité. Quand à Java, le choix de ce langage pour notre projet est dû à sa caractéristique la plus connue, à savoir la portabilité. Ce choix permettra à l'ensemble du projet de tourner sur de multiples plateformes. Cette caractéristique est encore accentuée par l'utilisation du XML.

1. eXtensible Markap Language – XML

XML est un langage de balises qui peut être considéré comme un métalangage permettant de définir d'autres langages, autrement dit, il permet de définir de nouvelles balises (markup). Il découle d'un langage défini en 1986, le SGML (Standard Generalized Markup Language). Ensuite, il est standardisé par W3C en 10/02/1998. XML est une norme puissante et acceptée par la majorité. Il permet d'archiver et de communiquer des informations sur les objets. Il est en train de devenir le langage universel d'échange des documents.

La puissance de XML vient de la liberté de définir des balises et également leur structure selon le besoin de l'application. Pour définir ces balises, nous allons utiliser la DTD qui nous permet de préciser les balises à utiliser dans l'application.

2. JAVA

Java est un langage de programmation de quatrième génération développé par Sun. Sa popularité évolue grâce à une caractéristique majeure, sa portabilité. Dans le cadre de développement d'une interface de gestion et d'exploitation d'une base de données XML, les bibliothèques Swing et JDOM sont les plus importantes. La bibliothèque Swing propose une série d'objets graphiques tels que des fenêtres, des boutons, etc. Elle est la base de toute l'interface que nous avons développée. La bibliothèque JDOM offre des classes spécialisées pour un document XML tels que Document, Element, etc. Elle est la base de tous nos accès à la base soit en mode écriture soit en mode lecture.

3. JDOM

JDOM est une API évoluée du langage Java développée indépendamment de Sun Microsystems et maintenant sous le maintien d'Oracle depuis 2009. Elle permet de manipuler des données XML plus simplement qu'avec les API classiques. Son utilisation est pratique pour tout développeur Java et repose sur les API XML de Sun.

JDOM utilise des collections SAX pour parser les fichiers XML. En outre, JDOM utilise DOM pour manipuler les éléments d'un Document Object Model spécifique (créé grâce à un constructeur basé sur SAX). Ainsi, JDOM nous permet de construire des documents, de naviguer dans leur structure, d'ajouter, de modifier, ou de supprimer leur contenu.

II. Conception

1. Choix de l'UML

Nous avons choisi la modélisation avec UML pour sa notation qui est la plus appropriée pour des projets orientés objets. Ce choix peut être justifié également par plusieurs raisons :

- La notation UML facilite la compréhension et la communication d'une modélisation objet.

- Le processus de développement adopté se base sur les diagrammes UML
- UML est aujourd'hui un standard, adopté par les grands constructeurs de logiciel du marché.

2. Diagramme de classes

Le diagramme suivant présente la structure statique de notre application en termes de classes et de relations classées.

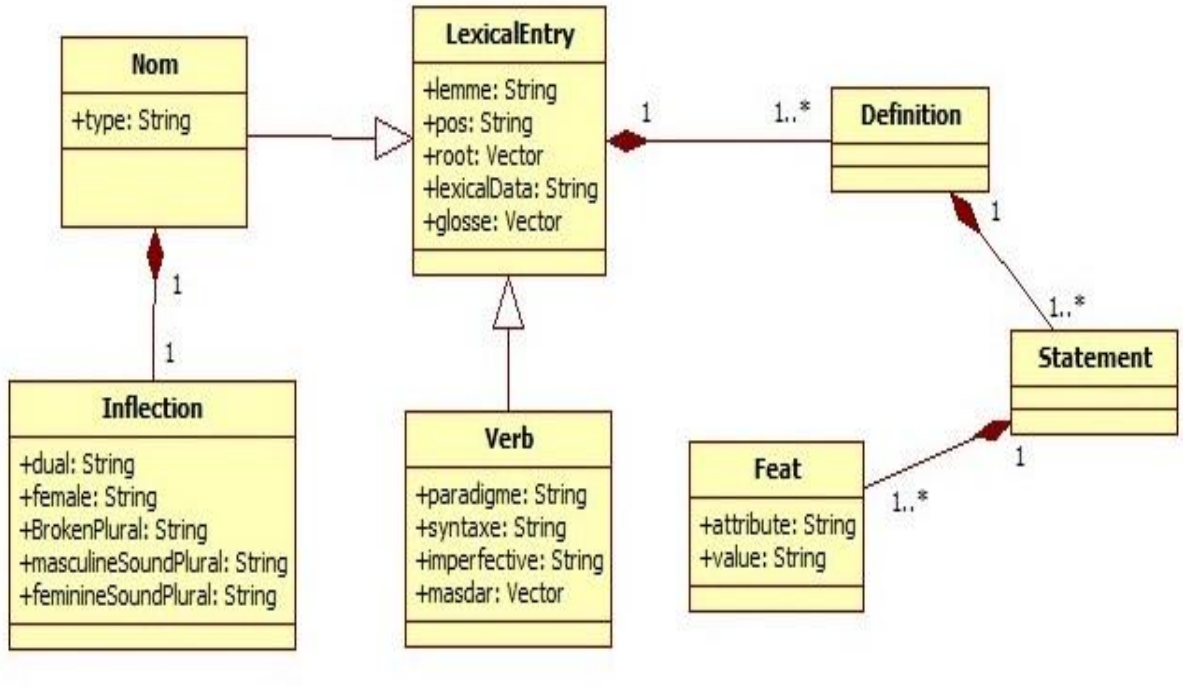

Figure 15 : Diagramme de classe

3. Diagramme de cas d'utilisation

Le digramme suivant montre les cas d'utilisation et l'interaction.

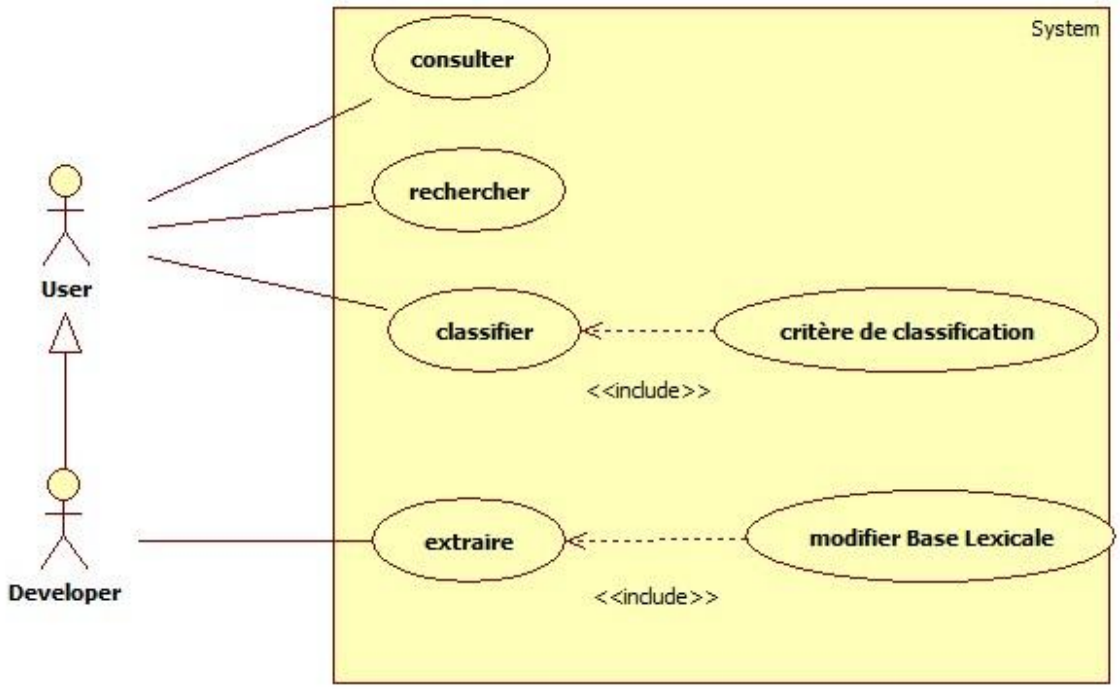

Figure 16 : Diagramme des cas d'utilisation

III. Réalisation

Cette partie de document concerne la couche de présentation et spécialement les interfaces Homme-Machine que nous avons développées.

Les figures suivantes présentent quelques exemples des interfaces :

Premièrement on choisit le fichier texte du dictionnaire numérisé, Après la fin de lecture du fichier, on génère le fichier XML, en choisissant extraire (استخراج) du menu (طبع).

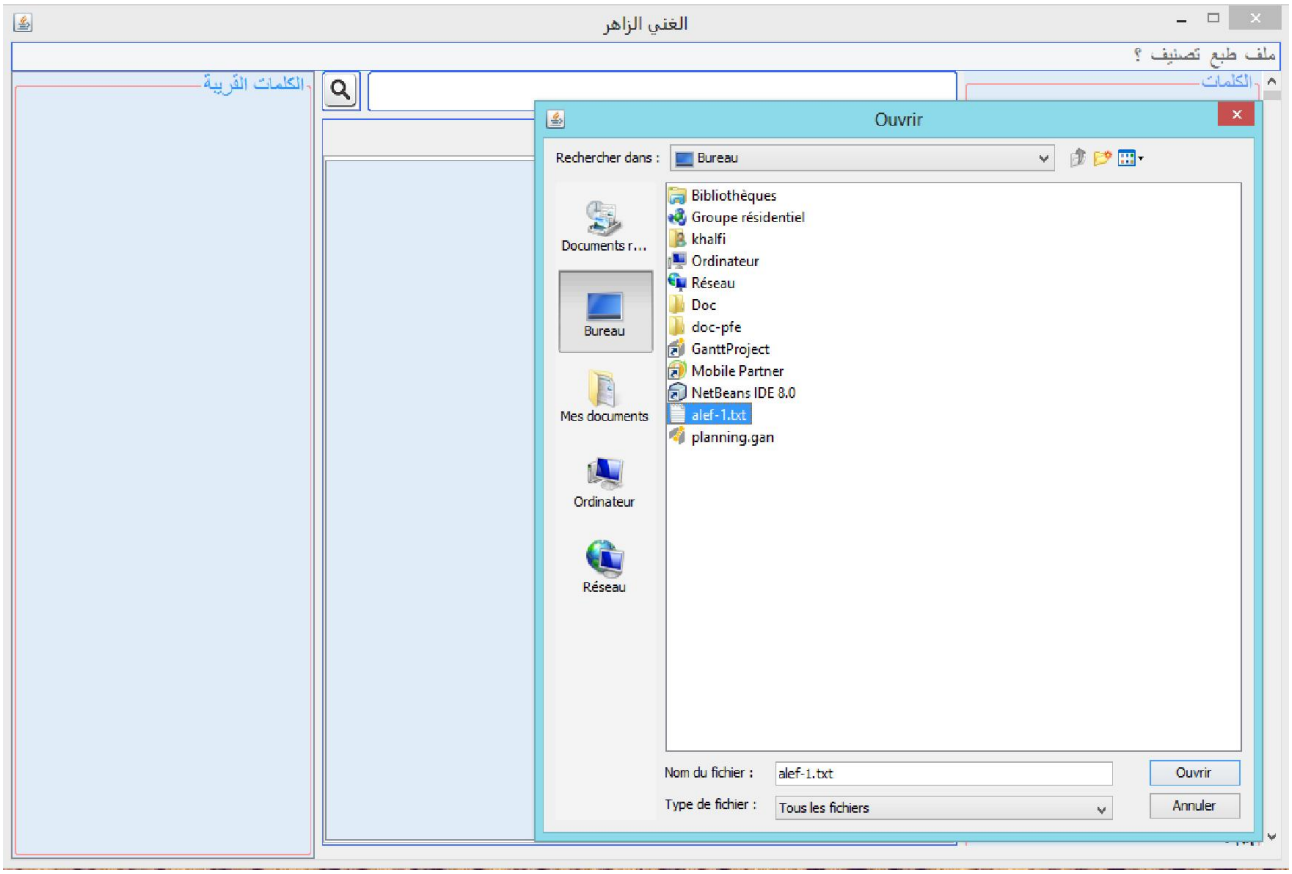

Figure 17 : Lecture fichier texte

La fenêtre principale se compose de trois parties, la première partie à droite pour l'affichage de tous les lemmes extraits, la deuxième à gauche pour afficher les mots proches du lemme sélectionné en se basant sur sa racine, et la troisième partie au milieu pour afficher les caractéristiques morphologiques, syntaxiques et sémantiques, aussi pour effectuer une recherche par mot.

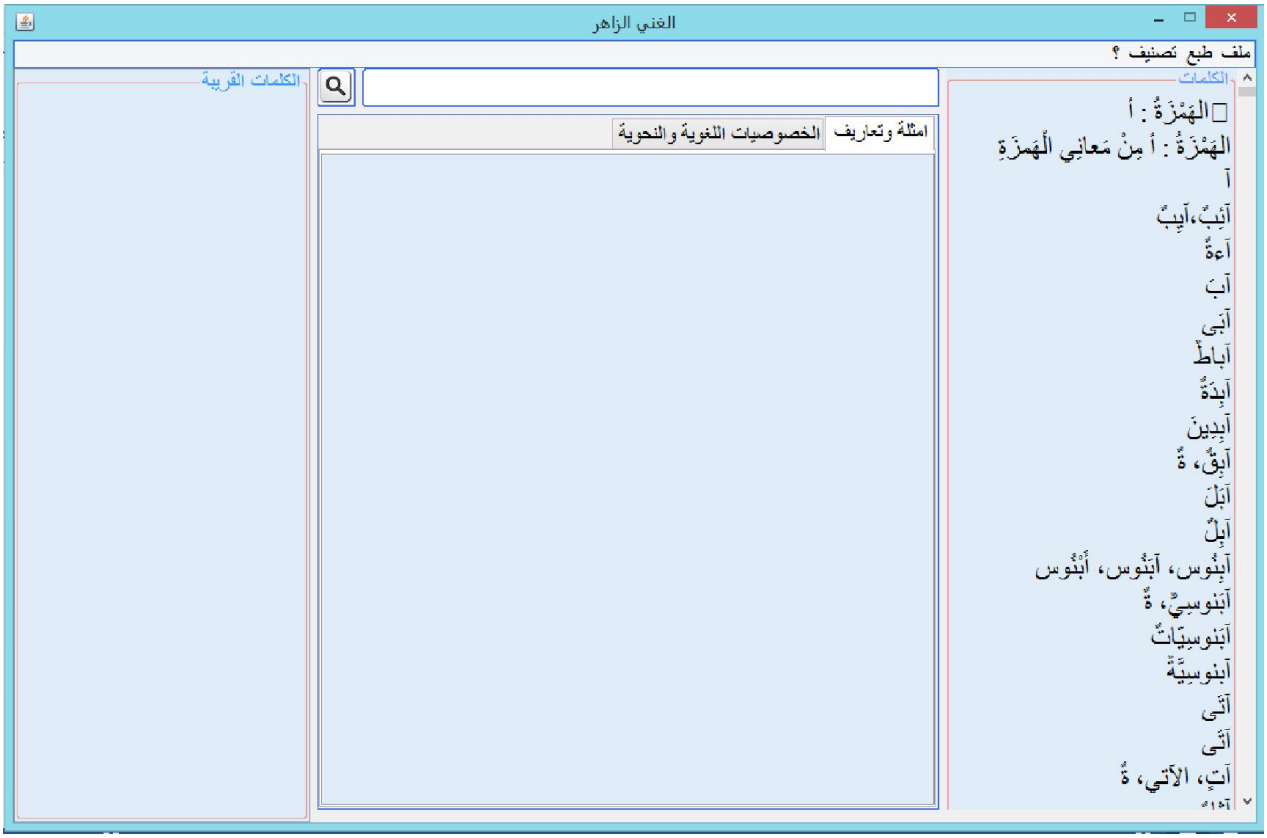

Figure 18 : Fenêtre principale

Notre application permet aussi de classifier ces données par POS, c-à-d par Noms, Adjectifs, ou Verbes, et pour ces derniers on peut les classifier aussi par nombre de consonne ou encore par paradigme.

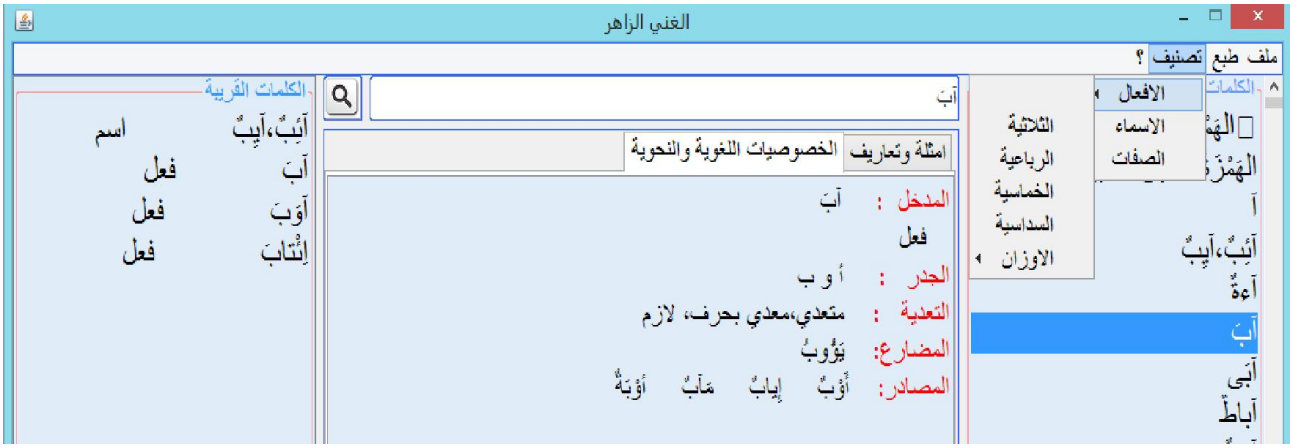

Figure 19 : Menu Classification

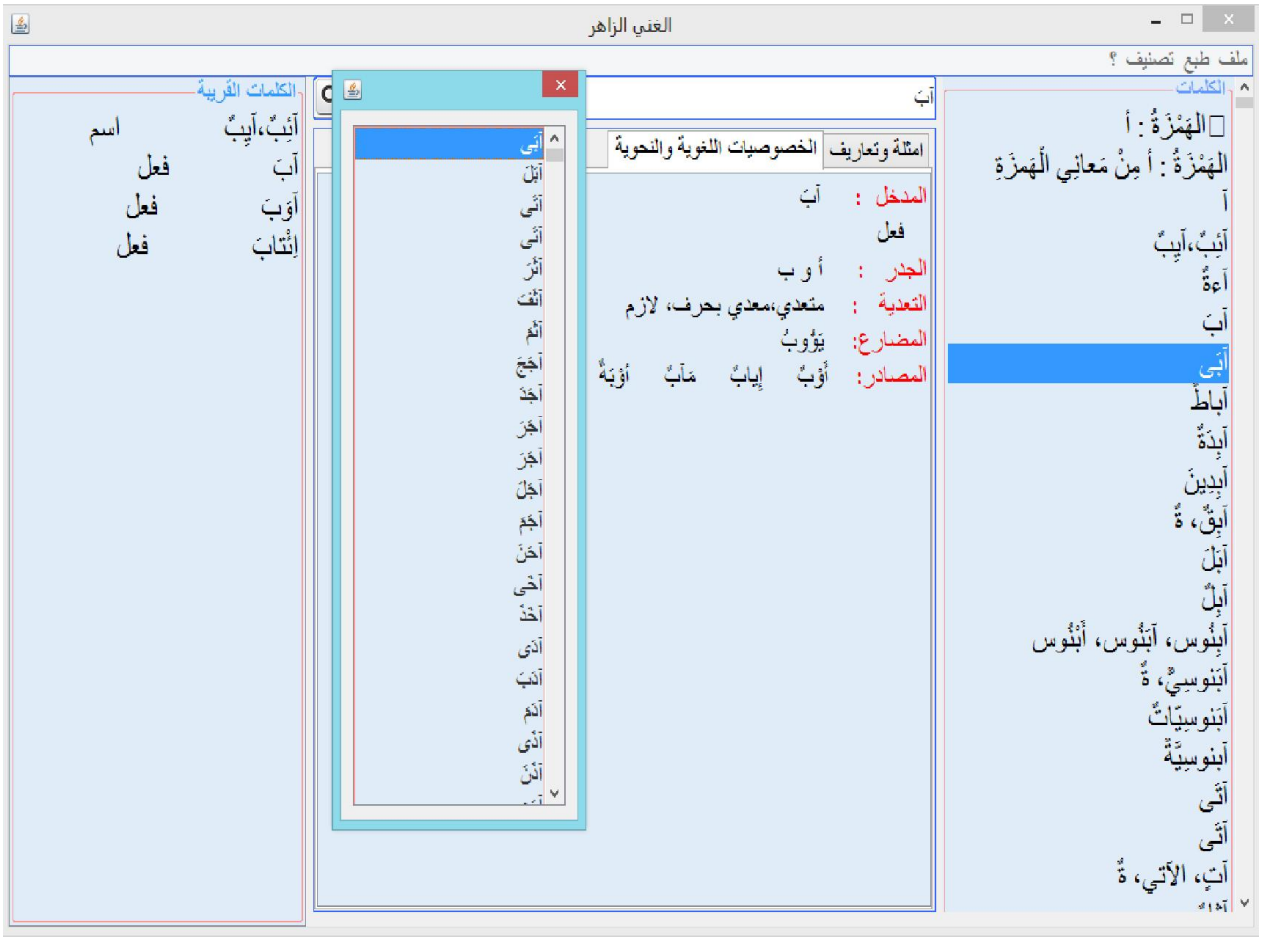

Figure 20 : Exemple de classification des verbes quadrilatères

Conclusion

Dans ce chapitre nous avons présenté la dernière phase du cycle de développement et nous avons donné une vue sur quelques interfaces réalisées du coté IHM pour l'exploitation des données extraites, en offrant des services de recherches et de classifications selon la POS, ainsi que le nombre de consonnes pour les verbes et aussi par paradigmes, et par Adjectif pour les noms.

Conclusion et Perspectives

Ce stage de fin d'études nous a permis d'intégrer le domaine de la recherche dans les meilleures conditions possibles. Nous y avons découvert ce monde des Traitements Automatiques de la Langue Arabe, les contraintes que cela implique et les méthodes à mettre en œuvre pour répondre aux problématiques.

Ce projet nous a été aussi d'une grande utilité sur le plan organisationnel. En effet, nous avons acquis des techniques de répartition de tâches et de communication au sein d'une équipe de plusieurs personnes afin d'améliorer le rendement du travail.

Durant notre stage, nous sommes sentis complètement impliquer dans le projet. Les encadrants soit au sein de la FST soit au sein de l'ILC nous ont fait confiance et nous ont soutenu, tout au long de notre stage.

En ce qui concerne la démarche, nous avons en premier lieu effectués étude bibliographique afin de comprendre les différents aspects de ce domaine de recherche. En deuxième lieu nous avons mis en place une solution de représentation des données morphosyntaxiques et sémantiques extraites du dictionnaire *Alghani AZZAHIR,* dans un format XML intermédiaire. En troisième lieu nous avons exploité ces données extraites en développant un module de recherche et de classification, en effectuant une recherche par lemme qui offre toutes les informations liées à celui-ci et une recherche par racine qui permet d'accéder à la base pour importer toutes les entrées qui présentent les formes dérivées, soit nominales ou verbales, de la racine et les proposer dans un champs des mots proches.
En ce qui concerne la classification proposée, elle permet la classification des données de la base lexicale suivant les noms, adjectifs et verbes, et pour ces derniers selon leurs nombres de consonnes, et aussi par paradigme.

Comme perspectives nous tenons de représenter tout le dictionnaire Alghani AZZAHIR sous format XML intermédiaire, et de continuer l'extraction du maximum des caractéristiques et ensuite de migrer vers une représentation LMF complète de la langue arabe, permettant d'enrichir le contenu numérique de cette langue.

Bibliographie

[1] abdesssalam, M. H. (1982). *كتاب سبويه، الجزء الرابع* Dar Rafai, Riad.

[2] AL Hajjar, a. (2010). A new system for evaluation of Arabic root extraction methods, The Fifth International Conference on Internet and Web Applications and Services. ICIW, Barcelona,Spain, pp. 489-495, 2010.

[3] Beesly. (1998). Arabic Morphology Using Only Finate-State Operations, Proceedings of the Workshop on Computational Approaches to Semetic languages. Montreal, Quebec, pp 50-57.

[4] Blachère, R., & Gaudefroy-Demombynes, M. (1975). "Grammaire de l'arabe lassique", Edition Maisonneuvre-Larose, Paris.

[5] BOUDLAL, A., LAKHOUAJA, A., MAZROUI, A., MEZIANE, A., OULD ABDALLAHI OULD BEBAH, M., & SHOUL, M. (2010). *Alkhalil Morpho Sys : A Morphosyntactic analysis system for Arabic text.*

[6] Buckwalter, T. (2002-2004). Buckwalter Arabic Morphological Analyzer Version 1.0. Linguistic Data Consortium, catalog number LDC2002L49 and ISBN 1-58563-257-0.

[7] CLN. (2009). *Compare le net (CLN), website: http://www.compare-le-net.com. 2009*.

[8] Cohen, D. (1970). *Études de linguistique sémitique et arabe Janua linguarum: Series practica. Walter de Gruyter GmbH & Co. KG .*

[9] Darwish. (2002). Building a Shallow Arabic Morphological Analyser in One Day". In Proceedings of the ACL-02Workshop on Computational Approaches to Semitic Languages.

[10] Dbili, F., Achour, H., & Souissi, E. (2002). *De l'étiquetage grammatical à la voyellation automatique de l'arabe.*

[11] Diab, M. (2009). Second Generation Tools (AMIRA2.0) : Fast and Robust Tokenization, POS tagging, and Base Phrase Chunking. .

[12] El Isbihani, Anas, Shahram, K., Oliver , B., & Hermann, N. (2006). Morpho-syntactic Arabic Preprocessing for Arabic to English Statistical Machine Translation". .

[13] El Kassas, D., & Kahane, S. (2004). *"Moélisation de l'ordre des mots en arabe standard" In: JEP-TALN 2004. Fès .*

[14] Francopoulo, G. a. (2006). Lexical Markup Framework (LMF), LREC 2006, Genoa.

[15] Goldsmith, J. (2001). Unsupervised learning of the morphology of a natural language. Computational Linguistics, 27(2), 153-198.

[16] Habash, & Rombow, O. (2005). "Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop".

[17] Habash, N., & Rambow, O. (2006). "MAGEAD: A Morphological Analyzer and Generator for the Arabic Dialects". In: In Proceedings of COLING-ACL. Sydney, Australia.

[18] Habash, Rambow, O., & Roth, R. (2009). *"MADA+TOKAN: A Toolkit for Arabic Tokenization, Diacritization, Morphological Disambiguation, POS Tagging, Stemming and Lemmatization".*

[19] Hegazi, E. (1986). Natural Arabic Language Processing, Proceedings of the 9th National Computer Conference and Exhibition, Riyadh, Saudi Arabia, 1-17.

[20] khoja, g. (1999). Stemming Arabic text. Computer Science Departement, Lancaster University, Lancaster, UK.

[21] Kulick, S. (2010). Simultaneous tokenization and part of speech tagging for Arabic with out a morphological analyser.

[23] Kulick, S. (2011). “Exploiting Separation of Closed-Class Categories for Arabic Tokeniza-tion and Part-of-Speech Tagging”. In: ACM Transactions on Asian Language Information Processing TALIP.

[24] Lee, Y.-s. (2004). Morphological analy-sis for statistical machine translation. (HLT-NAACL, pp.57–60.

[25] Léon, C. e. (2002). constitution du TAL, Étude historique des dénominations et des concepts, TAL, 43-3, 21-55.

[26] LJN. (2010). Le Journal du Net (LJN), website: http://www.journaldunet.com/ default.shtml 2010.

[27] Maamouri, Dave Graff, Basma Bouziri, Sondos Krouna, Ann Bies, & Seth Kulick. (2010). LDC Standard Arabic Morphological Analyzer (SAMA) Version 3.1 LDC2010.

[28] Mansour, S. (2010). MorphTagger: HMM Based Arabic Segmentation for Statistical Machine Translation”. In: International Workshop on Spoken Language Translation. Paris, France.

[29] Mansour, S., Siman'an, K., & Winter, Y. (2007). “Smoothing a lexicon-based POS tagger for Arabic and Hebrew”. In: Proceedings of the 2007 Workshop on Computational Approaches to Semitic Languages: Common Issues and Resources Semitic ’07. Prague, Czech Republic: Association for Computational Linguistics.

[30] Marsi, B. e. (2005). Memory-Based morphological analysis generation and part-of-speech tagging of Arabic.

[31] Mohamed, E., & Oflazer, B. M. (2012). “Annotating and Learning Morphological Segmentation of Egyptian Colloquial Arabic”. In: Proceedings of the Eighth International Conference on Language Resourcesand Evaluation (LREC-2012). .

[31] Romary L, e. A. (s.d.). "Action nationale INRIA Syntax (Décembre 2001 - Décembre 2003)", INRIA, 2003.

[32] Romary.L. (2003). "Action nationale INRIA Syntax (Décembre 2001 - Décembre 2003)", INRIA, 2003.

[33] Sakher_Lexicons. (2009). Sakher_Lexicons: Lisan Al-Arab, Al Qamous Al Mouhi, Al Wasit, Al Mouhit, Mouhit Al Mouhit, Al Ghani, Taj Al Arous, Najaat al Raed http://lexicons.sakhr.com, 2009.

[34] SEO. (2010). *SEO Search Consultants Directory, website: http://www. seoconsultants.com/search-engines/. 2010.*

[35] Shah, Rushin, Paramveer , S., Mark , L., Dean, Foster, Mohamed, Maamouri, & Lyle, Ungar . (2010). *A new approach to lexical disambiguation of Arabic text.*

[36] Soudi. (2002). A Computational Lexeme-Based, Treatment of Arabic Morphology. Doctorat d’état, Mohamed V University.

[37] WLP. (2007). Women's Learning Partnership (WLP), Faits et chiffres en technologies, website: http://www.learningpartnership.org/fr/node/1038.

[38] Yousfi. (2010). The morphological analysis of Arabic verbs by using the surface patterns. IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 3, No 11, May 2010.

Annexe

Annexe A : Tableau standard des Translations

Alphabet	Translation	Nom de la lettre
ء	***’***	*hamza*
ب	***b***	*bā’*
ت	***t***	*tā’*
ث	***ṯ***	*ṯā’*
ج	***ǧ***	*ǧīm*
ح	***ḥ***	*ḥā’*
خ	***ḫ***	*ḫā’*
د	***d***	*dāl*
ذ	***ḏ***	*ḏāl*
ر	***r***	*rā’*
ز	***z***	*zay*
س	***s***	*sīn*
ش	***š***	*šīn*
ص	***ṣ***	*ṣād*
ض	***ḍ***	*ḍād*
ط	***ṭ***	*ṭā’*
ظ	***ẓ***	*ẓā’*
ع	***‘***	*‘ayn*
غ	***ġ***	*ġayn*
ف	***f***	*fā’*
ق	***q***	*qāf*
ك	***k***	*kāf*
ل	***l***	*lām*
م	***m***	*mīm*
ن	***n***	*nūn*
ه	***h***	*hā’*
و	***w***	*wāw*
ا	***ā***	*’alif*
ي	***y***	*yā’*

Annexe B : Extrait du fichier texte

الـهَمْزَةُ : أ - الْحَرْفُ الأَوَّلُ مِنَ الْحُرُوفِ الـهِجائِيَّةِ (مؤنثة) وَتُرْسَمُ عَيْناً صَغِيرَةً (ء) وَيَتَعَيَّنُ مَوْقِعُها عَلَى الأَلِفِ أَوِ الـواوِ أَوِ الـياءِ أَوْ
الـهَمْزَةُ : أ مِنْ مَعانِي الْـهَمزَةِ - أن تَـأْتِيَ 1. حَرْفَ نِداءٍ لِلْقَرِيبِ: "أَخالِدُ خُذِ الْقَلَمَ". 2. حَرْفَ اسْتِفْهامٍ: "أَخَرَجْتَ هَذا الصَّباحَ ؟"، "أَه
آ - وَتُسمَّى الْمَدَّةَ - 1. وَهِيَ حَرْفُ نِداءٍ لِلْبَعِيدِ مَبْنِيٌّ عَلَى السُّكُونِ: "آخَمِيدُ تَمَهَّلْ". 2. تَـأْتِي لِتَنْبِيهِ الغافِلِ والسّاهِي: "آخالِدُ، إِنَّ
آبْبُ،آبِيبُ - [أ و ب] (فـا. آب). "آبِيبُ مِنْ عَمَلِهِ": 1. راجِعٌ 2. قـادِمٌ، 3. عائِدٌ. "فَـأَبْتُ إِلَى فَهْمٍ وَما كِدْتُ آبِيباً" (تـأَبَّط شَرّاً).
آءُ - [أ و أ] (نب) 1. نَباتٌ زِراعِيٌّ، مِنْ فَصِيلَةِ الجناحِيّاتِ، أَزْهارُهُ بِيضٌ، تُشْبِهُ السَّنابِلَ، دائِمُ الأَوْراقِ، طَعْمُها مُرٌّ، تَصْلُحُ لِلبَّباغَةِ
آبَ - [أ و ب] (ف: ث. لا. م. مج). يَؤُوبُ، مص. أَوْبٌ، إِيابٌ، مَآبٌ، أَوْبَةٌ. 1."آبَ مِنَ السَّفَرِ": رَجَعَ، عادَ. "وآبَ النـازِحونَ إِلى مقـارِّهِمْ و
آبَى - [أ ب ي] (ف: ر. م). يُؤْبِي، مص. إِبـاءٌ. 1."آبَى صاحِبُهُ الأَمْرَ": جَعَلَهُ يَـأْبـاهُ. 2."آبَى الطَّعامَ": اِمْتَنَعَ عَنْهُ.
آبـاطُ - [أ ب ط] (جمع إِبْطٍ) 1. "مَرَبَ آبـاطَ الأمورِ": عَرَفَ خفـايـاها وبَـواطِنَها .
آبِدَةُ - ج. أَوابِدُ. [أ ب د] 1."إِنَّها حَقّاً لآبِدَةٌ": أَمْرٌ عَظِيمٌ يُتَعَجَّبُ مِنهُ. "جاءَنـا بِـآبِدَةٍ". 2."آبِدَةُ الدَّهْرِ": الدّاهِيَةُ يَبْقَى ذِكْرُها أَ
آبِدِينَ - [أ ب د] 1. لَمْ يُسْتَعْمَلْ مُفْرَدُهُ. 2."لا أَفْعَلُ ذَلِكَ أَبَدَ الآبِدِينَ": أَبَدَ الدَّهْرِ
آبِقٌ، ةٌ - ج. أُبَّاق، أُبَّق. [أ ب ق] (فـا. مِنْ أَبَقَ) 1. عَبْدٌ هارِبٌ مِنْ سَيِّدِهِ. "كَما يَلوحُ السِّجْنُ لِلْآبِقِ الطَّلِيقِ" (ن. محفوظ)
آبَلَ - [أ ب ل] (ف: ر. لا). يُؤْبِلُ، مص. إِبـالٌ. 1. "آبَلَ الصَّحْراوِيُّ": كَثُرَتْ إِبِلُهُ
آبِلٌ - [أ ب ل] (فـا. من أَبَلَ، أَبِلَ) 1. "رَجُلٌ آبِلٌ": حاذِقٌ بِرِعايَةِ الإِبِلِ، وعالِمٌ عارِفٌ بِمَصْلَحَتِها. (شعر)"فَنَأَتْ وانْتَوَى بِها عَنْ هَواها
آبِنُوس، آبَنُوس، أَبْنُوس - (نب) (د) (يو).(abenos) مأخوذٌ مِنَ الاسْمِ العِلْمِيّ (abenus) 1. شَجَرَةٌ مِنْ مَجْموعَةِ الآبنوسِيّاتِ وَفَصِيلَةِ القرنب
آبَنوسِيٌّ، ةٌ - (مَنْسوبٌ إِلى آبنوس). كُلُّ ما لَهُ عَلاقَةٌ بِـالآبنوسِ. 1."شَعْرٌ آبَنُوسِيٌّ": ذو لَوْنٍ أَسْوَدَ حالِكٍ. 2."نَجّارٌ آبَنوسِيٌّ": مُخْتَصٌّ بِصِناعَ
آبَنوسِيّاتُ - 1. مَجْموعُ أَشْجارٍ وَشُجَيْراتٍ مِنْ فَصِيلَةِ القرنِيّاتِ، تَكْثُرُ في البِلادِ الحارَّةِ، أَلْواحُها شَدِيدَةُ الصَّلابَةِ
آبَنوسِيَّةُ - 1. مادَّةٌ سَوْداءُ صُلْبَةٌ، تُتَّخَذُ مِنْ خَلْطِ الكِبْرِيتِ بِالمَطّاطِ النَّقِيِّ، غَيْرُ مُوَصِّلَةٍ لِلْكَهْرَباءِ
آتَى - [أ ت ي] (ف: ر. م. مج). يُؤْتِي، مص. إِيتـاءٌ. 1."آتـاهُ اللهُ رِزْقاً عَمِيماً": أَعْطاهُ. "يُؤْتِي اللهُ الحِكْمَةَ مَنْ يَشاءُ". 2."آتـاهُ البُغْمَ
آتَى - [أ ت ي] (ف: ر. م. مج). يُؤاتِي، مص. مُؤاتـاةٌ. 1."آتـاه عَلَى ما فَعَلَ": وافَقَهُ عَلَيْهِ. 2."آتَتْهُ الفُرْصَةُ": وافَقَتْهُ، سَنَحَتْ لَهُ.
آتٍ، الآتي، ةٌ - [أ ت ي] (فـا. من أَتَى). 1."هُوَ آتٍ لا رَيْبَ فِي ذَلِكَ": مُقْبِلٌ، قـادِمٌ. "كـأَنَّهُ آتٍ مِنْ عُمْقِ سَحِيقٍ" (غسان كنفـاني). "كُلُّ آتٍ
آثـارُ - جمع أَثَرٍ. [أ ث ر] 1."آثـارُ النَّعْيِ": عَلاماتُهُ. 2."الآثـارُ النّـارِيخِيَّةُ": القُصُورُ والأَبْنِيَةُ القَدِيمَةُ، وَما فِي الْمَتـاحِفِ مِنْ ثَمـاثِي
آثَرَ - [أ ث ر] (ف: ر. م. مج). يُؤْثِرُ، مص. إِيثـارٌ. 1."آثَرَ الشَّيْءَ": فَضَّلَهُ، اخْتـارَهُ. "آثَرَ القَطِيعَةَ وَحَمِدَ مَغَبَّتَها واسْتَمْرَأَ مَذاقَـها"
آثَفَ - [أ ث ف] (ف: ر. م). يُؤْثِفُ، مص. إِيثـافٌ. 1. "آثَفَ الرَّحّالَةُ القِدْرَ قُرْبَ الْخَيْمَةِ": وَضَعَها عَلَى الأَثـافِيِّ
آثِلٌ، ةٌ - [أ ث ل] (فـا. من أَثَلَ) 1. "أَمْرٌ آثِلٌ": مُتَأَصِّلٌ، ثـابِتٌ. "أَثَّلَ اللهُ مُلْكاً آثِلاً". "رِبـابَةُ رُبَّ وَمُلْكاً آثِلاً" (رؤبة بن العجاج).
آثَمَ - [أ ث م] (ف: ر. م). يُؤْثِمُ، مص. إِيثـامٌ. 1."آثَمَ جارَهُ": أَوْقَعَهُ في الإِثْمِ، وَجَعَلَهُ آثِماً. 2."أَثَمَهُ القـاضِي": عَدَّهُ آثِماً.

Annexe C : Extrait du fichier XML généré

```
<texte_libre>آبَ - [أ و ب] (ف: ث. لا. م. مج). يَؤُوبُ، مص. أَوْبٌ، إِيابٌ، مَآبٌ، أَوْبَةٌ.</texte_libre>
<lemme>آبَ</lemme>
<racine1>أ و ب</racine1>
<pos>Verb</pos>
</ inflection>
<syntaxe>INT_T_T_PREP</syntaxe>
<imperfective>يَؤُوبُ</imperfective>
<masdar1>أَوْبٌ</masdar1>
<masdar2>إِيابٌ</masdar2>
<masdar3>مَآبٌ</masdar3>
<masdar4>أَوْبَةٌ</masdar4>
<glosse1>"آبَ مِنَ السَّفَرِ": رَجَعَ، عادَ. "وآبَ النازِحونَ إلى مقارِّهِمْ فِراراً مِنْ مُشارَكَتِكَ في هُمومِكَ" (عمر الدسوقى)</glosse1>
<definition>
</ "رَجَعَ، عادَ "=val "sens"=att feat>
<statement>
</ "آبَ مِنَ السَّفَرِ"=val "example"=att feat>
<statement/>
<statement>
</ "وآبَ النازِحونَ إلى مقارِّهِمْ فِراراً مِنْ مُشارَكَتِكَ في هُمومِكَ"=val "example"=att feat>
</ "عمر الدسوقى"=val "citation"=att feat>
<statement/>
<definition/>
<glosse2>"يَؤُوبُ إلَى اللهِ": يَتُوبُ</glosse2>
<definition>
</ "يَتُوبُ "=val "sens"=att feat>
<statement>
</ "يَؤُوبُ إلَى اللهِ"=val "example"=att feat>
<statement/>
<definition/>
<glosse3>"آبَتِ الشَّمْسُ": غابَتْ</glosse3>
<definition>
</ "غابَتْ "=val "sens"=att feat>
<statement>
</ "آبَتِ الشَّمْسُ"=val "example"=att feat>
<statement/>
<definition/>
<glosse4>"آبَ الْماءُ": وَرَدَهُ لَيْلاً.</glosse4>
<definition>
</ "وَرَدَهُ لَيْلاً."=val "sens"=att feat>
<statement>
</ "آبَ الْماءُ"=val "example"=att feat>
<statement/>
<definition/>
<DonneesLexicales>ف: ث. لا. م. مج</DonneesLexicales>
<ID>6</ID>
<Entree_lexical/>
```